STATISTICAL COMPUTING DENGAN PROGRAM R

UNTUK RISET SAINS TERAPAN

ZULFIKAR, SP. M.SI
MUNAWARAH, S.KOM-M.SI
AMBAR SUSANTI, SP. MP

STATISTICAL COMPUTING DENGAN PROGRAM R

Oleh:

Zulfikar, SP. M.Si
Ambar Susanti, SP. MP
Munawarah, S.Kom. M.Si

Penerbit :
Fakultas Pertanian Universitas KH. A. Wahab Hasbullah
Anggota IKAPI:
N0. 297/Anggota Luar Biasa/JTI/2021

STATISTICAL COMPUTING DENGAN PROGRAM R

Penulis:
Zulfikar, SP. M.Si
Ambar Susanti, SP. MP
Munawarah, S.Kom. M.Si

ISBN: 978-623-7540- 359

Perancang Sampul:
Sujono

Penata Letak:
Muhammad Misbakul Munir

Pracetak dan Produksi:
Tim UNWAHA Press

Penerbit:
Fakultas Pertanian, Universitas KH. A. Wahab Hasbullah Tambakberas Jombang

Redaksi: Jl. Garuda No. 9 Tambakberas Jombang – Jawa Timur
Telp: 0321 853533
e-mail: fapertapublisher@unwaha.ac.id
http://www.unwaha.ac.id

Cetakan Pertama, tahun 2022
i–xxiv + 232 hlm, 15.5 cm x 23.5 cm

PENGANTAR PENULIS

Statistika merupakan salah satu cabang ilmu matematika terapan yang sangat berperan dalam bidang riset. Setiap kegiatan riset tidak terlepas dari keberadaan statistika sebagai alat untuk analisis data-data riset. Kendala yang sering dihadapi oleh pelaku riset, baik mahasiswa, dosen maupun umum memiliki waktu terbatas untuk mempelajari statistika secara mendalam serta tingkat pemahaman yang beragam dalam mengintepretasi data hasil analisis, apalagi data riset yang didapatkan memiliki jumlah data yang cukup besar. Seringkali periset mengalami kesulitan dalam melakukan komputasi data risetnya, sehingga mereka terkadang mengandalkan aplikasi-aplikasi statistika yang lisensinya cukup mahal untuk dibeli. Bahkan jalan pintas sering dipakai dengan menggunakan software-software bajakan yang akibatnya kualitas risetnya menjadi rendah sehingga publikasi risetnya sulit diterima di jurnal internasional bereputasi.

Sebagai upaya untuk mengatasi mahalnya software statistika maka alternatifnya adalah menggunakan software statistika open access, yaitu software yang free tanpa membeli lisensi. Salah satu software statistika yang terkenal adalah program R, dimana software ini bisa didownload bebas. Keunggulan software statistika ini memiliki paket aplikasi yang komplit, memuat berbagai model analisis dari analisis statistika sederhana sampai pada tingkat analisis statistika untuk data-data yang memiliki banyak variabel (multivariate). Keunggulan software statistika ini terkoneksi Cloud sehingga mampu mengupdate paket program sewaktu-waktu terhadap paket-paket analisis statistika yang ingin digunakan periset. Namun masih banyak periset beranggapan aplikasi program R cukup rumit, karena menggunakan coding yang harus dibaca dalam R.

Program R memiliki sistem coding, sehingga pengguna bisa bebas berkreasi baik untuk fungsi statistikanya maupun tampilan visual yang dihasilkan dari analisis data. Hal ini menjadikan output visual yang dihasilkan program R sangat menarik dan cukup beragam, bahkan mampu menampilan sisi lain fungsi grafik secara detail. Sebagai upaya untuk memudahkan periset dalam mempelajari aplikasi sortware R ini makan disusunkan buku ini, dimana materi yang dikandung dalam buku ini mampu menjelaskan secara mendetail tahapan operasi program R. Buku ini menerangkan fungsi program R secara mendalam serta langkah-langkah operasi secara detail baik mulai dari download aplikasi,

cara mengorasikannya, input data riset sampai penggunaan coding R. Setiap langkah oprasi program R dilengkapi dengan gambar, serta informasi yang menjelaskan hasil analisis dengan interpretasi yang mudah dipahami.

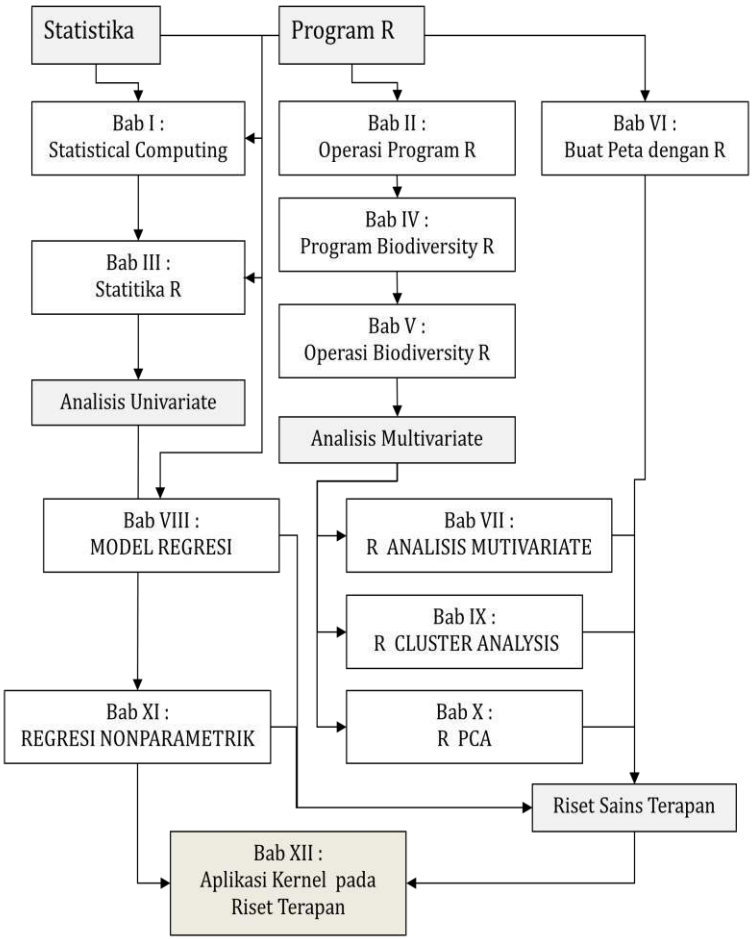
Buku ini disusun dalam rangka memenuhi luaran tambahan dari pelaksanaan hibah riset terapan multiyears (2021-2022) tahun pertama. Ucapan terima kasih disampaikan kepada Direktorat Jenderal Pendidikan Tinggi, Kementerian Pendidikan, Kebudayaan, Riset dan Teknologi atas bantuan dana riset sehingga buku ini bisa disusun dan diterbitkan. Ucapkan terima kasih tidak lupa disampaikan kepada Guru Besar Statistika ITS, Prof. I Nyoman Budiantara, atas saran dan masukkan. Tak lupa juga disampaikan ucapan yang sama kepada Guru Besar Pertanian Universitas Jember, Dr.Sc. Agr. Ir. Didik Sulistyanto, M.Sc. atas masukkan, dan saran yang diberikan demi kesempurnaan isi buku ini. Serta ucapan terima kasih kepada segenap civitas akademika Universitas KH. A. Wahab Hasbullah atas dukungan pada setiap proses penyusunan buku ini hingga selesai. Tak lupa dukungan do'a dan moril dari orang tua, saudara dan tim riset yang penuh semangat hingga buku ini sukses terbit. Semoga buku ini memberikan manfaat bagi semua pihak dan sebagai sumbangsih penulis untuk mendHarmabaktikan ilmunya bagi kemaslahatan umat dalam rangka mewujudkan riset Indonesia unggul dikancah internasional.

Jombang, Oktober 2021

Penulis:
Zulfikar, SP. M.Si
Munawarah, S.Kom, M.Si
Ambar Susanti, SP. MP.

KERANGKA KONSEP BUKU

Agar memudahkan pembaca dalam memahami isi buku ini, maka dibangun kerangka konsep yang berisikan alur dari materi yang akan disampaikan. Secara umum materi dalam buku ini membahas statistika dan program R, selanjutnya hubungan dari dua meteri dijabarkan dalam bab-bab yang bermuara pada aplikasinya di riset sains terapan. Terdapat 12 bab dimana masing-masing bab terkandung materi saling terkait dalam bentuk alur seperti yang ditunjukkan pada gambar 1.



Gambar 1. Kerangkan Konsep Buku

KERANGKAN OPERASIONAL BUKU

Pemaparan materi buku dari masing-masing bab selanjutnya akan dijelaskan materi pokoknya, sehingga pembaca memiliki gambaran awal dari ringkasan materi yang dibahas dalam buku ini. Kerangka operasional akan menjelaskan ringkasan materi dari masing-masing bab, dimulai dari bab awal sampai bab terakhir. Untuk tujuan ini maka dirancang kerangkan operasional buku seperti dijabarkan pada gambar 2.

Gambar 2. Kerangka Operasional Buku

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
STATISTICAL COMPUTING

1.1 Program R untuk Statistical Computing

Statistical Computing adalah bagian dari ilmu komputer yang memfokuskan diri untuk mengimplementasi metode statistika secara komputasi pada komputer. Bidang ini menjadi semacam penghubung antara ilmu statistika dan ilmu komputer. Beberapa contoh yang sering digunakan adalah algoritma genetika untuk optimasi, *principal component analysis* untuk klasifikasi, *discrete event simulation* untuk riset operasional dan lain sebagainya.

Perangkat lunak untuk *Statistical Computing* ini secara umum terbagi menjadi dua kelompok yaitu: kelompok perangkat lunak komersil dan kelompok *open source/freeware*. Kedua kelompok ini biasanya mendukung penggunaan secara *menu-driven* ataupun *line-command*. Beberapa contoh perangkat lunak statistik komersil yang popular di Indonesia adalah SPSS, MINITAB, STATA, SAS, dan Splus. Sedangkan contoh dari kelompok *open source/freeware* antara lain R, Vista, SalStat, PSPP, dan lain-lain. Dari semua alternatif yang ada, hanya R yang dapat dikategorikan bahasa pemrograman yang memenuhi spesifikasi untuk rekayasa perangkat lunak juga. Bahasa R ini telah masuk ke dalam TIOBE index (Februari 2012) pada posisi 20.

Gambar 3. Logo Program R

Proses komputasi statistik dapat dilakukan baik secara manual ataupun dengan menggunakan *statistical package*. Jika keduanya dibandingkan, komputasi dengan *stastical package* cenderung lebih cepat sehingga dapat menghemat waktu. Diantara beberapa *statistical package* yang ada, SPSS, Minitab dan SAS merupakan beberapa *statistical package* yang paling banyak digunakan. Hanya saja, banyak diantara kita yang tidak menggunakan perangkat lunak legal, sehinga cenderung menggunakan *statistical package* bajakan yang dapat melanggar hak cipta. Salah satu alasan mengapa hal ini terjadi adalah karena para pengguna *statistical package* tersebut merasa bahwa harga dari perangkat lunak yang ada tidak dapat dijangkau.

Sebagai salah satu alternatif dari *statistical package* berbayar, dewasa ini berkembang beberapa *open source statistical package* yang memiliki kemampuan yang sama dengan *statistical package* di atas. Karena *open source*, setiap calon pengguna dapat mengunduhnya secara gratis dan legal.Salah satu *open source statistical package* yang cukup popular adalah "R". R pertama kali dikembangkan oleh Ross Ihakaand Robert Gentleman di University of Auckland, Selandia Baru. Nama R sendiri diambil dari huruf pertama dari nama depan penemu R software (Hornik, 2016). Pada awalnya, R sendiri diciptakan oleh Ross Ihaka dan Robert Gentleman pada tahun 1995 sebagai implementasi dari bahasa program S. Tujuannya untuk mengembangkan bahasa yang fokus pada analisis data, statistik, dan model grafis.Selain gratis, R memiliki beberapa kelebihan seperti mampu membuat *object*, *function*, dan *packages* (Ulrich, 2010). R dapat memiliki kemampuan tersebut karena selain sebagai sebuah perangkat lunak, R juga merupakan sebuah bahasa pemrograman.

Dalam dunia teknlogi ada beragam jenis software bahasa pemrograman, salah satunya adalah R. Dikalangan para *data scientist* mungkin nama R sudah tidak asing lagi di dengar bahkan digunakan dalam menunjang pekerjaan sehari-hari. Selain sebagai software paket statistika, R juga merupakan lingkungan sistem pemrograman yang cukup lengkap. Artinya, R dapat digunakan sebagai alat untuk memecahkan berbagai masalah melalui pemrograman. Tentu saja masalah yang lebih tepat dipecahkan dengan R adalah yang terkait dengan analisis data dalam konteks statistika.

Para developer R menyadari bahwa analisis data kuantitatif melalui statistika memerlukan beberapa tahapan yang mana setiap tahapan dibutuhkan alat bantu untuk 'berbicara' dengan lingkungan disekitarnya. Secara singkat tahapan ini meliputi 1) persiapan data, 2) pemilihan metode/teknik, 3) eksekusi dan 4) penyajian informasi. Sebagai contoh, pada tahap persiapan data, alat bantu yang dibutuhkan bagi seorang statisi adalah kemudahan dalam menghubungkan *raw data* (*database*, *file* terstruktur, dsb) dengan lingkungan R. Solusi yang diberikan oleh developer dan kontributor R terkait dengan hal ini diantaranya adalah membuat library ODBC, JDBC, CSV dll.

Selain kemudahan, konsep yang sering digunakan oleh para programer adalah otomatisasi dan integrasi proses. Artinya dalam sistem yang kompleks, lingkungan R harus bisa menyediakan *interface* bahasa yang berfungsi sebagai penghubung antar fungsi-fungsi dalam

sistem tersebut. Dengan latar belakang itulah developer dan kontributor R membuat berbagai interface bahasa pemrograman yang berada diluar R seperti C, Java dan Tcl/Tk agar bisa memudahkan proses otomatisasi dan integrasi.

Dahulu, R hanya digunakan oleh para akademisi, namun lama-kelamaan R juga banyak digunakan oleh para praktisi di dunia bisnis. Hal inilah yang membuat R menjadi sangat terkenal di seluruh dunia.Salah satu keunggulan R adalah komunitas besar yang tergabung dalam satu mailing-list, dokumentasi para pengguna yang mudah diakses, grup Stack Overflow yang sangat aktif, dan koleksi packages R yang dibagikan oleh sesama pengguna.Di masa sekarang, R biasanya lebih banyak digunakan untuk analisis data yang dikerjakan pada server pribadi. R dapat difungsikan untuk pekerjaan eksplorasi hampir semua jenis data karena banyaknya jenis packages, test, dan tools yang dengan mudah bisa diadaptasi.Penggunaan rumus-rumus rumit dalam R juga mudah diatur. Pada penggunaan R, langkah pertama yang harus dilakukan adalah mengunduh RStudio IDE (*Integrated Development Environments*).

R sangat baik dan mudah digunakan dalam visualisasi data. Terdapat banyaki packages yang mendukung R untuk membangun visualisasi menarik, seperti GGPLOT2 untuk membuat grafik, lattice untuk menampilkan hubungan varibel, dan rCharts untuk menerbitkan visualisasi Java Script dengan R.R juga dibangun dalam ekosistem yang baik sehingga memudahkan penggunanya untuk menemukan packages dalam cran, bioconductor, dan github. R pun dibangun oleh statistisi untuk statistisi, sehingga siapapun yang tidak memiliki keahlian programming dalam dengan mudah beradaptasi dengan R.Namun, di balik semua itu, R bersifat lambat. Hal ini disebabkan oleh kondisi yang tidak jelas dimana R merupakan bahasa dan implementasi dari bahasa tersebut. Bahasa R juga mempunyai definisi abstrak, yang mengartikan maksud kode R dan bagaimana kode tersebut bekerja.Lalu cara mudah bagi pemula dalam menggunkan software ini adalah langkah pertama dengan meninstall software R, kemudian install R studio untuk melakukan coding dengan lebih mudah.

Setelah selesai melakukan proses coding, kita bisa menyimpan value di sebuah variable, namun perlu diperhatikan bahwa data yang tersimpan harus ada angka numeric, character, dates dan logical (boolean). Kita juga bisa mengelompokan tipe data yang sama melalui vectors dan functions untuk memudahkan memproses data. Data-data tersebut juga bisa diisi pada kolom data frame sejenis excel agar mudah

dibaca.Setelahnya bisa langsung membuat list container data yang berbeda, melalui matrix berisi index colomn dan row. Hampir mirip data frame, meskipun representai dan rulesnya jelas berbeda. Point point diatas harus dikuasai terlebih dahulu bagi para pemula sehingga nantinya akan mudah untuk programming yang automate.

1.2 Bahasa R

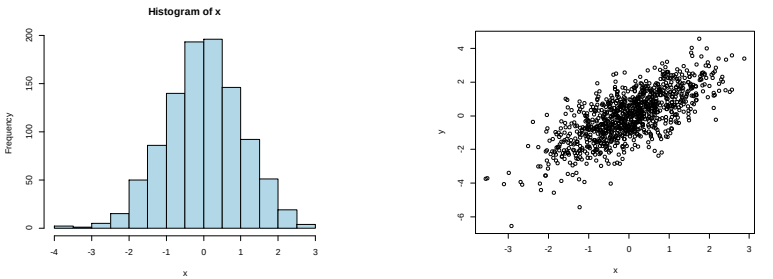
Bahasa R adalah suatu fasilitas perangkat lunak terpadu untuk manipulasi data, simulasi kalkulasi dan peragaan grafik. R memiliki kemampuan menganalisis data dengan sangat efektif dan dilengkapi dengan operator pengolahan array dan matriks. Tidak kalah menariknya R memiliki kemampuan penampilan grafik yang sangat canggih untuk peragaan datanya. Berikut adalah contoh code dalam bahasa R:

```
> x <- rnorm(1000)
> y <- rnorm(1000) + x
> summary(y)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-6.52621 -0.93644  0.05626 -0.00510  0.98113  4.56082
> var(y)
[1] 2.171591
> hist(x, col="lightblue")
```

Hasil running program R diperoleh histogram seperti ditunjukkan pada gambar 1.2

```
> plot(x,y)
```

Running program Runtuk fungsi plot (x,y) didapatkan out put sebagai berikut:



a b

Gambar 4. Tampilan hasil running code R dalam bentuk histogram (a) dan grafik scatter plot (b)

Bahasa R merupakan versi *open-source* dari bahasa pemrograman S (Azola dan Harrel, 2006). Bahasa R dapat diperoleh secara gratis dan jika berminat tinggal diunduh di http://cran.r-project.org. Versi komersial yang berbasis bahasa S adalah S plus. Bahasa R memiliki kemampuan yang tidak kalah dangan paket program pengolahan data komersial bahkan dalam beberapa hal kemampuannya lebih baik. Bahasa R mendapat sambutan yang baik dari kalangan statistikawan di seluruh dunia dan komunitas R sangat aktif dalam memberikan kontribusi paket aplikasi untuk R. Keunikan bahasa R adalah langsung terhubung dengan paket aplikasi yang dibangun oleh statistikawan di seluruh dunia ini dan jika membutuhkan dapat langsung diinstal dengan mencari paket yang sesuai.

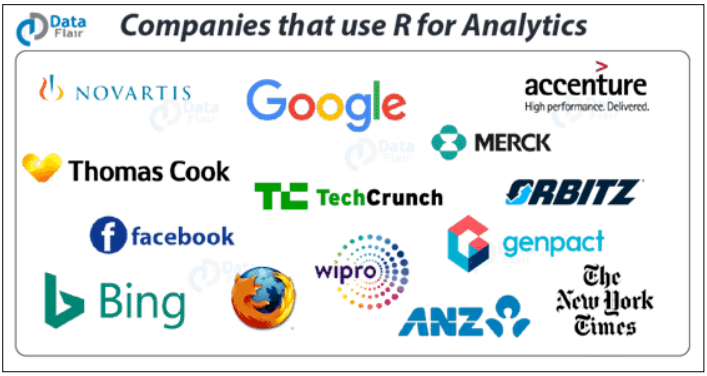
1.3 Sejarah Program R

Pada tahun 1976, John Chambers dan rekannya mengembangkan R di Bell Laboratories. Pada dasarnya, bahasa pemrograman R adalah implementasi dari bahasa pemrograman S. Kemudian, R menggabungkan bahasa pemrograman S dengan semantik pelingkupan leksikal yang terinspirasi dari *Scheme*.Namun, R diberi nama sesuai dengan nama depan dua penulis pertama bahasa pemrograman R, yaitu Ross Ihaka dan Robert Gentleman di University of Auckland, Selandia Baru. Selain itu, proyek R disusun pada tahun 1992**,** dengan versi awal yang dirilis pada tahun 1995 dan versi beta pada tahun 2000**.**

Pemrograman R merupakan bahasa pemrograman yang *open source*, sehingga bahasa pemrograman R sering diperbaharui sesuai kasus yang dibutuhkan. Tentu tak heran jika R merupakan aplikasi sistem statistik yang kaya. Hal ini disebabkan banyak sekali *packages* yang dikembangkan oleh *developers* dan komunitas untuk keperluan analisa statistik.R project pertama kali dikembangkan oleh Robert Gentleman dan Ross Ihaka (nama R untuk sofware ini berasal dari huruf pertama nama kedua orang tersebut) yang bekerja di departemen statistik Universitas Auckland tahun 1995. Sejak saat itu software ini mendapat sambutan yang luar biasa dari kalangan statistikawan, industrial engineering, peneliti, programmer dan sebagainya. Pada saat ini, source code kernel R dikembangkan terutama oleh R Core Team yang beranggotakan 17 orang statistisi dari berbagai penjuru dunia.

R Basics merupakan salah satu fasilitas R yang sedang populer dikalangan akademisi maupun praktisi didunia. Banyak perusahaan

besar yang telah menggunakan pemrograman R bagi kesuksesan perusahaan, diantaranya Google, facebook, NOVARTIS, Thomas Cook, Bing, TechCrunch, wipro, mozilla, ANZ, accenture, MERCK, ORBITZ, genpact, The New York Times, dan sebagainya.

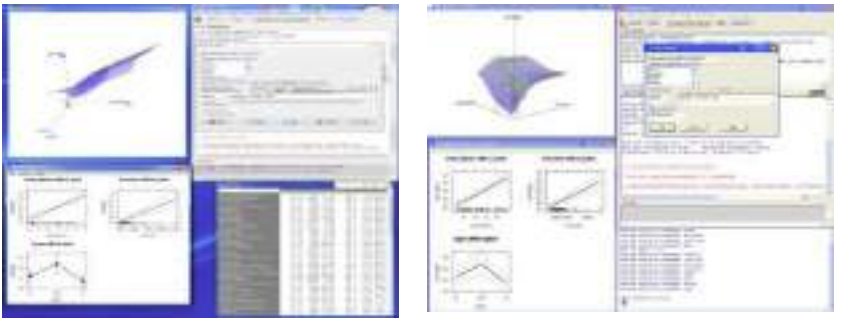


Gambar 5. Perusahaan Besar yang menggunakan Program R
Sumber: https://data-flair.training/blogs/r-applications/ (19 Des 2019)

Paket statistik R bersifat multiplatforms, dengan file instalasi binary/file tar tersedia untuk sistem operasi Windows, Mac OS, Mac OS X, Linux, Free BSD, NetBSD, irix, Solaris, AIX, dan HPUX. Fungsi dan kemampuan dari R sebagian besar dapat diperoleh melalui *Add-on packages/library*. Suatu *library* adalah kumpulan perintah atau fungsi yang dapat digunakan untuk melakukan analisis tertentu. Sebagai contoh library yang sangat powerful adalah *R-commander* dan *Rattle*. Meskipun R mengutamakan penggunaan bahasa pemrograman, bagi pengguna awam dengan metode statistik dan bahasa pemrograman, dapat memanfaatkan paket R-commander yang telah disediakan di *library*.

Dengan mengaktifkan paket *R-Commander*, pengguna dapat melakukan pengolahan data secara statistik dengan mudah, semudah menggunakan SPSS, Minitab ataupun software statistik berlisensi lainnya. Hal ini sangat dimungkinkan karena melalui R-Commander, pengguna bisa langsung melakukan pengolahan data dengan memilih menu-menu yang disediakan pada jendela *R-Commander*. Berikut adalah *screenshoot* dari R Commander.

(http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/):



Gambar 6. Tampilan R-Commander dalam program R

R Commander adalah antarmuka pengguna grafis (GUI) untuk perangkat lunak statistik open-source R yang gratis. R Commander diimplementasikan sebagai paket R, paket Rcmdr, yang tersedia secara bebas di CRAN (arsip paket R). Untuk informasi tentang R Commander GUI, lihat John Fox, Menggunakan R Commander (Chapman & Hall / CRC Press, 2017) dan buku pengantar yang didistribusikan bersama paket (dapat diakses melalui Bantuan -> Pengantar menu R.Commander). Versi sebelumnya dari R Commander dijelaskan dalam sebuah makalah di *Journal of Statistics Software* (yang sekarang ketinggalan zaman). Untuk menginstal paket Rcmdr, setelah menginstal R, lihat catatan instalasi R Commander, yang memberikan informasi spesifik untuk pengguna Windows, macOS, dan Linux / Unix.

Menurut Rexer's Annual Data Miner Survey 2010, R telah menjadi alat data mining yang digunakan oleh mayoritas pengguna (43%). Salah satu penyebabnya adalah adanya Rattle, suatu library yang khusus digunakan untuk Data Mining melalui GUI (*Graphic User Interface*). Rattle (the *R Analytical Tool To Learn Easily*) dapat menyajikan ringkasan data secara statistik dan secara visual dari berbagai sumber data (Excel, SQL, XML dll), selanjutnya dapat mentransformasi data ke dalam bentuk yang siap untuk dimodelkan. Untuk permodelannya dapat digunakan berbagai metode baik *supervised* maupun *unsupervised* dan sekaligus mampu membuat laporan secara grafis untuk unjuk kerja model yang dibangun.

R adalah sebuah program komputasi statistika dan grafis (R Core Team 2021). Saat ini R sudah dikenal luas sebagai salah satu *powerful software* untuk analisis data dan *Data Science*. Tentu saja selain R masih banyak *software* lain yang juga sering digunakan untuk analisis data, misalnya Python. R dibuat dengan tujuan awal untuk komputasi statistika dan grafis. Awalnya digunakan oleh para ilmuwan dalam riset

mereka dan para akademisi. Namun seiring perkembangan teknologi, cakupan kemampuan R sebagai bahasa pemrograman menjadi jauh lebih luas. Anda dapat membuat dan *update report* rutin menggunakan R Markdown. Anda juga dapat membuat aplikasi web interaktif atau dashboard dengan package shiny. Karena R didesain untuk analisis data dan perkembangan serta kemampuannya mencakup hampir semua lini dalam analisis data, tidak heran saat ini banyak analis data dan ilmuwan data (*data scientist*) menggunakan R untuk menyelesaikan berbagai masalah mereka.

Berikut ini beberapa kemampuan R.

a. **Gratis dan *Open Source***

Istilah *open source* merujuk kepada sesuatu yang bisa dimodifikasi dan dibagikan. *Open Source Software* (OSS) sendiri berarti *software* yang *source code*-nya dapat diperiksa, dimodifikasi, ditambahkan dan dibagikan oleh siapapun.

b. Tersedia banyak package

Karena R adalah *open source software*, hampir semua package yang ada pun dapat digunakan secara bebas. Package adalah kumpulan suatu script yang umumnya berupa function atau data yang dapat digunakan untuk kebutuhan tertentu.

c. Dibuat oleh statistisi untuk data analyst/data scientist

R adalah sebuah program yang awalnya dibuat untuk kebutuhan statistisi. Oleh karena itu banyak fungsi-fungsi dasar untuk statistika maupun eksplorasi data dan grafis sederhana sudah terdapat di R meskipun tanpa install package tambahan. Namun saat ini R sudah menjadi salah satu software yang digunakan dalam data science karena banyaknya package yang dapat mendukung.

d. Mudah dalam melakukan transformasi dan pemrosesan data

Karena R adalah program untuk analisis data, maka kemampuan R dalam transformasi data seperti penyiapan data, import dan export data dalam berbagai format, dan lain-lain.

e. Mampu menghasilkan grafik yang sangat bagus

Salah satu keunggulan yang dimiliki oleh R adalah kemampuannya untuk menghasilkan grafik yang sangat bagus. Salah satu yang diunggulkan adalah package {ggplot2}. Tentu saja masih banyak package untuk visualisasi selain {ggplot2}

f. **Membuat *Reproducible report***

Ketika Anda mempunyai pekerjaan untuk membuat laporan secara rutin, maka Anda dapat menggunakan R sebagai robot Anda. Dengan

package {*rmarkdown*} Anda dapat membuat laporan rutin dengan hanya satu baris perintah.

g. Dapat membuat aplikasi interaktif/dashboard berbasis web
Package {shiny} (dan semua pengembangannya) dan {*flexdashboard*} dapat Anda gunakan untuk membuat visualisasi interaktif ataupun sebagai sebuah produk dari *data science*.

h. Membuat REST API
Setelah membuat fungsi atau model prediktif dan ingin digunakan secara lebih luas, Selanjutnya dapat dibuat sebagai API menggunakan *package* {*plumber*}.Dan masih banyak lagi kemampuan R yang dapat dimanfaatkan untuk mendukung dan memudahkan pekerjaan kita dalam hal analisis data ataupun *data science*.

R benar-benar luar biasa. 12.000+ paket pemrograman statistik dan non-statistik telah dikembangkan untuk R. Banyak yang bagus. Mereka juga gratis. Kekuatan R untuk keperluan statistik luar biasa. Alat grafik saja luar biasa. R memiliki sesuatu yang disebut R-Studio. R-Studio juga gratis dan memukau. Kombinasi R dan R-Studio memungkinkan non-programmer untuk melakukan hampir semua analisis statistik…(– Peter Schaeffer)

Tabel 1. Perbandingan Harga Aplikasi olah data (Update 29 Maret 2020)

R	Stata	SPSS	AMOS	Eviews
Gratis	1,6 Juta	60 Juta	128 Juta	28 Juta

Sumber:https://ujistatistikhalal.com/blog/aplikasi-r-untuk-olah-data/

Program ini mampu mengolah data seperti di aplikasi SPSS, AMOS, eViews, Lisrel, atau STATA. Di luar negeri, program R / R Studio ini sudah sangat terkenal, terbukti:

a. Pemrograman R jadi bahasa terpopuler secara internasional pada tahun 2017 (sumber: IEEE Spectrum rank languages). Perlu diketahui bahwa IEEE adalah jurnal internasional reputasi sangat tinggi. Jika akademisi mau kirim jurnal ke sana sangat sulit diterima.
b. Punya forum internasional dan nasional (Indonesia)

Orang luar negeri beralih dari aplikasi berbayar ke R/R Studio karena mampu menghemat pengeluaran yang cukup fantastis. Karena perbandingan harga aplikasi olah data inilah mereka serius menguasai R / R Studio. Ada yang menggunakan program R/R Studio untuk statistika, ada pula yang non-statistika (sesuai kebutuhan).Namun program R juga

tidak terlepas dari adanya kekurangan, jasa olah data USH Indonesia telah membuat ringkasan:

1. Script/syntax kadang error tidak terduga
2. Butuh kesabaran untuk belajar
3. Di awal akan terasa tidak mudah

1.4 Download Program R

R dapat diperoleh secara gratis di CRAN-archive yaitu *The Comprehensice R Archive Network* di alamat http://cran.r-project.org. Pada server CRAN ini dapat didownload file instalasi binary dan source code dari R-base system dalam sistem operasi Windows (semua versi), beberapa jenis distro, linux, dan Macintosh.

Fungsi dan kemampuan dari R sebagian besar dapat diperoleh melalui *Add-on packages/library*. Suatu library adalah kumpulan perintah atau fungsi yang dapat digunakan untuk melakukan analisis tertentu. Sebagai contoh, fungsi untuk melakukan analisis time series dapat diperoleh di *library ts*. Instalasi standar dari R akan memuat berbagai library dasar, antara lain *base, datasets, graphics, utils,* dan*stats*. Library lain hasil kontribusi dari pengguna R (di luar yang standar) harus diinstal satu per satu sesuai dengan yang dibutuhkan untuk analisis. Daftar semua library yang tersedia dapat diakses dari link download CRAN di alamat http://cran.r-project.org.

1.4.1 Instalasi Program R

Instalasi program R ke dalam perangkat komputer dilakukan setelah tahapan download berhasil secara lengkap. Langkah-langkah tersebut seperti dijelaskan sebagai berikut:

a. Setelah download file **R-3.6.1-win.exe**, Klik dua kali (*double click*) file **R-3.6.1-win.exe**. maka akan muncul tampilan sebagai berikut:

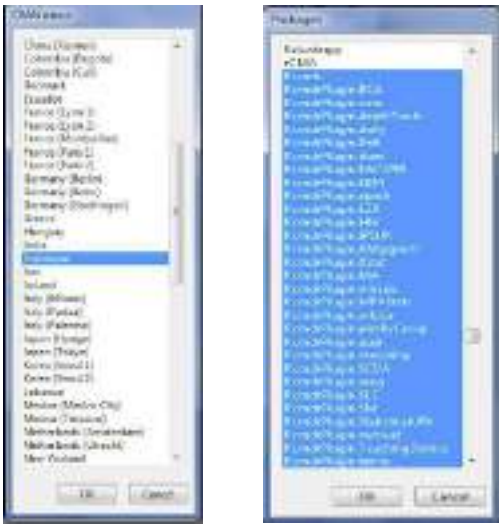
Gambar 7. Tampilan Setup Wizard hasil download program R-3.6.1 win.exe berbasis Windows.

b. Setelah itu, lanjutkan jalannya proses instalasi dengan mengikuti *Wizard* dan menggunakan pilihan-pilihan **default instalasi**.
c. Langkah terakhir jika instalasi R telah selesai adalah melakukan pengecekan atau pengujian apakah program R dapat berjalan dengan baik. Lakukan klik dua kali pada **shortcut R** di **Desktop** atau pada **Start Menu**. Jika instalasi berlangsung dengan baik, maka jendela program R akan terbuka seperti yang terlihat pada Gambar 1.6.

Gambar 8. Tampilan RGUI

Setelah instalasi selesai kita belum bisa melakukan analisis, walaupun bisa tapi pakai syntax sendiri, jadi disini kita mendownload lagi untuk *library/package* yang khusus untuk analisis statistik. Yang ane ketahui masih menggunakan *R Commander*. Cara instalnya bisa langsung dari R, jadi tidak perlu lagi buka websitenya. Langkah-langkahnya sebagai berikut:

a. Klik **packages**, terus pilih **install package(s)**. pertama akan muncul pilihan wilayah **download**, pilih saja Indonesia. Tampilannya seperti terlihat pada gambar 1.7.



(a) (b)

Gambar 9. Tampilan CRAN minor (a) dan Packeges dengan pemilihan paket yang diinginkan (b).

b. Kemudian muncul pilihan **packages** yang akan diinstal seperti terlihat pada gambar 1.8
c. Pilih **rcmdr** dan **plugin-**nya. Silahkan pilih pluginnya sesuai kebutuhan. Berikut sedikit gambar **plugin** tersebut.
d. Setelah semua selesai, kita coba buka R commander. Caranya packages terus pilih load packages kemuadian pilih rcmdr. Sesuai gambar 10 berikut:

Gambar 10. Tampilan load pakage dengan memilih Rcmdr untuk memunculkan R. Commander

e. Apabila hasilnya seperti ini maka install sudah sukses.

Gambar 11. Tampilan antar muka R.Commander

f. Apabila masih belum bisa dan ada tulisan missing maka coba install yang missing tersebut dengan cara yang diatas,cari sesuai nama yang hilang. Kalau belum bisa restart computer. Lalu dibuka kembali.

Setelah proses instalasi program R khususnya R commander selesai dengan sempurna, selanjutnya melalui antarmuka R commander ini bisa dilakukan analisis statistik.

1.5 Ringkasan

Program R merupakan software open source yang bebas diakses siapa saja, aplikasi ini mampu menstransfer dari data yang berasal dari aplikasi statsitika lain sehingga sangat fleksibel. Kekuatan R untuk keperluan statistik luar biasa baik alat grafik maupun coding yang digunakan cukup mudah bahkan selain bisa menyusun coding juga tersedia aplikasi statistika bawaan R. R memiliki sesuatu yang disebut R-Studio. R-Studio juga gratis dan memukau. Kombinasi R dan R-Studio memungkinkan non-programmer untuk melakukan hampir semua analisis statistic.

Refernsi

Budiharto, W dan Ro'fah N. R. 2013. Pengantar Praktis pemrograman R untuk Ilmu Komputer. Jakarta: Halaman Moeka Publishing.

Gio, P.U. dan E. Rosmaini, 2015. Belajar Olah Data dengan SPSS, Minitab, R, Microsoft Excel, EViews, LISREL, AMOS, dan SmartPLS. USUpress.

Hornik, K. 2016. R FAQ. Retrieved March 16, 2016, from https://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-is-R-named-R_003f

Ulrich, J. 2010, December 14). Why Use R?. Retrieved from http://www.r-bloggers.com/why-use-r/

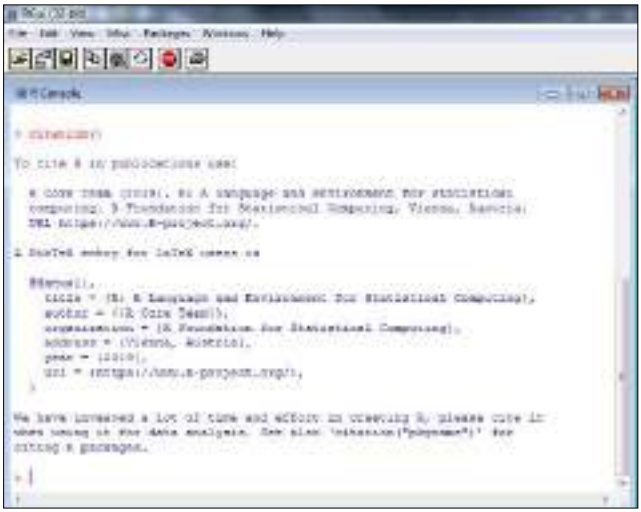
BAB II
OPERASI PROGRAM R

2.1 Memulai R

Setelah proses instalasi berjalan sukses, langkah berikutnya adalah memulai R dengan mengetikkan beberapa fungsi dasar pada R Console. Berikut contoh untuk mengetahui informasi cara citasi menggunakan fungsi citation() yang harus diketikkan sebagai berikut :

```
> citation()
```

Hasilnya sebagai berikut:



Gambar 12. Menuliskan fungsi citation() dan hasil yang diperoleh

Contoh begitu handalnya R ialah, jika kita ingin mengetahui umlah dari 2+400, cukup menulis sebagai berikut:

```
> 2+400
[1] 402
```

jika kita ingin mengetahui nilai dari log(5), cukup tulis:

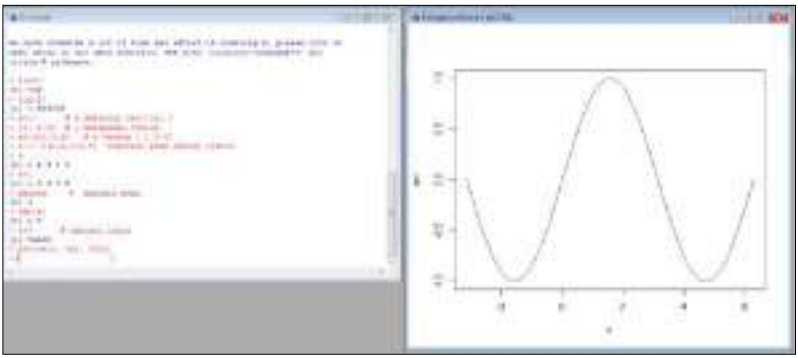
```
> log(5)
[1] 1.609438
```

Untuk memelajari variabel dasar, Anda dapat membayangkan sebuah variabel x, dengan diberi nilai 1, serta menggunakan symbol # sebagai komentar, berikut beberapa fungsi dasar yang penting:

```
> x<-1        # x sekarang bernilai 1
> y<- 1:10    # y merupakan vektor
> z<-c(1,3,5)    # z vector [ 1 3 5]
> a <- c(1,2,3,4,5)   #operasi pada sebuah vektor
> a
[1] 1 2 3 4 5
> a+1
[1] 2 3 4 5 6
> mean(a)     #  mencari mean
[1] 3
> var(a)
[1] 2.5
> 4>4       # operasi logis
[1] FALSE
>
```

Untuk penampilan grafik, dapat dicoba kode yang simpel dan menarik menggunakan fungsi plot berikut :

```
> plot(sin, -pi, 2*pi)
```

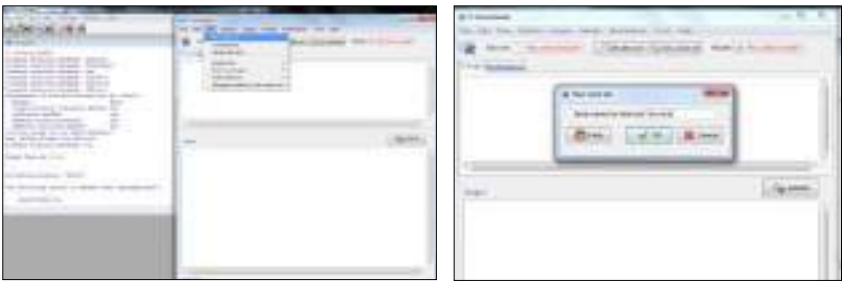


Gambar 13. Penulisan coding pada R. Console dan output pada R.Graphics untuk hasil penggambaran fungsi sinus

Jika sudah selesai bekerja dengan R, dapat mengetikkan q() untuk keluar dari aplikasi.

2.2 Entri Data menggunakan R Commander

Untuk menjalankan R Commander , ketikkan perintah **library(Rcmdr)** pada jendela konsol. Jika proses berjalan sukses maka akan nampak jendela R-Commander. Pengisian data secara langsung dengan R dengan menggunakan R-commander dapat dilakukan melalui menu Data, dan pilih New dataset ... . Setelah itu berna nama **Ujicoba1** seperti gambar 2.3.a



(a) (b)

Gambar 14. Membuat data set baru (a), pemberian nama data set (Ujicoba1)(b)

Kemudian klik OK, maka akan terbuka jendela Data Editor. Pengisian nama variabel dilakukan dengan cara klik pada kolom paling atas dari data editor. Sebagai contoh, masukkan data percobaan sebagai berikut:

Tabel 2. Data untuk Ujicoba1

Peserta	Statistika	Komputer	Matematika
Amir	75	80	70
Ani	85	80	85
Nita	80	85	85
Munir	80	85	85
Zahra	90	90	85
Yahya	85	85	90

Berikut ini hasil dari entri data dari table di atas, Untuk melihat hasil entri data maka klik view data set.

(a) (b)

Gambar 15. Hasil entri data (a), tampilan bila melihat data kembali dari view data set (b)

Untuk melakukan editing terhadap data Ujicoba1, dilakukan dengan mengklik tombol Edit data set. Setelah itu jendela Data Editor akan dibuka kembali, proses editing data dapat langsung dilakukan pada data yang ingin dirubah.

2.3 Fitur Statistika pada R Commader

Ada beberapa fitur Statistika yang memudahkan kita menganalisa data seperti Summary, means, variances dan fit models seperti gambar berikut :

Gambar 16. Beberapa fitur Statistika pada R Commander

Misalnya informasi summary dari data yang kita entri menggunakan fungsi summary seperti gambar di bawah ini:

2.4 Menggunakan Graph di R Commander

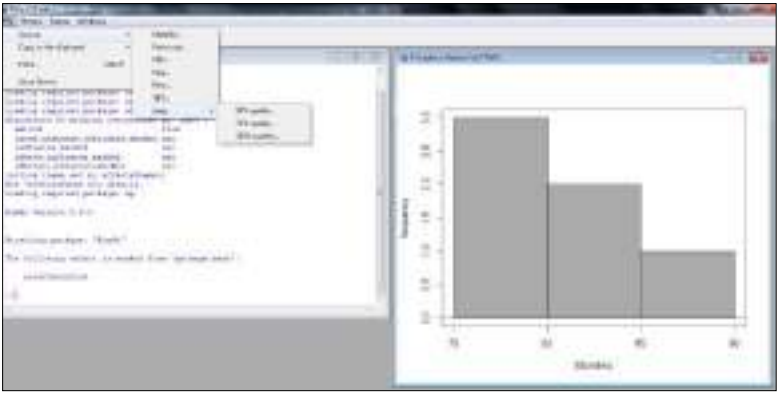
Data yang berhasil dientri atau diimport dari aplikasi lain selayaknya divisualisasikan pada grafik untuk analisa.Berbagai model plot dapat dihasilkan R, seperti Histogram, Boxplot, Scaterpot dan Pie chart sebagai contoh pada gambar 17 di R Commander:

Gambar 17. Fungsi summary yang diterapkan pada data set

Gambar 18. Berbagai pilihan grafik yang disediakan oleh R Commander

Untuk percobaan membuat grafik histogram menggunakan fitur Graphs di R Commander, buatlah data set bernama Nilai sebagai berikut:

Gambar 19. Tahapan pembuatan grafik di R Commander



Gambar 20. Tahapan penyimpanan output grafik

Setelah grafik histogram terbentuk, lalu gambar tersebut bisa disimpan sesuai dengan format yang diinginkan seperti terlihat pada gambar di atas.

2.5 Membuat Function

Fungsi yang ada dalam program R tersedia dalam bentuk pembuatan loop dan user defined function melalui R.Console.

a. Loop dan Vektorisasi

R memiliki beberapa fitur untuk pemrograman yang mirip dengan bahasa C. Format loop dan pengecekan kondisi pada R adalah :

```
for (name in expr_1) expr_2
if (expr_1) expr_2 else expr_3
```

Misal, jika kita memiliki sebuah vektor x, dan tiap elemen x dengan nilai b, kita ingin memberikan nilai 0 ke variable y, maka programnya sebagai berikut:

```
>y <- numeric(length(x))
for (i in 1:length(x)) if (x[i]
== b) y[i] <- 0 else y[i] <- 1
```

Beberapa instruksi dapat dieksekusi berbarengan jika diletakkan di dalam kurung kurawal :

```
for (i in 1:length(x)) {
y[i] <- 0
...
}
if (x[i] == b) {
y[i] <- 0
...
}
```

Selain itu, kita dapat mengeksekusi instruksi selagi kondisi true menggunakan while :

```
while (myfun > minimum) {
...
}
```

Kita juga dapat menggunakan vektorisasi untuk loop pada pemberian nilai berulang, misalnya:

```
> z <- numeric(length(x))
> for (i in 1:length(z)) z[i] <- x[i] + y[i]
```

b. Membuat Fungsi Sendiri

Membuat fungsi sendiri sangat dibutuhkan pada pemrograman tingkat lanjut R. Biasanya fungsi kita buat agar program utama kita mampu lakukan berbagai instruksi dengan baik pada 1 file R. Format penulisan fungsi adalah sebagai berikut:

```
name <- function(arg_1, arg_2, ...)
```

Sebagai contoh, jika kita ingin membuat fungsi bernama myfun yang membutuhkan 2 parameter untuk plot data, dan melakukan beberapa aksi, dapat digunakan contoh berikut yang dibuat pada script baru dan disimpan dengan nama file **buatfungsi.r.**

```
myfun <- function(S, F)
{
data <- read.table(F)
plot(data$V1, data$V2, type="l")
title(S)
}
```

Lalu memanggil fungsi terebut dapat digunakan perintah di bawah ini:

layout(matrix(1:3, 3, 1))
myfun("swallow", "Swal.dat")
myfun("wren", "Wrenn.dat")
myfun("dunnock", "Dunn.dat")

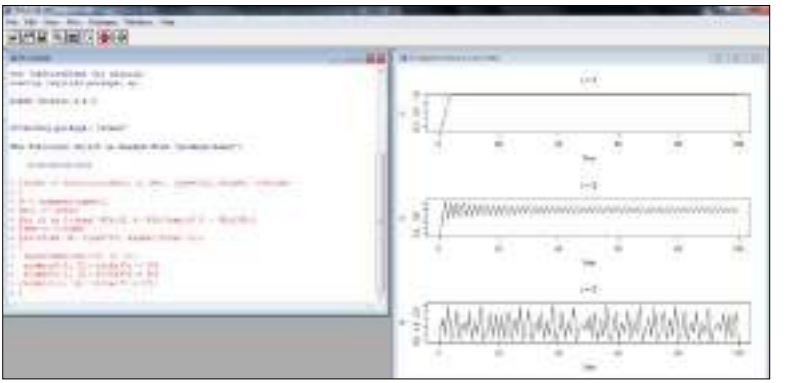
Contoh lainnya dengan parameter fungsi yang sudah kita tentukan sebagai berikut:

```
ricker <- function(nzero, r, K=1, time=100, from=0,
to=time)
{
N <- numeric(time+1)
N[1] <- nzero
for (i in 1:time) N[i+1] <- N[i]*exp(r*(1 - N[i]/K))
Time <- 0:time
plot(Time, N, type="l", xlim=c(from, to))
```

Lalu jalankan fungsi diatas dengan memanggilnya sebagai berikut:

```
> layout(matrix(1:3, 3, 1))
> ricker(0.1, 1); title("r = 1")
> ricker(0.1, 2); title("r = 2")
> ricker(0.1, 3); title("r = 3")
```

Setelah dilakuakan running program didapatkan output seperti yang ditunjukkan pada gambar berikut:



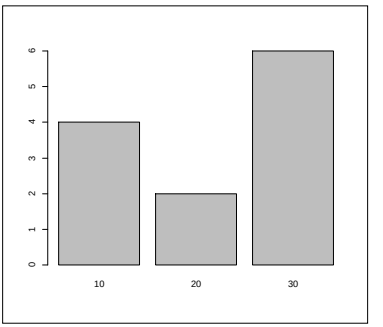
Gambar 21. Output dari fungsi program R.

Dari gambar diatas, terlihat bahwa dengan memiliki kemampuan pemrograman fungsi, program yang dijalankan bs lebih dinamis dan handal.

c. Fungsi Barplot

Fungsi barplot dalam R berfungsi untuk menyajikan data dalam bentuk diagram batang. Misalkan variabel A menyimpan data 10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30. Berikut, akan disajikan data pada variabel A dalam bentuk diagram batang.

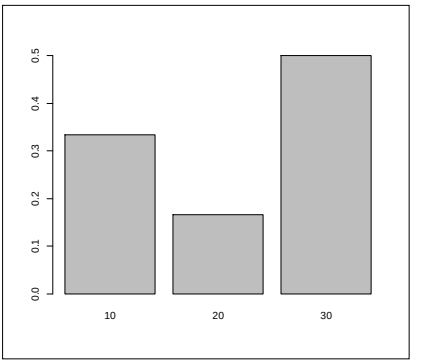
```
A=c(10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30)
barplot(table(A))
```



Gambar 22. Tampilan Ouput dari Fungsi Barplot

Perhatikan bahwa untuk data dengan nilai 10 mempunyai frekuensi sebanyak 4, data dengan nilai 20 mempunyai frekuensi sebanyak 2, dan data dengan nilai 30 mempunyai frekuensi sebanyak 6. Grafik batang di atas dapat diatur agar disajikan secara proporsi.

```
A=c(10, 10, 10, 10, 20, 20, 30, 30, 30, 30, 30, 30)
barplot(table(A)/length(A))
```



Gambar 23. Tampilan Output Barplot Proporsi

Perhatikan bahwa nilai 0,3, 0,2, dan 0,5 masing-masing merupakan proporsi dari nilai 10, 20, dan 30.

d. Fungsi Plot

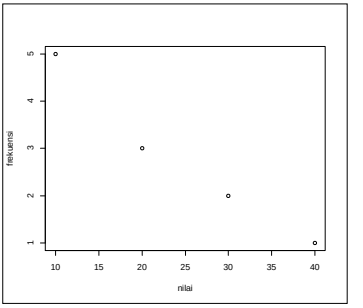
Misalkan variabel bernama A menyimpan data 10,10,10,10,10,20,20,20,30,30,40. Berikut akan digunakan fungsi table untuk mengetahui frekuensi dari masing-masing nilai data.

```
A=c(10,10,10,10,10,20,20,20,30,30,40)
table(A)
A
10 20 30 40
```

```
 5  3  2  1
```

Diketahui nilai 10 muncul sebanyak 5, nilai 20 sebanyak 3, nilai 30 sebanyak 2, dan nilai 40 sebanyak 1. Berikut akan digunakan fungsi plot ( ) untuk memplot data yang tersimpan dalam variabel A.

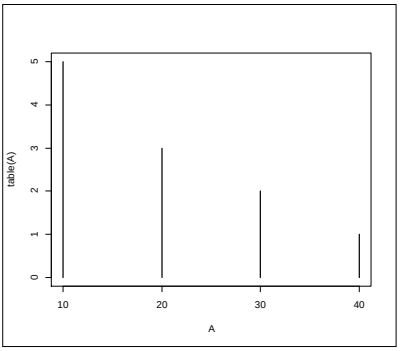
```
nilai=c(10,20,30,40)
frekuensi=c(5,3,2,1)
plot(nilai,frekuensi)
```



Gambar 24. Fungsi Scatter Plot

Alternatif lain untuk menyajikan data.

```
A=c(10,10,10,10,10,20,20,20,30,30,40)
plot(table(A))
```



Gambar 25. Fungsi histogram garis

Ilustrasi dalam R diperlihatkan pada Gambar 23 dan Gambar 24.

2.6 Input data dari file data Excel

Berikut cara memasukkan data di R:

a. Download file data tinggi mahasiswa berikut;

b. Bila file dengan Excel

c. Selanjutnya simpan file tersebut ke dalam format *.csv dengan cara:

- Pilih *menu > file > save as* dan lanjutkan dengan memilih direktori penyimpanan file

- setelah kotak dialoq *save as* muncul pada drop-down *save as type* pilih CSV (*Comma delimited*)
- Simpan dengan nama yang sama lalu klik *save*.

Gambar 26. Tampilan jendela pengambilan data dari Excel yang disimpan dalam format CSV.

d. Buka R

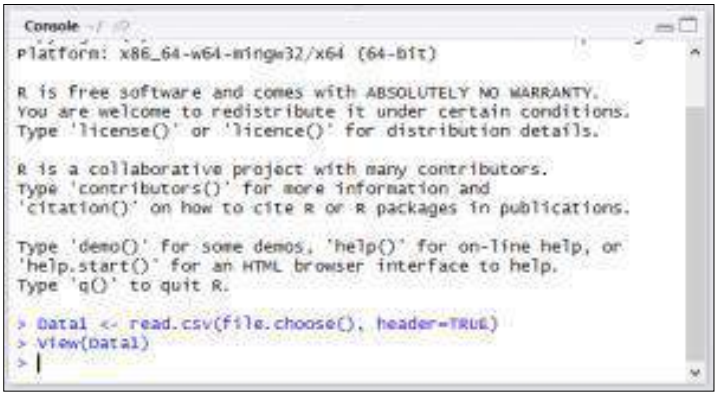
e. Tulis code berikut Data1 <- read.csv(file.choose(), header=TRUE) pada jendela *console* dan tekan *enter*.

Gambar 27. Tampilan jendela R Console

f. Setelah muncul kotak dialoq *select file*, pilih file data tinggi mahasiswa.csv, lalu klik *open*.

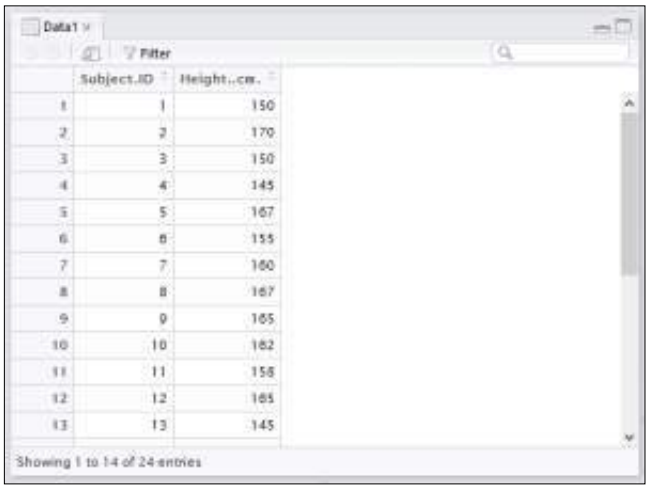
Gambar 28. Tampilan Jendela data Excel tersimpan

g. Untuk melihat data yang diinputkan tulis code View(Data1) pada jendela Console.



Gambar 29. Tampilan Jendela Console dari proses untuk melihat data

h. Apabila data berhasil dimasukkan, data tersebut akan terlihat di jendela *view file and data*.



Data1

Filter

	Subject.ID	Height..cm.
1	1	150
2	2	170
3	3	150
4	4	145
5	5	167
6	6	155
7	7	160
8	8	167
9	9	165
10	10	162
11	11	158
12	12	165
13	13	145

Showing 1 to 14 of 24 entries

Gambar 30. Tampilan data

Penjelasan dari kode **Data1 <- read.csv(file.choose(), header=TRUE)**:

1. **Data1** merupakan identifier (nama) dari data yang anda masukan. Saudara dapat menggantinya dengan nama lain sesuai keinginan saudara.
2. **read.csv** berarti file yang akan dibuka berformat csv.
3. **file.choose()** berarti file yang akan dibuka ditentukan dengan cara memilih
4. **Header=TRUE** berarti baris pertama pada file csv tersebut akan digunakan sebagai *header*.

Referensi

Budiharto, W dan Ro'fah N. R. (2013). Pengantar Praktis pemrograman R untuk Ilmu Komputer. Jakarta: Halaman Moeka Publishing.

Gio, P.U. dan E. Rosmaini, 2015. Belajar Olah Data dengan SPSS, Minitab, R, Microsoft Excel, EViews, LISREL, AMOS, dan SmartPLS. USUpress.

Hornik, K. (2016). R FAQ. Retrieved March 16, 2016, from https://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-is-R-named-R_003f

Ulrich, J. (2010, December 14). Why Use R?. Retrieved from http://www.r-bloggers.com/why-use-r/

BAB III
STATISTIKA R

3. 1 Pengantar

Program R menyediakan banyak menu metode analisis statistika, mulai dari statistika deskriptif, statistika parametric sampai nonparametrik. Selain aplikasi statistika yang siap pakai, program R juga menyediakan pemrograman koding sehingga memberikan keleluasaan bagi para ilmuan untuk membangun metode statistic yang ingin digunakan. Tampilan hasil analisis dari program R cukup menarik dan mampu menampilkan visualisasi data yang mampu memnjelaskna interetasi dan makna data hasil analisis. Program R memiliki sistem yang mampu menyediakan paket baru secara online bila ingin update paket maupun menambah paket lewat aplikasi Cloud. Pada bab ini akan dibahas beberapa metode statistic dasar dengan program R serta tahapan analisis yang dirancang secara runtut yang dilengkapi dengan visual jendela menu yang ada di program R.

3.2. Analisis RMSE

3.2.1. Cara Menghitung *Root Mean Square Error* (RMSE) di R

Root Mean Square Error (RMSE) dalam R memungkinkan kita untuk mengukur seberapa jauh nilai prediksi dari nilai yang diamati dalam analisis regresi.Dengan kata lain, seberapa terkonsentrasi data di sekitar garis yang paling sesuai.

$$RMSE = \sqrt{\left[\frac{\sum (P_i - O_i)^2}{n}\right]}$$

di mana:

$\sum$ = simbol menunjukkan “jumlah”

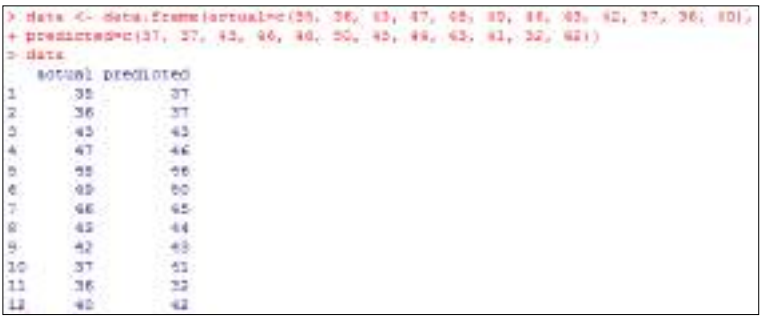
P_i = adalah nilai prediksi untuk pengamatan ke-i dalam dataset

O_i = adalah nilai yang diamati untuk pengamatan ke-i dalam kumpulan data

n = adalah ukuran sampel

Metode 1: Fungsi

Mari buat bingkai data dengan nilai prediksi dan nilai yang diamati.



```
> data <- data.frame(actual=c(35, 36, 43, 47, 46, 60, 66, 44, 42, 37, 36, 40),
+ predicted=c(37, 37, 43, 46, 46, 60, 65, 44, 43, 41, 32, 42))
> data
   actual predicted
1      35        37
2      36        37
3      43        43
4      47        46
5      46        46
6      60        60
7      66        65
8      44        44
9      42        43
10     37        41
11     36        32
12     40        42
```

Kami akan membuat fungsi kami sendiri untuk perhitungan RMSE:

```
> sqrt(mean((data$actual - data$predicted)^2))
[1] 2.041241
```

Nilai error akar rata-rata kuadrat (RMSE) adalah 2.041241.

Metode 2: Paket

rmse() fungsi yang tersedia dari paket Metrik, Mari kita gunakan yang sama.

rmse (aktual, diprediksi)

```
library (Metric)
rmse(data$actual, data$predicted)
2.041241
```

Nilai RMSEnyat adalah 2.041241.

Kesimpulan

Kesalahan kuadrat rata-rata adalah cara yang berguna untuk menentukan sejauh mana model regresi mampu mengintegrasikan kumpulan data. Semakin besar perbedaan menunjukkan kesenjangan yang lebih besar antara nilai yang diprediksi dan diamati, yang berarti model regresi yang buruk cocok. Dengan cara yang sama, semakin kecil RMSE yang menunjukkan model yang lebih baik. Berdasarkan RMSE kita dapat membandingkan dua model yang berbeda satu sama lain dan dapat mengidentifikasi model mana yang lebih cocok dengan data.

3.3 Cara Membuat Boxplot di R-Quick Start Guide

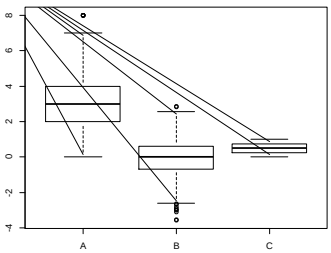
Boxplot adalah plot yang menampilkan ringkasan lima digit dari kumpulan data. Ringkasan lima digit adalah nilai terendah, kuartil pertama, median, kuartil ketiga, dan nilai maksimum.Kita dapat menggunakan boxplot untuk memvisualisasikan sekumpulan data dengan mudah.

3.3.1 Boxplot di R

Mari buat bingkai data untuk pembuatan boxplot .

```
> data <- data.frame( A = rpois(900, 3),
+                     B = rnorm(900),
+                     C = runif(900))
> boxplot(data)
```

Misalkan jika ingin membuat boxplot tunggal maka sintaks berikut akan berguna, output:



Gambar 31. Tampilan boxplot dari sintax R

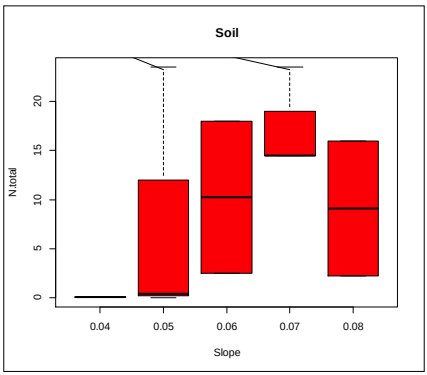
Mari kita ambil kerangka data bio untuk pembuatan Boxplot.

```
head(Soil)
```

```
> head(Soil)
   Slope elevation temperature Radar.air pH.H2O pH.KCl C.organic N.total P2O5
A1  23.5        68        30.6      2.10    6.0    5.9      2.43    0.07   65
A2  23.5        68        30.1      2.03    6.1    6.0      2.31    0.05   60
A3  16.0        67        31.5      1.89    6.2    6.0      1.96    0.06   62
B1  14.5        62        32.7      2.02    5.4    5.2      2.11    0.07   67
B2  16.0        56        33.0      2.09    5.6    5.5      2.04    0.08   73
B3  19.5        56        34.5      2.02    5.2    5.0      1.86    0.07   70
    K2O   CaO2 Mg2O Mn2O   ETM
A1 0.27 10,00 2.01 0.08 10.00
A2 0.23  9.05 1.87 0.07  9.02
A3 0.25  9.76 1.98 0.05  9.16
B1 0.31 10.70 2.05 0.10 10.00
B2 0.32 12.00 2.01 0.13  9.67
B3 0.30 11.20 1.98 0.12  9.51
```

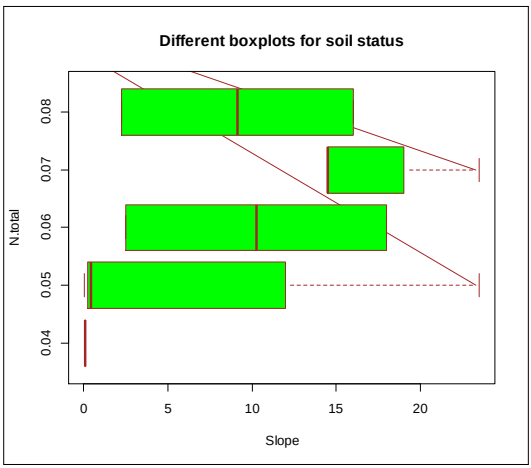
Misalkan jika kita ingin menghasilkan satu *boxplot* untuk setiap *supp* dalam dataset. Buat *boxplot* yang menampilkan distribusi *len* untuk setiap supp dalam *dataset*.

```
> boxplot(Slope~N.total,
+ data= Soil,
+ main="Soil",
+ xlab="Slope",
+ ylab="N.total",
+ col="red",border="black"
+ )
```



Gambar 32. Boxplot dari data Soil

```
> data <- data.frame( A = rpois(900, 3),
+                     B = rnorm(900),
+                     C = runif(900))
> boxplot(data)
> boxplot(Slope~N.total,
+         data=Soil,
+         main="Different boxplots for soil status",
+         xlab="Slope",
+         col="green",
+         border="brown",
+         horizontal=TRUE
+ )
```

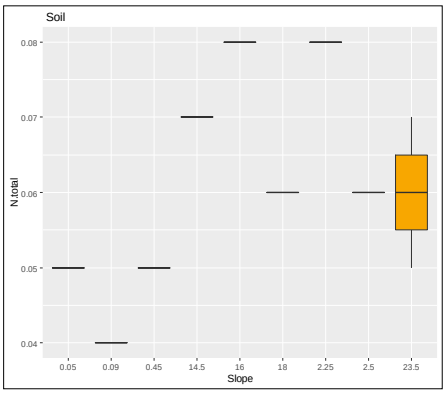


Gambar 33. Boxplot bentuk horizontal

3.3.2 Boxplot di R-ggplot2

Kita dapat menggunakan ggplot2 untuk pembuatan boxplot. Mari kita memuat library.

```
> library(ggplot2)
> ggplot(data = Soil, aes(x=as.character(Slope), y=N.total)) +
+      geom_boxplot(fill="orange") +
+      labs(title=" Soil ", x="Slope", y="N.total")
>
```



Gambar 34. Tampilan output boxplot melalui ggplot2

3. 4. Bagaimana cara mengukur heteroskedastisitas dalam regresi?

Heteroskedastisitas dalam regresi, salah satu cara termudah untuk mengukur heteroskedastisitas adalah saat menggunakan tes Breusch-Pagan. Tes ini terutama digunakan untuk mengidentifikasi jika heteroskedastisitas hadir dalam analisis regresi.Tutorial ini menjelaskan cara mengeksekusi tes breusch-pagan di R.Heteroskedastisitas dalam Regresi.

3.4.1 Langkah 1:

Sesuaikan model regresi.

Pertama, kita akan memasukkan model regresi menggunakan angin sebagai variabel respons dan temp dan bulan sebagai dua variabel penjelas.

memuat dataset Soil

```
data(Soil)
```

cocok (fit) dengan model regresi

```
model <- lm(C.organic~Slope+elevation, data= Soil)
```

lihat ringkasan model

```
summary(model)
```

Koefisien:

```
Call:
lm(formula = C.organic ~ Slope + elevation, data = Soil)

Residuals:
      Min        1Q    Median        3Q       Max
-0.301018 -0.081231  0.004629  0.112442  0.235042

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.728707   0.451028   6.050 0.000306 ***
Slope        0.026403   0.007914   3.336 0.010291 *
elevation   -0.016974   0.008575  -1.979 0.083121 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1707 on 8 degrees of freedom
Multiple R-squared:  0.5869,    Adjusted R-squared:  0.4836
F-statistic: 5.683 on 2 and 8 DF,  p-value: 0.02912
```

3.4.2 Langkah 2:

Lakukan Tes Breusch-Pagan.Di sini kita akan mengukur heteroskedastisitas untuk itu kita dapat menggunakan Uji Breusch-Pagan.Memuat perpustakaan lmtest:

```
library(lmtest)
```

```
> library(lmtest)
Loading required package: zoo

Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

Jalankan Tes Breusch-Pagan

```
bptest(model)
```

```
> bptest(model)

        studentized Breusch-Pagan test

data:  model
BP = 3.9143, df = 2, p-value = 0.1413
```

studentized Breusch-Pagan test

```
data:  model
BP = 3.9143, df = 2, p-value = 0.1413
```

Statistik uji adalah 3,9143 dan nilai p yang sesuai adalah 0,1413. Karena nilai p lebih besar dari 0,05, kita tidak dapat menolak hipotesis nol.Ini menunjukkan bahwa kami tidak memiliki cukup bukti untuk

menolak hipotesis nol atau bukti yang cukup untuk mengatakan heteroskedastisitas hadir dalam model regresi.

3.5 Analisis Variansi (ANOVA)

Analisis Varians dalam R, kita akan dapat mengidentifikasi alasan untuk menggunakan uji Analisis Varians (atau ANOVA) dalam analisis data setelah menyelesaikan materi ini serta kita juga akan belajar bagaimana menganalisis temuan uji-f ANOVA.

Bayangkan jika ingin melihat variabel kategori dan melihat bagaimana hubungannya dengan variabel lain.Ambil, misalnya, kumpulan data Maskapai.

Langkah 1. Loading data

```
library(tidyverse)
library(dplyr)
library(ggplot2)
data<-read.csv("D:/RStudio/ADI.csv",1)
head(data)
```

Langkah 2: Hipotesis Nol

Akibatnya, hipotesis nol untuk ANOVA adalah rata-rata (nilai rata-rata ADI) sama untuk semua grup.Hipotesis alternatif atau penelitian adalah bahwa rata-rata untuk semua kelompok tidak sama.

```
summary(AnovaModel.1)
            Df Sum Sq Mean Sq F value   Pr(>F)
Blok.Situs   3  60.25  20.083    24.1 0.000233 ***
Residuals    8   6.67   0.833
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Skor uji-F dan nilai-p dikembalikan oleh uji ANOVA.Uji-F menentukan rasio varians antara rata-rata setiap kelompok sampel dan variasi dalam setiap kelompok sampel.Nilai p menunjukkan apakah hasilnya signifikan secara statistik atau tidak.Secara umum, kita dapat menganggap varians signifikan secara statistik jika nilai p kurang dari 0,05.Hubungan tersebut substansial jika skor uji-F tinggi dan tidak ada hubungan jika nilai uji-F rendah.

Langkah 3: Perbandingan ANOVA

Uji ANOVA menghasilkan skor uji F yang signifikan dan nilai P yang kecil, Anda dapat menyimpulkan bahwa ada hubungan yang kuat antara variabel kategori dan faktor lainnya. Kita ketahui bahwa uji ANOVA dapat

digunakan untuk mengidentifikasi korelasi antara kelompok yang berbeda dari variabel kategori dan bahwa skor uji-F dan nilai-p dapat digunakan untuk mengidentifikasi signifikansi statistik.

3.5.1. Analisis Ragam/Analysis of variance (Anova) satu arah (one way)

Kali ini akan dibahas analisis ragam satu arah menggunakan software R. Contoh kasus Anova satu arah. Ada 4 situs pengukuran indeks dominasi apical pohon kelengkeng yang diukur pada 4 kecamatan dengan masing-masing diambil 3 sampel seperti ditunjukkan pada tabel berikut.

Tabel 3. Hasil pengukuran nilai ADI dari empat situs yang diamati.

Sampel	*Apical Domination Index* (ADI)			
	Situs 1 (Plandaan)	Situs 2 (Kabuh)	Situs 3 (Ploso)	Situs 4 (Kudu)
Sampel 1	8	4	3	2
Sampel 2	9	5	4	3
Sampel 3	7	6	3	1

Apakah keempat situs tersebut memberikan nilai rata-rata ADI yang sama? Uji pendapat tersebut dengan taraf nyata 5%.

Solusi kasus Anova satu arah.

Identifikasi Metode statistic yang digunakan

Pertama, berdasarkan hipotesis yang digunakan yaitu membandingkan rata-rata lebih dari dua kelompok maka metode yang mungkin adalah Anova. Kedua, sampel digunakan tiap kelompok berbeda perlakuan sehingga tipe anova yang cocok adalah anova satu arah.

Dalam metode Anova yang perlu diperhatikan ada empat, asumsi normal dan homgenitas antar varians kelompok harus terpenuhi. Dalam contoh ini kita asumsikan terpenuhi, karena focus pada langkah-langkah Anova satu arah, kemudian kelompok yang dianalisis berasal dari kelompok saling bebas dan data yang digunakan merupakan data rasio. Seteleh asumsi ini terpenuhi maka bisa lanjut ke perhitungan selanjutnya, kalau tidak ganti metode.

Langkah-langkah dalam Uji Hipotesis Anova Satu Arah (*One Way*)

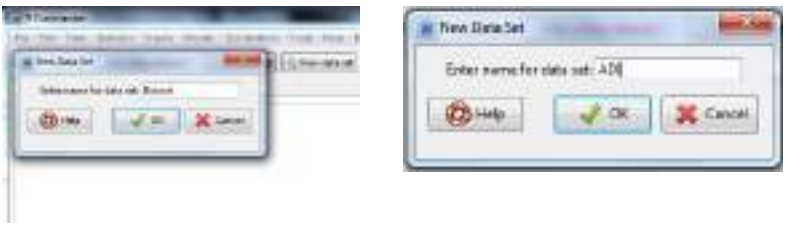
1. **Load Packege Rcmdr**. Hal ini dilakukan karena paket yang digunakan dalam R adalah R commander. Caranya klik **Package** kemudian pilih **Load Package** terus muncul tampilan dan pilih **Rcmdr**.

2. Masukkan data. Caranya yaitu buat dua kolom yang pertama berupa semua nilai. Kedua yaitu kolom yang menyatakan penjelasan kelompok dari kolom pertama.

Tabel 4. Bentuk data yang akan diinput dalam R commader

No.	ADI	Situs
1	8	1
2	9	1
3	7	1
4	4	2
5	5	2
6	6	2
7	3	3
8	4	3
9	3	3
10	2	4
11	3	4
12	1	4

Cara memasukkan data melalui jendela Rcommander, dengan tahapan berikut:
Klik data >New data set > lalu tampil jendela berikut. Beri nama data yang akan diinput, misalnya data ADI.



Gambar 35. Tampilan menu R commader pada proses input data

Kemudian kill **ok**. Langkah selanjutnya, lalukan input data dengan menambah baris dan kolom yang diinginkan.

Gambar 36. Tampilan input data pada R commander

Selanjutnya klik ok. Data akan tersimpan dalam program R dengan nama file data ADI yang terlihat pada jendela R commander. Konversi data ke Factor. Tujuannya yaitu memberi nama kategori kolom 2 tadi. Caranya pilih Data, kemudian pilih manage variables in active data set dan klik convert numeric variables to factors. Kemudian akan muncul seperti berikut.

Gambar 37. Tahapan konversi data ke faktor

3. Pilih **Blok Situs** dan pada **factor level** pilih **suplay level names**. Kemudia isi data seperti berikut:

Gambar 38. Tampilan pemberian lama level kategori

4. Kemudian pilih statistics, lalu pilih means dan klik Anova one way, maka akan muncul tmpilan berikut:

Gambar 39. Proses analisis varians

5. Cek list pairwise comparisons of means kalau mau uji lanjut (Tukey), kemudian klik ok.

```
> AnovaModel.1 <- aov(ADI ~ Blok.Situs, data = ADI)

> summary(AnovaModel.1)
             Df Sum Sq Mean Sq F value   Pr(>F)
Blok.Situs    3  60.25  20.083    24.1 0.000233 ***
Residuals     8   6.67   0.833
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> with(ADI, numSummary(ADI, groups = Blok.Situs, statistics =
c("mean",
 "sd")))
           mean        sd data:n
Blok 1 8.000000 1.0000000      3
Blok 2 5.000000 1.0000000      3
Blok 3 3.333333 0.5773503      3
Blok 4 2.000000 1.0000000      3
```

Intepretasi hasil uji Anova satu arah dengan R commander.

Summary:

Pada bagian ini menjelaskan tabel Anova, dengan memperhatikan nilai Pr(>F) bernilai 0,000233. Karena nilainya lebih kecil dari 0,05(alpha)

maka keputusan gagal tolak Ho, sehingga disimpulkan terdapat perbedaan nilai ADI antar blok.

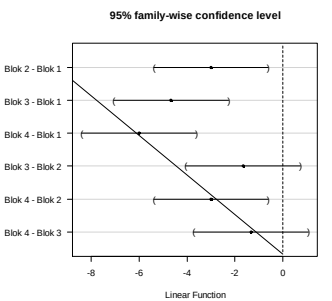
NumSummary:

Bagian ini menjelaskan deskripsi dari tiap kategori yang dalam hal ini adalah kategori blok. Deskrispsinya hanya rata-rata (mean), standar deviasi(sd) dan jumlah pengulangan (n).

Selanjutnya kita bisa melihat tingkat perbedaan antar blok melalui hasil analisis berikut:

```
> local({
+      .Pairs <- glht(AnovaModel.1, linfct = mcp(Blok.Situs =
"Tukey"))
+     print(summary(.Pairs))  # pairwise tests
+     print(confint(.Pairs))  # confidence intervals
+     print(cld(.Pairs))  # compact letter display
+     old.oma <- par(oma = c(0, 5, 0, 0))
+     plot(confint(.Pairs))
+     par(old.oma)
+ })

      Simultaneous Tests for General Linear Hypotheses
Multiple Comparisons of Means: Tukey Contrasts
Fit: aov(formula = ADI ~ Blok.Situs, data = ADI)
Linear Hypotheses:
                     Estimate Std. Error t value Pr(>|t|)
Blok 2 - Blok 1 == 0  -3.0000     0.7454  -4.025  0.01626 *
Blok 3 - Blok 1 == 0  -4.6667     0.7454  -6.261  0.00105 **
Blok 4 - Blok 1 == 0  -6.0000     0.7454  -8.050  < 0.001 ***
Blok 3 - Blok 2 == 0  -1.6667     0.7454  -2.236  0.19326
Blok 4 - Blok 2 == 0  -3.0000     0.7454  -4.025  0.01596 *
Blok 4 - Blok 3 == 0  -1.3333     0.7454  -1.789  0.34432
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
(Adjusted p values reported -- single-step method)
```

Untuk melihat tingkat perbedaan dari masing-masing blok ditampilkan grafik berikut:

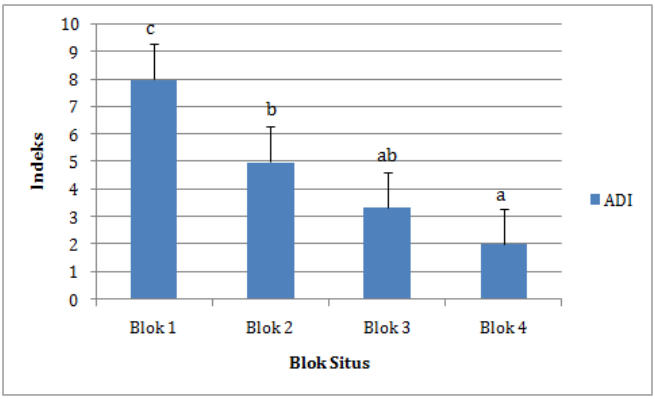


Gambar 40. Tampilan grafik tingkat perbedaan masing-masing blok

```
Simultaneous Confidence Intervals
```

```
Multiple Comparisons of Means: Tukey Contrasts
Fit: aov(formula = ADI ~ Blok.Situs, data = ADI)
Quantile = 3.1979
95% family-wise confidence level

Linear Hypotheses:
                     Estimate lwr     upr
Blok 2 - Blok 1 == 0 -3.0000  -5.3835 -0.6165
Blok 3 - Blok 1 == 0 -4.6667  -7.0502 -2.2831
Blok 4 - Blok 1 == 0 -6.0000  -8.3835 -3.6165
Blok 3 - Blok 2 == 0 -1.6667  -4.0502  0.7169
Blok 4 - Blok 2 == 0 -3.0000  -5.3835 -0.6165
Blok 4 - Blok 3 == 0 -1.3333  -3.7169  1.0502

Blok 1 Blok 2 Blok 3 Blok 4
   "c"    "b"   "ab"    "a"

> oneway.test(ADI ~ Blok.Situs, data = ADI)  # Welch test

One-way analysis of means (not assuming equal variances)
data:  ADI and Blok.Situs
F = 16.907, num df = 3.0000, denom df = 4.2857, p-value =
0.007887
```

Hasil analisis Rcommander menunjukkan bahwa terdapat perbedaan nilai ADI dari 4 blok yang diteliti, sehingga bisa ditampilkan histogram dari tingkat perbedaan yang ada seperti pada gambar berikut;



Gambar 41. Output grafik tingkat perbedaan nilai ADI pada masing-masing blok penelitian

Pada grafik di atas terlihat bahwa secara umum semua blok memberikan nilai ADI yang berbeda secara signifikan, namun blok 3 tidak

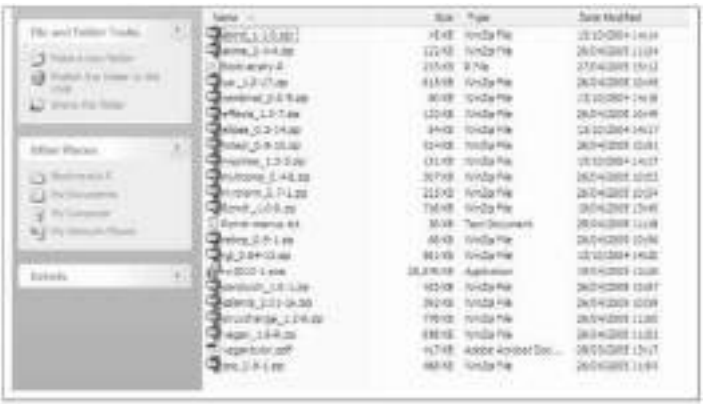
berbeda secara signifikan dengan blok 2 dan blok 4. Nilai ADI terbesar ada di blok 4 yang berbeda secara signifikan dengan 3 blok lainnya.

BAB IV
PROGRAM BIODIVERSITY R

4.1 Pengantar

Bab ini menjelaskan bagaimana analisis yang disajikan dalam manual ini dapat dilakukan dengan perangkat lunak Biodiversity.R.Perhatian khusus pada Biodiversity.R ini tidak berarti bahwa paket perangkat lunak lain tidak dapat digunakan untuk analisis keanekaragaman hayati. Pada tahun 2003, CD-ROM Paket Analisis Keanekaragaman Hayati diproduksi untuk menyediakan beberapa paket perangkat lunak yang sangat baik untuk analisis keanekaragaman hayati. Namun, beberapa paket perangkat lunak memiliki cakupan yang lebih terbatas dalam analisis yang didukungnya. Beberapa paket perangkat lunak belum mengembangkan antarmuka pengguna grafis, yang menyebabkan beberapa masalah dalam pengajaran analisis. Beberapa jenis analisis tidak dapat dilakukan dalam perangkat lunak yang disediakan pada CD-ROM. Untuk alasan ini, perangkat lunak Biodiversity.R dikembangkan, termasuk antarmuka pengguna grafis. Semua analisis yang dijelaskan dalam manual ini dapat dilakukan dengan Biodiversity.R.

Biodiversity.R adalah perangkat lunak yang melakukan semua analisis keanekaragaman hayati yang dijelaskan dalam manual ini. Itu perlu dimuat ke dalam perangkat lunak statistik R. R adalah perangkat lunak yang dikembangkan untuk memungkinkan berbagai jenis analisis statistik. Ini sangat mirip dengan perangkat lunak statistik S dan S-Plus. Ini adalah perangkat lunak bebas, seperti Biodiversity.R. Perangkat lunaknya juga terbuka, sehingga Anda dapat memeriksa bagaimana perhitungan dilakukan, dan grafiknya cukup canggih.



Gambar 42. File-file yang digunakan selama instalasi disediakan di folder File instalasi

Setiap kali Anda menjalankan R, Anda perlu memuat paket Rcmdr untuk mengakses antarmuka pengguna grafis Biodiversity.R.Dua opsi disediakan untuk menginstal Biodiversity.R. Ketika kita mengklik Install-Biodiversity.bat, maka semua file akan diinstal di bawah folder Program Files dari drive C Anda.Metode alternatif adalah menginstal Biodiversity.R secara bertahap. File yang digunakan selama instalasi semuanya terdaftar di CD-ROM di Biodiversity. Hal pertama yang perlu dilakukan adalah menginstal perangkat lunak R. Kita juga dapat memperoleh perangkat lunak dari situs web berikut: http://cran.r-project.org. Jika menggunakan Windows, dapat mengunduh R dari: http://cran.r-project.org/bin/windows/base. Selanjutnya akan ditemukan file di sana dengan nama yang mirip dengan rw2010-1.exe. Kita perlu menjalankan file untuk menginstal R. Perhatikan bahwa mungkin lebih aman untuk menutup semua program lain saat menginstal R.

Selama instalasi, pastikan kita memilih untuk menginstal file dukungan untuk library(tcltk). Setelah kita menginstal R, maka kita dapat menjalankannya. R adalah bahasa yang dijalankan dengan mengetikkan perintah. Itu tidak memiliki antarmuka pengguna yang luas. Beberapa perangkat lunak tambahan dapat dimuat ke R. Addin ini disebut paket atau library. Beberapa paket sudah datang dengan versi instalasi R. Beberapa paket lain perlu diunduh, kita memiliki berbagai opsi untuk menginstal paket tambahan ini. Opsi pertama adalah menginstal paket tambahan dari CD-ROM. Dalam program R, kita harus pergi ke menu atas, pilih Paket, lalu pilih opsi menu: Instal paket dari file zip lokal…. Gambar 3.2 menunjukkan bagaimana hal ini dapat dilakukan. Perhatikan bahwa kami tidak akan menempatkan banyak gambar menu dalam manual ini, sehingga beberapa ruang akan dihemat.

Disini akan dijelaskan pilihan yang ditunjukkan pada Gambar 3.2 sebagai: “Kita memilih opsi menu R: Packages > Install package(s) from local zip files…”.Kita akan mendapatkan daftar paket yang tersedia di bawah folder File instalasi CD-ROM. Anda dapat menginstal semua paket dengan memilih semuanya secara bersamaan(klik tombol CTRL dan A secara bersamaan).

Gambar 43. Menginstal paket lain ke R. Opsi menu ini dijelaskan dalam teks sebagai: "Paket > Instal paket dari file zip lokal...".

Alternatifnya adalah mengunduh paket dari R dan kemudian menginstalnya secara manual setelahnya, atau menginstal paket langsung dari situs web CRAN. Kita dapat mengunduh file untuk paket yang berbeda dari situs web yang sama dengan yang diunduh dari paket R itu sendiri dari: http://cran.r-project.org. Kita akan melihat tautan ke paket yang dikontribusikan". Jika kita menggunakan Windows, maka kita dapat mengunduhpaket dari: http://cran.r-project.org/bin/windows/contrib.

Kita hanya perlu menginstal paket satu kali setelah menginstal R. Saat menjalankan R, kita tidak akan dapat mengakses fungsi paket yang berkontribusi, kecuali jika kita mengikuti opsi menu: Paket > Muat paket…, atau jika kita memberikan yang sesuai perintah di R memuat paket. Untuk pustaka vegan, kita perlu memilih paket ini setelah opsi menu: Paket > Muat paket… atau kita perlu mengetik pustaka(vegan) setiap kali kita mulai menggunakan R dan ingin mengakses fungsi pustaka ini. Perhatikan bahwa kita akan menggunakan font yang berbeda untuk hasil dan perintah di R dan Biodiversity.R. Misalnya, library(vegan)menunjuk ke perintah di R. Informasi apa pun dalam font ini akan muncul di R console.

Paket yang dibutuhkan untuk menginstal ke dalam R adalah:

- abind
- akima
- car
- combinat
- effects
- ellipse
- lmtest
- maptree
- multcomp
- mvtnorm
- rcmdr

• relimp
• rgl
• sandwich
• splancs
• strucchange
• vegan
• zoo

Ini adalah beberapa paket, tetapi ini juga berarti bahwa berbagai jenis analisis dapat dilakukan di R. Yang paling penting adalah vegan karena memungkinkan banyak analisis Biodiversity.R. Rcmdr juga merupakan paket penting karena memungkinkan antarmuka grafis R-commander. (Untuk pengguna tingkat lanjut: R-commander berjalan paling baik di bawah Antar muka Dokumen Tunggal. Kita dapat mengatur opsi ini dengan menyetel "MDI = no" di file C:\Program Files\R\rw2010\etc\Rconsole.).

Langkah terakhir (dan mungkin yang paling rumit) dalam menginstal Biodiversity.R adalah menyalin dua file ke folder library\Rcmdr\etc dari R. Jika mau menginstal R di bawah file program drive C, maka kita dapat menemukan Direktori Rcmdr di bawah C:\Program files\R\rw2010\library\Rcmdr\etc.

Kita perlu memasukkan file-file berikut ke dalam direktori ini: Biodiversity.R dan Rcmdr-menus.txt. Rcmdr-menus.txt sudah ada di perpustakaan, jadi kita perlu mengganti Rcmdr-menus.txt dengan file yang disediakan di CD-ROM. Kita dapat menggunakan program seperti Windows Explorer untuk menyalin file. Ketika kita telah menyelesaikan semua langkah ini, Biodiversity.R akan terinstal.Singkatnya, kita perlu mengikuti langkah-langkah ini untuk menginstal dan menjalankan Biodiversity.R:

1. Instal R
2. Instal paket kontribusi yang diperlukan
3. Salin Biodiversity.R dan *Rcmdr-menus.txt* ke dalam*library\Rcmdr\etc*

Jelas kita harus menjalankan R terlebih dahulu. Untuk menjalankan Biodiversity.R dengan opsi menu, kita perlu memuat R-commander, baik dengan perintah library(Rcmdr) atau dengan opsi menu: Packages > Load package…Setiap kali kita ingin menggunakan Biodiversity.R, kita perlu memuat paket Rcmdr lagi setelah meluncurkan R.

4.2 Instalasi Program BiodiversityR berbasis Windows

Ini adalah panduan cepat untuk menginstal BiodiversityR di komputer dan mulai menggunakan paket.

4.2.1 Langkah 1:

Unduh file instalasi untuk lingkungan statistik R (misalnya R-3.5.1-win.exe) dari situs web CRAN (URL http://cran.r-Project.org/bin/windows/base/), sebaiknya melalui situs mirror (URL http://cran.r-project.org/mirrors.html; misalnya https://pbil.univ-lyon1.fr/CRAN/bin/windows/).

4.2.2. Langkah 2:

Instal lingkungan statistik R di komputer.

a. Tutup semua program lainnya
b. Klik pada file instalasi (misalnya klik pada R-3.6.1-win.exe)
c. Ikuti instruksi dan klik berikutnya, kecuali untuk layar di bawah ini (bagian c – d)
d. Pilih startup yang disesuaikan

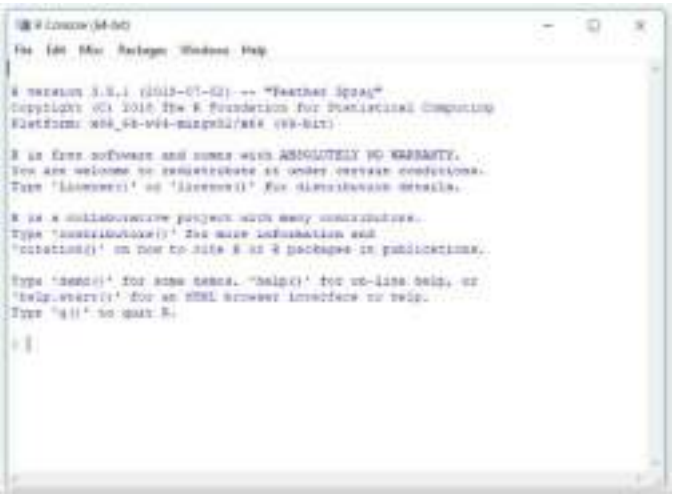
Gambar 44. Tampilan Jendela Setup-R for Windows 3.5.1

e. Pilih Mode Tampilan SDI (jendela terpisah)

Gambar 45. Tampilan Display Mode

4.2.3 Langkah 3: Instal semua paket setelah meluncurkan R

a. Luncurkan R



Gambar 46. Tampilan R Console

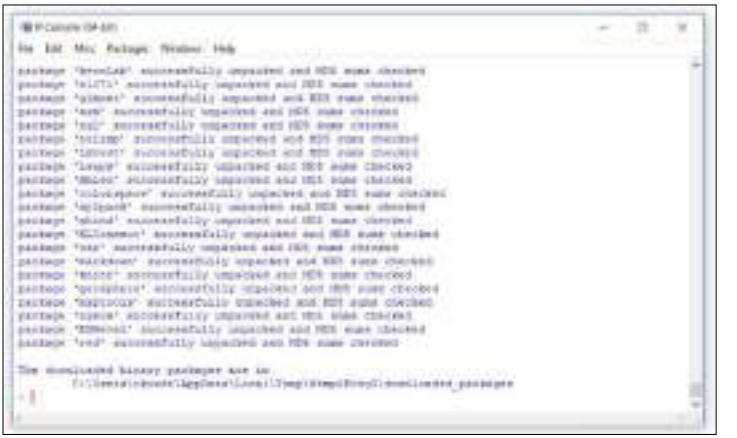
b. Disarankan untuk tidak menginstal BiodiversityR secara langsung di RStudio untuk menghindari masalah dengan instalasi beberapa paket…

c. Rekatkan perintah berikut di R Console:

```
install.packages(pkgs=c("BiodiversityR", "vegan",
  "Rcmdr", "MASS", "mgcv",
  "cluster", "RODBC", "rpart", "effects", "multcomp",
  "ellipse", "maptree", "sp", "splancs", "spatial",
  "akima", "nnet", "dismo", "raster", "rgdal",
  "bootstrap", "PresenceAbsence",
  "maxlike", "gbm", "randomForest", "gam", "earth", "mda",
  "kernlab", "e1071", "glmnet", "sem", "rgl", "relimp",
  "lmtest", "leaps", "Hmisc", "colorspace", "aplpack",
  "abind", "XLConnect", "car", "markdown", "knitr",
  "geosphere", "maptools", "rgeos", "ENMeval", "red"),
  dependencies=c("Depends", "Imports"))
```

Gambar 47. Tampilan Jendela R Console dengan Secure CRAN mirrors

d. Tekan tombol 'ENTER' pada keyboard Anda setelah menempelkan perintah. (Berpotensi memilih situs mirror HTTP yang tersedia melalui antarmuka jendela bawah 'Secure CRAN Mirrors'). Jika folder utama R read-only, pilih untuk menggunakan personal library.



Gambar 48. Tampilan Secure CRAN mirrors yang terbaca di R Console

e. Ketika proses instalasi untuk paket yang berbeda telah selesai dengan sukses, Anda akan melihat layar di atas. Jika ada beberapa masalah selama proses instalasi, keluar dari R Console dan coba lagi

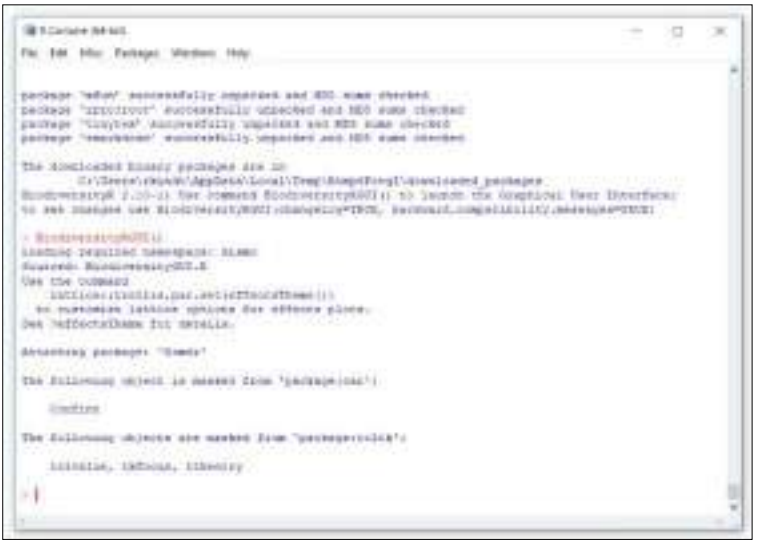
langkah (a) dan (b). Jika Anda memiliki peringatan kesalahan khusus untuk paket tertentu, Anda dapat mengubah perintah dengan menghapus paket yang terdaftar sebelumnya. Jika Anda memiliki masalah dalam menginstal paket besar, alternatifnya adalah mencoba mengunduh paket langsung dari CRAN(mis., https://pbil.univ-lyon1.fr/CRAN/bin/windows/contrib/3.5/rgdal_1.3-4.zip) dan kemudian instal secara manual melalui opsi menu R GUI: Packages > Install package(s) from local files…

4.2.4 Langkah 4: Mulai gunakan BiodiversityR dan antarmuka pengguna grafisnya (GUI) Mulai gunakan BiodiversityR dan antarmuka pengguna grafisnya (GUI) dengan mengetik (atau menempelkan) dua perintah berikut:

```
library(BiodiversityR)
BiodiversityRGUI()
```

Perhatikan bahwa R adalah *case-specific*, jadi jangan gunakan huruf kapital di mana ini tidak ditampilkan.(Saat meluncurkan BiodiversityR pertama kali, ada kemungkinan bahwa R-Commander menyarankan untuk menginstal beberapa paket tambahan – izinkan instalasi ini).

Ulangi langkah 4 setiap kali Anda ingin menggunakan BiodiversityR.



Gambar 49. Tampilan perintah menjalankan program BiodiversityR

Jika semuanya berjalan dengan baik, maka Anda sekarang seharusnya meluncurkan R Commander:

Gambar 50. Tampilan jendela R Commader

Antarmuka menu untuk BiodiversityR tersedia dari opsi menu paling kanan (BiodiversityR).Berdasarkan pilihan dari jendela menu, R Commandermenghasilkan skrip gaya perintah yang dikirimkan ke R di "Scrip Windows". Hasil ditampilkan di "OutputWindows". Sebagai alternatif untuk menghasilkan skrip melalui antarmuka menu, kita juga dapat mengetik langsung di Scrip Windows, sorot skrip yang diinginkan, lalu kirim ke R melalui tombol "Submit".

Bantuan penggunaan BiodiversityR tersedia dari pilihan menu: **BiodiversityR > Help about BiodiversityR > Help about Biodiversity**. Metode alternatif untuk mengakses file bantuan adalah dengan mengirimkan perintah berikut:

```
help("BiodiversityRGUI", help_type="html")
```

4.3 Kumpulan data spesies dan lingkungan

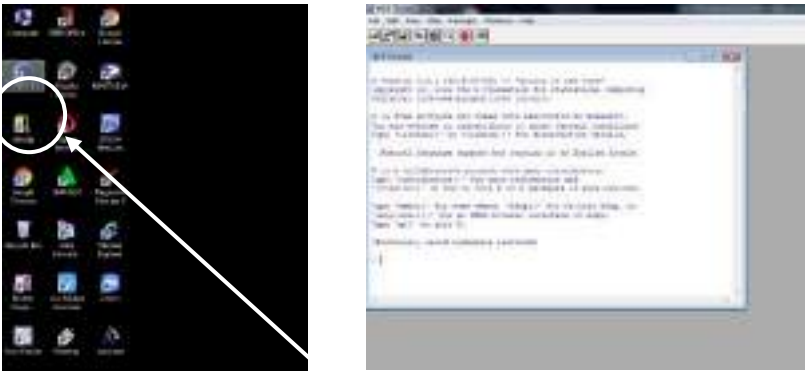
Setelah kita memuat Biodiversity.R, Anda tidak akan dapat melakukan analisis jenis apa pun sebelum kita memilih kumpulan data spesies dan lingkungan. Seperti yang kita perlu persiapan data, analisis dalam manual ini akan dilakukan dengankedua kumpulan data ini.

R-commander dirancang untuk hanya menggunakan satu set data. Dataset ini disebut dataset aktif. Biodiversity.R menggunakan dua dataset, yaitu spesies dan dataset lingkungan. Saat mengembangkannya, keputusan diambil untuk membuat dataset lingkungan Biodiversity.R selalu menjadi dataset aktif dari R-commander. Ketika dataset baru dipilih untuk menjadi dataset aktif baru untuk R-commander, maka itu juga menjadi dataset lingkungan baru Biodiversity.R. Dengan opsi menu R-commander, berbagai manipulasi dapat dilakukan pada dataset aktif, termasuk mengimpor dataset aktif dan menyimpan dataset aktif. Kita dapat melakukan manipulasi yang sama pada kumpulan data spesies, tetapi pertama-tama kita perlu mengatur kumpulan data spesies menjadi kumpulan data aktif (dan dengan demikian juga kumpulan data lingkungan).Uraian tentang cara di mana setiap jenis analisis dapat dilakukan dalam R disediakan di akhir setiap bab. Sebagai contoh dan uji apakah Anda menginstal Biodiversity.R dengan benar, Anda bisamenjalankan analisis berikut. Contoh ini menghitung kurva akumulasi spesies untuk kumpulan data bukit pasir. Dataset ini disediakan dengan vegan. Hasil yang akan Anda peroleh ditunjukkan dalam bab tentang kekayaan spesies.

BAB V
OPERASI BIODIVERSITY R

5.1. Pengantar

Bila program R sudah terinstall maka kita bisa mengoperasikan perangkat lunat ini dengan mengklik logo R yang tercantum dalam destop computer anda. Aplikasi program yang digunakan R versi 3.6.1. Selanjutnya akan muncul tampila R.Console

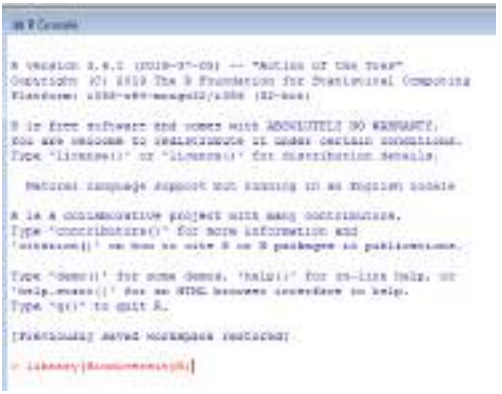


(a) (b)

Gambar 51. Tampilan logo program R di destop (a), dan tampilan R.Console (b)

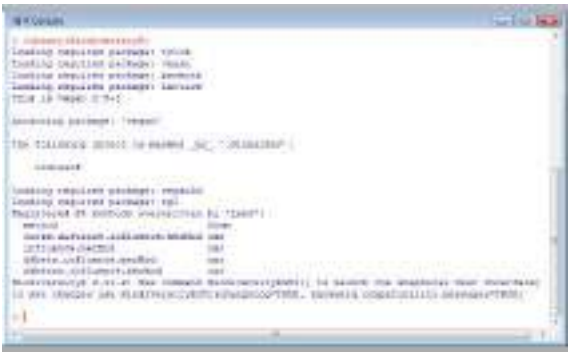
Karena program R akan menggunakan Biodiversity.R maka pada R.Console kita ketik:

```
> library(BiodiversityR)
```



Gambar 52. Tampilan R Console untuk memulai operasi BiodiversityR

Lalu tekan enter, maka akan muncul tampilan seperti gambar berikut:

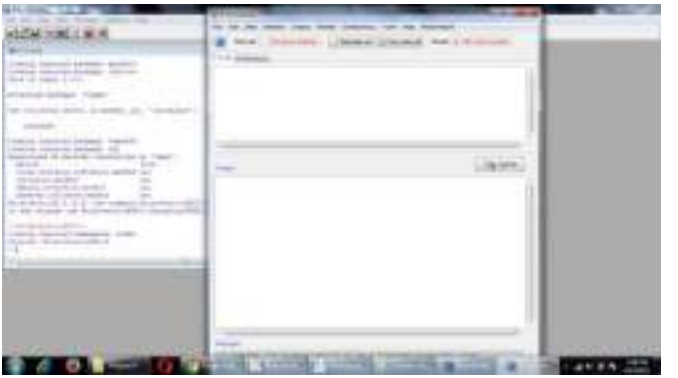


Gambar 53. **1**R Console

Selanjutnya di ruang R.Console ketik:

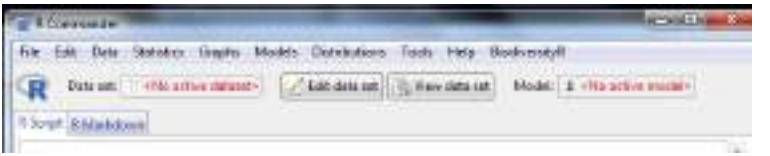
```
>BiodiversityRGUI()
```

Lalu enter, maka akan muncul tampilan berikut:



Gambar 54. Tampilan R Console dan R Commader

Akan muncul tampilan R.Commander, sebagai lembar kerja R. Pada R.Commader terdapat menu File, Edit, Data, Statistics, Graphs, Models, Distributions, Tools, Help, dan BiodiversityR.



Gambar 55. Bar menu R Commander

Pada masing-masing menu memiliki beberapa submenu, yaitu:

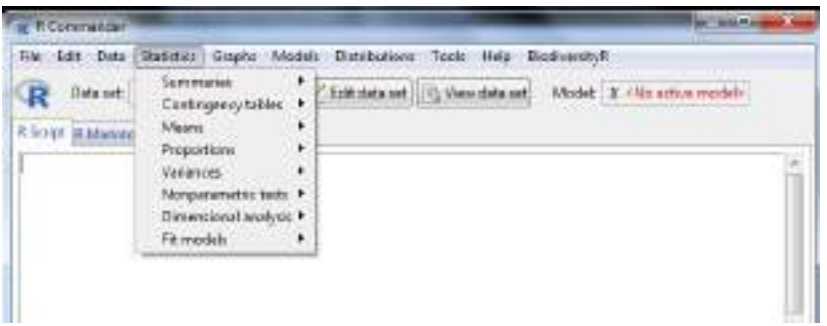
a. Submenu Data

Gambar 56. Tampilan isi dari submenu data

Fungsi dari menu data, misalnya untuk mengimport data dari file tersimpan dalam format text clipboard atau URL, SPSS, SAS, Minitab, STATA dan Excel.

b. Submenu Statistics

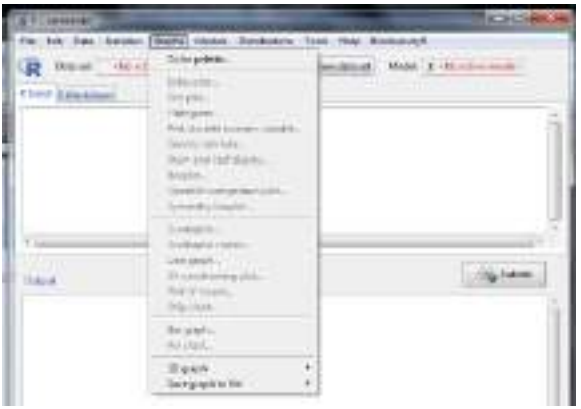
Submenu statistics menyediakan bentuk analisi summaries, contingency tables, means, proportions, variances, nonparametric test, dimensional analysis dan fit model.



Gambar 57. Tampilan dari bentuk analisis data statistic

c. Submenu Graphs

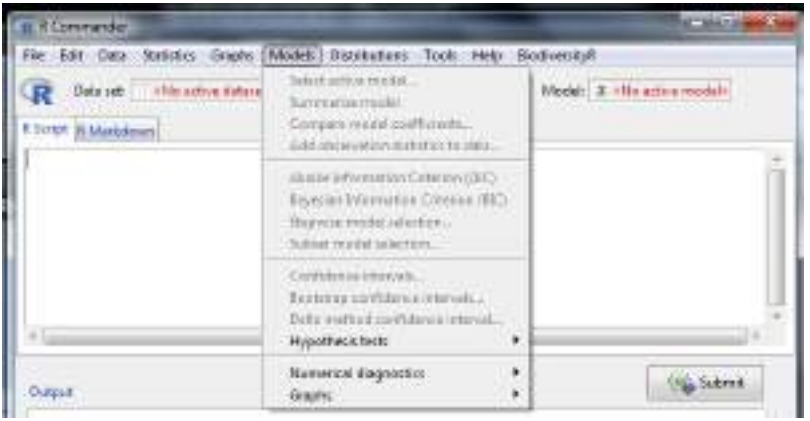
Submenu graphs menyediakan berbagai bentuk grafik untuk visualisasi data.



Gambar 58. Tampilan submenu Graphs

d. Submenu Models

Submenu model menyediakan berbagai bentuk model yang akan dibangun dari data penelitian yang ada.



Gambar 59. Tampilan submenu Model

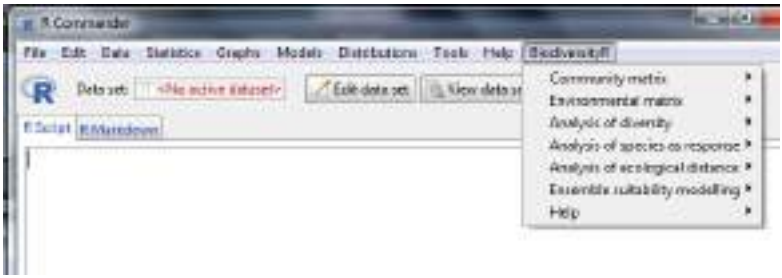
e. Submenu Distributions

Submenu ini menyediakan alternative bentuk distribusi data yang akan dianalisis dengan program R.

Gambar 60. Tampilan submenu Distributions

f. Submenu BiodiversityR

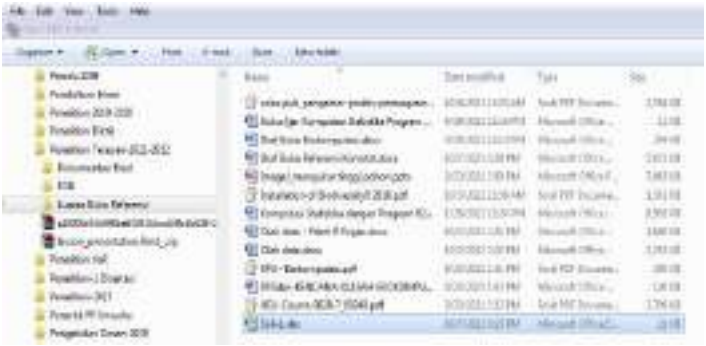
Pada submenu ini menyediakan berbagai bentuk analisis untuk data penelitian bidang biologi dan pertanian.



Gambar 61. Tampilan submenu BiodiversityR

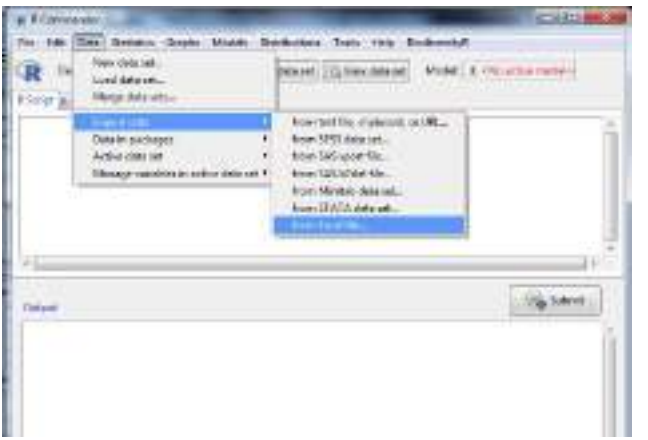
5.2 Import Data dari Excel

Pada program R untuk menginputkan data disediakan banyak cara, salah satunya dengan mengambil data dari file Excel. Proses input data dilakukan melalui tampilan jendela R.Commander pada menu Bar Data. Input data dengan cara ini pertama-tama kita siapkan data excel sesuai nama file di excel yang kita pilih.



Gambar 62. Penentuan file data format Excel untuk dimasukkan ke program R

Selanjutnya buka jendela R.Commander dengan mengklik menu Data, dengan tahapan **Data > Import data > from excel file** dan secara jelas ditunjukkan pada gambar 5.13 berikut:

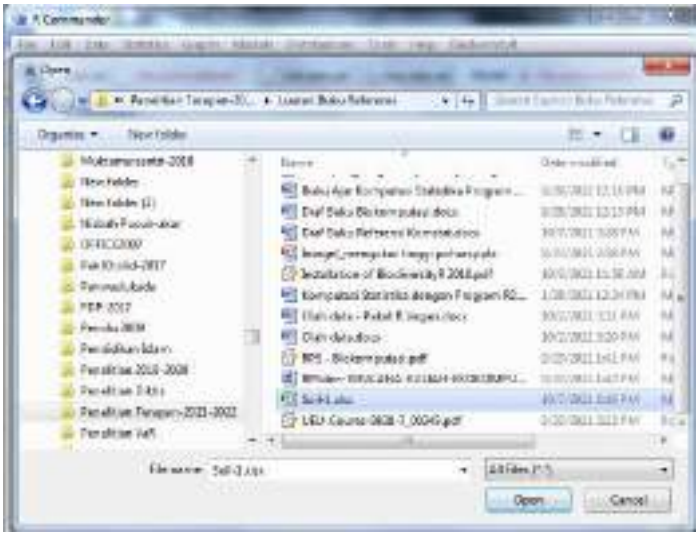


Gambar 63. Tampilan R.Commander untuk proses Import Data dari Excel

Kemudian klik **import data > from Excel File.** Selanjutnya muncul jendela **Import Excel Data Set**, lalu ber inama data set dengan **tanah** (sesuai nama file data yang ingin dianalisis), klik ok.

Gambar 64. Tampilan Jendela Import Excel Data Set

Selanjutnya pilih folder data yang ingin diimport, seperti tampak pada gambar berikut:

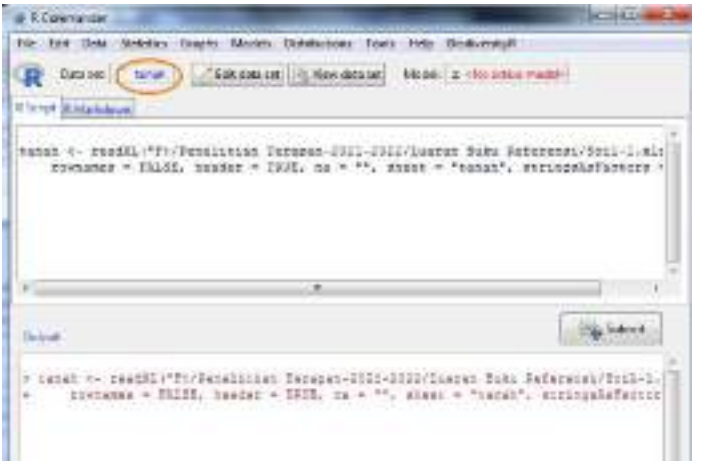


Gambar 65. Tahapan pemilihan folder data Excel

Kemudian akan muncul kotak dialog untuk memlih salah data yang mau diimport, kemudian klik ok.

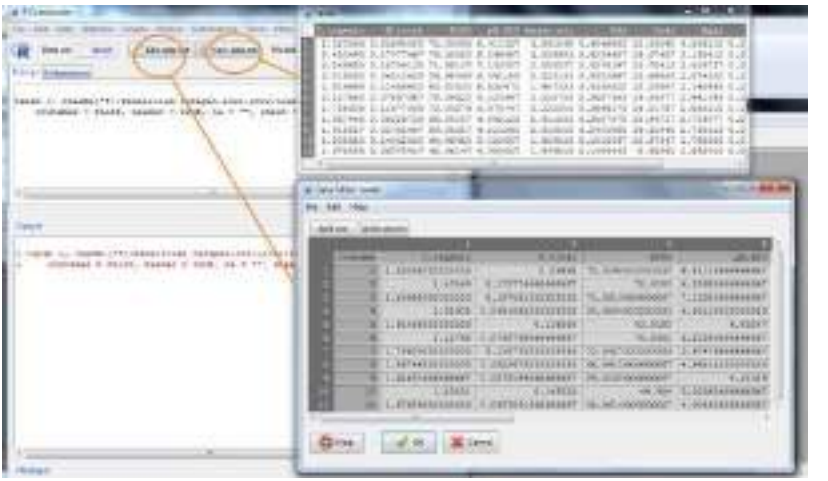
Gambar 66. Tampilan Jendela pada proses pemilihan data set dengan memilih satu tabel

Apabila proses import data berhasil, selanjutnya akan tampil di R.Commader pada Data set dengan nama data yang telah kita import



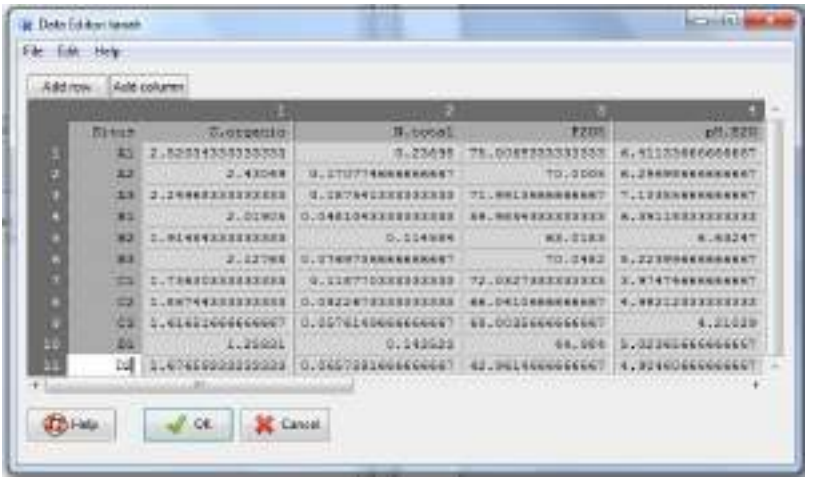
Gambar 67. Proses Import data sukses jika pada kolom Data set muncul nama data set yang kita import.

Selanjutnya data bisa kita lihat kembali dengan mengklik**View Data Set**, atau melakukan editing data dengan mengklik **Edit Data Set**.

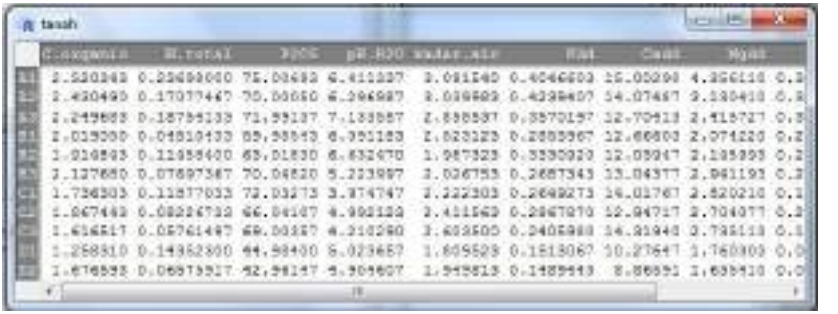


Gambar 68. Tampilan jendela R.Commander tentang tahapan melihat dan mengedit data

Editing data untuk penulisan situs penelitian, dengan langsung mengetik kolom yang ingin diedit, kemudian klik ok. Editing yang disediakan juga bisa dalam bentuk penambahan dan pengurangan baris dan kolom atau mengedit nominal data.



Gambar 69. Tampilan jendela Data Editor untuk data tanah



Gambar 70. Tampilan data tanah setelah dilakukan editing

Setelah proses import data dari Excel ke dalam program R maka langkah selanjutnya proses analisis data sesuai model analisis yang digunakan.

BAB VI
MENGGAMBAR PETA PENELITIAN
DENGAN PROGRAM R

6. 1 Pengantar

Peta termasuk bagian penting dari manuskrip khususnya penelitian yang terkait dengan ekologi dan konservasi atau penelitian yang melibatkan pengambilan sampel organisme di lapangan. Keberadaan peta diperlukan sebagai informasi yang menunjukkan lokasi penelitian dilakukan atau lokasi pengambilan sampel organisme. Peta untuk manuskrip dapat dibuat dengan menggunakan berbagai macam program (software), seperti ArcGIS dan QGIS. Walaupun saya pernah menggunakan program berbayar dalam analisis spasial, kita lebih memilih menggunakan program gratisan karena untuk saat ini belum mampu membeli lisensi program berbayar.

Salah satu program gratisan yang juga memiliki kemampuan untuk membuat peta adalah R. Program ini mampu membuat peta yang tidak kalah informatif dan menarik. Melalui tulisan ini, berharap kode R untuk membuat peta ini akan tersedia secara daring sehingga tidak akan lupa dan mungkin saja dapat digunakan oleh mahasiswa, dosen atau peneliti untuk membuat peta yang akan ditampilkan di skripsi, tesis, disertasi atau manuskrip untuk jurnal nasional maupun internasional.

Jika tertarik untuk menggunakan kode R di panduan ini, disarankan untuk memasang beberapa paket R ini terlebih dahulu, yaitu ggplot2, ggmap, ggsn dan ggthemes. Kode R untuk membuat peta ini bisa disesuaikan jika akan membuat peta di luar wilayah Jawa Timur. Petunjuk untuk penyesuaian disampaikan di awal setiap bagian kode. Terakhir, pastikan memiliki akses internet yang cepat untuk mengunduh peta dari maps.stamen.com.

Memuat Paket:

```
Library(ggplot2) # Paket untuk visualisasi data
## Warning: package 'ggplot2' was built under R version 3.6.1
Library(ggmap) # Paket untuk membuat peta
## warning: package 'ggmap' was built under R version 3.6.1
## Google's Terms of Service: https://cloud.google.com/maps-
platform/term/.
## Please cite ggmap if you use it! See citation ("ggmap")for
details.
Library(ggsn) # Paket untuk menambahkan symbol utara dan balok
skala
## Warning: package 'ggsn' was built under R version 3.6.1
## Loading required package: grid # Paket untuk menambah tema
peta
```

6.2 Membuat titik data spasial yang bisa dibaca oleh R

Dalam panduan ini, titik data yang akan dibuat berada di Nanga Leboyan, sekitar wilayah Danau Sentarum, Taman Nasional Betung Kerihun - Danau Sentarum, Kalimantan Barat. Jika akan membuat titik data spasial di daerah lain, sesuaikan koordinat latitude dan longitude dari kode di bawah ini:

```
> library(ggplot2)
> library(ggmap)
Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
Please cite ggmap if you use it! See citation("ggmap") for details.
> library(ggsn)
Warning: package 'ggsn' was built under R version 3.6.3
Loading required package: grid
> site = c("Jombang District")
> lat = c(-7.13)
> lon = c(112.12)
> data = cbind.data.frame(site, lon, lat)
> print(data)
             site    lon   lat
1 Jombang District 112.12 -7.13
```

6.3 Mengunduh peta dari Stamen Maps

Setelah dutentukan titik koordinat dari kota lokasi penelitian, maka kita buat kode untuk mengunduh peta. Selanjutnya kita tentukan batas peta lokasi penelitian sesuai dengan skala peta yang ingin diunduh.

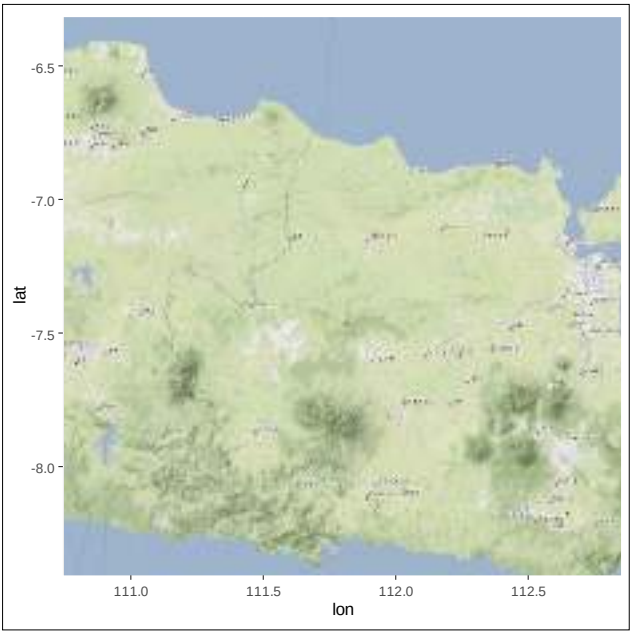
```
# Kode di bawah ini digunakan untuk mengunduh peta dengan
batas latitude dan longitude tertentu (left, bottom, right,
dan top).
# Jika akan membuat peta di luar wilayah jawa Timur,
sesuaikan batas latitude dan longitude dengan cara mengganti
nilai left, bottom, right dan top.
```

```
> myMap <- get_stamenmap(bbox = c(left = 111.05,
+                                 bottom = -8.25,
+                                 right = 112.75,
+                                 top = -6.5),
+            maptype = "terrain",
+            crop = FALSE,
+            zoom = 10)
Source : http://tile.stamen.com/terrain/10/827/530.png
```

```
> myMap
1536x1536 terrain map image from Stamen Maps.
See ?ggmap to plot it.
> ggmap(myMap)
```

Hasil unduhan peta dari coding R ini didapatkan tampilan peta sangat menarik, gambarnya cukup jelas dan berwarna. Gambar peta ini tidak kalah kualitasnya dengan peta yang dihasilkan pada aplikasi map berbayar lainnya. Seperti yang terlihat pada gambar 6.1 merupakan

wilayah penelitian di kabupaten Jombang yang batasan wilayah peta dilihat dari jarak latitude dan longitudenya.



Gambar 71. Peta Penelitian yang berhasil diunduh

Setelah peta berhasil diunduh dengan melalui coding R, maka selanjutnya peta dilengkapi dengan nama lokasi peta, titik-titik lokasi sampel penelitian, arah mata angin, skala peta dan peta inset untuk memberikan informasi lokasi penelitian.

6.4 Menambahkan titik spasial

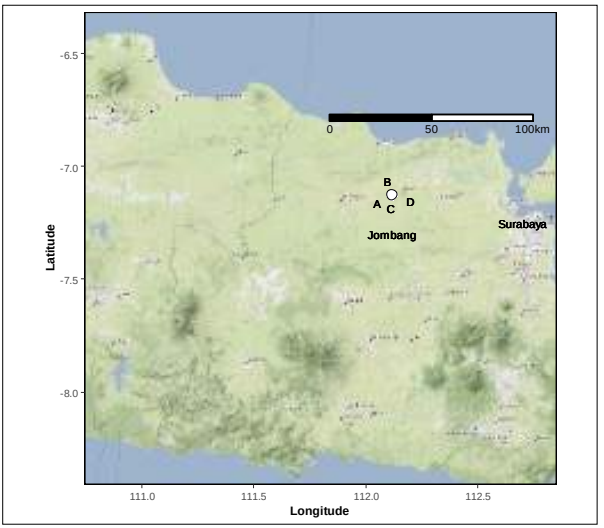
Kode di bawah ini menguraikan cara menambah titik spasial dari dataset di atas ke dalam peta yang telah diunduh. Posisi balok skala (Scalebar) dapat diubah dengan mengganti nilai x.min, x. max, y.min dan y.max.Keberadaan nama kota tertentu dip eta sangat penting khususnya untuk memudahkan pembaca yang awam dengan lokasi penelitian. Nilai x dan y dari kota-kota tertentu (dalam panduan ini Surabaya sebagai Ibukota Propinsi) dapat diubah dengan menyesuaikan latitude dan longitudenya. Google map bisa digunakan untuk membantu menentukan x dan y dari suatu kota.

Pada peta yang ingin dibuat, ada 4 sampel lokasi penelitian yang akan ditentukan dalam peta, yaitu lokasi A, B, C dan D dengan titik-titik

koordinat tertentu. Pada pembuatan peta ini juga ditentukan skala peta yang ingin diunduh.

```
> ggmap(myMap)
> ml <- ggmap(myMap) +
+       geom_point(data=data, aes(x=lon, y=lat), pch=21, size=4, fill="white") +
+       xlab("Longitude") + ylab("Latitude") +
+       annotation_scale(location= "topright", st.bottom= TRUE, st.size = 3, x.min = 111.0,
+             x.max = 112.75,y.min = -6.25, y.max = -6.5,dist = 50, dist_unit = "km",
+             transform = TRUE, model = 'WGS84') +
+       geom_text(x = 112.05, y = -7.16, label = "A", size = 3) +
+       geom_text(x = 112.10, y = -7.26, label = "B", size = 3) +
+       geom_text(x = 112.11, y = -7.19, label = "C", size = 3) +
+       geom_text(x = 112.20, y = -7.25, label = "D", size = 3) +
+       geom_text(x = 112.12, y = -7.30, label = "Jombang", size = 3) +
+       geom_text(x = 112.70, y = -7.25, label = "Surabaya", size = 3) +
+       theme_bw()+
+       theme(axis.text=element_text(size=8),
+             axis.title=element_text(size=10,face="bold"),
+             panel.border = element_rect(colour = "black", fill=NA, size=1))
> ml
```

Bentuk bulatan dapat diubah menjadi bentuk yang lain dengan mengubah kode pch styles. Pilih kode pch styles yang lain adalah 1, 8, 10, 23.

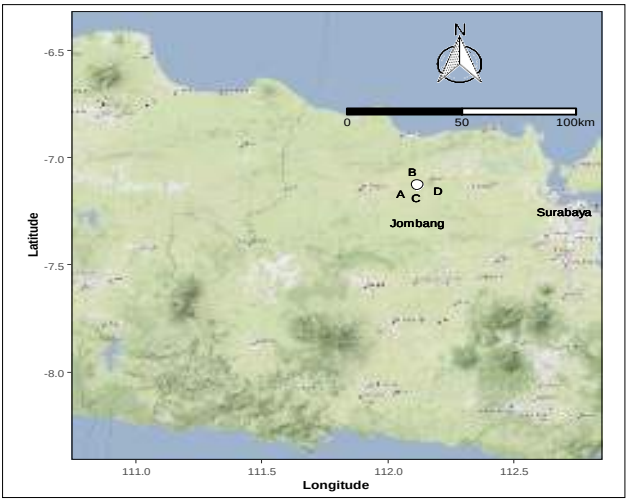


Gambar 72. Peta Penelitian setelah penambahan titik sampel lokasi penelitian dan skala peta.

6.5 Menambahkan Simbul Arah Mata Angin:

Selanjutnya dengan coding north2 yang disalin dalam RGUI, maka dihasilkan symbol arah mata angin dalam peta yang sudah dibangun.

```
> north2(myMap, x = 0.73, y=0.9, scale = 0.14, symbol = 1)
1536x1536 terrain map image from Stamen Maps.
See ?ggmap to plot it.
```

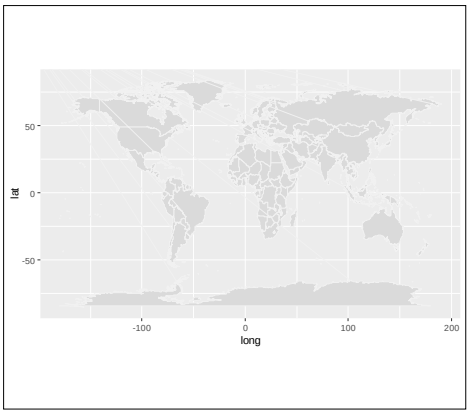


Gambar 73. Peta setelah dilengkapi dengan arah mata angin

6.6 Membuat Peta Insert (Pulau Jawa)

Kode R berikut digunakan untuk mengunduh data latitude dan longitude di dunia.

```
> dunia <- map_data("world")
>
> # Visualisasi data peta dunia yang diunduh
> gl <- ggplot() + geom_polygon(data = dunia, aes(x=long, y = lat, group = group),
+                 fill = "gray65", color = "gray95") + coord_fixed(1.3)
> gl
```



Gambar 74. Hasil Unduhan Peta Dunia

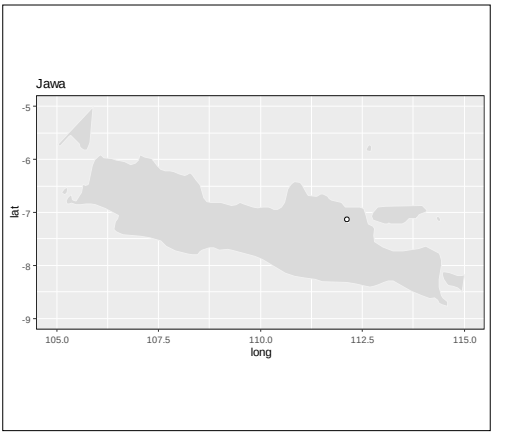
Lalu kita susun kode untuk membuat peta Jawa, jika akan membuat peta insert daerah lain, misalnya Sumatera, Sulawesi, dan

Kalimantan dapat dilakukan dengan mengubah nilai xlim, ylim, dan title pada kode labs.

```
Jawa<-g1 + xlim(105,115) + ylim(-9,-5) + labs(title = "Jawa")
```

Kode ini digunakan untuk menambahkan titik spasial ke dalam peta insert.

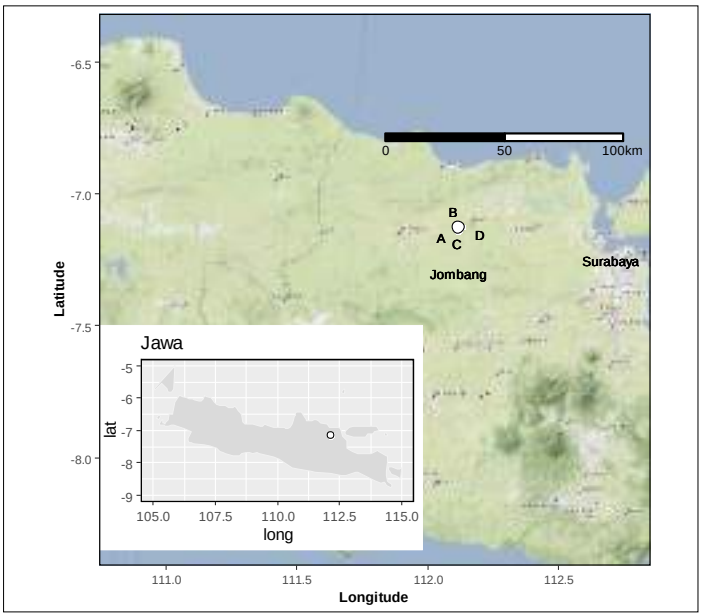
```
M2 <- Jawa + 68eom._point(data=data, aes(x=lon, y=lat), pch=21,
size=2, fill="white") + theme(panel.border = element_rect(colour
 = "black", fill=NA, size=1))
```



Gambar 75. Hasil unduhan peta Jawa dari peta dunia

Masukkan peta inset dan symbol utara, kode di bawah ini digunakan untuk menggabungkan peta detil dan peta insert serta menambahkan symbol utara. Posisi inset dapat diubah dengan menyesuaikan nilai xmin, xmax, ymin, dam ymax.

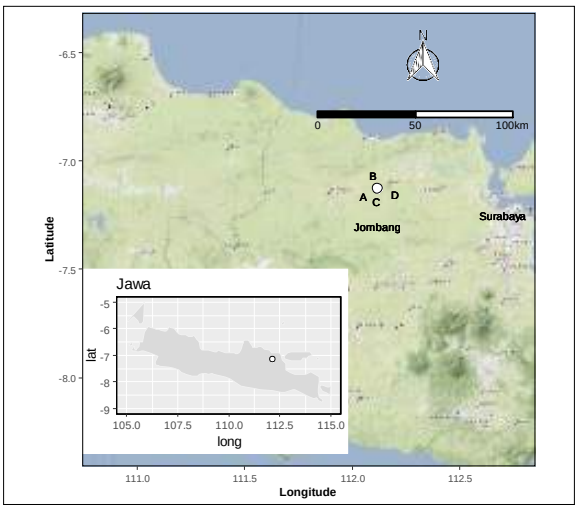
```
peta <- m1 + inset(ggplotGrob(m2), xmin = 110.7, xmax = 112.0, y
min = -7.5, ymax = -8.35)
peta
```



Gambar 76. Hasil inset peta Jawa pada peta penelitian (Jombang Distric)

Selanjutnya masukkan symbol arah mata angin, dengan memberikan coding berikut;

```
> north2 (peta, x = 0.73, y=0.9, scale = 0.12, symbol = 1)
```



Gambar 77. Gambar peta penelitian lengkap

Hasil akhir dari pembuatan peta penelitian ini terlihat sudah dilengkapi dengan inset peta pulau Jawa, titik lokasi sampel penelitian, skala peta dan arah angin. Hal ini sangat membantu bagi peneliti untuk memberikan informasi yang lengkap tentang lokasi penelitian.

BAB VII
ANALISIS MULTIVARIAT KOMUNITAS EKOLOGIS
DENGAN PROGRAM R: PAKET VEGAN

7.1. Pengantar

Langkah awal analisis multivariate dengan program R menggunakan paket Vegan adalah mempersiapkan data yang kita import dari Excel. Ada dua file Excel yang berisi data multivariate yang digunakan dalam buku ini yaitu dataset **fruit2** dan **geo2,** lalu lakukan pengecakan data apakah data tersebut sudah terimport kedalam program R. Program R yang digunakan berbasis BiodiversityR. Perlu diingat bahwa dari dua dataset tersebut harus memiliki jumlah baris yang sama, karena sistem analysis dari paket Vegan menggunankan transformasi matrik.

(a)

(b)

Gambar 78. Pengecekan ketersediaan data dalam program BiodiversityR untuk dataset **fruit2** sebagai variabel spesies (a) dan **geo2** sebagai variabel lingkungan(b)

Paket Vegan untuk analisis data multivariat ini menunjukkan alur kerja khas dalam analisis penahbisan multivariat komunitas biologis. Langkah pertama membahas analisis dasar tak terbatas dan interpretasi lingkungan dari hasilnya. Kemudian memperkenalkan pentahbisan terbatas menggunakan analisis korespondensi terbatas sebagai contoh: metode alternatif seperti analisis terbatas dari pendekatan dan analisis redundansi dapat digunakan (hampir) sama. Terakhir menjelaskan analisis spesies dan hubungan lingkungan tanpa penahbisan, dan brie y menyentuh klasifikasi komunitas.

Contoh-contoh dalam tutorial ini diuji: Ini adalah dokumen Sweave. File sumber asli hanya berisi teks dan perintah R: output dan grafiknya dihasilkan saat menjalankan sumber melalui Sweave. Namun, kita mungkin memerlukan versi vegan terbaru. Dokumen ini dibuat menggunakan vegan versi 3.6.1. Analisis ini mencakup metode penahbisan dalam Vegan. Itu tidak membahas banyak metode lain dalam Vegan. Misalnya, ada beberapa fungsi untuk analisis keanekaragaman hayati: indeks keanekaragaman (keanekaragaman, renyi, fisher.alpha), kekayaan spesies yang diekstrapolasi (specpool, estimasiR),kurva akumulasi spesies (specaccum), model kelimpahan spesies (radfit, fisherfit, prestonfit) dll. Baik Vegan, satu-satunya paket R untuk penahbisan komunitas ekologis. Basis R memiliki alat statistik standar, labdsv melengkapi Vegan dengan beberapa metode canggih dan menyediakan versi alternatif dari beberapa metode, dan ade4 menyediakan implementasi alternatif untuk seluruh program metode ordinasi.

Proses analiis menjelaskan hanya metode yang paling penting dan menunjukkan alur kerja yang khas. Saya melihat penahbisan terutama sebagai alat grafis, dan saya tidak menunjukkan hasil numerik yang terlalu tepat. Sebagai gantinya, ada sketsa kecil hasil plot di margin dekat dengan tempat Anda melihat perintah plot. Saya menyarankan Anda mengulangi analisis, mencoba berbagai alternatif dan memeriksa hasilnya dengan lebih saksama di waktu luang Anda. Fungsi-fungsi dijelaskan hanya secara singkat, dan sangat berguna untuk memeriksa halaman bantuan yang sesuai untuk penjelasan metode yang lebih menyeluruh. Metode juga hanya dijelaskan secara singkat. Yang terbaik adalah berkonsultasi dengan buku teks tentang metode penahbisan, atau ceramah saya, untuk latar belakang teoritis pertama.

7.2. Ordinasi: Metode Dasar

7.2.1 Metode NMDS (*Nonmatric Multidimensional Scaling*)

Penskalaan multidimensi non-metrik dapat dilakukan menggunakan fungsi isoMDS dalam paket MASS. Fungsi ini membutuhkan perbedaan sebagai input. Fungsi vegdist dalam vegan mengandung perbedaan yang ditemukan baik dalam ekologi komunitas. Standarnya adalah ketidaksamaan Bray-Curtis, yang sekarang sering dikenal dengan Steinhaus dissimilarity, atau di Finlandia disebut Sorensen index. Langkah-langkah dasarnya adalah:

```
> BiodiversityRGUI()
Loading required namespace: diamo
Sourced: BiodiversityGUI.R
Error in structure(.External(.C_dotTcl, ...), class = "tclObj") :
  [tcl] bad window path name ".3".

> library(vegan)
> library(MASS)
> data(fruit2)
```

```
> vare.dis <- vegdist(fruit2)
> vare.mds0 <- isoMDS(vare.dis)
initial  value 29.854298
iter   5 value 22.568619
final  value 21.401072
converged
>
```

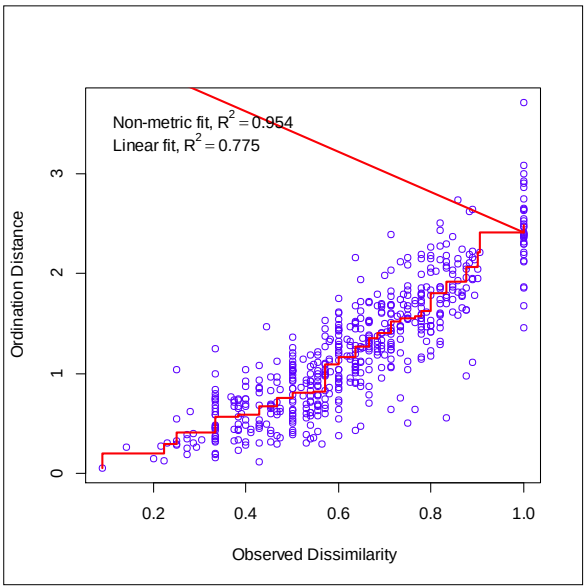
Standarnya adalah menemukan dua dimensi dan menggunakan penskalaan metrik (cmdscale) sebagai solusi awal. Solusinya adalah iteratif, seperti yang dapat dilihat dari informasi penelusuran (ini dapat ditekan dengan melacak jejak = F). Hasil isoMDS adalah daftar (poin item, stres) untuk konguration dan stres. Stres S adalah statistik kebaikan t, dan itu adalah fungsi dan transformasi monoton non-linear dari perbedaan yang diamati (d) dan jarak pentahbisan ~ d. Peta NMDS mengamati perbedaan komunitas secara nonlinier ke ruang pentahbisan dan dapat menangani respons spesies nonlinier dalam bentuk apa pun.

$$S = \sqrt{\frac{\sum_{i \neq j} \left[\theta(d_{ij}) - \tilde{d}_{ij} \right]^2}{\sum_{i \neq j} \tilde{d}_{ij}^2}}$$

Kita dapat memeriksa pemetaan menggunakan fungsi Shepard dalam paket MASS, atau pembungkus stres sederhana di vegan:

```
> stressplot(vare.mds0, vare.dis)
```

Function stress plot menggambar plot *Shepard* di mana jarak penahbisan diplot terhadap perbedaan komunitas, dan t ditunjukkan sebagai garis langkah monoton. Selain itu, stressplot menunjukkan dua korelasi seperti statistik kebaikan t. Korelasi berdasarkan stres adalah $R^2 = 1- S^2$.

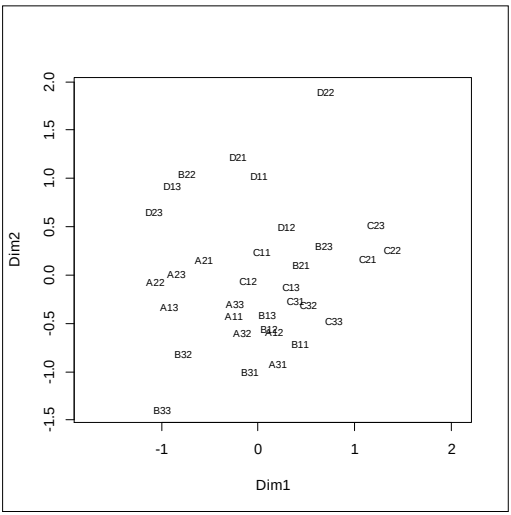


Gambar 79. Output dari fungsi Stressplot

R^2 berbasis \ "adalah korelasi antara nilai fitted (d) dan jarak pentahbisan ~ d, atau antara garis langkah dan titik. Ini harus linier bahkan ketika t sangat melengkung dan sering dikenal sebagai \ linear t ". Kedua korelasi ini keduanya didasarkan pada residu dalam plot Shepard, tetapi keduanya berbeda dalam model nolnya. Pada t linier, model nol adalah bahwa semua jarak pentahbisan adalah sama, dan t adalah pada garis horizontal. Ini kedengarannya masuk akal, tetapi Anda membutuhkan dimensi N 1 untuk model nol titik N, dan model nol ini secara geometris tidak mungkin dalam ruang pentahbisan. Tegangan dasar menggunakan model nol di mana semua pengamatan diletakkan di titik yang sama , yang secara geometris dimungkinkan.

Akhirnya kata peringatan: Anda kadang-kadang melihat bahwa orang menggunakan korelasi antara perbedaan komunitas dan jarak penahbisan. Ini berbahaya dan menyesatkan karena NMDS adalah metode nonlinier: peningkatan koordinasi dengan hubungan yang lebih nonlinier akan tampak lebih buruk dengan kriteria ini.Skor fungsi dan ordiplot dalam vegan dapat digunakan untuk menangani hasil NMDS:

```
> ordiplot(vare.mds0, type = "t")
Species scores not available
```

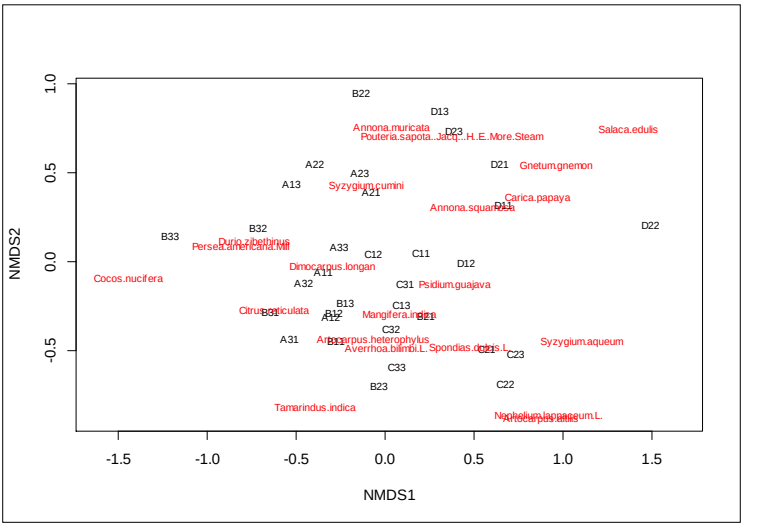


Gambar 80. Output dari fungsi ordiplot

Hanya skor situs yang ditunjukkan, karena perbedaan tidak memiliki informasi tentang spesies. Pencarian berulang sangat sulit di NMDS, karena hubungan nonlinear antara penahbisan dan perbedaan asli. Iterasi dengan mudah terperangkap ke dalam optimum lokal alih-alih menemukan optimum global. Oleh karena itu disarankan untuk menggunakan beberapa start acak, dan pilih di antara solusi serupa dengan tekanan terkecil. Ini mungkin membosankan, tetapi vegan memiliki fungsi metaMDS yang melakukan ini, dan banyak hal lainnya. Output penelusuran panjang, dan kami menekannya dengan jejak = 0, tetapi biasanya kami ingin melihat sesuatu terjadi, karena analisisnya dapat memakan waktu lama:

```
> vare.mds <- metaMDS(fruit2, trace = FALSE)
> vare.mds

Call:
metaMDS(comm = fruit2, trace = FALSE)

global Multidimensional Scaling using monoMDS

Data:     fruit2
Distance: bray

Dimensions: 2
Stress:     0.1839458
Stress type 1, weak ties
No convergent solutions - best solution after 20 tries
Scaling: centring, PC rotation, halfchange scaling
Species: expanded scores based on 'fruit2'

> plot(vare.mds, type = "t")
```

Out put dari analisis ini seperti ditunjukkan pada gambar berikut:



Gambar 81. Output dari hubungan lokasi (A – D) dengan spesies pohon buah.

Kami tidak menghitung perbedaan dalam langkah terpisah, tetapi kami memberikan matriks data asli sebagai input. Hasilnya lebih rumit dari sebelumnya, dan memiliki beberapa komponen selain yang ada di isoMDS hasil: nobj, nfix, ndim, ndp, ngrp, diss, iidx, jidx, xinit, istart, isform, ities, iregn, iscal, maxits, sratmx, strmin, sfgrmn, dist, dhat, points, stress, grstress, iters, icause, call, model, distmethod, distcall, data, jarak, konvergen, percobaan, mesin, spesies. Fungsi ini membungkus prosedur yang disarankan menjadi satu perintah. Jadi apa yang terjadi di sini?

1. Kisaran nilai data begitu besar sehingga data ditransformasikan dengan akar kuadrat, dan kemudian diserahkan ke Wisconsin standardisasi ganda, atau spesies dibagi dengan maksimalnya, dan berdiri standar untuk total yang sama. Kedua standardisasi ini sering meningkatkan kualitas penahbisan, tetapi kami lupa untuk memikirkannya dalam analisis awal.
2. Fungsi yang digunakan ketidaksamaan Bray-Curtis.
3. Fungsi jalankan isoMDS dengan beberapa mulai acak dan berhenti baik setelah sejumlah percobaan, atau setelah menemukan dua konfigurasi serupa dengan tekanan minimum. Bagaimanapun, itu mengembalikan solusi terbaik.
4. Fungsi memutar solusi sehingga varian situs terbesar skor akan berada di sumbu pertama.

5. Fungsi meningkatkan skala solusi sehingga satu unit sesuai dengan separuh kesamaan komunitas dari kesamaan tiruan.
6. Fungsi menemukan skor spesies sebagai rata-rata tertimbang dari skor situs, tetapi memperluasnya sehingga spesies dan skor situs memiliki varian yang sama. Perluasan ini dapat diurungkan menggunakan shrink = TRUE dalam perintah tampilan.

Halaman bantuan untuk metaMDS akan memberikan rincian lebih lanjut, dan arahkan ke penjelasan fungsi yang digunakan dalam fungsi tersebut.

7.2.2 Perbedaan Komunitas

Penskalaan multidimensi non-metrik adalah metode penahbisan yang baik karena dapat menggunakan cara-cara yang bermakna secara ekologis untuk mengukur perbedaan komunitas. Ukuran perbedaan yang baik memiliki hubungan urutan peringkat yang baik dengan jarak sepanjang gradien lingkungan. Karena nmds hanya menggunakan informasi peringkat dan peta memberi peringkat non-linear ke ruang pentahbisan, nmds dapat menangani respons spesies non-linear dalam bentuk apa pun dan secara efektif dan kuat menemukan gradien yang mendasarinya.

Ukuran perbedaan yang paling alami adalah jarak Euclidean secara inheren digunakan oleh metode penahbisan eigen. Ini adalah jarak dalam ruang spesies. Ruang spesies berarti bahwa setiap spesies merupakan poros ortogonal untuk semua spesies lain, dan situs adalah titik dalam ruang multidimensi ini. Namun, jarak Euclidean didasarkan pada perbedaan kuadrat dan sangat didominasi oleh perbedaan besar tunggal. Kebanyakan perbedaan yang secara ekologis bermakna adalah tipe Manhattan, dan menggunakan perbedaan daripada perbedaan kuadrat. Ciri lain dalam indeks perbedaan yang baik adalah bahwa mereka proporsional: jika dua komunitas tidak memiliki spesies, mereka memiliki perbedaan maksimum = 1. Perbedaan Euclidean dan Manhattan akan bervariasi sesuai dengan kelimpahan total meskipun tidak ada spesies bersama.

Paket vegan memiliki fungsi vegdist dengan indeks Bray-Curtis, Jaccard dan Kulczy nski. Semua ini adalah tipe Manhattan dan hanya menggunakan istilah pesanan pertama (jumlah dan perbedaan), dan semuanya din Relativiasi dengan total lokasi dan mencapai nilai maksimumnya (1) ketika tidak ada spesies bersama antara dua komunitas yang dibandingkan. Function vegdist adalah pengganti drop-

in untuk dist fungsi R standar, dan salah satu dari fungsi ini dapat digunakan dalam analisis perbedaan.

Ada banyak aspek membingungkan dalam indeks ketidaksamaan. Salah satunya adalah bahwa indeks yang sama dapat ditulis dengan persamaan tampak sangat berbeda: dua formulasi alternatif perbedaan Manhattan di margin berfungsi sebagai contoh. Komplikasi lain adalah penamaan. Function vegdist menggunakan nama sehari-hari yang mungkin tidak sepenuhnya benar. Indeks default di vegan disebut Bray (atau Bray-Curtis), tetapi mungkin harus disebut indeks Steinhaus. Di sisi lain, nama yang benar seharusnya adalah indeks Czekanowski beberapa tahun yang lalu (tapi sekarang ini dianggap sebagai indeks lain), dan itu juga dikenal sebagai indeks Sorensen (tetapi biasanya salah eja). Sebenarnya, indeks Jaccard adalah biner, dan varian kuantitatif dalam vegan harus disebut indeks Ruzicka. Namun, vegan menemukan varian kuantitatif atau biner dari indeks apa pun dengan nama yang sama.

Ketiga indeks dasar ini dianggap baik dalam mendeteksi gradien. Selain itu, fungsi vegdist memiliki indeks yang harus memenuhi kriteria lain. Indeks Morisita, Horn-Morisita, Raup-Cric, Binomial dan Mountford harus dapat membandingkan unit sampel dengan ukuran yang berbeda. Indeks Euclidean, Canberra dan Gower harus memiliki sifat teoritis yang lebih baik.

Fungsi metaMDS menggunakan ketidaksamaan Bray-Curtis sebagai standar, yang biasanya merupakan pilihan yang baik. Indeks Jaccard (Ruzicka) memiliki urutan peringkat yang sama, tetapi memiliki sifat metrik yang lebih baik, dan mungkin lebih disukai.

Indeks peringkat fungsi di vegan dapat digunakan untuk mempelajari indeks mana yang memisahkan komunitas terbaik di sepanjang gradien yang diketahui menggunakan korelasi peringkat sebagai default. Contoh berikut menggunakan semua variabel lingkungan dalam himpunan data geo2, tetapi menstandarkan ini ke varian unit:

```
> data(geo2)
```

```
> rankindex(scale(geo2), fruit2, c("euc","man","bray","jac","kul"))
      euc       man      bray       jac       kul
0.2779183 0.2941919 0.3623004 0.3623004 0.3746298
```

Terkait non-linear, tetapi mereka memiliki urutan peringkat yang sama, dan korelasi peringkat mereka identik. Secara umum, tiga indeks yang direkomendasikan cukup sama.Perlu diambil pendekatan yang sangat praktis pada indeks yang menekankan kemampuan mereka untuk memulihkan gradien lingkungan yang mendasarinya. Banyak buku teks

menekankan sifat metrik indeks. Ini penting dalam beberapa metode, tetapi tidak dalam NMDS yang hanya menggunakan informasi urutan peringkat. Properti metrik hanya mengatakan itu:
1. jika dua situs identik, jaraknya nol,
2. jika dua situs berbeda, jaraknya lebih besar dari nol,
3. jarak simetris, dan
4. jarak terpendek antara dua situs adalah garis, dan Anda tidak bisa tingkatkan dengan mengunjungi situs lain.

Ini semua kedengarannya kondisi yang sangat alami, tetapi tidak sepenuhnya dipenuhi oleh semua perbedaan. Sebenarnya, hanya jarak Euclidean - dan mungkin indeks Jaccard - memenuhi semua kondisi di antara perbedaan yang dibahas di sini, dan merupakan metrik. Banyak perbedaan lain memenuhi tiga kondisi pertama dan bersifat semimetrik.

Ada sebuah studi yang mengatakan bahwa kita harus menggunakan indeks metrik, dan yang paling alami, jarak Euclidean. Salah satu kelemahan mereka adalah mereka tidak memiliki batas tetap, tetapi dua situs tanpa spesies yang sama dapat bervariasi dalam perbedaan, dan bahkan terlihat lebih mirip daripada dua situs yang berbagi beberapa spesies. Ini dapat disembuhkan dengan standarisasi data. Karena jarak Euclidean didasarkan pada perbedaan kuadrat, transformasi alami adalah untuk menstandardisasi situs dengan jumlah kuadrat yang sama, atau dengan norma vektornya menggunakan fungsi decostand:

```
> dis <- vegdist(decostand(fruit2, "norm"), "euclid")
```

Ini memberikan jarak chord yang mencapai batas maksimum $\sqrt{2}$ ketika tidak ada spesies bersama antara dua situs. Alternatif lain yang direkomendasikan adalah jarak Hellinger yang didasarkan pada akar kuadrat dari situs yang distandarisasi untuk unit total:

```
> dis <- vegdist(decostand(fruit2, "hell"), "euclidean")
```

Meskipun ada standardisasi, ini masih merupakan jarak Euclidean dengan semua sifatnya yang baik, tetapi untuk data yang diubah. Sebenarnya, seringkali berguna untuk mengubah atau membakukan data bahkan dengan indeks lain. Jika ada perbedaan besar antara jumlah terkecil non-nol dan jumlah terbesar, kami ingin mengurangi perbedaan ini. Biasanya transformasi akar kuadrat cukup untuk menyeimbangkan data. Wisconsin standardisasi ganda sering meningkatkan kemampuan deteksi gradien indeks ketidaksamaan; ini dapat dilakukan dengan menggunakan perintah wisconsin di vegan. Di sini kita pertama-tama

membagi semua spesies dengan maksimalnya, dan kemudian membakukan lokasi menjadi satuan total.

Setelah standarisasi ini, banyak indeks ketidaksamaan menjadi identik dalam urutan peringkat dan harus memberikan hasil yang sama dalam nmds.

Anda tidak dibatasi untuk hanya menggunakan indeks vegdist di vegan: vegdist mengembalikan struktur ketidaksamaan yang sama sebagai dist fungsi R standar yang juga dapat digunakan, serta fungsi kompatibel lainnya dalam paket apa pun. Beberapa fungsi yang kompatibel adalah dsvdis (paket labdsv), daisy (paket cluster), dan jarak (paket analog), dan indeks keanekaragaman beta di betadiver dalam vegan. Terlebih lagi, vegan memiliki daftar fungsi di mana Anda dapat menentukan indeks ketidaksamaan Anda sendiri dengan menulis persamaannya menggunakan notasi untuk A, B dan J di atas, atau dengan data biner, notasi tabel kontingensi 2 x 2 di mana a adalah jumlah spesies yang ditemukan pada kedua situs yang dibandingkan, dan b dan c adalah jumlah spesies yang hanya ditemukan di salah satu situs. Tiga persamaan berikut menunjukkan indeks Sorensen yang sama di mana jumlah spesies yang dibagi dibagi dengan kekayaan spesies rata-rata dari situs yang dibandingkan:

```
> d <- vegdist(fruit2, "bray", binary = TRUE)
> d <- designdist(fruit2, "(A+B-2*J)/(A+B)")
> d <- designdist(fruit2, "(b+c)/(2*a+b+c)", abcd=TRUE)
```

Fungsi betadiver baik mendefinisikan beberapa indeks ketidaksamaan yang lebih biner dalam vegan.

$$d_{jk} = \sqrt{\sum_{i=1}^{N}(x_{ij} - x_{ik})^2} \quad \text{Euclidean}$$

$$d_{jk} = \sum_{i=1}^{N}|x_{ij} - x_{ik}| \quad \text{Manhattan}$$

$$A = \sum_{i=1}^{N} x_{ij} \qquad B = \sum_{i=1}^{N} x_{ik}$$

$$J = \sum_{i=1}^{...} \min(x_{ij}, x_{ik})$$

$$d_{jk} = A + B - 2J \qquad \text{Manhattan}$$

$$d_{jk} = \frac{A + B - 2J}{A + B} \qquad \text{Bray}$$

$$d_{jk} = \frac{A + B - 2J}{A + B - J} \qquad \text{Jaccard}$$

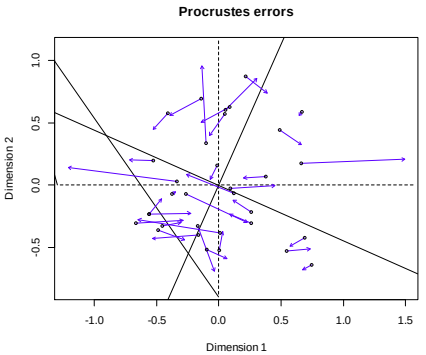
$$d_{jk} = 1 - \frac{1}{2}\left(\frac{J}{A} + \frac{J}{B}\right) \qquad \text{Kulczyński}$$

Kebanyakan indeks perbedaan yang dipublikasikan dapat dinyatakan sebagai formula designdist. Namun, jauh lebih mudah dan aman untuk menggunakan alternatif kalengan dalam fungsi yang ada: sangat mudah untuk membuat kesalahan dalam menulis persamaan ketidaksamaan.

7. 2.3 Membandingkan pentahbisan: Perputaran procrustes

Dua penahbisan bisa sangat mirip, tetapi ini mungkin sulit dilihat, karena sumbu memiliki orientasi dan penskalaan yang sedikit berbeda. Sebenarnya, dalam nmds tanda, orientasi, skala dan lokasi sumbu tidak ditentukan, meskipun metaMDS menggunakan metode sederhana untuk x tiga komponen terakhir. Cara terbaik untuk membandingkan pentahbisan adalah dengan menggunakan rotasi Procrustes. Rotasi procrustes menggunakan penskalaan yang seragam (ekspansi atau kontraksi) dan rotasi untuk meminimalkan perbedaan kuadrat antara dua ordinasi. Paket vegan memiliki fungsi procrustes untuk melakukan analisis Procrustes. Berapa banyak yang kita peroleh dengan menggunakan metaMDS daripada isoMDS default?

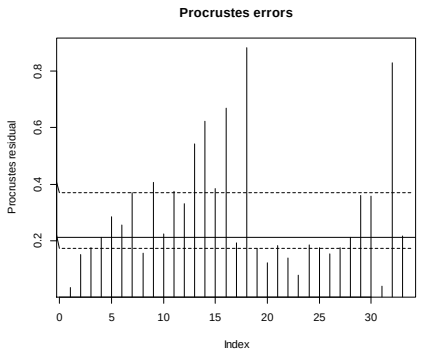
```
> tmp <- wisconsin(sqrt(fruit2))
> dis <- vegdist(tmp)
> vare.mds0 <- isoMDS(dis, trace = 0)
> pro <- procrustes(vare.mds, vare.mds0)
> pro

Call:
procrustes(X = vare.mds, Y = vare.mds0)

Procrustes sum of squares:
4.236

> plot(pro)
```



Gambar 82. Output Hasil Perputaran Procuster

Dalam hal ini, perbedaannya cukup kecil, dan terutama menyangkut dua poin. Anda dapat menggunakan fungsi identifikasi untuk mengidentifikasi titik-titik tersebut dalam sesi interaktif, atau Anda dapat menanyakan plot perbedaan residual saja:

```
> plot(pro, kind = 2)
```



Gambar 83. Output Procuster error

Statistik deskriptif adalah \ Procrustes jumlah kuadrat "atau jumlah panah kuadrat dalam plot Procrustes. Rotasi procrustes adalah non-simetris, dan statistik akan berubah dengan membalik urutan urutan penahbisan dalam panggilan. Dengan argumen symmetric = TRUE, keduanya solusinya pertama diskalakan ke varian unit, dan statistik yang lebih mandiri dan berskala ditemukan (sering dikenal sebagai Procrustes m^2).

7.2.4 Metode Vektor Eigen

Penskalaan multidimensi non-metrik adalah tugas yang sulit, karena segala bentuk ketidaksamaan dapat digunakan dan ketidaksamaan dipetakan secara nonlinier menjadi tahbisan. Jika kita hanya menerima jenis ketidaksamaan tertentu dan membuat pemetaan linier, penahbisan menjadi tugas sederhana rotasi dan proyeksi. Dalam hal ini kita dapat menggunakan metode vektor eigen. Analisis komponen utama (PCA) dan analisis korespondensi (CA) adalah metode vektor eigen yang paling penting dalam penahbisan komunitas. Selain itu, analisis koordinat utama a.k.a. metrik scaling (NMDS) digunakan sesekali. PCA didasarkan pada jarak Euclidean, CA didasarkan pada jarak Chi-square, dan koordinat utama (PC) dapat menggunakan perbedaan apa pun (tetapi dengan jarak Euclidean sama dengan PCA).

method	metric	mapping
NMDS	any	nonlinear
MDS	any	linear
PCA	Euclidean	linear
CA	Chi-square	weighted linear

$$d_{jk} = \sqrt{\sum_{i=1}^{N} (x_{ij} - x_{ik})^2}$$

PCA adalah metode statistik standar, dan dapat dilakukan dengan fungsi dasar R prcomp atau prinsip. Analisis korespondensi (CA) tidak ada di mana-mana, tetapi ada beberapa implementasi alternatif untuk itu juga. Dalam tutorial ini saya menunjukkan bagaimana menjalankan analisis ini dengan fungsi vegan rda dan cca yang sebenarnya dirancang untuk analisis terbatas.

Analisis komponen utama (PCA) dapat dijalankan sebagai:

```
> vare.pca <- rda(fruit2)
> vare.pca
Call: rda(X = fruit2)

              Inertia Rank
Total           6.854
Unconstrained   6.854   21
Inertia is variance

Eigenvalues for unconstrained axes:
   PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
1.9348 1.3848 0.9706 0.7127 0.4211 0.2827 0.2464 0.1695
(Showing 8 of 21 unconstrained eigenvalues)
```

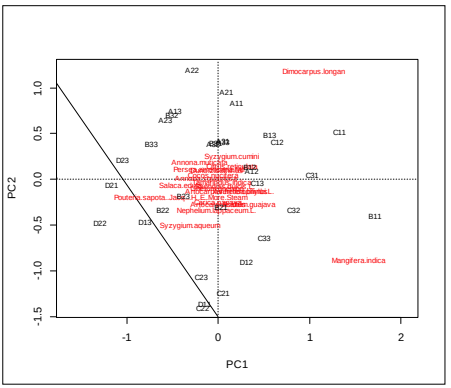
Output mengatakan bahwa inersia total adalah 6.85, dan inersia adalah varians. Jumlah dari semua 21 (peringkat) nilai eigen akan sama dengan inersia total. Dengan kata lain, solusi menguraikan total varians menjadi komponen linier. Kita dapat dengan mudah melihat bahwa varians sama dengan inersia:

```
> sum(apply(fruit2, 2, var))
[1] 6.854167
```

Fungsi berlaku, gunakan var fungsi atau varians untuk dimensi 2 atau kolom (spesies), dan kemudian jumlah mengambil jumlah nilai-nilai ini. Inersia adalah jumlah dari semua varian spesies. Nilai-nilai eigen merangkum inersia total. Dengan kata lain, mereka masing-masing \ menjelaskan "proporsi tertentu dari total varian. Sumbu pertama \ menjelaskan" 1.9348/6.85417 = 28.228 % dari total varian, sedangkan sumbu kedua menjelaskan "1.3848/6.854 17= 20,203%.

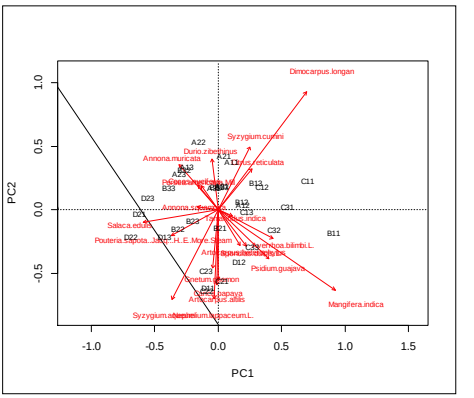
Perintah plot penahbisan standar menggunakan titik atau label untuk spesies dan situs. Beberapa orang lebih suka menggunakan panah biplot untuk spesies di pca dan mungkin juga untuk situs. Ada fungsi biplot khusus untuk tujuan ini:

```
> plot(vare.pca)
```



Gambar 84. Output plot vare.pca

```
> biplot(vare.pca, scaling = -1)
```

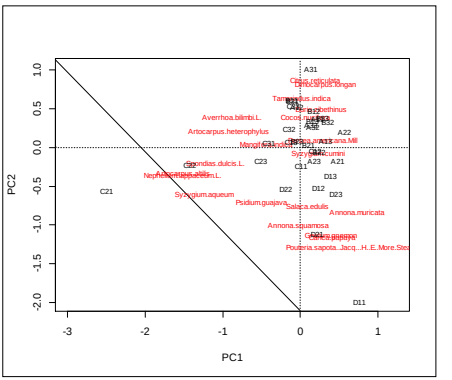


Gambar 85. Output PCA skala -1

Untuk grafik ini, kami menetapkan skala = -1. Hasilnya diskalakan hanya saat diakses, dan kami dapat mengubah penskalaan dalam plot, biplot, dan perintah lainnya secara ekslusif. Nilai negatif berarti bahwa skor spesies dibagi dengan standar deviasi spesies sehingga spesies yang melimpah dan langka akan berada sejauh yang jauh dari asal.

Penahbisan spesies terlihat agak tidak memuaskan: hanya kelengkeng (*Dimocarpus longan*) dan mangga (*Mangifera indica*) terlihat, dan spesies yang lainnya ramai di pusat. Ini terjadi karena inersia adalah varians, dan hanya spesies berlimpah dengan varians tinggi yang layak dijelaskan (tetapi kita bisa menyembunyikan ini dalam plot dengan menetapkan skala negatif). Standarisasi semua spesies untuk varian unit, atau menggunakan koefisien korelasi sebagai ganti kovariansi akan memberikan penahbisan yang lebih seimbang:

```
> vare.pca <- rda(fruit2, scale = TRUE)
> vare.pca
Call: rda(X = fruit2, scale = TRUE)

              Inertia Rank
Total              21
Unconstrained      21   21
Inertia is correlations

Eigenvalues for unconstrained axes:
  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
3.816 3.198 2.355 2.320 1.505 1.464 1.206 1.010
(Showing 8 of 21 unconstrained eigenvalues)

> plot(vare.pca, scaling = 3)
```



Gambar 86. Output plot korelasi antara variabel

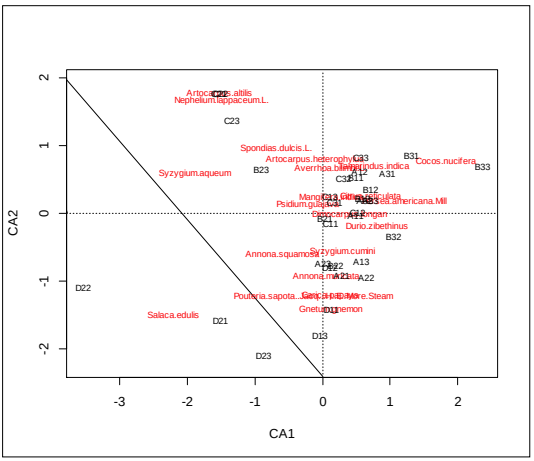
Sekarang inersia adalah korelasi, dan korelasi variabel dengan dirinya sendiri adalah satu. Dengan demikian total inersia sama dengan jumlah variabel (spesies). Pangkat atau jumlah total vektor eigen sama dengan sebelumnya. Peringkat maksimum yang mungkin ditentukan oleh dimensi data: itu adalah satu kurang dari jumlah spesies atau jumlah situs yang lebih kecil.

```
> dim(fruit2)
[1] 33 21
```

Jika ada spesies atau situs yang mirip satu sama lain, peringkat akan berkurang bahkan dari ini. Persentase yang dijelaskan oleh sumbu pertama menurun dari pca sebelumnya. Ini wajar karena sebelumnya kita perlu menjelaskan "hanya spesies yang melimpah dengan varian tinggi, tetapi sekarang kita harus menjelaskan semua spesies secara sama. Kita seharusnya tidak melihat secara membuta pada persentase, tetapi hasilnya kita dapatkan. Analisis korespondensi sangat mirip dengan pca:

```
> vare.ca <- cca(fruit2)
> vare.ca
Call: cca(X = fruit2)

              Inertia Rank
Total           2.784
Unconstrained   2.784   20
Inertia is scaled Chi-square

Eigenvalues for unconstrained axes:
    CA1     CA2     CA3     CA4     CA5     CA6     CA7     CA8
0.4949 0.4280 0.3855 0.2957 0.2197 0.1955 0.1498 0.1087
(Showing 8 of 20 unconstrained eigenvalues)

> plot(vare.ca)
```

Sekarang inersia disebut koefisien kuadrat kontingensi rata-rata. Analisis korespondensi didasarkan pada jarak Chi-kuadrat, dan inersia adalah statistik Chi-kuadrat dari matriks data standar untuk unit total:

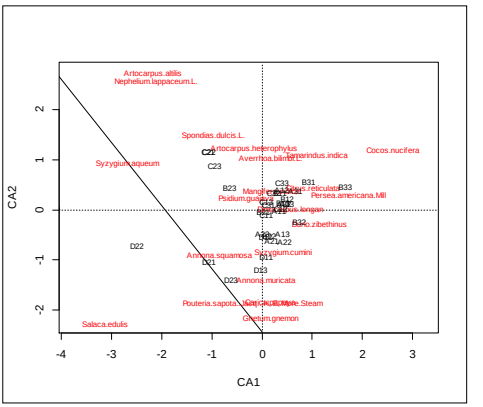


Gambar 87. Out Analisis Korespondensi (CA)

```
> chisq.test(fruit2/sum(fruit2))
Warning in chisq.test(fruit2/sum(fruit2)) :
  Chi-squared approximation may be incorrect

        Pearson's Chi-squared test

data:  fruit2/sum(fruit2)
X-squared = 2.7842, df = 640, p-value = 1
```

Anda seharusnya tidak memperhatikan nilai-P yang tentu saja menyesatkan, tetapi perhatikan bahwa X-squared yang dilaporkan sama dengan inersia di atas. Analisis korespondensi adalah metode rata-rata tertimbang. Dalam grafik di atas skor spesies adalah rata-rata tertimbang dari skor situs. Dengan penskalaan hasil yang berbeda, kami dapat menampilkan skor situs sebagai rata-rata tertimbang dari skor spesies:

```
> plot(vare.ca, scaling = 1)
```



Gambar 88. Output Ca skala 3

Kami telah melihat contoh penskalaan = 3 atau penskalaan simetris dalam pca. Dua bilangan bulat lainnya berarti bahwa salah satu spesies adalah rata-rata tertimbang dari lokasi (2) atau situs adalah rata-rata tertimbang dari spesies (1). Ketika kita mengambil rata-rata tertimbang, kisaran rata-rata menyusut dari nilai aslinya. Faktor susut sama dengan nilai eigen ca, yang memiliki maksimum teoritis 1.

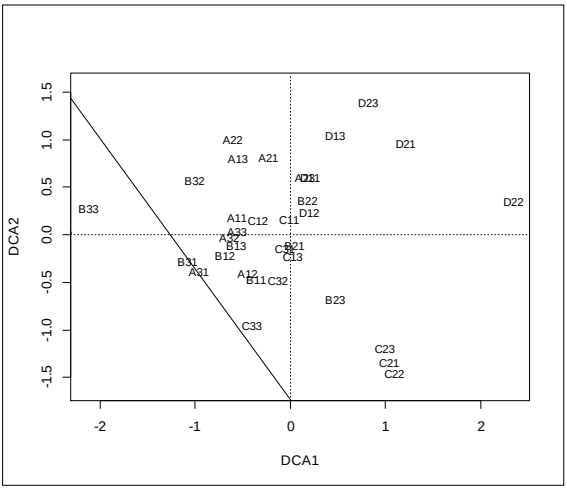
7. 2.5 Analisis Korespondensi Detrended

Analisis korespondensi adalah metode yang jauh lebih baik dan lebih kuat untuk penahbisan komunitas daripada analisis komponen utama. Namun, dengan gradien ekologi yang panjang, ia mengalami beberapa kelemahan atau \ "kesalahan yang diperbaiki dalam analisis korespondensi detrended (dca):

1. Gradien panjang tunggal muncul sebagai kurva atau lengkungan dalam pentahbisan (efek busur): solusinya adalah dengan meringkas sumbu selanjutnya dengan membuat berarti sama di sepanjang segmen sumbu sebelumnya.
2. Situs dikemas lebih dekat pada gradien ekstrem daripada di pusat: solusinya adalah mengubah skala sumbu menjadi varian skor spesies yang sama.
3. Spesies langka tampaknya memiliki pengaruh yang terlalu tinggi pada hasil: solusi untuk spesies langka berat badan.

Ketiga trik terpisah ini tergabung dalam fungsi decoranayang merupakan pelabuhan setia program asli Mark Hill dengan nama yang sama. Penggunaannya sederhana:

```
> vare.dca <- decorana(fruit2)
> vare.dca

Call:
decorana(veg = fruit2)

Detrended correspondence analysis with 26 segments.
Rescaling of axes with 4 iterations.

                 DCA1   DCA2   DCA3   DCA4
Eigenvalues     0.4591 0.4015 0.2360 0.1972
Decorana values 0.4949 0.3076 0.1744 0.1097
Axis lengths    4.4632 2.8426 2.4244 1.9574

> plot(vare.dca, display="sites")
```



Gambar 89. Output plot CCA

Fungsi dekorasi hanya empat sumbu. Nilai eigen didefinisikan sebagai nilai penyusutan dalam rata-rata tertimbang, sama seperti dalam cca di atas. Nilai\Decorana adalah angka yang dikembalikan oleh program asli sebagai\nilai eigen. Paling sering orang mengomentari panjang sumbu, yang kadang-kadang disebut\panjang gradien. Etimologinya tidak jelas: ini bukan gradien, tetapi sumbu penahbisan. Sering dikatakan bahwa jika panjang sumbu lebih pendek dari dua unit, datanya linear. , dan pca harus digunakan. Ini hanya cerita rakyat dan tidak berdasarkan penelitian yang menunjukkan bahwa ca setidaknya sama baiknya dengan pca dengan gradien pendek, dan biasanya lebih baik. Set data saat ini adalah homogen, dan efek dca tidak sangat besar. Dalam data heterogen dengan efek busur jernih, perubahan sering kali

lebih dramatis. Penyelamatan mungkin memiliki pengaruh lebih besar daripada detrending dalam banyak kasus.

Analisis standarnya adalah tanpa menurunkan bobot spesies langka: lihat halaman bantuan untuk argumen yang diperlukan. Sebenarnya, downweight adalah fungsi independen yang dapat digunakan dengan cca juga. Ada aliran pemikiran yang menganggap dca sebagai metode pilihan dalam pentahbisan tak terbatas. Namun, itu tampaknya merupakan trik yang rapuh dan tidak jelas yang lebih baik dihindari.

7.2.6 Grafik Ordinasi

Kami telah melihat banyak diagram pentahbisan dalam tutorial ini dengan satu fitur yang sama: mereka berantakan dan label sulit dibaca. Diagram pentahbisan sulit untuk digambar dengan rapi karena kita harus meletakkan sejumlah besar label dalam plot kecil, dan seringkali tidak mungkin menggambar plot yang bersih dengan semua item berlabel. Dalam bab ini kita akan melihat pembuatan plot yang lebih bersih. Untuk ini, kita harus melihat anatomi fungsi merencanakan dalam vegan dan melihat bagaimana mendapatkan kontrol yang lebih baik dari fungsi default. Fungsi penahbisan di vegan memiliki fungsi plot khusus yang menyediakan plot sederhana. Misalnya, hasil dari decorana ditampilkan oleh function plot.decorana yang di belakang layar disebut oleh fungsi plot kami. Sebagai alternatif, kita dapat menggunakan fungsi ordiplot yang juga berfungsi dengan banyak fungsi penahbisan non-vegan, tetapi menggunakan titik alih-alih teks sebagai default. Fungsi plot.decorana (atau ordiplot) sebenarnya berfungsi dalam tiga tahap:

1. Ini menggambar plot kosong dengan sumbu berlabel, tetapi tanpa simbol untuk situs atau spesies.
2. Menggunakan teks fungsi atau titik untuk menambahkan spesies ke bingkai kosong.
3. Jika pengguna tidak meminta secara spesifik, fungsi akan menggunakan teks dalam kumpulan data kecil dan titik dalam kumpulan data besar.
4. Ia menambahkan situs yang sama.

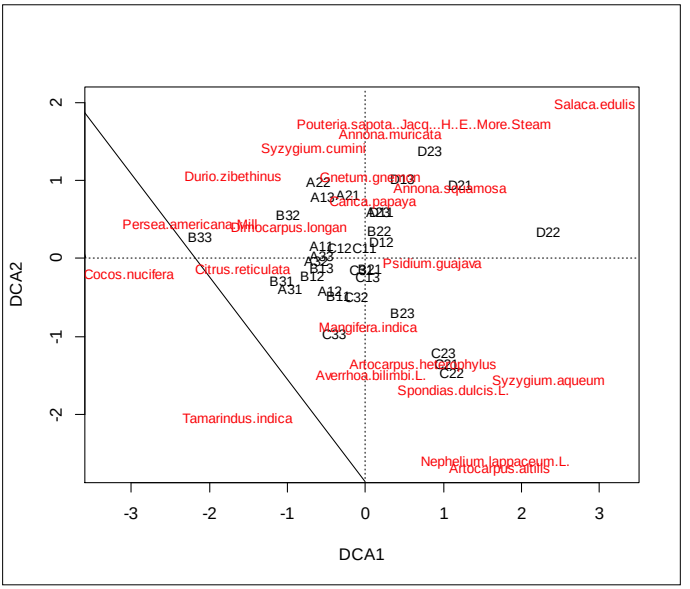
Untuk kontrol plot yang lebih baik, kita harus mengulangi tahap-tahap ini dengan tangan: menggambar plot kosong dan kemudian menambahkan situs dan / atau spesies yang diinginkan.

Dalam bab ini kita mempelajari kasus yang sulit: merencanakan penahbisan biodiversitas pohon buah di lahan kritis.

```
> data(fruit2)
```

Ini adalah kumpulan data yang sulit untuk plot: memiliki 229 spesies dan tidak ada cara untuk memberi label mereka semua dengan bersih {kecuali kita menggunakan area plot yang sangat besar dengan teks kecil. Kita harus menunjukkan hanya pilihan spesies atau bagian kecil plot. Pertama sebuah pentahbisan dengan decorana dan alur awalnya:

```
>  mod <- decorana(fruit2)
>  plot(mod)
```



Gambar 90. Output DCA

Ada masalah tambahan dalam merencanakan penahbisan spesies dengan data ini:

```
>  names(fruit2)[1:21]
 [1] "Annona.squamosa"
 [2] "Annona.muricata"
 [3] "Cocos.nucifera"
 [4] "Salaca.edulis"
 [5] "Durio.zibethinus"
 [6] "Carica.papaya"
 [7] "Gnetum.gnemon"
 [8] "Artocarpus.heterophylus"
 [9] "Artocarpus.altilis"
[10] "Psidium.guajava"
[11] "Syzygium.aqueum"
[12] "Syzygium.cumini"
```

```
[13] "Citrus.reticulata"
[14] "Dimocarpus.longan"
[15] "Mangifera.indica"
[16] "Pouteria.sapota..Jacq...H..E..More.Steam"
[17] "Averrhoa.bilimbi.L."
[18] "Tamarindus.indica"
[19] "Persea.americana.Mill"
[20] "Nephelium.lappaceum.L."
[21] "Spondias.dulcis.L."
```

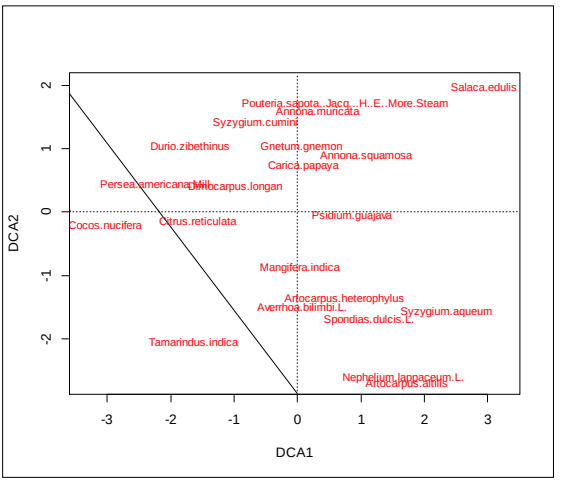
Kumpulan data menggunakan nama spesies lengkap, dan tidak ada cara untuk menyebutkannya dalam grafik penahbisan. Ada fungsi utilitas membuat cepnames di vegan untuk menyingkat nama Latin:

```
> shnam <- make.cepnames(names(fruit2))
> shnam[1:21]
 [1]  "Annosqua"  "Annomuri"  "Coconuci"  "Salaedul"  "Durizibe"
"Caripapa"
 [7]  "Gnetgnem"  "Artohete"  "Artoalti"  "Psidguaj"  "Syzyaque"
"Syzycumi"
[13]  "Citrreti"  "Dimolong"  "Mangindi"  "PoutStea"  "AverL"
"Tamaindi"
[19] "PersMill" "NephL"    "SponL"
```

Cara termudah untuk secara selektif memberi label spesies adalah dengan menggunakan fungsi identifikasi interaktif: ketika Anda mengklik di sebelah suatu titik, labelnya akan muncul di sisi yang Anda klik. Anda dapat menyelesaikan pelabelan dengan mengklik tombol mouse kanan, atau dengan mouse satu tombol yang cacat, Anda dapat menekan tombol esc.

```
>  pl <- plot(mod, dis="sp")
```

Semua fungsi plot penahbisan vegan mengembalikan objek ordiplot yang tidak terlihat yang berisi informasi tentang titik-titik yang diplot. Hasil yang tidak terlihat ini dapat ditangkap dan digunakan sebagai input untuk mengidentifikasi. Selektif berikut memberi label beberapa spesies ekstrem sebagai diklik:



Gambar 91. Output ordiplot

```
>  pl <- plot(mod, dis="sp")
>  identify(pl, "sp", labels=shnam)
```

Ada fungsi \ ordinasi teks atau poin "orditorp di vegan. Fungsi ini akan memberi label suatu item hanya jika ini dapat dilakukan tanpa menimpa label sebelumnya. Jika suatu item tidak dapat dilabeli dengan teks, itu akan ditandai sebagai titik. Item diproses baik dari margin ke tengah, atau dalam urutan prioritas yang menurun. Berikut ini memberikan prioritas lebih tinggi untuk spesies yang paling melimpah:

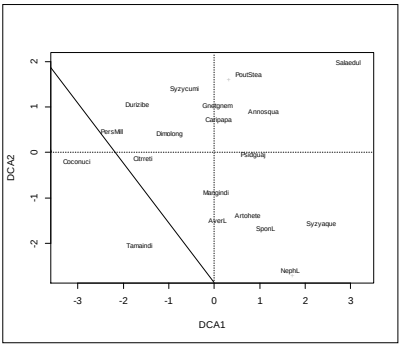
```
>  stems <- colSums(fruit2)
>  plot(mod, dis="sp", type="n")
>    sel <- orditorp(mod, dis="sp", lab=shnam, priority=stems,
pcol = "gray", pch="+")
```

Kita juga dapat memperbesar ke beberapa bagian diagram penahbisan dengan mengatur xlim dan ylim, dan kita dapat melihat lebih detail. Alternatif untuk orditorp adalah fungsi ordilabel yang menggambar teks pada label buram yang menutupi label lain di bawahnya. Semua label tidak dapat ditampilkan, tetapi setidaknya yang paling atas dapat dibaca. Prioritas argumen berfungsi sama seperti pada orditorp dan dapat digunakan untuk memilih label mana yang paling penting untuk ditampilkan:

```
> plot(mod, dis="sp", type="n")
> ordilabel(mod, dis="sp", lab=shnam, priority = stems)
```

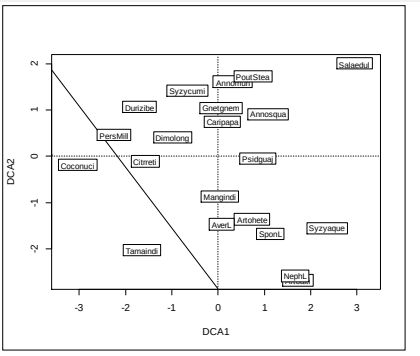
Akhirnya, ada fungsi ordipointlabel yang menggunakan poin dan label ke titik-titik ini. Poin berada di posisi tetap, tetapi label terletak berulang untuk meminimalkan tumpang tindih mereka. Kumpulan data Barro Colorado Island memiliki terlalu banyak nama untuk fungsi

ordipointlabel, tetapi dapat berguna dalam banyak kasus. Selain fungsi-fungsi otomatis ini, fungsi orditkplot memungkinkan pengeditan plot. Ini memiliki poin di posisi tetap dengan label yang dapat diseret ke tempat yang lebih baik dengan mouse. Fungsi ini menggunakan toolset grafis yang berbeda (Tcl/Tk) dari grafik R biasa, tetapi hasilnya dapat diteruskan ke fungsi plot R standar untuk mengedit atau disimpan langsung sebagai file grafik. Selain itu, ouput ordipointlabel dapat diedit menggunakan ordiplot.



Gambar 92. Output ordipointlabel

```
> plot(mod, dis="sp", type="n")
> ordilabel(mod, dis="sp", lab=shnam, priority = stems)
```



Gambar 93. Output ordiplot editing

Fungsi mengidentifikasi, orditorp, ordilabel dan ordipointlabel dapat menyediakan cara yang cepat dan mudah untuk memeriksa hasil penahbisan. Seringkali kita membutuhkan kontrol grafis yang lebih baik, dan memilih spesies yang diberi label. Dalam hal ini, kita dapat terlebih dahulu menggambar plot kosong (dengan tipe = "n"), dan kemudian menggunakan argumen pilih dalam fungsi penahbisan teks dan poin.

Argumen pilih dapat berupa vektor numerik yang mencantumkan indeks item yang dipilih. Indeks tersebut ditampilkan dari fungsi identifikasi yang dapat digunakan untuk membantu dalam memilih item. Atau, pilih dapat menjadi vektor logis yang BENAR ke item yang dipilih. Daftar seperti itu diproduksi secara kasat mata dari orditorp. Anda tidak dapat melihat hasil yang tidak terlihat secara langsung dari metode ini, tetapi Anda dapat menangkap hasilnya seperti yang kami lakukan di atas dalam panggilan orditorp pertama, dan menggunakan vektor ini sebagai dasar untuk grafik yang sepenuhnya dikontrol. Dalam hal ini item pertama adalah:

```
>  sel[1:21]
Annosqua Annomuri Coconuci Salaedul Durizibe Caripapa Gnetgnem Artohete
    TRUE    FALSE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
Artoalti Psidguaj Syzyaque Syzycumi Citrreti Dimolong Mangindi PoutStea
   FALSE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
   AverL Tamaindi PersMill    NephL    SponL
    TRUE     TRUE     TRUE     TRUE     TRUE
```

7. 3. Interpretasi lingkungan

Sering kali mungkin untuk menjelaskan "penahbisan menggunakan pengetahuan ekologi di lokasi yang diteliti, atau pengetahuan tentang karakteristik ekologis spesies. Biasanya lebih baik menggunakan variabel lingkungan eksternal untuk menafsirkan penahbisan. Ada banyak cara untuk melapiskan informasi lingkungan ke dalam diagram penahbisan. Salah satu yang paling sederhana adalah mengubah ukuran karakter plot sesuai dengan variabel lingkungan (argumen cex dalam fungsi plot). Paket vegan memiliki beberapa fungsi yang berguna untuk menyesuaikan variabel lingkungan.

7.3.1 Pemasangan vektor

Metode penafsiran yang paling umum digunakan adalah t vektor lingkungan ke pentahbisan. Vektor yang dipasang adalah panah dengan interpretasi:

1. Panah menunjuk ke arah perubahan paling cepat dalam variabel lingkungan. Seringkali ini disebut arah gradien.
2. Panjang panah sebanding dengan korelasi antara penahbisan dan variabel lingkungan. Seringkali ini disebut kekuatan gradien.

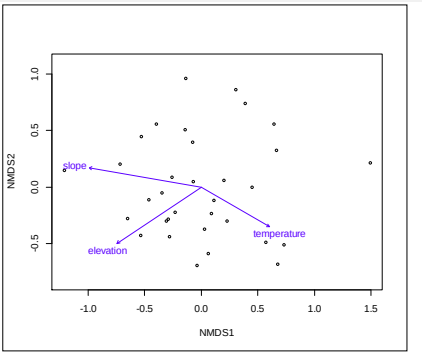
Pas vektor lingkungan mudah menggunakan fungsi envfit. Contoh ini menggunakan hasil nmds sebelumnya dan variabel lingkungan di set data varechem:

```
> data(geo2)
> ef <- envfit(vare.mds, geo2, permu = 999)
```

```
> ef
***VECTORS
               NMDS1   NMDS2    r2 Pr(>r)
elevation   -0.8317 -0.5553 0.386  0.001 ***
slope       -0.9853  0.1711 0.484  0.001 ***
temperature  0.8634 -0.5046 0.234  0.019 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Permutation: free
Number of permutations: 999
```

Dua kolom pertama memberikan cosinus arah vektor, dan r2 memberikan koefisien korelasi kuadrat. Untuk merencanakan, sumbu harus diskalakan oleh akar kuadrat dari r2. Fungsi plot melakukan ini secara otomatis, dan Anda dapat mengekstrak nilai yang diskalakan dengan skor (ef, "vektor"). Signifikansi (Pr> r), atau nilai-P didasarkan pada permutasi acak data: jika Anda sering mendapatkan R2 yang baik atau lebih baik dengan data yang diotorisasi secara acak, nilai Anda tidak mencolok. Anda dapat menambahkan vektor yang dipasang ke pentahbisan menggunakan perintah plot.Anda dapat membatasi plot untuk variabel yang paling signifikan dengan argumen p.max. Seperti biasa, lebih banyak opsi dapat ditemukan di halaman bantuan.

```
> plot(vare.mds, display = "sites")
> plot(ef, p.max = 0.1)
```



Gambar 94. Output Plot variabel geografis

7.3.2 Pemasangan permukaan

Pemecahan vektor populer, dan menyediakan cara ringkas untuk secara bersamaan menampilkan sejumlah besar variabel lingkungan. Namun, itu menyiratkan hubungan linier antara penahbisan dan lingkungan: hanya arah dan kekuatan yang perlu Anda ketahui. Ini mungkin tidak selalu tepat. Fungsi mengatur permukaan variabel lingkungan ke ordinasi. Ini menggunakan model aditif umum dalam

fungsi gam paket mgcv. Fungsi gam menggunakan splines pelat tipis dalam dua dimensi, dan secara otomatis memilih tingkat perataan dengan validasi silang umum.

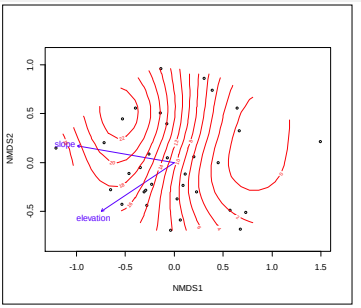
Jika responsnya benar-benar linier dan vektornya sesuai, permukaan yang diperlihatkan adalah bidang yang gradiennya sejajar dengan panah, dan kontur yang dipasang sama-sama diberi garis paralel sejajar dengan panah.Dalam contoh berikut ini saya memperkenalkan dua fitur R baru:

1. Fungsi envfit dapat dipanggil dengan antarmuka formula. Rumus memiliki karakter khusus tilde (), dan sisi kiri memberikan hasil penahbisan, dan sisi kanan mencantumkan lingkungan variabel. Selain itu, kita harus mendefinisikan nama data yang berisi variabel pas.
2. Variabel dalam bingkai data tidak terlihat oleh sesi R kecuali bingkai data terlampir pada sesi.

Kami mungkin tidak ingin membuat semua variabel terlihat oleh sesi, karena mungkin ada sinonim nama, dan kami dapat menggunakan variabel yang salah dengan nama yang sama dalam beberapa analisis. Kita dapat menggunakan fungsi yang membuat bingkai data yang diberikan hanya terlihat oleh perintah berikut.

Sekarang kita siap untuk contohnya dengan membuat vektor yang pas untuk variabel yang dipilih dan menambahkan permukaan yang pas di plot yang sama.

```
> ef <- envfit(vare.mds ~ slope + elevation, geo2)
> plot(vare.mds, display = "sites")
> plot(ef)
> tmp <- with(geo2, ordisurf(vare.mds, slope, add = TRUE))
```

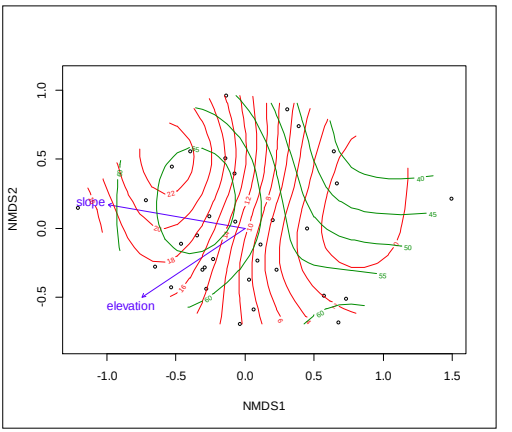


Gambar 95. Output fungsi ordisuf

Fungsi ordisurf mengembalikan hasil penetapan gam. Jika kita menyimpan hasil itu, seperti yang kita lakukan pada awalnya dengan Al,

kita dapat menggunakannya untuk analisis lebih lanjut, seperti pengujian statistik dan prediksi nilai-nilai baru. Misalnya, dipasang (ef) akan memberikan nilai pas yang sebenarnya untuk situs.

```
> with(geo2, ordisurf(vare.mds, elevation, add = TRUE, col = "green4"))
Family: gaussian
Link function: identity
Formula:
y ~ s(x1, x2, k = 10, bs = "tp", fx = FALSE)
Estimated degrees of freedom:
7.57  total = 8.57
REML score: 102.313
```



Gambar 96. Output fungsi ordisurf variabel slope dan elevasi

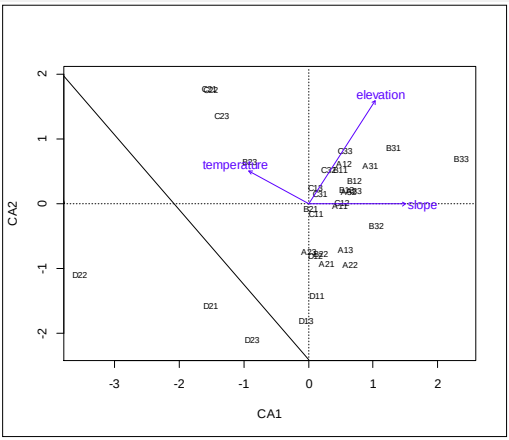
7.3.3 Faktor

Kelas centroid adalah pilihan alami untuk variabel faktor, dan R2 dapat digunakan sebagai statistik yang baik. \ Signifikansi "dapat diuji dengan permutasi seperti dalam pemasangan vektor. Variabel dapat didefinisikan sebagai faktor dalam R, dan mereka akan diperlakukan sesuai tanpa trik khusus.

Sebagai contoh, kita akan memeriksa data lahan kritis yang memiliki beberapa variabel kelas. *Function envfit* juga bekerja dengan faktor:

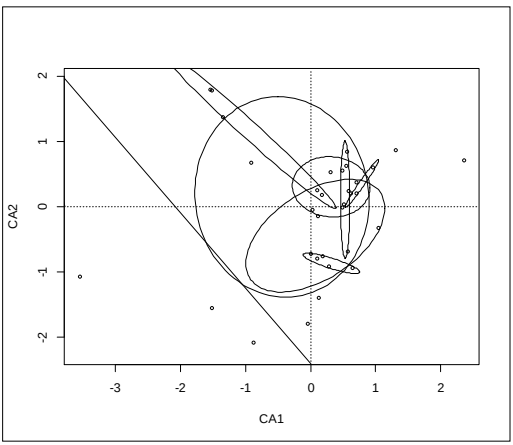
```
> data(fruit2)
> data(geo2)
> fruit2.ca <- cca(fruit2)
> ef <- envfit(fruit2.ca, geo2, permutations = 999)
> ef

***VECTORS

               CA1     CA2    r2 Pr(>r)
elevation    0.5444  0.8388 0.402  0.002 **
```

```
slope        1.0000 -0.0024 0.250  0.024 *
temperature -0.8763  0.4818 0.122  0.200
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Permutation: free
Number of permutations: 999
> plot(fruit2.ca, display = "sites")
> plot(ef)
```



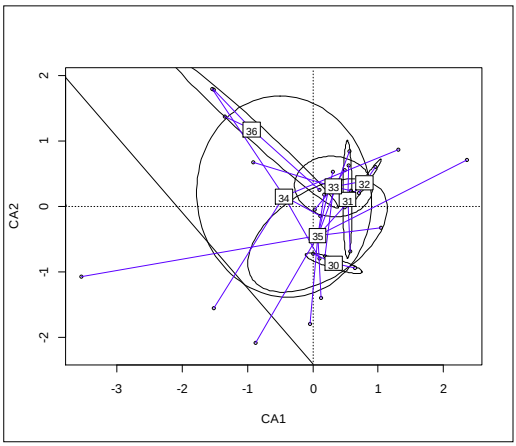
Gambar 97. Output biplot fungsi CA antara variabel lingkungan dan situs

```
> plot(fruit2.ca, display = "sites", type = "p")
> with(geo2, ordiellipse(fruit2.ca, temperature, kind = "se",
conf = 0.95))
```



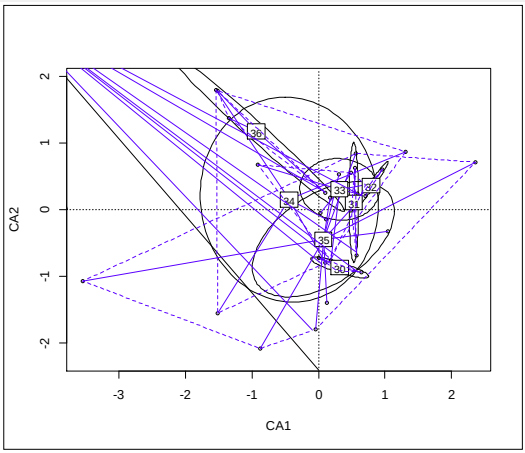
Gambar 98. Output ordiellipse

```
with(geo2, ordispider(fruit2.ca, temperature, col = "blue",
label= TRUE))
```



Gambar 99. Output Ordispider

```
> with(geo2, ordihull(fruit2.ca, temperature, col="blue", lty=2))
```



Gambar 100. Output Ordihull

Nama-nama centroid faktor dibentuk dengan menggabungkan nama faktor dan nama level. Sekarang sumbu menunjukkan centroid untuk tingkat, dan nilai R2 untuk seluruh faktor, sama seperti uji signifikansi. Plotnya terlihat padat, dan kita dapat menggunakan trik x2.6 untuk membuat plot yang lebih bersih, tetapi jelas tidak semua faktor diperlukan dalam interpretasi. Paket vegan memiliki beberapa fungsi untuk menampilkan faktor-faktor grafis. Fungsi ordihull menggambar lambung cembung terlampir untuk item dalam kelas, ordispider menggabungkan item ke centroid kelasnya (tertimbang), dan ordiellipse

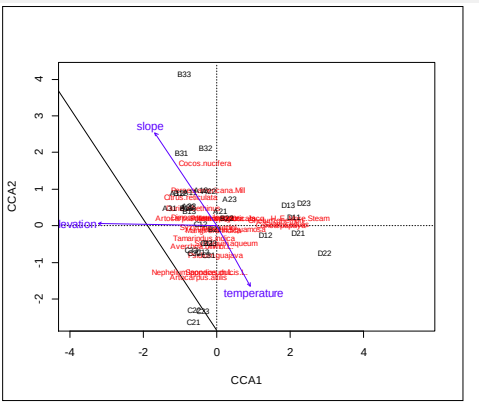
menggambar elips untuk penyimpangan standar kelas, kesalahan standar atau area kepercayaan. Contoh ini menampilkan semua ini untuk tipe Manajemen di pentahbisan sebelumnya dan secara otomatis memberi label pada grup-grup dalam perintah ordispider:

7.4 Constrained ordination

7.4.1 Spesifikasi Model

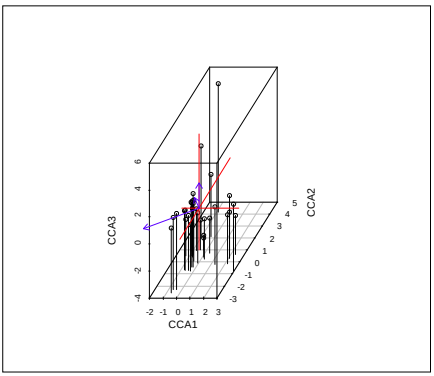
```
>   vare.cca <- cca(fruit2 ~ slope + elevation + temperature,
geo2)
>  vare.cca
Call: cca(formula = fruit2 ~ slope + elevation + temperature,
data =
geo2)

              Inertia Proportion Rank
Total           2.784      1.000
Constrained     0.732      0.263    3
Unconstrained   2.052      0.737   20
Inertia is scaled Chi-square

Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3
0.322 0.239 0.171

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.410 0.289 0.253 0.226 0.166 0.126 0.114 0.100
(Showing 8 of 20 unconstrained eigenvalues)
```

```
>  plot(vare.cca)
```



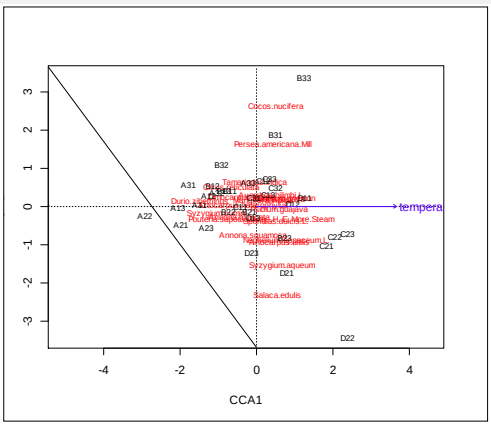
Gambar 101. Output Triplot CCA

```
> library(vegan3d)
> ordiplot3d(vare.cca, type = "h")
```



Gambar 102. Output CCA tiga dimensi

```
> fruit2.cca <- cca(fruit2 ~ temperature, geo2)
> plot(fruit2.cca)
```



Gambar 103. Output CCA dengan faktor pembatas temperatur

```
> fruit2.cca
Call: cca(formula = fruit2 ~ temperature, data = geo2)

              Inertia Proportion Rank
Total          2.7842     1.0000
Constrained    0.1969     0.0707    1
Unconstrained  2.5873     0.9293   20
Inertia is scaled Chi-square

Eigenvalues for constrained axes:
  CCA1
0.1969

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
```

```
0.473 0.421 0.357 0.260 0.199 0.186 0.119 0.101
(Showing 8 of 20 unconstrained eigenvalues)
```

7.4.2 Uji Permutasi

Fungsi hubungan spesies dengan lingkungan dengan uji ChiSquare dengan pendekatan permutasi, tujuannya untuk merduksi variabel yang kurang memberikan respon.

```
>  anova(vare.cca)
Permutation test for cca under reduced model
Permutation: free
Number of permutations: 999
Model: cca(formula = fruit2 ~ slope + elevation + temperature,
data = geo2)
         Df ChiSquare     F Pr(>F)
Model     3    0.7321 3.449  0.001 ***
Residual 29    2.0521
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil uji Anova terlihat bahwa model yang dibangun signifikan (Pr>F = 0,001) sehingga semua variabel lingkungan berpengaruh terhadap spesies.

Selanjutnya dilakukan uji untuk masing-masing variabel lingkungan seperti terlihat pada analisis berikut:

```
> mod <- cca(fruit2 ~ slope + elevation + temperature, geo2)
> anova(mod, by = "term", step=200)
Permutation test for cca under reduced model
Terms added sequentially (first to last)
Permutation: free
Number of permutations: 999

Model: cca(formula = fruit2 ~ slope + elevation + temperature,
data = geo2)
            Df ChiSquare     F Pr(>F)
slope        1    0.2433 3.438  0.001 ***
elevation    1    0.2928 4.138  0.001 ***
temperature  1    0.1960 2.770  0.001 ***
Residual    29    2.0521
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil analisis menunjukkan semua variabel lingkungan memberikan respon yang signifikan terhadap spesies.

```
> anova(mod, by = "margin", perm=500)
Permutation test for cca under reduced model
Marginal effects of terms
Permutation: free
Number of permutations: 999
```

```
Model: cca(formula = fruit2 ~ slope + elevation + temperature,
data = geo2)
            Df ChiSquare     F Pr(>F)
slope        1    0.2229 3.150  0.001 ***
elevation    1    0.2925 4.133  0.001 ***
temperature  1    0.1960 2.770  0.002 **
Residual    29    2.0521
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
>  anova(mod, by="axis", perm=1000)
Permutation test for cca under reduced model
Forward tests for axes
Permutation: free
Number of permutations: 999
Model: cca(formula = fruit2 ~ slope + elevation + temperature,
data = geo2)
         Df ChiSquare     F Pr(>F)
CCA1      1    0.3221 4.552  0.001 ***
CCA2      1    0.2387 3.374  0.001 ***
CCA3      1    0.1712 2.420  0.004 **
Residual 29    2.0521
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

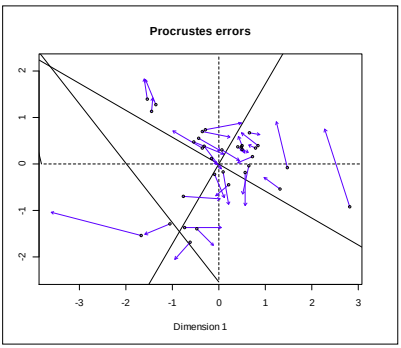
7.4.3 Membangun Model

Model yang dibentuk antara hubungan spesies dengan lingkungan bisa dibangun dengan fungsi berikut:

```
> mod1 <- cca(fruit2 ~ ., geo2)
> mod1
Call: cca(formula = fruit2 ~ elevation + slope + temperature,
data = geo2)
              Inertia Proportion Rank
Total           2.784      1.000
Constrained     0.732      0.263    3
Unconstrained   2.052      0.737   20
Inertia is scaled Chi-square
Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3
0.322 0.239 0.171
Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.410 0.289 0.253 0.226 0.166 0.126 0.114 0.100
(Showing 8 of 20 unconstrained eigenvalues)
```

Selanjutnya dilakukan uji residu dengan pendekatan procruster CCA, dengan langkah berikut:

```
>plot(procrustes(cca(fruit2), mod1))
```



Gambar 104. Output CCA Procruster fungsi respon lingkungan

Selanjutnya dilakukan uji validasi tingkat respon lingkungan dengan spesies berdasarkan nilai AIC terendah.

```
> mod0 <- cca(fruit2 ~ 1, geo2)
> mod <- step(mod0, scope = formula(mod1), test = "perm")
Start:  AIC=99.72
fruit2 ~ 1
             Df   AIC     F Pr(>F)
+ elevation    1 97.92 3.780  0.005 **
+ slope        1 98.70 2.968  0.005 **
+ temperature  1 99.30 2.359  0.010 **
<none>           99.72
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Step:  AIC=97.92
fruit2 ~ elevation
             Df   AIC     F Pr(>F)
+ slope        1 96.66 3.116  0.005 **
+ temperature  1 97.05 2.724  0.010 **
<none>           97.92
- elevation    1 99.72 3.780  0.005 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Step:  AIC=96.66
fruit2 ~ elevation + slope
             Df   AIC     F Pr(>F)
+ temperature  1 95.65 2.770  0.005 **
<none>           96.66
- slope        1 97.92 3.116  0.005 **
- elevation    1 98.70 3.907  0.005 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Step:  AIC=95.65
fruit2 ~ elevation + slope + temperature
```

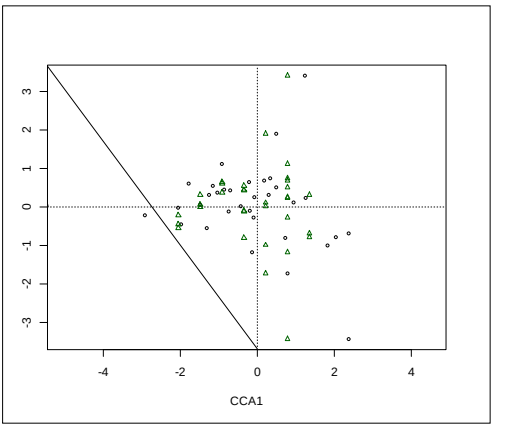
```
              Df   AIC     F Pr(>F)
<none>           95.65
- temperature  1 96.66 2.770  0.005 **
- slope        1 97.05 3.150  0.005 **
- elevation    1 98.05 4.133  0.005 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>modb <- step(mod1, scope = list(lower = formula(mod0), upper =
formula(mod1)), trace = 0)
>  modb
Call: cca(formula = fruit2 ~ elevation + slope + temperature,
data = geo2)
              Inertia Proportion Rank
Total           2.784      1.000
Constrained     0.732      0.263    3
Unconstrained   2.052      0.737   20
Inertia is scaled Chi-square

Eigenvalues for constrained axes:
 CCA1  CCA2  CCA3
0.322 0.239 0.171

Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.410 0.289 0.253 0.226 0.166 0.126 0.114 0.100
(Showing 8 of 20 unconstrained eigenvalues)

> modb$anova
  Step Df Deviance Resid. Df Resid. Dev     AIC
1      NA       NA        29    469.931 95.6506

> vif.cca(mod1)
  elevation       slope temperature
    1.68148     4.96453     4.26574
```

Hasil analisis ditunjukkan validasi model terbaik pada CCA1 dengan nilai AIC terendah.

7.4.4 Kombinasi Linier dan Rata-rata Terbobot

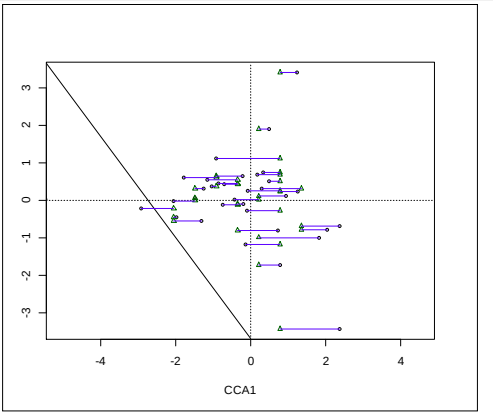
Model yang dibangun antara respon lingkungan dengan spesies bisa dilakukan analisis kombinasi linier dan pendekatan rata-rata terbobot dengan tahapan berikut:

```
> spenvcor(mod)
    CCA1     CCA2     CCA3
0.890929 0.798805 0.727631
>  fruit2.cca <- cca(fruit2 ~ temperature, geo2)
>  plot(fruit2.cca, display = c("lc", "wa"), type = "p")
```



Gambar 105. Output CCA1 fungsi kombinasi linier temperature dengan spesies

```
> ordispider(fruit2.cca, col="blue")
```



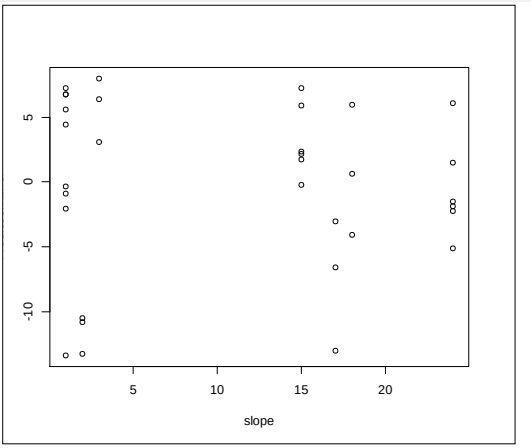
Gambar 106. Output fungsi CCA ordispider

7.4.5 Biplot arrows and environmental calibration

Berikut dilakukan analisis untuk memprediksi hubunga variabel lingkungan terhadap responnya dengan spesies dengan output biplot bentuk panah dan kalibrasi lingkungan.

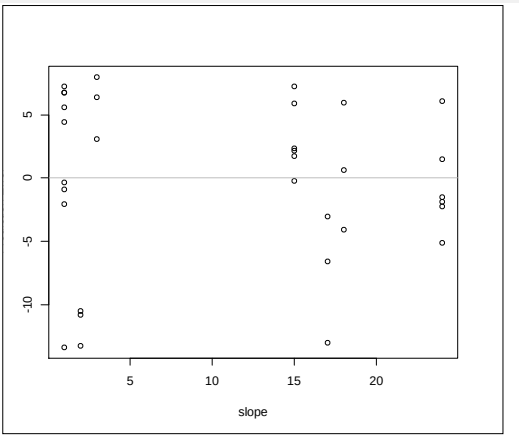
```
> pred <- calibrate(mod)
> head(pred)
    elevation   slope temperature
A11   66.9378 22.5237     31.1950
A12   68.2012 21.7379     31.8148
A13   67.3753 25.4769     29.9472
A21   65.0512 22.1403     29.7191
A22   69.6234 30.0535     28.1247
```

```
A23   60.0237 18.8672     31.0567
>       with(geo2,    plot(slope,    pred[,"slope"]    -    slope,
ylab="Prediction Error"))
```



Gambar 107. Output biplot dengan kalibrasi

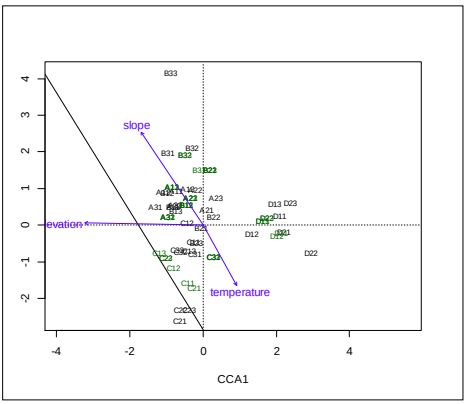
```
>  abline(h=0, col="grey")
```



Gambar 108. Output biplot dengan garis horizontal sebagai batas prediksi error

Langkah berikut untuk menampilkan variabel lingkungan dalam bentuk garis vector yang menggambarkan arah dan kekuatan hubungan dengan langkah berikut:

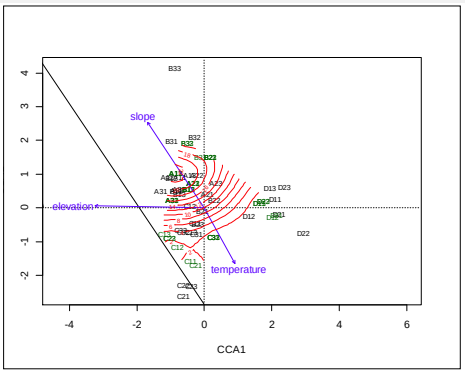
```
>plot(mod, display = c("bp", "wa", "lc"))
```



Gambar 109. Output hubungan variabel lingkungan berbasis lokasi

Tampilan grafik di atas menunjukkan bahwa temperature memberikan arah yang berlawanan dengan slope dan elevasi. Selanjutnya bisa dibentuk tampilan smooth berbasis lokasi.

```
> ef <- with(geo2, ordisurf(mod, slope, display = "lc", add =
TRUE))
```

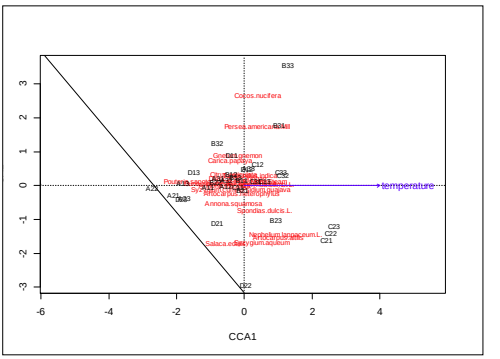


Gambar 110. Out biplot hubungan variabel lingkungan dengan lokasi penelitian

7.4.6 Model Terkondisi dan Parsial

Model yang dibentuk antara faktor lingkungan dengan spesies bisa dibentuk dengan mengkondisikan variabel lingkungan secara parsial. Berikut membangun model dengan mengkondisikan variabel elevasi.

```
>        fruit2.cca     <-     cca(fruit2     ~     temperature     +
Condition(elevation), geo2)
>  plot(fruit2.cca)
```



Gambar 111. Output grafik triplot respon temperature terhadap spesies dan lokasi dengan mengkondisikan elevasi.

Berikut bisa dilihat tingkat variasi berdasarkan nilai inersia dari analisis CCA.

```
> fruit2.cca
Call: cca(formula = fruit2 ~ temperature + Condition(elevation),
data =
geo2)
              Inertia Proportion Rank
Total          2.7842     1.0000
Conditional    0.3026     0.1087    1
Constrained    0.2066     0.0742    1
Unconstrained  2.2750     0.8171   20
Inertia is scaled Chi-square

Eigenvalues for constrained axes:
  CCA1
0.2066
```

Nilai eigen untuk faktor kendala menunjukkan nilai variasi 0,2066 pada CCA1, hal ini berarti model yang dibangun memiliki variasi 20,66% faktor kendala.

```
Eigenvalues for unconstrained axes:
  CA1   CA2   CA3   CA4   CA5   CA6   CA7   CA8
0.465 0.357 0.261 0.236 0.193 0.138 0.115 0.101
(Showing 8 of 20 unconstrained eigenvalues)
```

Pada faktor tanpa kendala pada CA1 menunjukkan nilai variasi 46,5%. Kondisi model ini perlu mengurangi variabel yang memberikan variasi terendah melalui uji Anova.

```
> anova(fruit2.cca, perm.max = 2000)
Permutation test for cca under reduced model
Permutation: free
```

```
Number of permutations: 999

Model:     cca(formula     =     fruit2     ~     temperature     +
Condition(elevation), data = geo2)
         Df ChiSquare     F Pr(>F)
Model     1    0.2066 2.724  0.004 **
Residual 30    2.2750
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil uji Anova menunjukkan bahwa model yang dibangun signifikan dengan nilai Pr(>) = 0.004, sehingga mengkondisikan elevasi adalah tepat.

```
>  with(geo2, anova(fruit2.cca, strata = elevation))
Permutation test for cca under reduced model
Blocks:  strata
Permutation: free
Number of permutations: 999
Model:     cca(formula     =     fruit2     ~     temperature     +
Condition(elevation), data = geo2)
         Df ChiSquare     F Pr(>F)
Model     1    0.2066 2.724  0.076 .
Residual 30    2.2750
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.5 Dissimilaritas dan Lingkungan

7. 5.1 Adonis: Multivariate ANOVA Berbasis pada Dissimilaritas

Analisis berikut ini berdasarkan dissimilaritas Anova multivariate dengan fungsi Adonis.

```
> betad <- betadiver(fruit2, "z")
> adonis(betad ~ slope, geo2, perm=200)

Call:
adonis(formula = betad ~ slope, data = geo2, permutations = 200)
Permutation: free
Number of permutations: 200
Terms added sequentially (first to last)

          Df SumsOfSqs MeanSqs F.Model     R2  Pr(>F)
slope      1     0.877  0.8767   4.091 0.1166 0.00498 **
Residuals 31     6.644  0.2143         0.8834
Total     32     7.521                 1.0000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil uji Adonis menunjukkan bahwa variabel slope memberikan respon yang signifikan sebagai variabel dissimilaritas lingkungan terhadap

spesies. Selanjutnya dilakukan kombinasi variabel slope dengan elevasi dengan langkah analisis berikut:

```
>  adonis(betad ~ elevation*slope, geo2, perm = 200)
Call:
adonis(formula = betad ~ elevation * slope, data = geo2,
permutations = 200)
Permutation: free
Number of permutations: 200

Terms added sequentially (first to last)
               Df SumsOfSqs MeanSqs F.Model     R2  Pr(>F)
elevation       1     0.614  0.6138   3.244 0.0816 0.00498 **
slope           1     1.002  1.0019   5.296 0.1332 0.00498 **
elevation:slope 1     0.418  0.4182   2.210 0.0556 0.02488 *
Residuals      29     5.487  0.1892         0.7296
Total          32     7.521                 1.0000
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil analisis menunjukkan bahwa model yang dibangun menunjukkan bahwa variabel elevasi, slope dan kombinasi elevasi dan slope memberikan respon yang signifikan, serta model yang dibangun sudah

7.5.2 Homogeneitas pada Grup dan Diversitas Beta

Berikut dilakukan uji homogeneitas per grup dan beta diversitas, misalnya digunakan slope sebagai grup.

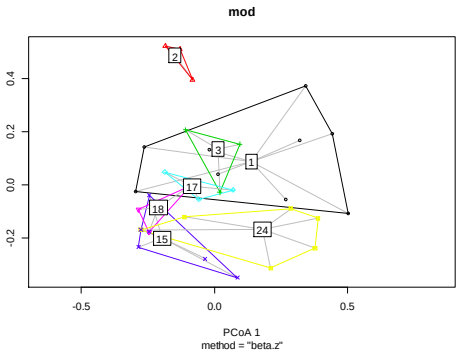
```
> mod <- with(geo2, betadisper(betad, slope))
> mod
        Homogeneity of multivariate dispersions
Call: betadisper(d = betad, group = slope)
No. of Positive Eigenvalues: 14
No. of Negative Eigenvalues: 16

Average distance to median:
    1     2     3    15    17    18    24
0.456 0.197 0.194 0.360 0.333 0.320 0.375
```

Hasil analisis menunjukkan bahwa model yang dibangun dengan slope sebagai grup menghasilkan nilai eigen positif 14 dan negative 16, dengan jarak rata-rata median ke 1 senilai 0,436, ke 2 senilai 0,197.

```
Eigenvalues for PCoA axes:
(Showing 8 of 30 eigenvalues)
PCoA1 PCoA2 PCoA3 PCoA4 PCoA5 PCoA6 PCoA7 PCoA8
1.965 1.590 1.048 0.943 0.672 0.584 0.512 0.451
```

Selanjutnya dibangun plot model grup yang dibentuk dengan langkah berikut:

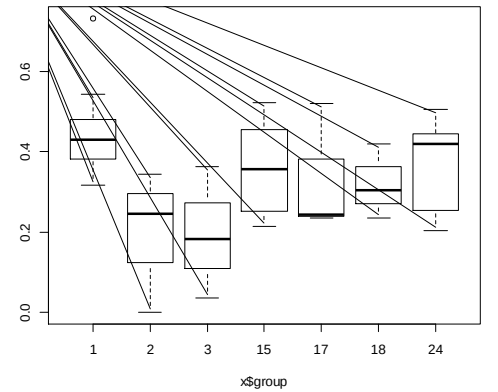
```
> plot(mod)
```



Gambar 112. Plot model dengan slope sebagai grup

Berikut model boxplot dengan slope sebagai kelompok.

```
>  boxplot(mod)
```



Gambar 113. Boxplot model dengan Slope sebagai kelompok

Berikut uji Anova untuk melihat tingkat signifikansi respon lingkungan terhadap spesies dengan slope sebagai grup.

```
> anova(mod)
Analysis of Variance Table
Response: Distances
          Df Sum Sq Mean Sq F value Pr(>F)
```

```
Groups     6 0.2522 0.04204   2.423  0.054 .
Residuals 26 0.4511 0.01735
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hasil analisis Anova menunjukkan bahwa model yang dibentuk dengan slope sebagai grup adalah signifikan pada taraf ($\alpha = 0.1$) pada nilai Pr(>F) = 0,054.

```
> TukeyHSD(mod)
  Tukey multiple comparisons of means
    95% family-wise confidence level
Fit: aov(formula = distances ~ group, data = df)
$group
             diff       lwr       upr    p adj
2-1   -0.25921015 -0.539339 0.0209190 0.083336
3-1   -0.26212988 -0.542259 0.0179992 0.077738
15-1  -0.09642844 -0.317890 0.1250331 0.802551
17-1  -0.12346034 -0.403589 0.1566688 0.793778
18-1  -0.13618738 -0.416316 0.1439417 0.712779
24-1  -0.08153192 -0.302993 0.1399296 0.897408
3-2   -0.00291973 -0.346006 0.3401670 1.000000
15-2   0.16278171 -0.134340 0.4599035 0.592264
17-2   0.13574980 -0.207337 0.4788365 0.862510
18-2   0.12302277 -0.220064 0.4661095 0.908193
24-2   0.17767823 -0.119444 0.4748000 0.493286
15-3   0.16570144 -0.131420 0.4628232 0.572695
17-3   0.13866954 -0.204417 0.4817562 0.850580
18-3   0.12594250 -0.217144 0.4690292 0.898645
24-3   0.18059796 -0.116524 0.4777198 0.474361
17-15 -0.02703190 -0.324154 0.2700899 0.999941
18-15 -0.03975894 -0.336881 0.2573629 0.999444
24-15  0.01489652 -0.227702 0.2574955 0.999994
18-17 -0.01272704 -0.355814 0.3303597 1.000000
24-17  0.04192842 -0.255193 0.3390502 0.999247
24-18  0.05465546 -0.242466 0.3517773 0.996682
```

7.5.3 Mantel Test

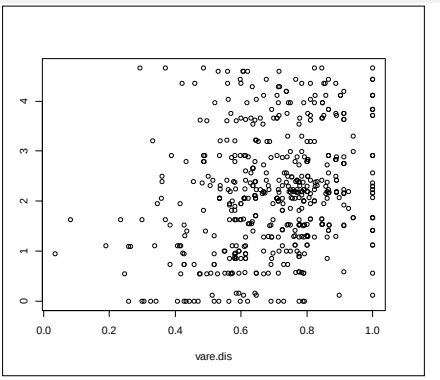
Berikut adalah uji Mantel untuk melihat tingkat hubungan berbasis korelasi Perason's Product Moment (PPM)

```
> pc <- prcomp(geo2, scale = TRUE)
> pc<- scores(pc, display = "sites", choices = 1:4)
> edis <- vegdist(pc, method = "euclid")
> vare.dis <- vegdist(wisconsin(sqrt(fruit2)))
> mantel(vare.dis, edis)
Mantel statistic based on Pearson's product-moment correlation
Call:
mantel(xdis = vare.dis, ydis = edis)
Mantel statistic r: 0.257
```

```
      Significance: 0.001
Upper quantiles of permutations (null model):
   90%    95%  97.5%    99%
0.0783 0.0976 0.1158 0.1374
Permutation: free
Number of permutations: 999
```

Hasil uji menunjukkan nilai korelasi mantel sebesar 0,257, selanjutnya dilakukan plot.

```
>plot(vare.dis, edis)
```

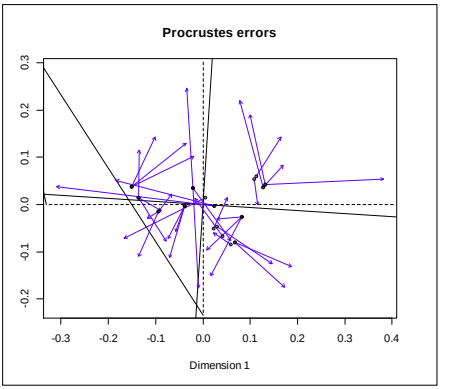


Gambar 114. Output plot mantel Test

7.5.4 Protest: Procrustes test

Berikut melihat tingkat error dengan menggunakan Uji Procrutes:

```
> pc <- scores(pc, choices = 1:2)
> pro <- protest(vare.mds, pc)
> plot(pro)
```



Gambar 115. Output plot protest

Selanjutnya dihitung nilai erornya, dengan langkah berikut:

```
> pro
Call:
protest(X = vare.mds, Y = pc)
Procrustes Sum of Squares (m12 squared):        0.663
Correlation in a symmetric Procrustes rotation: 0.58
Significance:  0.001
Permutation: free
Number of permutations: 999
```

Hasil analisis menunjukkan bahwa nilai error dari model yang dibangun sebesar 0,662, denga korelasi dalam rotasi procrustes simetrik sebesar 0,58.

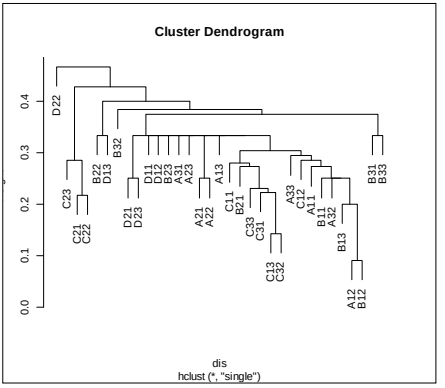
7.6 Classification

7.6.1 Cluster analysis

Tahapan ini menjelaskan tentang pengelompokkan variabel dengan menggunakan analissi kluster.

a. Metode Single

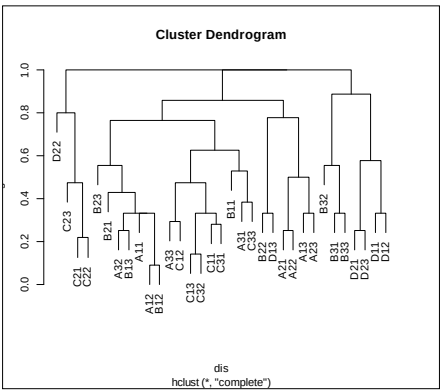
```
> dis <- vegdist(fruit2)
> clus <- hclust(dis, "single")
> plot(clus)
```



Gambar 116. Dendrogram dengan metode Single

b. Metode Complete

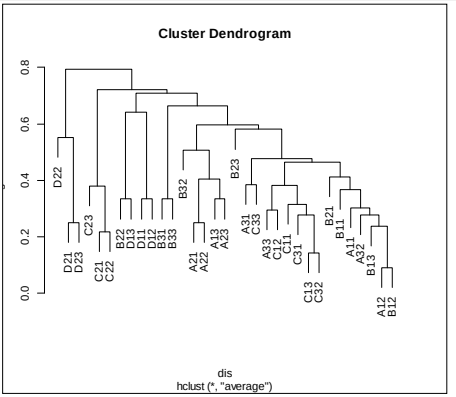
```
>  cluc <- hclust(dis, "complete")
>  plot(cluc)
```



Gambar 117. Dendrogram dengan metode Complete

c. Metode Average

```
>  clua <- hclust(dis, "average")
>  plot(clua)
```



Gambar 118. Dendrogram dengan metode Average

Berikut dilakukan perhitungan jarak (range) untuk mendapatkan nilai korelasi kopenetik.

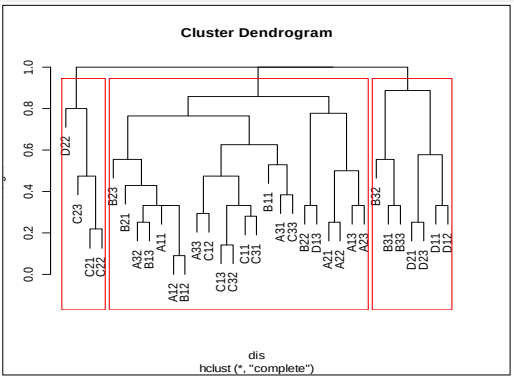
```
>  range(dis)
[1] 0.0909091 1.0000000
>  cor(dis, cophenetic(clus))
[1] 0.662887
> cor(dis, cophenetic(cluc))
[1] 0.641601
> cor(dis, cophenetic(clua))
[1] 0.738114
```

Sebagai dasar penentukan klasifikasi terbaik ditentukan berdasarkan nilai yang mendekati 0,60.

7.6.2 Display and interpretation of classes

Untuk memperjelas tampilan klasifikasi dalam pengelompokan dilakukan langkah berikut:

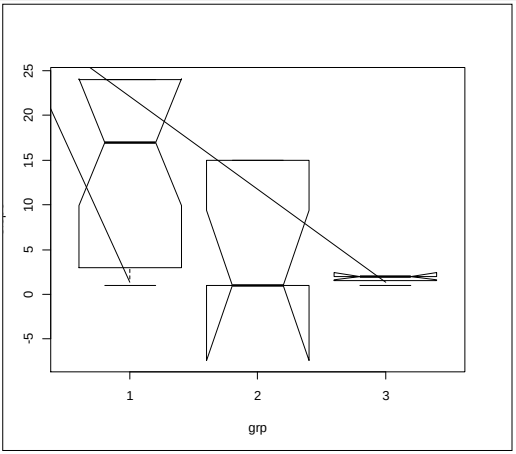
```
> plot(cluc)
> rect.hclust(cluc, 3)
```



Gambar 119. Output klasifikasi dengan 3 kelompok

Berikut tampilan klasifikasi dengan bentuk pohon.

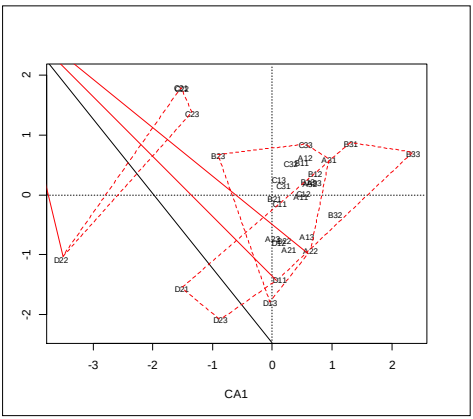
```
>  grp <- cutree(cluc, 3)
>  boxplot(slope ~ grp, data=geo2, notch = TRUE)
```



Gambar 120. Output klasifikasi dengan 3 kelompok bentuk pohon

Berikut klasifikasi dengan metode CCA:

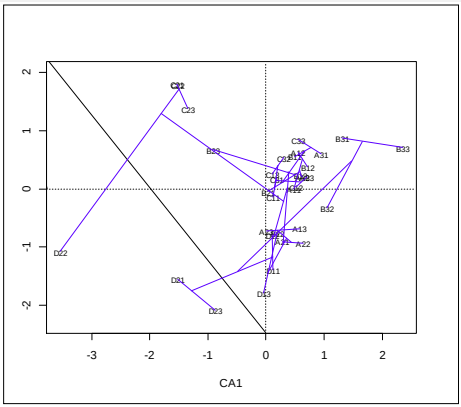
```
> ord <- cca(fruit2)
> plot(ord, display = "sites")
ordihull(ord, grp, lty = 2, col = "red")
```



Gambar 121. Output klasifikasi dengan metode CCA ordihull

Bentuk klasifikasi dengan metode CCA ordicluster

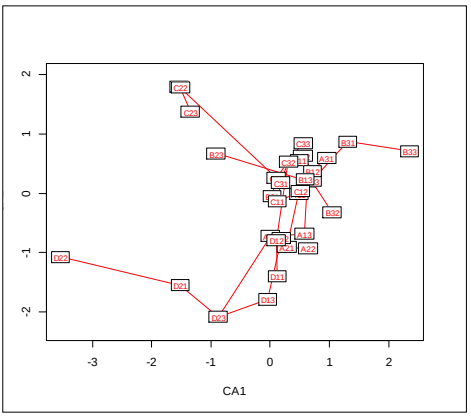
```
> plot(ord, display="sites")
> ordicluster(ord, cluc, col="blue")
```



Gambar 122. Output klasifikasi metode CCA ordicluster

Selanjutnya dilakukan klasifikasi dengan bentuk spantree.

```
> mst <- spantree(dis, toolong = 1)
> plot(mst, ord=ord, pch=21, col = "red", bg = "yellow", type =
"t")
```

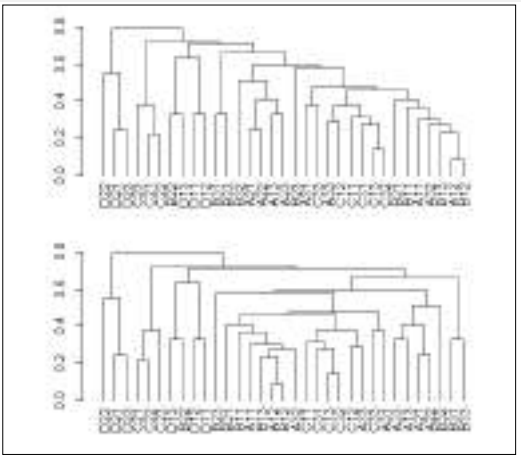


Gambar 123. Output klasifikasi metode CCA spantree

7.6.3 Classified Community Tables

Tabel komunitas yang terkelompokkan mengikuti langkah berikut:

```
> wa <- scores(ord, display = "sites", choices = 1)
> den <- as.dendrogram(clua)
> oden <- reorder(den, wa, mean)
> op <- par(mfrow=c(2,1), mar=c(3,5,1,2)+.1)
> plot(den)
> plot(oden)
```



Gambar 124. Output klasifikasi berdasarkan lokasi plot oden

```
> par(op)
> vegemite(fruit2, use = oden, zero = "-")
DDDCCCDBDDBBBABABACCCCCACAAAAABBB
22222212112211111313131333212233 3
21312332213111322211322331331221 3
 Salaca.edulis                    221-----------------------------
 Artocarpus.altilis               ---21----------------------------
```

```
Syzygium.aqueum                           21-221----1----------------------
Nephelium.lappaceum.L.                    ---211-------------------------
Carica.papaya                             --------12----------------------
Gnetum.gnemon                             ---------1----------------------
Spondias.dulcis.L.                        ---1--------------1------------
Annona.squamosa                           1211-----1-1-1----2-1-----1-1----
Pouteria.sapota..Jacq...H..E..More.Steam -22---2-23---------1------1-111--
Psidium.guajava                           -1-111--11--------211121---------
Mangifera.indica                          ---22211331251222133232131--1--1-
Artocarpus.heterophylus                   ---11-------1--1----------1----1-
Annona.muricata                           -11---11-1--------1---11--1111---
Averrhoa.bilimbi.L.                       ---11-------------1111-111-----1-
Citrus.reticulata                         ------------11-11------1-1----1--
Dimocarpus.longan                         -111----1-11333222432232122233221
Syzygium.cumini                           ------------------11--------21---
Tamarindus.indica                         ------------------------11-------
Persea.americana.Mill                     ---------1------------11------122
Durio.zibethinus                          ------------------------1---1---
Cocos.nucifera                            ------------------------------11
 sites species
    33      21
```

Daftar Pustaka

Alain F. Zur et.al, A Beginner's Guide to R, Springer, 2009.

Nicholas Walliman, Research Method Basics, Rouledge Publisher, 2011

Emanuel Paradis, R for the Beginner, Institut des Sciences de l' _ Evolution, Paris, 2005

Suhartono, Analisis Data Statistik dengan R, Lab Statistika ITS, 2010

Annette J. Dobson , An Introduction to Generalized Linear Models, Chapman and Hall, London., 1990

Peter McCullagh and John A. Nelder, Generalized Linear Models. Second edition, Chapman and Hall, London, 1989

John A. Rice (1995), Mathematical Statistics and Data Analysis. Second edition. Duxbury Press, Belmont, CA, 1995.

http://cran.r-project.org/

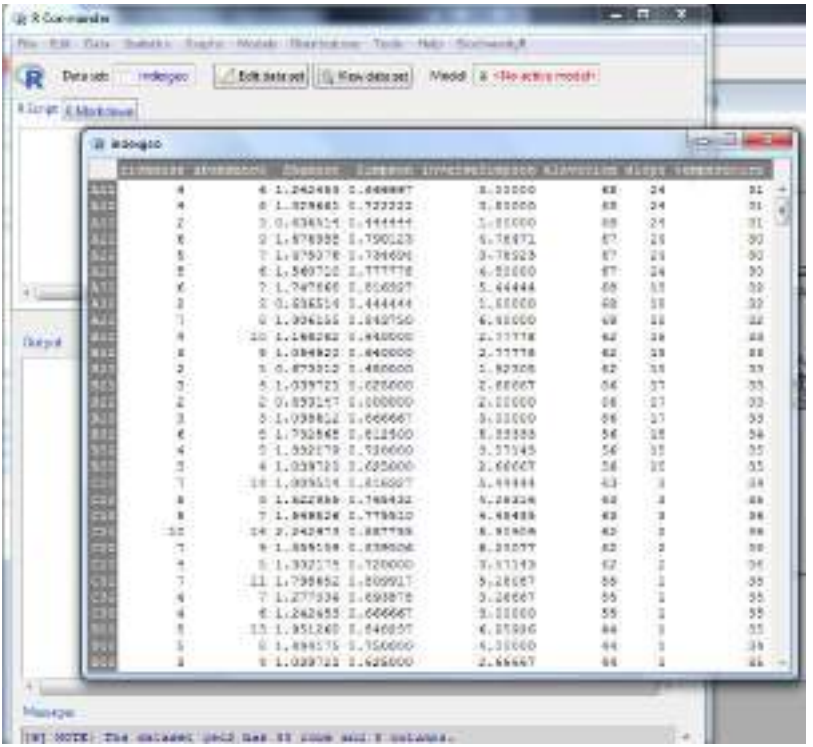
www.Widodo.com

http://cran.r-project.org/doc/manuals/r-release/R-intro.html

BAB VIII
MODEL REGRESI

8.1 Pengantar

Untuk melakukan analisis regresi pada bab ini, digunakan dataset indexgeo, sebelumnya lakukan pelacakan data apakah sudah tersimpan dalam data R, jika belum lakukan import data dari file Excel.



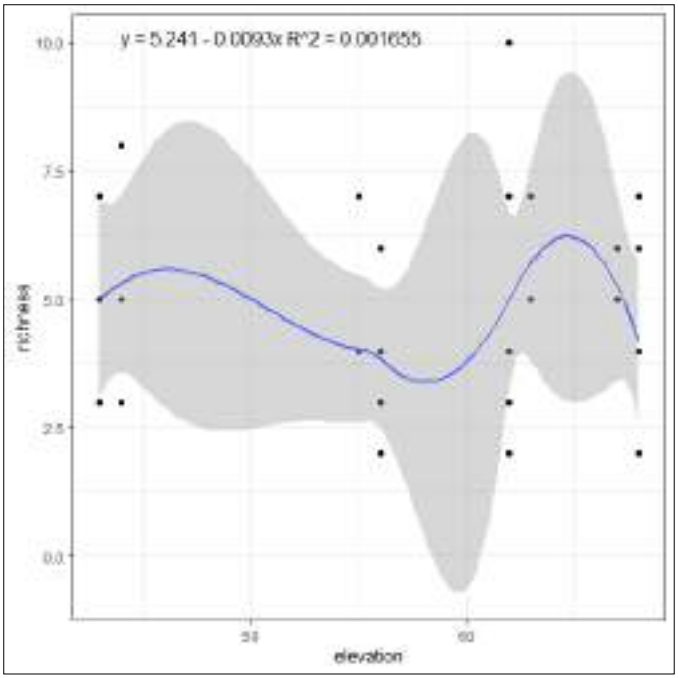
Gambar 125. Hasil pengecekan dataset indexgeo di data R

Setelah dataset ditemukan, maka lakukan uji regresi dengan memilih variabel ujinya.

Loading Required R packages

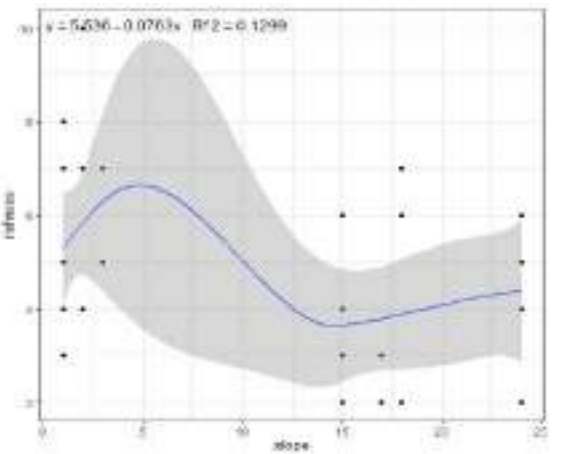
```
> plot(vare.dca, display="sites")
> model <- lm(richness ~ elevation, data = indexgeo)
> summary(model)

Call:
lm(formula = richness ~ elevation, data = indexgeo)

Residuals:
    Min      1Q  Median      3Q     Max
-2.7206 -1.7206 -0.6091  1.3614  5.3351

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.241098   2.424502   2.162   0.0385 *
elevation   -0.009294   0.040996  -0.227   0.8221
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.874 on 31 degrees of freedom
Multiple R-squared:  0.001655,  Adjusted R-squared:  -0.03055
F-statistic: 0.0514 on 1 and 31 DF,  p-value: 0.8221
```

```
> ggplot(indexgeo, aes(x = elevation, y = richness)) +
geom_point() +stat_smooth()`geom_smooth()` using method =
'loess' and formula 'y ~ x')
```



Gambar 126. Output grafik hubungan antara kekayaan spesies dengan elevasi

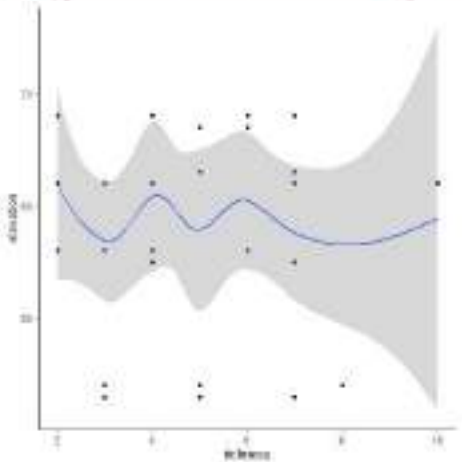
```
> model <- lm(richness ~ slope, data = indexgeo)
> summary(model)

Call:
lm(formula = richness ~ slope, data = indexgeo)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4599 -1.3918 -0.3918  1.5401  4.6164

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.53623    0.50502  10.962 3.42e-12 ***
slope       -0.07690    0.03546  -2.252   0.0393 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.843 on 31 degrees of freedom
Multiple R-squared:  0.1299,    Adjusted R-squared:  0.1019
F-statistic: 4.629 on 1 and 31 DF,  p-value: 0.03934
```



Gambar 127. Grafik hubungan antara kekayaan spesies dengan slope

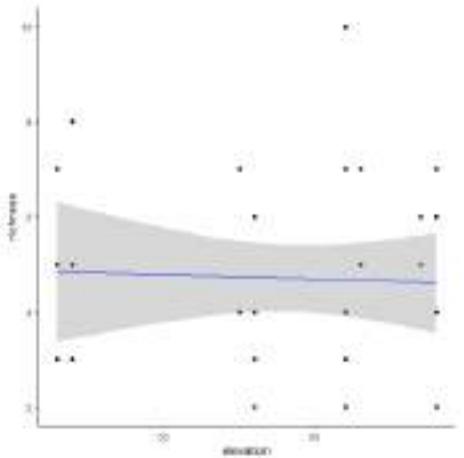
```
> library(tidyverse)
-- Attaching packages --------------------------------------- tidyverse 1.2.1 --
v tibble  2.1.3     v purrr   0.3.2
v tidyr   0.8.3     v dplyr   0.8.3
v readr   1.3.1     v stringr 1.4.0
v tibble  2.1.3     v forcats 0.4.0
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x dplyr::recode() masks car::recode()
x dplyr::select() masks MASS::select()
x purrr::some()   masks car::some()
> library(caret)

Attaching package: 'caret'

The following object is masked from 'package:purrr':

    lift
```

```
> theme_set(theme_classic())
> ggplot(indexgeo, aes(richness, elevation) ) +geom_point() +stat_smooth()
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
> `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Gambar 128. Grafik hubungan natara elevasi dan kekayaan spesies

```
> model <- lm(richness ~ elevation, data = indexgeo)
> predictions <- model %>% predict(indexgeo)
> data.frame(RMSE = RMSE(predictions, indexgeo$richness),
+ R2 = R2(predictions, indexgeo$richness))
      RMSE          R2
1 1.913029 0.00165534
>  ggplot(indexgeo,  aes(elevation,  richness)  )  +geom_point()
+stat_smooth(method = lm, formula = y ~ x)
```



Gambar 129. Grafik hubungan antara kekayan spesies dengan elevasi

8.2 Model linier

Pada tahapan ini dilakukan analisis dengan model linier antara kekayaan spesies dengan lingkungan (Slope dan elevasi).

```
> model1 <- lm(richness ~ elevation + slope, data = indexgeo)
> summary(model1)

Call:
lm(formula = richness ~ elevation + slope, data = indexgeo)

Residuals:
    Min      1Q  Median      3Q     Max
-2.5855 -1.6780  0.0225  1.2611  3.8717

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.26730    2.63594   0.481   0.6342
elevation    0.08251    0.05005   1.648   0.1097
slope       -0.12736    0.04638  -2.746   0.0101 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.794 on 30 degrees of freedom
Multiple R-squared:  0.2022,    Adjusted R-squared:  0.149
F-statistic: 3.801 on 2 and 30 DF,  p-value: 0.03377
> predictions <- model1 %>% predict(indexgeo)
> RMSE(predictions, indexgeo$richness)
[1] 1.710139
> R2(predictions, indexgeo$richness)
[1] 0.2021888
```

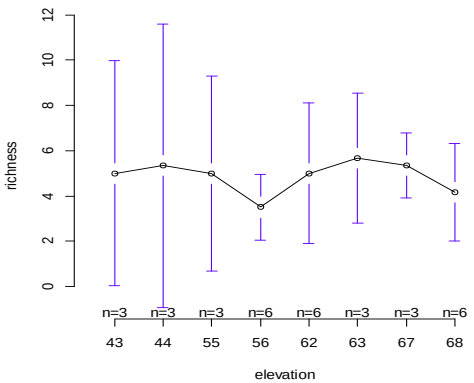
Hasil analisis menunjukkan bahwa slope memberikan respon yang signifikan dibandingkan elevasi.

8.3 Efek Interaksi

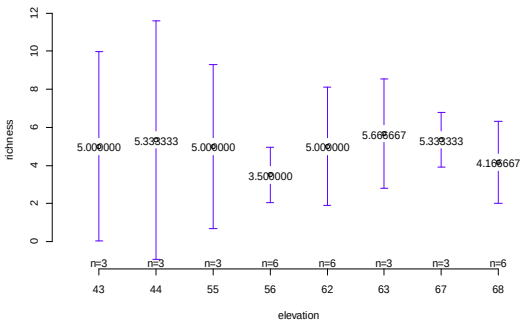
Berikut untuk melihat interaksi dari variabel yang berpengaruh.

```
> model2 <- lm(richness ~ elevation + slope + elevation:slope,data=indexgeo)
> summary(model2)

Call:
lm(formula = richness ~ elevation + slope + elevation:slope,
    data = indexgeo)

Residuals:
    Min      1Q  Median      3Q     Max
-2.8290 -1.2683 -0.0919  1.0758  4.0128

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     3.056200   3.069925   1.005    0.323
elevation       0.031055   0.056937   0.397    0.377
slope          -0.505975   0.336265  -1.609    0.142
elevation:slope 0.006028   0.005287   1.140    0.264

Residual standard error: 1.785 on 29 degrees of freedom
Multiple R-squared:  0.2364,    Adjusted R-squared:  0.1574
F-statistic: 2.993 on 3 and 29 DF,  p-value: 0.04701
```

```
> library(gplots)

Attaching package: 'gplots'

The following object is masked from 'package:stats':

    lowess

> plotmeans(richness ~ elevation, data = indexgeo, frame = FALSE)
```



(a)

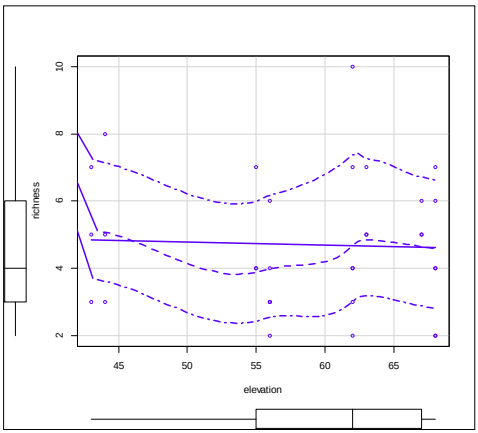


(b)

Gambar 130. Grafik plot rata-rata interaksi antara kekayaan spesies dengan elevasi (a) dan tampilan nilai interasksi (b)

Selanjutnya dibentuk grafik hubungan kekayaan spesies dengan elevasi yang dilengkapi dengan boxplot dari bentuk distribusi data.

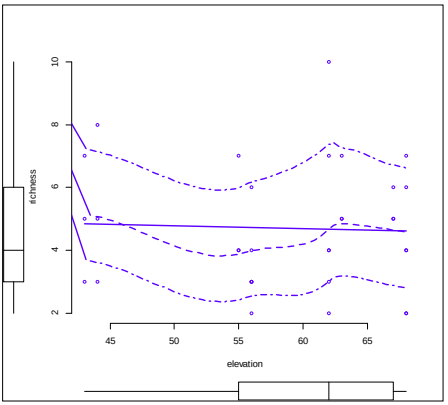
```
> library("car")
> scatterplot(richness ~ elevation, data = indexgeo)
```



Gambar 131. Output grafik dengan boxplot

Selanjutnya ditampilkan scatterplot dengan menghilangkan grid.

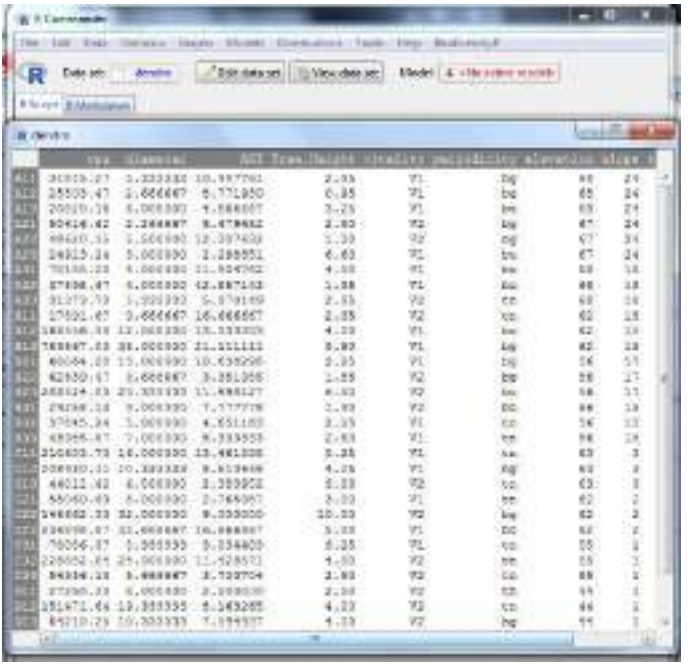
```
> scatterplot(richness ~ elevation, data = indexgeo,smoother = FALSE, grid = FALSE, frame = FALSE)
```



Gambar 132. Output grafik tanpa Grid

8.4 Analisis Regresi dengan Tambahan Data Kategori

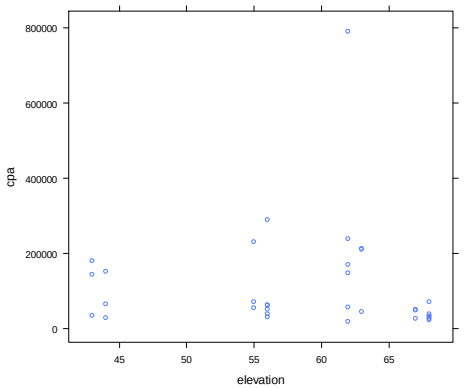
Pada subbab ini akan dibahas bagaimana variabel data yang dianalisis terdapat data kategori, dalam hal ini kita gunakan dataset **dendro**. Bentuk analisis regresi ini mengkombinasikan data kuantitatif dengan data kualitatif (kategori) yang bisa dianalisis secara bersamaan.



Gambar 133. Hasil pengecekatan dataset dendro pada R.Commander

Langkah berikut untuk menampilkan data yang sudah tersimpan di program R.

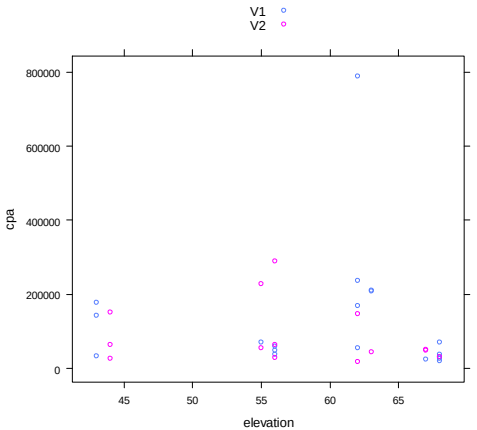
```
> my_data <- dendro
> head(my_data)
       cpa diameter       ADI Tree.Height vitality periodicity elevation
A11 31815.27 5.333333 10.447761        2.95       V1          bg        68
A12 25503.47 2.666667  5.771930        0.95       V1          bg        68
A13 20820.16 6.000000  4.666667        3.25       V1          bu        68
A21 50416.62 2.266667  5.479452        2.50       V2          bg        67
A22 48620.15 5.500000 12.907692        1.30       V2          bg        67
A23 24913.24 3.000000  2.296851        6.60       V1          bu        67
    slope temperature
A11    24          31
A12    24          31
A13    24          31
A21    24          30
A22    24          30
A23    24          30
> |
> xyplot(cpa ~ elevation, data = my_data)
```



Gambar 134. Output plot antara variabel cpa dan elevasi

Selanjutnya dibangun model hubungan dengan variabel kategori sebagai grup (Vitalitas).

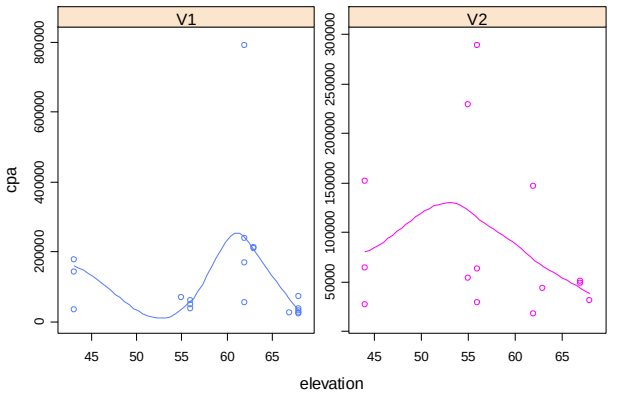
```
> xyplot(cpa ~ elevation, group = vitality,data = my_data, auto.key = TRUE)
```



Gambar 135. Output plot cpa dan elevasi dengan vitalitas sebagai grup.

Selanjutnya dibangun grafik hubungan dengan grup vitalitas yang terpisah.

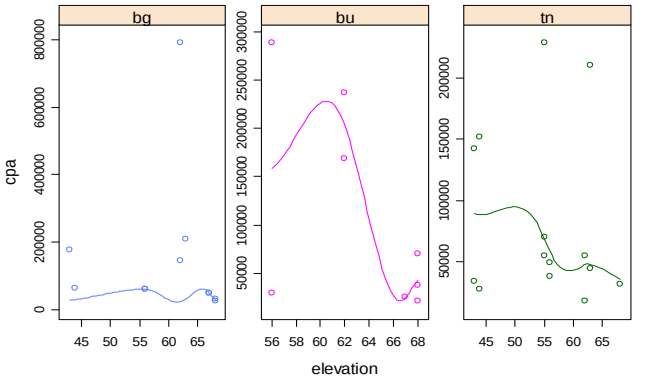
```
> xyplot(cpa ~ elevation | vitality,group = vitality, data =
my_data,type = c("p", "smooth"),scales = "free")
```



Gambar 136. Output grafik hubungan cpa dan elevasi dengan grup terpisah.

Berikut hubungan yang dibentuk antara cpa dan elevasi dengan variabel periodisitas (kategori) sebagai grup yang terpisah.

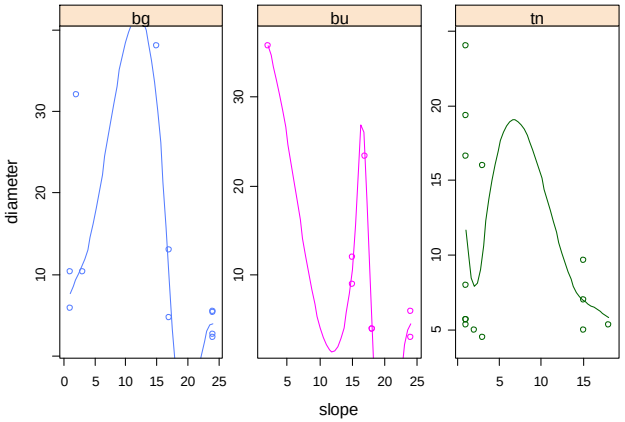
```
> xyplot(cpa ~ elevation | periodicity,group = periodicity, data
= my_data,type = c("p", "smooth"),scales = "free")
```



Gambar 137. Output hubungan antara cpa dan elevasi dengan variabel kategori periodisitas

Berikut hubungan natara diameter dan slope dengan periodisitas.

```
xyplot(diameter ~ slope | periodicity,group = periodicity, data
= my_data,type = c("p", "smooth"),scales = "free")
```

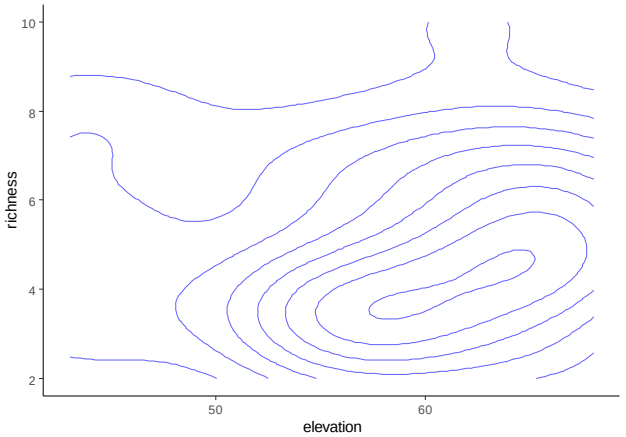


Gambar 138. Output grafik hubungan diameter dan slope dengan periodisitas (kategori)

Pada bagian ini ditampilkan bentuk fungsi densitas dari hubungan elevasi dan kekayaan spesies.

geom_density_2d():

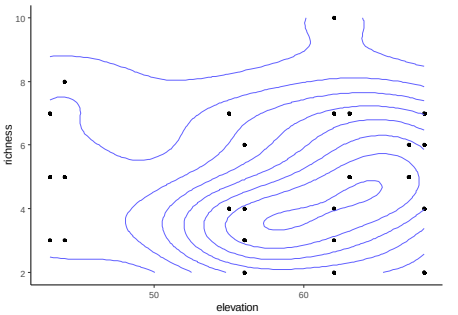
```
> sp <- ggplot(indexgeo, aes(x=elevation, y=richness)
> sp + geom. Density 2d()
```



Gambar 139. Grafik hubungan antara kekayaan spesies dan elevasi dengan fungsi densitas.

Berikut tampilan fungsi densitas denga tampilan lokasi.

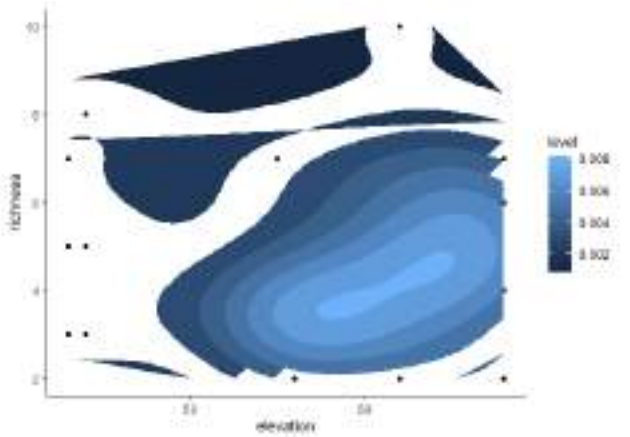
```
> sp + geom_point() + geom._density_2d()
```



Gambar 140. Grafik hubungan kekayaan spesies dengan elevasi dengan fungsi densitas berbasis lokasi.

Berikut tampilan grafik dengan kontur warna.

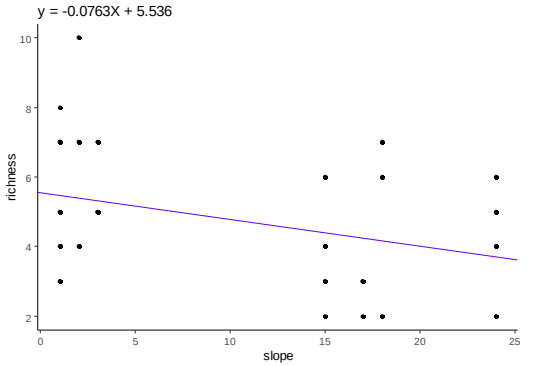
```
> sp + geo_point() + stat_density_2d(fill = ..level..), geom.=”polugon”)
```



Gambar 141. Grafik hubungan antara kekayaan spesies dengan elevasi berbasis kontur warna.

Berikut tampilan grafik garis dari hubungan kekayaan spesies dan slope.

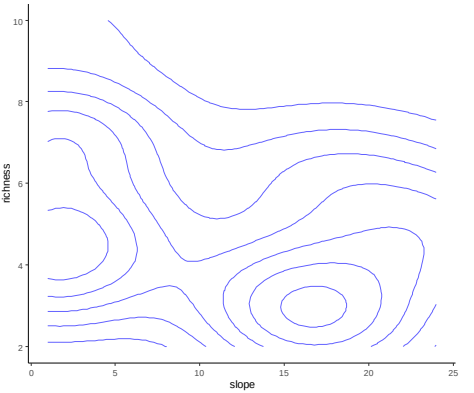
```
> # Simple scatter plot
> sp <- ggplot(data=indexgeo, aes(x=slope, y=richness)) + geom_point()
> sp + geom_abline(intercept = 5.536, slope = -0.0763, color="blue")+ggtitle("y = -0.0763X + 5.536")
```



Gambar 142. Grafik berbentuk linier dari hubungan kekayaan spesies dan slope.

Kemudian bisa dibentuk grafik fungsi slope dan kekayaan spesies dengan tampilan kontur garis yang smooth, dengan langkah berikut:

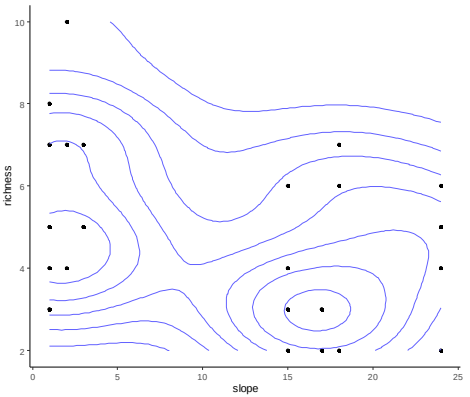
```
> sp <- ggplot(indexgeo, aes(x=slope, y=richness))
> sp + geom_density_2d()
```



Gambar 143. Grafik fungsi hubungan slope dan kekayaan spesies bentuk kontur garis

Dari grafik fungsi hubungan tersebut bisa ditetapkan titik-titk lokasi penelitian, dengan langkah berikut:

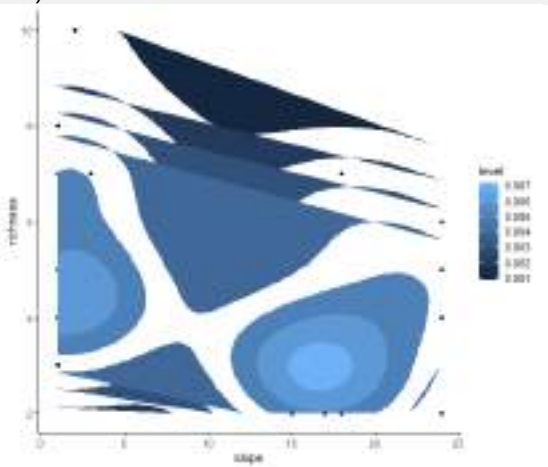
```
> sp + geom_point() + geom_density_2d()
```



Gambar 144. Grafik fungsi hubungan slope dan kekayaan spesies berbentuk kontur garis dan scatterplot

Ragam bentuk fungsi hubungan antar variabel tersebut juga bisa dibangun dengan menampilkan kontur warna.

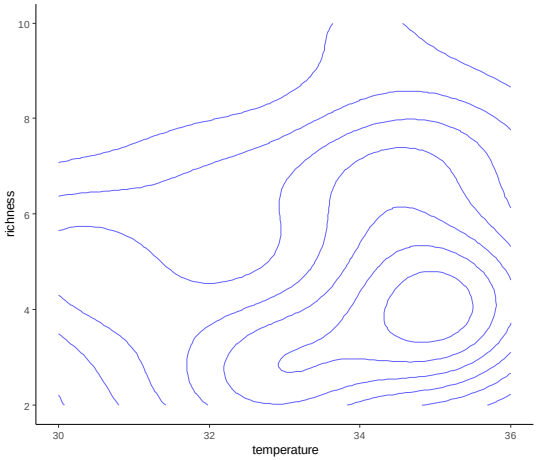
```
> sp + geom_point() + stat_density_2d(aes(fill = ..level..),
geom="polygon")
```



Gambar 145. Grafik hubungan slope dan kekayaan spesies dengan level kontur warna

Selanjutnya akan dibangkitkan fungsi hubungan antara variabel temperature dan kekayaan spesies.

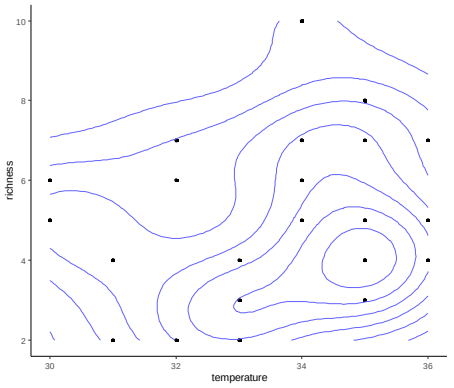
```
> sp <- ggplot(indexgeo, aes(x=temperature, y=richness))
> sp + geom_density_2d()
```



Gambar 146. Grafik fungsi hubungan antara temperature dan kekayaan spesies berbentuk kontur garis.

Berikutnya ditampilakn bentuk grafik dengan dilengkapi scatter plot

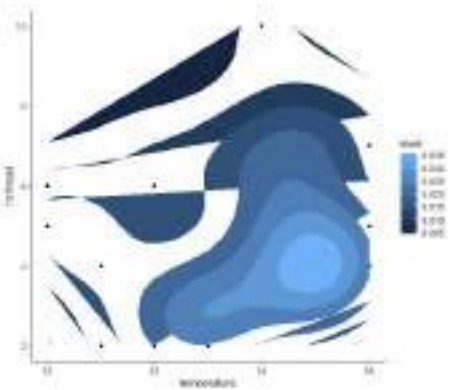
```
> sp + geom_point() + geom_density_2d()
```



Gambar 147. Grafik fungsi hubungan dengan tambahan scatterplot

Langkah berikut ditampilkan grafik fungsi hubungan dengan level kontur berwarna.

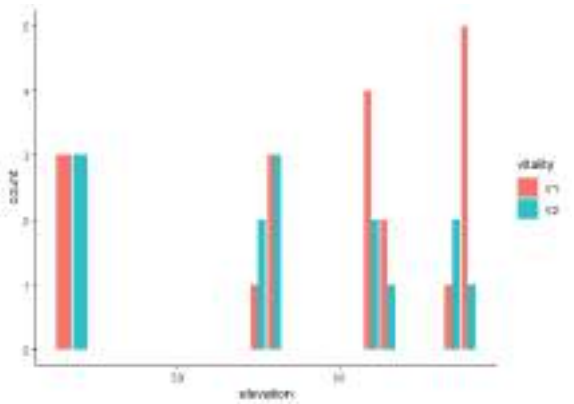
```
>sp + geom_point() + stat_density_2d(aes(fill = ..level..),
geom="polygon")
```



Gambar 148. Grafik fungsi hubungan dengan level kontur warna

Variasi visual hasil analisis dengan program R juga bisa ditampilkan dalam bentuk histogram warna dari variabel kategori yang dikelompokkan.

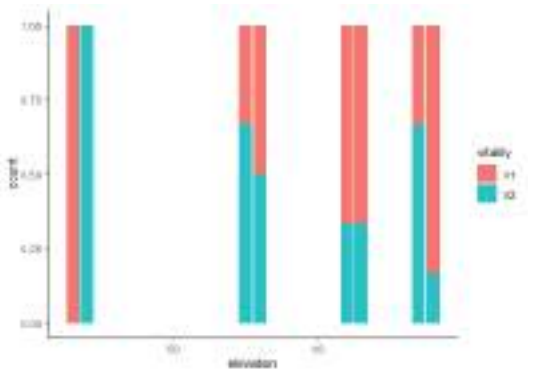
```
> p <- ggplot(dendro, aes(elevation, fill = vitality))
> p + geom_bar(position = "dodge")
```



Gambar 149. Histogram hubungan antara elevasi dengan pengelompokkan variabel vitalitas.

Selanjutnya bisa dibangun histrogram kumulatifnya dengan langkah berikut:

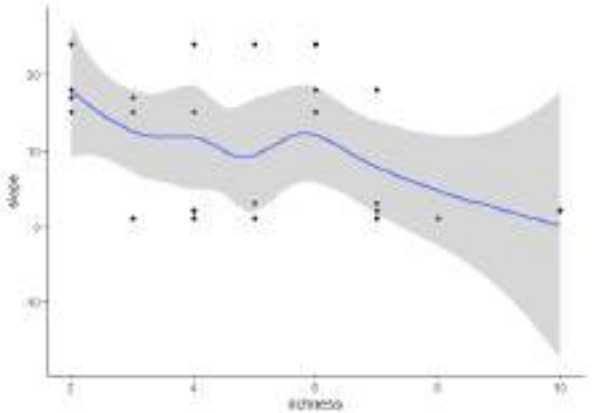
```
> # Stack objects on top of one another,
> # and normalize to have equal height
> p + geom_bar(position = "fill")
```



Gambar 150. Histogram kumulatif pengelompokan vitalis berdasarkan elevasi

Berikut menampilkan grafik dengan error bar, dengan langkah berikut:

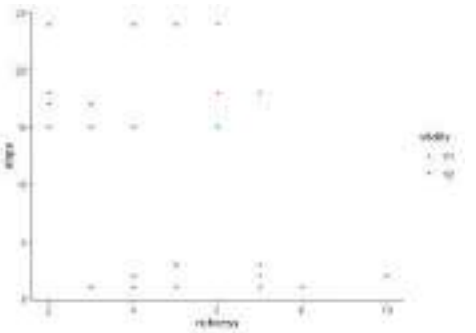
```
> qplot(richness, slope, data = dendro,geom = c("point",
"smooth"))
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Gambar 151. Grafik fungsi hubungan kekayaan spesies dan slope dengan error bar

Berikut tampilan scatter plot hubungan antara kekayaan spesies dan slope dengan vitalitas sebagai kelompok.

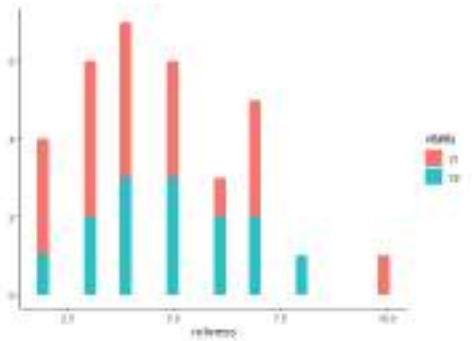
```
> qplot(richness, slope, data = dendro, colour = vitality, shape
= vitality)
```



Gambar 152. Scatter plot kekayaan spesies dengan slope berdasarkan pengelompokkan vitalitas

Tampilan histogram kekayaan spesies berdasarkan pengelompokkan vitalitas dengan langkah berikut:

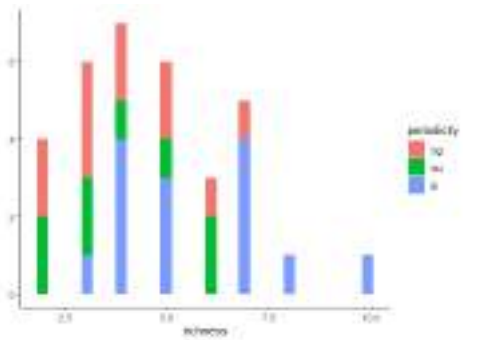
```
> qplot(richness, data = dendro, geom =
"histogram",fill=vitality)
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



Gambar 153. Tampilan histogram kekayaan spesies dengan pengelompokkan vitalitas

Langkah berikut menampilkan histogram kekayaan spesies dengan pengelompokkan periodisitas.

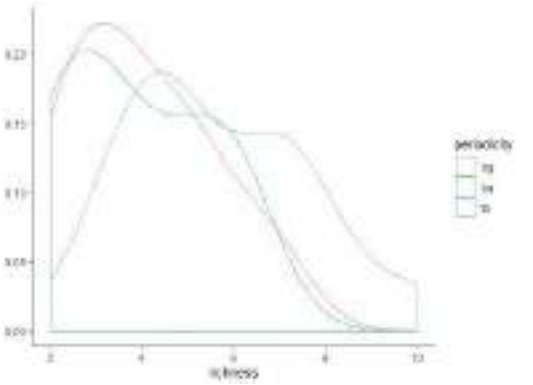
```
> qplot(richness, data = dendro, geom =
"histogram",fill=periodicity)
`stat_bin()` using `bins = 30`. Pick better value with
`binwidth`.
```



Gambar 154. Histogram kekayaan spesies dengan pengelompokkan periodisitas.

Berikut langkah untuk menampilkan grafik kekayaan spesies dengan periodisitas sebagai kelompok.

```
> qplot(richness, data = dendro, geom = "density",color = periodicity, linetype = periodicity)
```

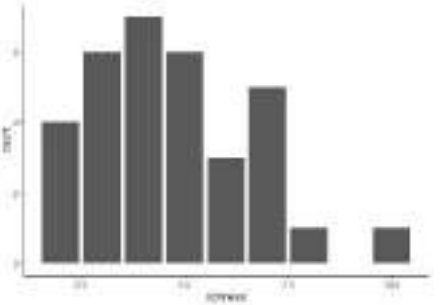


Gambar 155. Grafik kekayaan spesies dengan periodisitas sebagai kelompok

Langkah berikut memberikan tampilan berbagai pilihan warna histogram.

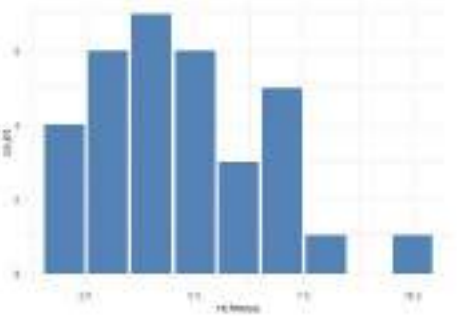
```
> attach(dendro)
The following objects are masked from indexgeo:
    elevation, slope, temperature
The following objects are masked from dendro2:
    ADI, cpa, diameter, elevation, periodicity, slope, temperature,
    Tree.Height, vitality
> b <- ggplot(dendro, aes(richness))
> # Basic plot
```

```
> b + geom_bar()
```



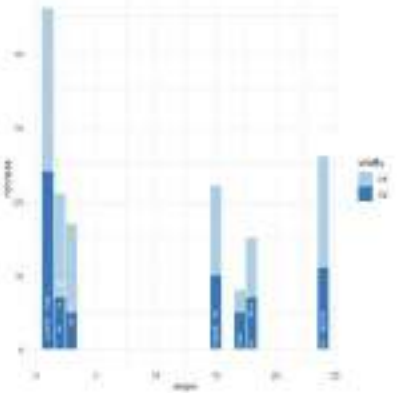
Gambar 156. Histogram kekayaan spesies dengan warna hitam

```
> b + geom_bar(fill = "steelblue", color ="steelblue")
+theme_minimal()
```



Gambar 157. Histogram warna biru dengan grid

```
> ggplot(data=dendro, aes(x=slope, y=richness, fill=vitality))
+geom_bar(stat="identity")+geom_text(aes(y=richness,label=richne
ss), vjust=1.6, color="white",
size=3.5)+scale_fill_brewer(palette="Paired")+theme_minimal()
```



Gambar 158. Histogram fungsi hubungan slope dan kekayaan spesies dengan vitalitas sebagai kelompok

BAB IX
HIERARCHICAL CLUSTERING ANALYSIS

9.1 Analisis Cluster

9.1.1 Pengantar

Analisis *cluster* merupakan suatu teknik analisis multivariat yang bertujuan untuk mengclusterkan data observasi ataupun variabel-variabel ke dalam *cluster* sedemikian rupa sehingga masing-masing *cluster* bersifat homogen sesuai dengan faktor yang digunakan untuk melakukan pengclusteran.

Ide dasar di balik k-means clustering terdiri dari pendefinisian cluster sehingga total variasi intra-cluster (dikenal sebagai variasi total dalam-cluster) diminimalkan. Ada beberapa algoritma k-means yang tersedia. Algoritma standar adalah algoritma Hartigan-Wong (Hartigan dan Wong 1979), yang mendefinisikan variasi total dalam-kluster sebagai jumlah jarak kuadrat Jarak Euclidean antara item dan centroid yang sesuai:

$$W(C_k) = \sum_{x_0 \in C_k} \left(x_i - \mu_k \right)^2$$

a. x_i mendesain titik data milik cluster C_k
b. μ_k adalah nilai rata-rata dari titik-titik yang ditetapkan untuk cluster C_k

Setiap pengamatan (x_i) ditugaskan untuk cluster yang diberikan sehingga jumlah jarak kuadrat (SS) pengamatan ke pusat cluster yang ditugaskan μ_k adalah minimum.

$$tot.withinss = \sum_{k=1}^{k} W(C_k) = \sum_{k=1}^{k} \sum_{x_0 \in C_k} \left(x_i - \mu_k \right)^2$$

Total dalam-kluster dari kuadrat mengukur kekompakan (yaitu *goodness*) dari pengelompokan dan kami ingin sekecil mungkin.

9.1.2. Algoritma K-means

Langkah pertama saat menggunakan k-means clustering adalah menunjukkan jumlah cluster (k) yang akan dihasilkan dalam solusi akhir. Algoritme dimulai dengan secara acak memilih objek k dari kumpulan data untuk dijadikan sebagai pusat awal untuk cluster. Objek yang dipilih juga dikenal sebagai sarana kluster atau centroid. Selanjutnya, masing-masing objek yang tersisa ditugaskan ke centroid terdekatnya, di mana terdekat didefinisikan menggunakan jarak Euclidean antara objek dan mean kluster. Langkah ini disebut "langkah penugasan klaster".

Perhatikan bahwa, untuk menggunakan jarak korelasi, data dimasukkan sebagai z-score.

Setelah langkah penugasan, algoritma menghitung nilai rata-rata baru dari setiap cluster. Istilah "pembaruan centroid" klaster digunakan untuk merancang langkah ini. Sekarang pusat telah dihitung ulang, setiap pengamatan diperiksa lagi untuk melihat apakah mungkin lebih dekat ke cluster yang berbeda. Semua objek ditugaskan kembali menggunakan sarana cluster diperbarui.

Langkah-langkah penugasan cluster dan pembaruan centroid diulangi secara berulang sampai penugasan cluster berhenti berubah (yaitu sampai konvergensi tercapai). Artinya, cluster yang terbentuk dalam iterasi saat ini sama dengan yang diperoleh pada iterasi sebelumnya. Algoritma K-means dapat diringkas sebagai berikut:

a. Tentukan jumlah cluster (K) yang akan dibuat (oleh analis)
b. Pilih objek k acak dari dataset sebagai pusat cluster awal atau sarana
c. Tetapkan setiap pengamatan pada centroid terdekat, berdasarkan jarak Euclidean antara objek dan centroid
d. Untuk setiap cluster k perbarui centroid kluster dengan menghitung nilai rata-rata baru dari semua titik data dalam kluster. Centoid dari cluster Kth adalah vektor dengan panjang p yang berisi nilai rata-rata dari semua variabel untuk pengamatan di cluster kth; p adalah jumlah variabel.
e. Secara minimum meminimalkan total dalam jumlah kuadrat. Yaitu, ulangi langkah c dan d sampai tugas cluster berhenti berubah atau jumlah iterasi maksimum tercapai. Secara default, perangkat lunak R menggunakan 10 sebagai nilai default untuk jumlah iterasi maksimum.

9.2 Komputasi K-Means Clustering di R

Metode hierarki (*hierarchical method*) yaitu metode yang memulai pengelompokkannya dengan dua atau lebih objek yang mempunyai kesamaan paling dekat, kemudian proses dilanjutkan ke obyek lain yang mempunyai kedekatan kedua. Biasanya pengelompokkan ini disajikan dalam bentuk dendogram, yang mirip dengan “struktur diagram pohon” (*tree diagram*). Dendogram adalah respentasi visual dari langkah-langkah analisis *cluster* yang menunjukkan bagaimana *cluster* terbentuk dan nilai koefisien jarak pada setiap langkah. Dalam hiraki terdapat beberapa macam:

a. *Divisive* (penyebaran). Dalam *divisive* ada 2 yaitu:
   1. *A Splintar Average Distance Method*
   2. *Automatic Interaction Detection*

b. *Aglomerative* (pemusatan). Ada 5 macam, diantaranya:

1. *Single linkage* (mengelompokan berdasarkan jarak terkecil antar objek)
2. *Complete linkage* (jarak terjauh)
3. *Average linkage* (rata-rata jarak seluruh individu dalam *cluster* dengan jarak seluruh individu *cluster* lain)
4. *Ward method* (total sum of square tiap dua cluster dalam masing-masing variabel)
5. *Centroid method* (jarak pusat dua *cluster*).

9.2 .1 Pengukuran Jarak Clustering

Klasifikasi pengamatan ke dalam kelompok memerlukan beberapa metode untuk menghitung jarak atau kesamaan (dis) antara setiap pasangan pengamatan. Hasil perhitungan ini dikenal sebagai dissimilarity atau distance matrix. Ada banyak metode untuk menghitung informasi jarak ini; pilihan ukuran jarak adalah langkah penting dalam pengelompokan. Ini mendefinisikan bagaimana kesamaan dua elemen (x, y) dihitung dan itu akan mempengaruhi bentuk cluster.

Pilihan ukuran jarak adalah langkah penting dalam pengelompokan. Ini mendefinisikan bagaimana kesamaan dua elemen (x, y) dihitung dan itu akan mempengaruhi bentuk cluster. Metode klasik untuk pengukuran jarak adalah jarak Euclidean dan Manhattan, yang didefinisikan sebagai berikut:

a. Jarak Euclidean:

$$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

b. Jarak Manhattan:

$$d_{man}(x,y) = \sum_{i=1}^{n}\left|(x_i - y_i)\right|$$

Di mana, x dan y adalah dua vektor panjang n.

Langkah-langkah ketidaksamaan lain ada seperti jarak berbasis korelasi, yang banyak digunakan untuk analisis data ekspresi gen. Jarak berbasis korelasi didefinisikan dengan mengurangi koefisien korelasi dari 1. Berbagai jenis metode korelasi dapat digunakan seperti:

c. Jarak korelasi Pearson:

$$d_{cor}(x,y) \equiv 1 - \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

d. Jarak korelasi Spearman:

Metode korelasi spearman menghitung korelasi antara pangkat x dan pangkat variabel y.

$$d_{spear}(x,y) = 1 - \frac{\sum_{i=1}^{n}(x_i' - \bar{x}')(y_i' - \bar{y}')}{\sqrt{\sum_{i=1}^{n}(x_i' - \bar{x}')^2 \sum_{i=1}^{n}(y_i' - \bar{y}')^2}}$$

Dimana: $x_i' = rank(x_i)$ dan $y_i' = rank(y_i)$

e. Jarak korelasi kendall:

Metode korelasi Kendall mengukur korespondensi antara peringkat variabel x dan y. Jumlah total pasangan yang mungkin dari x dengan pengamatan y adalah n (n - 1) / 2, di mana n adalah ukuran x dan y. Mulailah dengan memesan pasangan dengan nilai x. Jika x dan y berkorelasi, maka mereka akan memiliki urutan urutan relatif yang sama. Sekarang, untuk setiap yi, hitung jumlah yj> yi (pasangan konkordan (c)) dan jumlah yj <yi (pasangan sumbang (d)).

Jarak korelasi kendall didefinisikan sebagai berikut:

$$d_{kend}(x,y) = 1 - \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

Pilihan pengukuran jarak sangat penting, karena memiliki pengaruh kuat pada hasil pengelompokan. Untuk sebagian besar perangkat lunak pengelompokan umum, ukuran jarak default adalah jarak Euclidean. Namun, tergantung pada jenis data dan pertanyaan penelitian, langkah-langkah ketidaksamaan lainnya mungkin lebih disukai dan Anda harus menyadari pilihannya.

9.2.2 Metode Cluster

Sebelum kita menentukan metode cluster yang akan digunakan maka lakukan input data. *Copy* semua data, lalu gunakan *syntax* berikut untuk meng-*input* data ke program R.

```
> data<-read.delim("veg2")
> data(veg2)
```

Selanjutnya melakukan *cluster* hirarki *aglomerative.* Berikut syntax dari kelima metode :

```
#Average linkage
metode_a1<-hclust(dist(scale(data)),method = "ave")
plot(metode_a1)
#Single linkage
metode_s1<-hclust(dist(scale(data)),method = "single")
plot(metode_s1)
#Ward method
metode_ward<-hclust(dist(scale(data)),method = "ward.D")
plot(metode_ward)
#Centroid method
metode_centroid<-hclust(dist(scale(data)),method = "centroid")
plot(metode_centroid)
#Complete linkage
metode_c1<-hclust(dist(scale(data)),method = "complete")
plot(metode_c1)
```

Kali ini, hanya akan membandingkan 2 metode saja, yaitu metode *Complete linkage* dan *Ward method.* Berikut adalah output *cluster* dendogram dengan metode berbagai metode yang digunakan, diawali dengan menampilkan data di R Commander pada dataset sesuai dengan identitas data yang akan dianalisis. Pada kasus ini dataset yang digunakan adalah **veg2**.

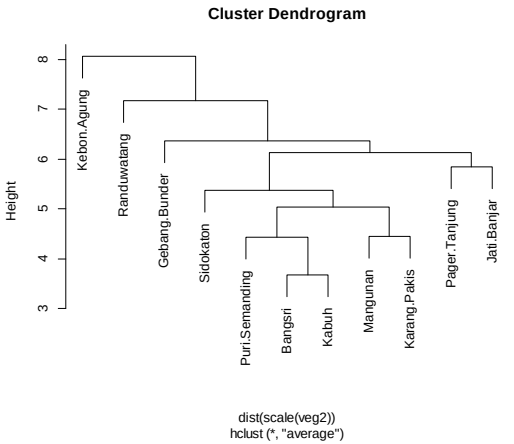
```
> data(veg2)
> veg2
             AS AM CN SE DZ CP GG AH AA PG SA SC CR DL MI PS AB TI PA NL SD
Bangsri       1  1  0  0  0  0  0  1  0  0  0  0  2  7  3  0  0  0  1  0  0
Puri.Semanding  2  3  0  0  0  0  0  1  0  0  1  2  0  8  1  2  0  0  0  0  0
Gebang.Bunder  0  0  0  0  1  0  0  0  0  1  0  0  2  6  3  0  3  1  0  0  0
Mangunan      0  0  0  0  0  1  0  2  0  0  0  0  0  8  9  0  0  0  0  0  0
Kabuh         2  0  0  0  0  0  0  0  0  0  1  0  0  2  1  0  0  0  0  0  0
Karang.Pakis  0  0  2  0  0  1  0  1  0  0  0  1  0  5  4  1  1  0  0  1  0
Pager.Tanjung  4  4  0  0  0  0  0  0  0  5  0  1  0  9  7  0  2  0  2  0  0
Kebon.Agung   1  1  0  0  0  0  1  2  3  3  3  0  0  1  6  0  2  0  1  4  0
Jati.Banjar   0  0  0  0  0  1  0  0  0  2  1  1  0  6  8  1  4  1  3  0  0
Sidokaton     1  2  0  0  0  0  0  0  0  3  0  0  0  1  7  7  0  0  0  0  0
Randuwatang   4  0  0  5  0  0  0  0  0  1  2  0  1  2  1  4  0  0  0  0  1
```

1) Metode Average Lingkage

Kode yang diinputkan ke R Consolenya adalah:

```
> metode_a1<-hclust(dist(scale(veg2)),method = "ave")
> plot(metode_a1)#Single lingked
```

Output Cluster Dendrogram:

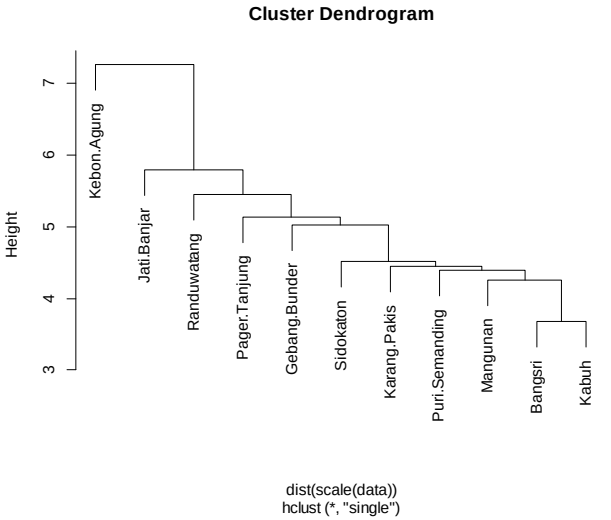


Gambar 159. Dendrogram metode Average Lingkage

2) Metode Single Lingkage

Kode yang diinputkan ke R Consolenya adalah:

```
> metode_s1<-hclust(dist(scale(veg2)),method = "single")
> plot(metode_s1)#Ward method
```

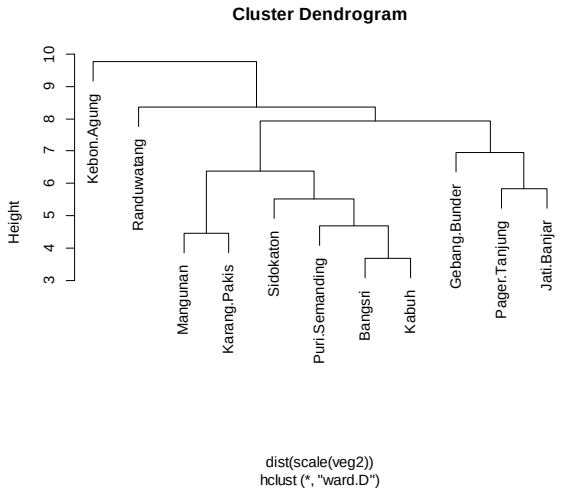


Gambar 160. Dendrogram metode Single Lingkage

3) Metode Ward.D

Kode yang diinputkan ke R Consolenya adalah:

```
> metode_ward<-hclust(dist(scale(veg2)),method = "ward.D")
> plot(metode_ward)
```

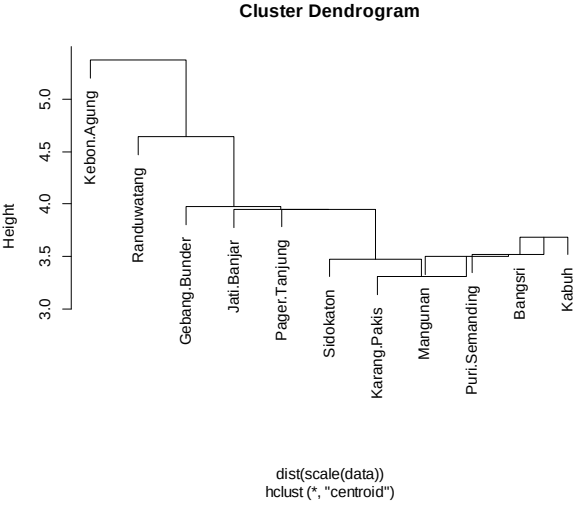


Gambar 161. Dendrogram metode Ward.D

4) Metode Centroid

Kode yang diinputkan ke R Consolenya adalah:

```
> metode_centroid<-hclust(dist(scale(veg2)),method = "centroid")
> plot(metode_centroid)
```

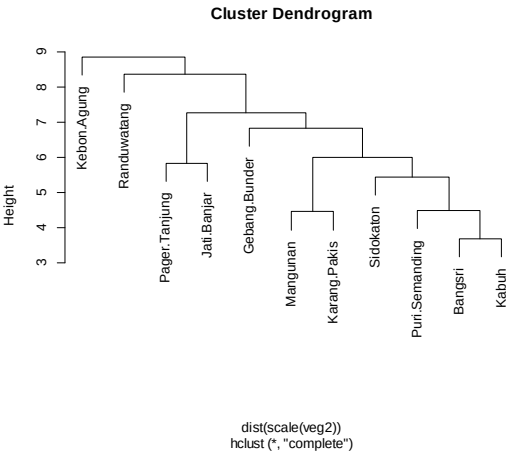


Gambar 162. Dendrogram metode Centroid

5) Metode Complete Lingkage

Kode yang diinputkan ke R Consolenya adalah:

```
> metode_c1<-hclust(dist(scale(veg2)),method = "complete")
> plot(metode_c1)
```

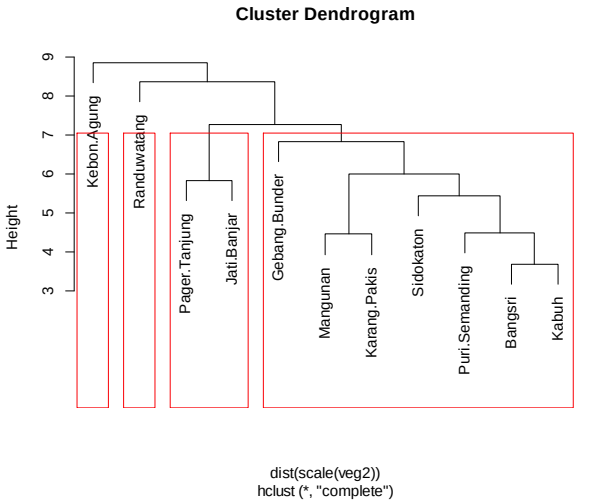


Gambar 163. Dendrogram metode Complete Lingkage

Berdasarkan dendogram tersebut, kita akan membagi desa-desa tersebut kedalam 4 kelompok, menggunakan syntax berikut :

```
> rect.hclust(metode_c1, 4)
```

Sehingga diperoleh output :



Gambar 164. Dendrogram dengan 4 Cluster

Berdasarkan *output* di atas, dari 11 desa yang ada terbagi menjadi empat kelompok. Untuk melihat desa mana saja yang termasuk ke dalam kelompok-kelompok tersebut, maka gunakan syntax sebagai berikut:

```
> kelompok<-cutree(metode_c1, 4)
> kelompok
```

Output pembagian kelompok desa sebagai berikut:

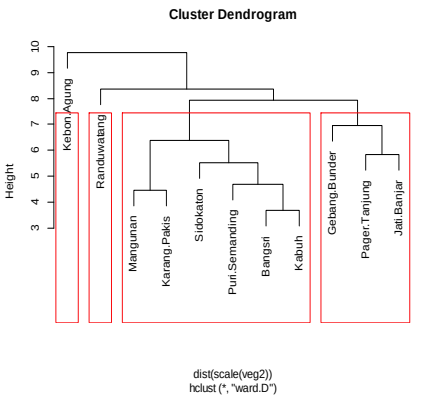
```
       Bangsri Puri.Semanding  Gebang.Bunder       Mangunan          Kabuh
             1              1              1              1              1
  Karang.Pakis  Pager.Tanjung    Kebon.Agung    Jati.Banjar      Sidokaton
             1              2              3              2              1
   Randuwatang
             4
> tabel=data.frame(kelompok)
> tabel
```

Output tabel kelompok berdasarkan hasil analaisis kluster dengan metode Complete Lingkage adalah:

```
               kelompok
Bangsri               1
Puri.Semanding        1
Gebang.Bunder         1
Mangunan              1
Kabuh                 1
Karang.Pakis          1
Pager.Tanjung         2
Kebon.Agung           3
Jati.Banjar           2
Sidokaton             1
Randuwatang           4
```

Berdasarkan dendogram tersebut, akan membagi desa-desa tersebut kedalam 4 kelompok, menggunakan syntax berikut :

```
> plot(metode_ward)
> rect.hclust(metode_ward, 4)
```



Gambar 165. Dendrogram 4 Cluster dengan metode Complete Lingkage

Berdasarkan *output* di atas, dari 11 desa yang ada terbagi menjadi empat kelompok. Untuk melihat desa mana saja yang termasuk ke dalam kelompok-kelompok tersebut, maka gunakan syntax sebagai berikut:

```
> kelompok<-cutree(metode_ward, 4)
```

```
> kelompok
```

Output pembagian kelompok desa sebagai berikut:

```
Bangsri Puri.Semanding  Gebang.Bunder       Mangunan          Kabuh
             1              1              2             1             1
  Karang.Pakis  Pager.Tanjung    Kebon.Agung    Jati.Banjar      Sidokaton
             1              2              3             2             1
   Randuwatang
             4
> tabel=data.frame(kelompok)
> tabel
```

Output tabel kelompok desa dengan metode Ward.D adalah sebagai berikut:

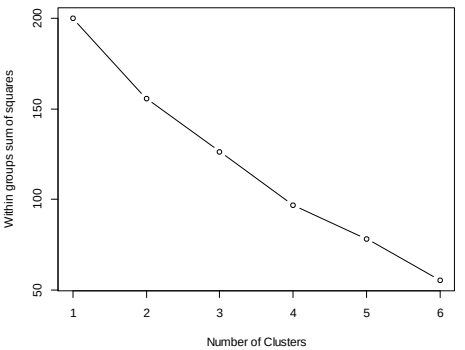
```
kelompok
Bangsri               1
Puri.Semanding        1
Gebang.Bunder         2
Mangunan              1
Kabuh                 1
Karang.Pakis          1
Pager.Tanjung         2
Kebon.Agung           3
Jati.Banjar           2
Sidokaton             1
Randuwatang           4
```

9.2.3 Clustering menggunakan Paket Rattle

Langkah berikutnya akan dilakukan analisis kluster dengan menggunakan paket Rattle.

```
> library(rattle)
Warning: package 'rattle' was built under R version 3.6.3
Loading required package: tibble
Loading required package: bitops
Rattle: A free graphical interface for data science with R.
Version 5.4.0 Copyright (c) 2006-2020 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
> veg2.stand <- scale(veg2[-1]) # untuk menstandarkan variabel
> # K-Means
> k.means.fit <- kmeans(veg2.stand, 3) # k = 3
> attributes(k.means.fit)

$names
[1] "cluster"      "centers"      "totss"        "withinss"
"tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"

$class
[1] "kmeans"

> # Cluster
> k.means.fit$cluster

      Bangsri Puri.Semanding  Gebang.Bunder       Mangunan          Kabuh
             2              2              1             2             2
  Karang.Pakis  Pager.Tanjung    Kebon.Agung    Jati.Banjar      Sidokaton
             2              1              3             1             2
   Randuwatang
             2
```

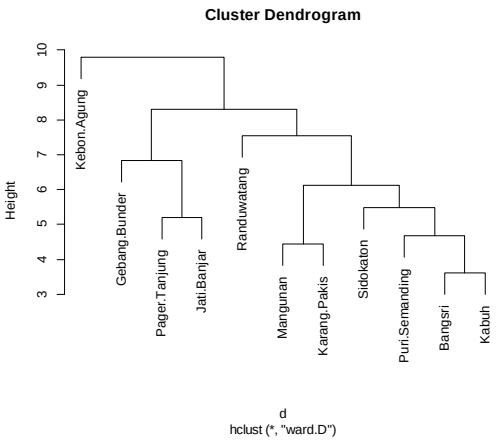
```
> # Cluster size:
> k.means.fit$size
[1] 3 7 1

> wssplot <- function(data, nc=15, seed=1234){
+   wss <- (nrow(data)-1)*sum(apply(data,2,var))
+   for (i in 2:nc){
+     set.seed(seed)
+     wss[i] <- sum(kmeans(data, centers=i)$withinss)}
+   plot(1:nc, wss, type="b", xlab="Number of Clusters",
+        ylab="Within groups sum of squares")}
> wssplot(veg2.stand, nc=6)
```



Gambar 166. Grafik penentuan jumlah Cluster Optimal

Yang mana nilai optimal dalam kasus ini? Mangapa?
Library cluster memperkenankan kita untuk menghadirkan (dengan tujuan PCA) solusi cluster dalam 2 dimensi.

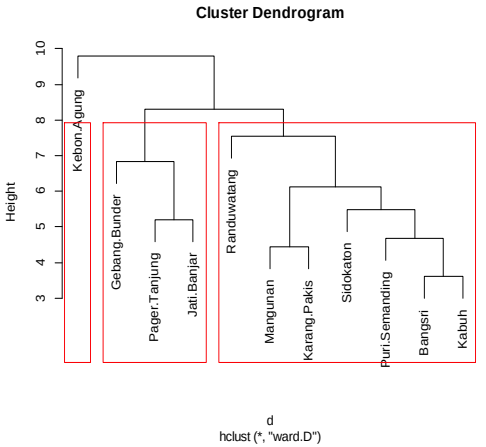
```
> library(cluster)
>     clusplot(veg2.stand,     k.means.fit$cluster,     main='2D
representation of the Cluster solution',
+          color=TRUE, shade=TRUE,labels=2, lines=0)

> table(veg2[,1],k.means.fit$cluster)

   1 2 3
  0 2 2 0
  1 0 2 1
  2 0 2 0
  4 1 1 0
> d <- dist(veg2.stand, method = "euclidean") # Euclidean distan
ce matrix.
> H.fit <- hclust(d, method="ward")
The "ward" method has been renamed to "ward.D"; note new "ward.D
2"
> plot(H.fit) # display dendogram
```



Gambar 167. Dendrogram dengan metode Ward.D

Selanjutnya dibangun dendrogram dengan membentuk 3 Cluster.

```
> groups <- cutree(H.fit, k=3) # cut tree into 3 clusters
> # draw dendogram with red borders around the 3 clusters
> rect.hclust(H.fit, k=3, border="red")
```



Gambar 168. Dendrogram metode Ward.D dengan 3 Cluster

```
> table(veg2[,1],groups)
   groups
     1 2 3
  0 2 2 0
  1 2 0 1
  2 2 0 0
  4 1 1 0
```

9.2.4 Jumlah Optimal Cluster

Sebagai upaya untuk menemukan jumlah cluster optimal untuk k-means, upaya ini direkomendasikan untuk memilih optimalisasi cluster berdasarkan pada:

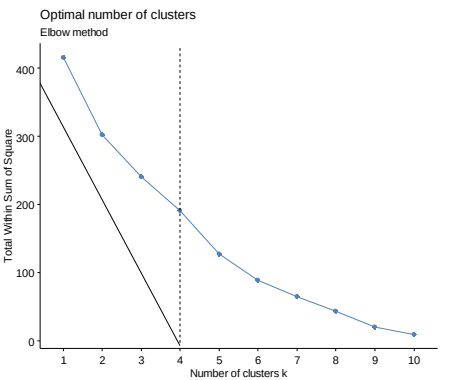
1) Kontek pada permasalahan yang ditangani, untuk cepatnya jika diketahui bahwa permasalahan tersebut pada jumlah kelompok yang spesifik dalam data (ini adalah opsi yang subyektif), atau:
2) mengikuti empat pendekatan berikut:
   a. Metode Elbow (yang menggunakan dalam jumlah cluster pada kuadrat)
   b. Metode rata-rata silhouette
   c. MMetode statistic gap
   d. Fungsi NbClust()

Selanjutnya kan tunjukkan kode R untuk 4 metode berikut, untuk informasi yang lebih teorikal .

a. Metode Elbow

Metode ini terlihat pada jumlah cluster dalam kuadrat (*Within - cluster sum of square* = WSS) sebagai fungsi pada jumlah cluster.

```
> # load required packages
> library(factoextra)
Loading required package: ggplot2
Welcome! Related Books: `Practical Guide To Cluster Analysis in
R` at https://goo.gl/13EFCZ
> library(NbClust)
> # Elbow method
> fviz_nbclust(veg2, kmeans, method = "wss") +
+   geom_vline(xintercept = 4, linetype = 2) + # add line for
better visualisation
+   labs(subtitle = "Elbow method") # add subtitle
>
```



Gambar 169. Grafik penentapan Cluster Optimum

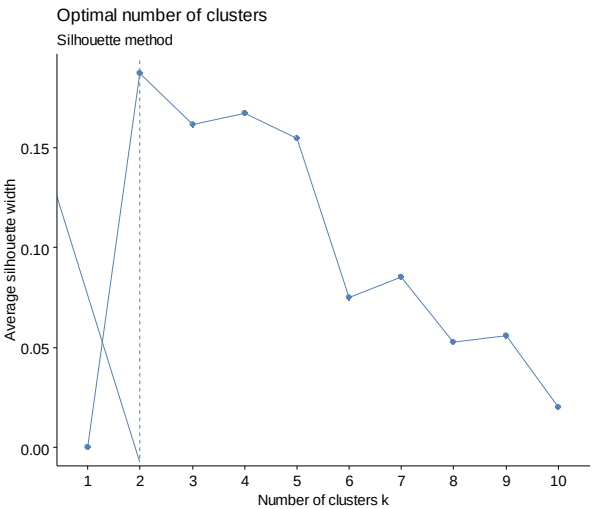
Lokasi pada garis putus-putus tegak lurus pada plot bisanya dipertimbangkan sebagai indikator untuk ptendekatan jumlah cluster, karena hal ini berarti penambahan cluster lainnya tidak memperbaiki

lebih baik secara terpisah. Metode ini terlihat untuk menyarankan 4 cluster. Metode Elbow kadang-kadang membingunkan dan sebagai alternatifnya digunakan metode rata-rata Silhouette.

b. Metode Silhouette

Metode ini mengukur kualitas cluster dan menentukan seberapa baik tiap-tiap garis titik dalam cluster tersebut.

```
> # Silhouette method
> fviz_nbclust(veg2, kmeans, method = "silhouette") +
+   labs(subtitle = "Silhouette method")
```



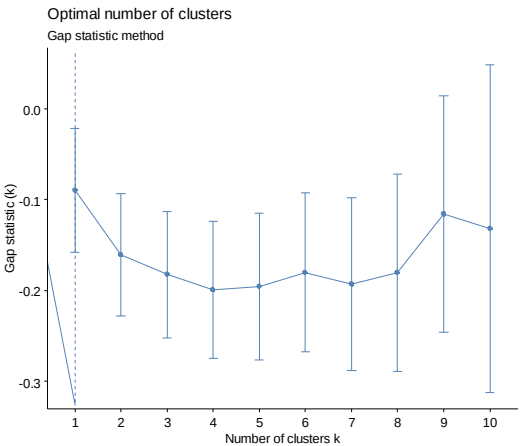
Gambar 170. Grafik penetapan jumlah Cluster optimum dengan metode Silhouette

Hasil tampilan plot metode Silhouette menyarankan pengelompokkan dalam 2 cluster.

c. Metode Statistik Gap

```
> # Gap statistic
> set.seed(42)
> fviz_nbclust(veg2, kmeans,
+   nstart = 25,
+   method = "gap_stat",
+   nboot = 500
+ ) + # reduce it for lower computation time (but less precise
results)
+   labs(subtitle = "Gap statistic method")
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 500)  [one "." per sample]:
.................................................. 50
.................................................. 100
.................................................. 150
.................................................. 200
```

```
.................................................. 250
.................................................. 300
.................................................. 350
.................................................. 400
.................................................. 450
.................................................. 500
```



Gambar 171. Grafik penentapan Custer Optimum dengan metode Statistic Grap

Jumlah optimum cluster adalah satu yang memaksimalkan statistic gap. Metode ini menyarankan hanya satu cluster (yang mana bahwa pengklateran tidak berguna). Seperti yang kita lihat bahwa ketiga metode tidak begitu penting menuju pada hasil yang sama. Dimana, 3 pendekatan menyarankan jumlah yang berbeda pada cluster.

d. Metode fungsi NbClust()

Alternatif keempat adalah menggunakan fungsi NbClust(), yang menyediakan 30 tanda untuk pemilihan jumlah cluster terbaik.

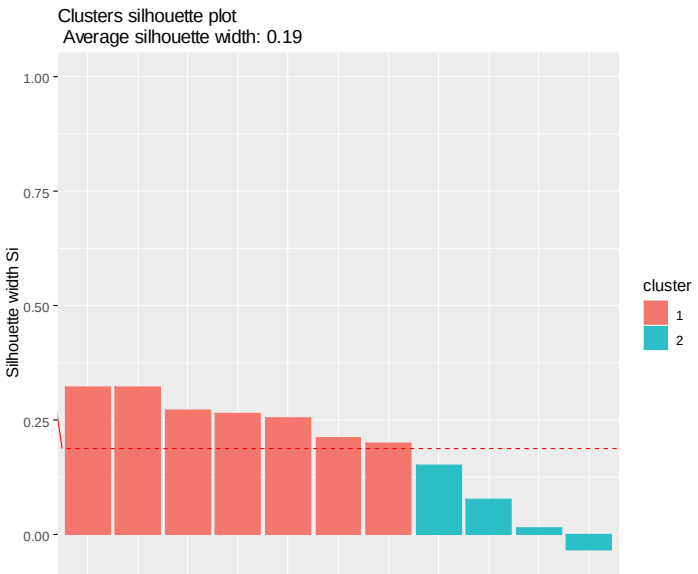
```
> nbclust_out <- NbClust(
+   data = veg2,
+   distance = "euclidean",
+   min.nc = 1, # minimum number of clusters
+   max.nc = 3, # maximum number of clusters
+   method = "kmeans" # one of: "ward.D", "ward.D2", "single",
"complete", "average", "mcquitty", "median", "centroid",
"kmeans"
+ )
```

9.2.5 Visualisasi

Untuk mengkonfirmasikan bahwa jumlah kelas secara mendalam optimal, disini ada cara untuk menilai kualitas cluster kamu melalui plot silhouette (yang menunjukkan koefisien silhouette pada aksis x). Kami

menggambarkan plot silhouette untuk 2 cluster, seperti yang disarankan oleh metode silhouette rata-rata.

```
> library(cluster)
> set.seed(42)
> km_res <- kmeans(veg2, centers = 2, nstart = 20)
>
> sil <- silhouette(km_res$cluster, dist(veg2))
> fviz_silhouette(sil)
  cluster size ave.sil.width
1       1    7          0.26
2       2    4          0.05
```



Gambar 172. Grafik penentapan Cluster Optimum dengan nilai koefisien Silhouette

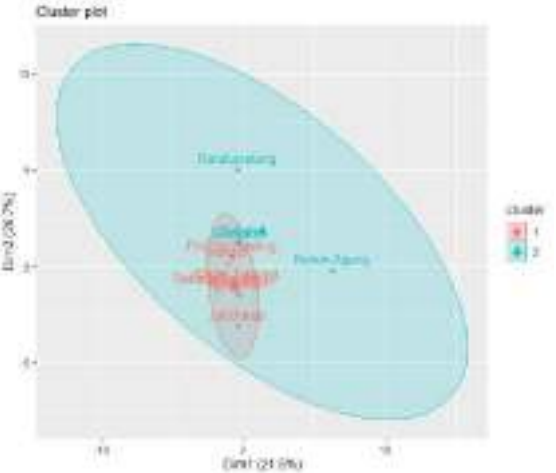
Sebagai pengingat bahwa, interpretasi koefisien silhouette adalah sebagai berikut:

1. Nilai : 0 > 0 berarti bahwa pengamatan adalah terkelompokkan dengan baik. Koefisien tertutup adalah 1, pengamatan terbaik adalah sudah terkelompokkan.
2. Nilai : <0 < 0 berarti bahwa pengamatan telah ditempatkan pada cluster yang salah
3. Nilai : =0 = 0 berarti bahwa pengamatan adalah antara dua cluster.

Plot silhouette tersebut dan koefisien silhouette rata-rata membantu untuk menentukan apakah cluster kamu baik atau tidak. Jika secara luas koefisien silhouette positif, ini menandakan bahwa pengamatan ditempatkan pada kelompok yang tepat. Plot silhouette selain itu dapat digunakan dalam pemilihan jumlah optimal kelas. Jika

dimungkinkan untuk mengplot cluster oleh penggunaan fungsi fviz_cluster(). Catatan bahwa analisis principal komponen (PCA) dibentuk untuk menghadirkan variabel dalam plane 2 dimensi.

```
> library(factoextra)
> fviz_cluster(km_res, veg2, ellipse.type = "norm")
```



Gambar 173. Grafik fungsi Cluster dengan pendekatan PCA

9.3 Normalisasi

Normalisasi adalah sangat penting dalam analisis cluster, kadang-kadang kita memiliki variabel dengan skala berbeda, perlu dinormalkan didasarkan dalam fungsi skala sebelum pengklasteran pada dataset.

Normalisasi didasarsi untuk analisis cluster.

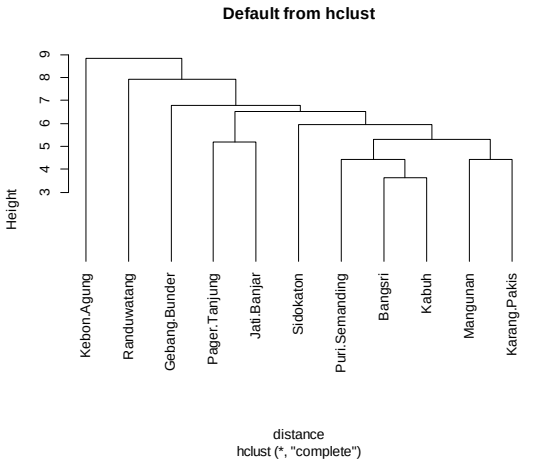
```
> mydata <- veg2
z <- mydata[,-c(1,1)]
means <- apply(z,2,mean)
sds <- apply(z,2,sd)
nor <- scale(z,center=means,scale=sds)
```

Menghitung jarak matrik

```
> distance = dist(nor)
```

Clustering Hierarchical Agglomerative

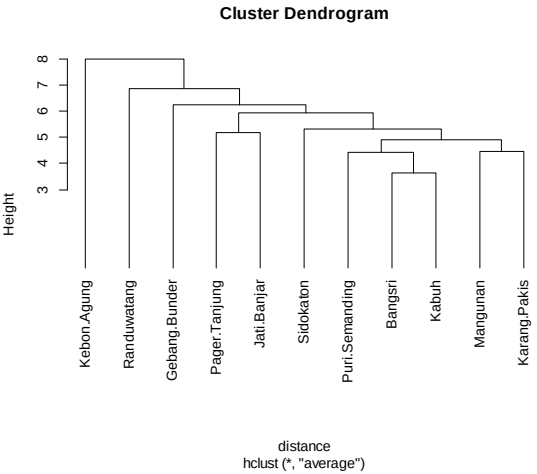
```
mydata.hclust = hclust(distance)
plot(mydata.hclust)
plot(mydata.hclust,labels=mydata$Company,main='Default from
hclust')
plot(mydata.hclust,hang=-1,   labels=mydata$Company,main='Default
from hclust')
```



Gambar 174. Dendrogram dengan metode Hierarchical Agglomerative

Clustering Hierarchical Agglomerative menggunakan "Average" Lingkage

```
> mydata.hclust<-hclust(distance,method="average")
> plot(mydata.hclust,hang=-1)
```



Gambar 175. Dendrogram hclust dengan Average Lingkage

Pengelompokkan Cluster

```
> member = cutree(mydata.hclust,3)
> table(member)
member
1 2 3
9 1 1
```

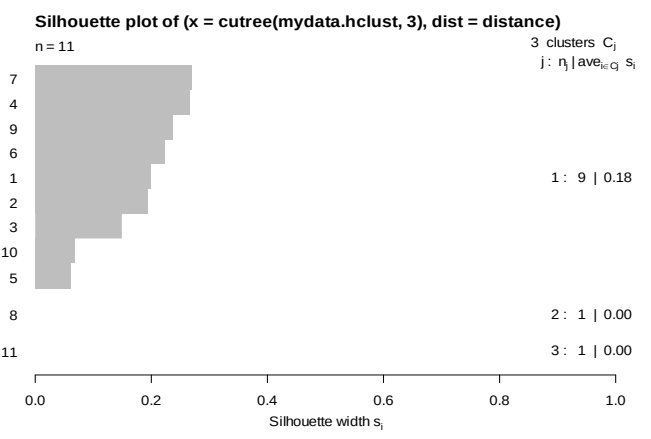
Cluster Karakteristik

```
aggregate(nor,list(member),mean)
```

```
Group.1          AM          CN          SE          DZ
CP         GG
1       1  0.07856742  0.06700252 -0.3015113  0.06700252
0.1297498 -0.3015113
2       2  0.00000000 -0.30151134 -0.3015113 -0.30151134 -
0.5838742  3.0151134
3       3 -0.70710678 -0.30151134  3.0151134 -0.30151134 -
0.5838742 -0.3015113
           AH          AA          PG          SA          SC
CR
1 -0.09988146 -0.3015113 -0.08369979 -0.3904062  0.1469127 -
0.01231531
2  1.68549966  3.0151134  0.96852612  2.2523437 -0.6611074 -
0.55418887
3 -0.78656651 -0.3015113 -0.21522803  1.2613124 -0.6611074
0.66502665
          DL          MI          PS          AB          TI
PA
1  0.2592593  0.07799452 -0.06290017  0.01397099  0.09988146
0.02950893
2 -1.3333333  0.48831350 -0.60653731  0.62869461 -0.44946657
0.35410712
3 -1.0000000 -1.19026416  1.17263880 -0.75443354 -0.44946657 -
0.61968745
          NL          SD
1 -0.2829975 -0.3015113
2  2.9215327 -0.3015113
3 -0.3745555  3.0151134
```

Silhouette Plot

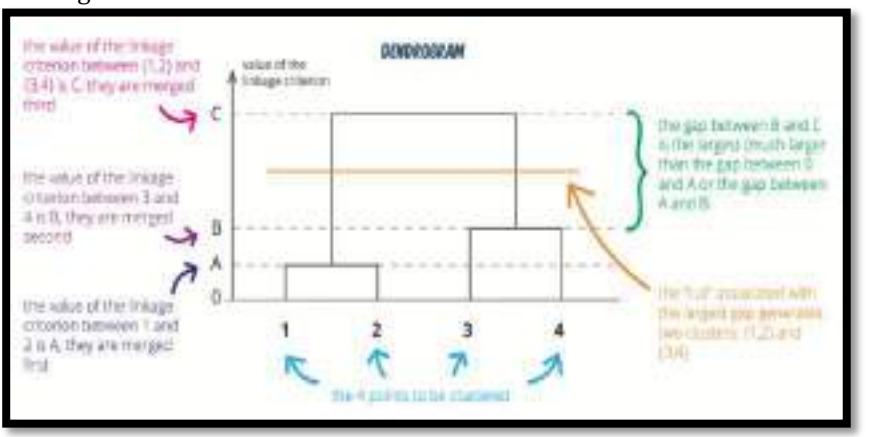
```
> library(cluster)
> plot(silhouette(cutree(mydata.hclust,3), distance))
```



Gambar 176. Silhouette Plot dengan 3 Cluster

Berdasarkan plot diatas, jika beberapa bar menjadi sebagai sisi negative, maka kita bisa memasukkan data utama adalah sebuah outlier dapat diubah dari data kita.

Plotiing K-means



Gambar 177. Bagan dari Dendrogram

Kesimpulan

K-mean clustering adalah sangat mudah dan algoritmanya cepat dan bisa efisien dengan dataset sangat besar. K-,mean clustering perlu untuk menyediakan sejumlah kluster sebagai luaran. Hierarchical clustering adalah sebuah pendekatan alternative yang tidak memerlukan persyaratan yang kita komit untuk pemilihan utama kkluster.

Referensi

Usman, H., & Sobari, N. (2013). *Aplikasi Teknik Multivariate Untuk Riset Pemasaran.* Jakarta: PT Grafindo Persada.

BAB X
ANALISIS KOMPONEN UTAMA DALAM R

10.1 Pendahuluan

Dalam tutorial ini, akan dipelajari cara menggunakan PCA untuk mengekstraksi data dengan banyak variabel dan membuat visualisasi untuk menampilkan data itu. *Principal Component Analysis* (PCA) adalah teknik yang berguna untuk analisis data eksplorasi, yang memungkinkan untuk memvisualisasikan variasi hadir dalam kelompok data dengan lebih banyak variabel. Ini sangat membantu dalam kasus dataset "lebar", di mana memiliki banyak variabel untuk setiap sampel. Dalam tutorial ini, akan ditemukan PCA di R. Lebih khusus, menangani topik-topik berikut:

1. Pertama-tama tentang pengantar PCA: belajar tentang komponen utama dan bagaimana berhubungannya dengan nilai eigen dan vektor eigen.
2. Kemudian, mencoba PCA sederhana dengan kumpulan data yang sederhana dan mudah dipahami.
3. Selanjutnya, menggunakan hasil dari bagian sebelumnya untuk merencanakan PCA pertama - Visualisasi sangat penting!
4. Melihat bagaimana dapat mulai menafsirkan hasil visualisasi ini dan Cara mengatur parameter grafis plot dengan paket ggbiplot!
5. Tentu saja, diharapkan visualisasi sesesuaikan mungkin, dan itulah sebabnya juga akan membahas beberapa cara untuk melakukan penyesuaian tambahan pada plot !
6. Melihat bagaimana cara menambahkan sampel baru ke plot dan memproyeksikan sampel baru ke PCA asli.

10.2 PCA

Seperti yang sudah dijelaskaan di pendahuluan, PCA sangat berguna ketika bekerja dengan kelompok data "lebar". Dalam kasus seperti itu, di mana banyak variabel hadir, tidak dapat dengan mudah memplot data dalam format mentahnya, sehingga sulit untuk memahami tren yang ada di dalamnya. PCA memungkinkan kita melihat "bentuk" keseluruhan data, mengidentifikasi sampel mana yang mirip satu sama lain dan mana yang sangat berbeda. Ini dapat memungkinkan untuk mengidentifikasi kelompok sampel yang serupa dan mencari tahu variabel mana yang membuat satu kelompok berbeda dari yang lain.

Matematika yang mendasarinya agak rumit, jadi tidak akan membahas terlalu banyak detail, tetapi dasar-dasar PCA adalah sebagai berikut: mengambil kelompok data dengan banyak variabel, dan menyederhanakan kelompok dataset tersebut dengan mengubah variabel asli menjadi angka yang lebih kecil dari "Komponen Utama".

Tapi apa sebenarnya ini? Komponen Utama adalah struktur yang mendasari dalam data. PCA adalah arah di mana ada varian yang paling, arah di mana data paling tersebar. Ini berarti bahwa mencoba menemukan garis lurus yang paling baik menyebarkan data ketika diproyeksikan sepanjang itu. Ini adalah komponen utama pertama, garis lurus yang menunjukkan varians paling substansial dalam data.

PCA adalah jenis transformasi linear pada kelompok data yang diberikan yang memiliki nilai untuk sejumlah variabel (koordinat) untuk jumlah ruang tertentu. Transformasi linear ini cocok dengan kemlompok data ini ke sistem koordinat baru sedemikian rupa sehingga varians paling signifikan ditemukan pada koordinat pertama, dan setiap koordinat berikutnya adalah ortogonal ke yang terakhir dan memiliki varian yang lebih rendah. Dengan cara ini, bisa mentransformasikan sekumpulan variabel berkorelasi x atas sampel y ke sekumpulan komponen utama tidak berkorelasi pada sampel yang sama.

Di mana banyak variabel berkorelasi satu sama lain, semuanya akan berkontribusi kuat pada komponen utama yang sama. Setiap komponen utama merangkum persentase tertentu dari total variasi dalam kelompok data. Di mana variabel awal sangat berkorelasi satu sama lain dapat memperkirakan sebagian besar kompleksitas dalam kelompok data hanya dengan beberapa komponen utama. Saat menambahkan lebih banyak komponen utama, akan meringkas lebih banyak dan lebih banyak dari kelompok data asli. Menambahkan komponen tambahan membuat perkiraan dari total kelompok data lebih akurat, tetapi juga lebih berat.

10.3 Nilai eigen dan vektor Eigen

Sama seperti banyak hal dalam hidup, vektor eigen, dan nilai eigen berpasangan: setiap vektor eigen memiliki nilai eigen yang sesuai. Sederhananya, vektor eigen adalah arah, seperti "vertikal" atau "45 derajat", sedangkan nilai eigen adalah angka yang menunjukkan berapa banyak variasi dalam data ke arah itu. Oleh karena itu vektor eigen dengan nilai eigen tertinggi adalah komponen utama pertama. Jadi

mungkin ada lebih banyak nilai eigen dan vektor eigen yang bisa ditemukan dalam satu set data.

Itu benar! Jumlah nilai eigen dan vektor eigen yang keluar sama dengan jumlah dimensi yang dimiliki kumpulan data. Dalam contoh yang terlihat di atas, ada 2 variabel, sehingga kumpulan data dua dimensi. Itu berarti ada dua vektor eigen dan nilai eigen. Demikian pula, akan ditemukan tiga pasangan dalam kelompok data tiga dimensi. Kita dapat membingkai ulang dataset dalam hal vektor eigen dan nilai eigen ini tanpa mengubah informasi yang mendasarinya. Perhatikan bahwa membingkai ulang kumpulan data mengenai serangkaian nilai eigen dan vektor eigen tidak berarti mengubah data itu sendiri, hanya melihatnya dari sudut yang berbeda, yang seharusnya mewakili data dengan lebih baik.

Sekarang setelah melihat beberapa teori di balik PCA, maka siap untuk melihat semuanya beraksi!

10.4 Fungsi untuk melakukan Analisis Komponen Utama dalam R

Principal Component Analysis (PCA) adalah teknik multivariat yang memungkinkan kami untuk merangkum pola variasi sistematis dalam data.Dari sudut pandang analisis data, PCA digunakan untuk mempelajari satu tabel pengamatan dan variabel dengan gagasan utama mengubah variabel yang diamati menjadi satu set variabel baru, komponen utama, yang tidak berkorelasi dan menjelaskan variasi dalam data. Untuk alasan ini, PCA memungkinkan untuk mereduksi data "kompleks" yang ditetapkan ke dimensi yang lebih rendah untuk mengungkapkan struktur atau jenis variasi dominan dalam pengamatan dan variabel.

a. PCA dalam R

Di R, ada beberapa fungsi dari paket yang berbeda yang memungkinkan untuk melakukan PCA. Dalam penulisan ini ditunjukkan 5 cara berbeda untuk melakukan PCA menggunakan fungsi berikut (dengan paket yang sesuai dalam tanda kurung):

- `prcomp () (statistik)`
- `princomp () (statistik)`
- `PCA () (FactoMineR)`
- `dudi.pca () (ade4)`
- `acp () (amap)`

Catatan singkat: Bukan kebetulan bahwa tiga paket eksternal ("FactoMineR", "ade4", dan "amap") telah dikembangkan oleh analis

data, yang memiliki tradisi panjang dan preferensi untuk PCA dan teknik eksplorasi terkait lainnya.

Apa pun fungsi yang diputuskan untuk digunakan, hasil PCA tipikal harus terdiri dari sekumpulan nilai eigen, tabel dengan skor atau Komponen Utama (PC), dan tabel pemuatan (atau korelasi antara variabel dan PC). Nilai eigen memberikan informasi tentang variabilitas dalam data. Skor memberikan informasi tentang struktur pengamatan. Pemuatan (atau korelasi) memungkinkan untuk merasakan hubungan antara variabel, serta hubungannya dengan PC yang diekstraksi.

b. Data

Untuk mempermudah, kami akan menggunakan dataset USArrests yang sudah dilengkapi dengan R. Ini adalah kerangka data dengan 11 baris (desa) dan 4 kolom yang berisi informasi tentang karakteristik geografis dan kandungan hara tanah. Karena sebagian besar kali variabel diukur dalam skala yang berbeda, PCA harus dilakukan dengan data terstandarisasi (rata-rata = 0, varians = 1). Kabar baiknya adalah bahwa semua fungsi yang melakukan PCA dilengkapi dengan parameter untuk menentukan bahwa analisis harus diterapkan pada data standar.

Opsi 1: menggunakan prcomp ()

Fungsi prcomp () dilengkapi dengan paket "stats" default, yang berarti tidak perlu menginstal apa pun. Ini mungkin cara tercepat untuk melakukan PCA jika tidak ingin menginstal paket lain. Misalkan dalam operasi ini digunakan dataset **enviro2**.

```
> # PCA with function prcomp
> pca1 = prcomp(enviro2, scale. = TRUE)
> # sqrt of eigenvalues
> pca1$sdev
[1] 2.980871e+00 1.409889e+00 8.880204e-01 7.437252e-01 5.479946e-01
 [6] 4.637139e-01 3.854768e-01 2.720776e-01 2.017983e-01 7.900791e-02
[11] 3.437833e-16
> # loadings
> head(pca1$rotation)
                   PC1        PC2        PC3         PC4         PC5
elevation   -0.3063813  0.1272401  0.1789403 -0.03221748  0.47416133
slope       -0.2875052 -0.3132254  0.1292585  0.18020406 -0.29204353
temperature  0.2729373  0.3402986  0.2166932  0.10811270  0.08846613
C.organic   -0.3085687 -0.1254693  0.1417838  0.06150329 -0.39702088
N.total     -0.2187087 -0.1765794 -0.6937518  0.38633936  0.36329429
P2O5        -0.2732845  0.3161576  0.2538088  0.30657610  0.24992423
                   PC6         PC7         PC8          PC9        PC10
elevation    0.14133664 -0.31995222 -0.27659107 -0.610938674 0.05924415
slope        0.03258273  0.26032390  0.08091774 -0.153878059 0.56674221
temperature -0.40825892 -0.39582569  0.19519324  0.007011539 0.12024750
```

```
C.organic    0.19629981 -0.56280347 -0.12788152  0.117852955 0.20432186
N.total     -0.20582930 -0.11221924  0.02712222  0.046791502 0.14778441
P2O5         0.01695202 -0.03770765  0.17637454  0.572209123 0.19611280
                   PC11
elevation    0.13391617
slope       -0.27575068
temperature -0.57914858
C.organic   -0.01537493
N.total     -0.14675732
P2O5         0.40434449

> # PCs (aka scores)
> head(pca1$x)
                      PC1        PC2         PC3         PC4         PC5
Bangsri        -5.01118134  0.2373628 -1.31000424  0.33905040 -0.57265530
Puri.Semanding -3.86199299 -0.6871719 -0.43424954 -0.40994314 -0.22419009
Gebang.Bunder  -2.49765984 -1.3318864  0.09605264 -0.01442784  1.04354587
Mangunan       -0.20742279 -0.7777755  1.27211339 -0.95373821 -0.04371489
Kabuh           0.03791742 -1.5612459  0.88140143  0.44568884  0.14208968
Karangpakis     0.31290537  0.3130755  1.14163231  1.15519300 -0.77416887
                      PC6        PC7        PC8         PC9        PC10
Bangsri        -0.3389827 -0.1393525  0.1955473 -0.14583630 -0.08120780
Puri.Semanding  0.6322102  0.3472085 -0.4200570  0.10262349  0.08480774
Gebang.Bunder   0.1053847 -0.4676475  0.3122381  0.18468206  0.03352180
Mangunan        0.2217403  0.1359093  0.1817175 -0.41581331 -0.01454251
Kabuh          -0.6057007  0.4827838 -0.2111776  0.16683656 -0.10868498
Karangpakis    -0.1142516 -0.1572012  0.1489883  0.05510395  0.12157962
                       PC11
Bangsri        -8.414764e-17
Puri.Semanding -6.577041e-17
Gebang.Bunder   4.439046e-17
Mangunan        2.526445e-16
Kabuh          -3.199074e-17
Karangpakis     7.902885e-16
```

Opsi 2: menggunakan princomp ()

Fungsi princomp () juga dilengkapi dengan paket "stats" default, dan sangat mirip dengan prcomp sepupunya (). Apa yang saya tidak suka dari princomp () adalah bahwa kadang-kadang itu tidak akan menampilkan semua nilai untuk memuat, tetapi ini adalah detail kecil.

```
> #PCA with function princomp
> pca2 = princomp(enviro2, cor = TRUE)
> # sqrt of eigenvalues
> pca2$sdev
> pca2$sdev
   Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6
1.4888315 1.0918157 0.9513448 0.9388848 0.7400989 0.5069627

> # loadings
> unclass(pca2$loadings)
                 Comp.1     Comp.2       Comp.3      Comp.4      Comp.5
soil.moisture  0.39510582  0.1279316  0.004282522  0.77350269  0.40684465
pH            -0.46765370 -0.3411175  0.152054512 -0.15157832  0.77868517
temperatur     0.11551781 -0.5421301 -0.825701027  0.01198174 -0.01922247
```

```
humidity       0.09106389  0.7048254 -0.449453038 -0.38770373  0.35887760
elevation      0.57812587 -0.1701055  0.095873039 -0.20531673  0.31414227
Slope          0.51896521 -0.2182948  0.289619105 -0.43139877 -0.01669806
                    Comp.6
soil.moisture  0.2523442
pH             0.1117662
temperatur     0.1022514
humidity       0.1175297
elevation     -0.6977124
Slope          0.6424636
> # PCs (aka scores)
> head(pca2$scores)
    Comp.1     Comp.2    Comp.3     Comp.4      Comp.5     Comp.6
1 3.908165 -0.8167572 0.6894553 -0.2598157  0.21808339 0.29120043
2 2.173148  0.4710774 1.0659892 -0.5978624  0.13019778 0.46280661
3 2.384654 -0.2367685 0.3607130 -1.2135627 -0.32724146 1.21252394
4 2.156307  0.0542255 1.2069491 -1.2170805  0.02923532 0.07831919
5 1.231379  0.9217543 0.3502849 -0.6514198  0.41753433 0.65604772
6 1.883411 -0.6047958 0.2446145 -1.4801082 -0.65740125 0.08868541
```

Opsi 3: menggunakan PCA ()

Opsi yang sangat direkomendasikan, terutama jika Anda menginginkan hasil yang lebih rinci dan menilai alat, adalah fungsi PCA () dari paket "FactoMineR". Sejauh ini, ini merupakan fungsi PCA terbaik dalam R dan ia hadir dengan sejumlah parameter yang memungkinkan Anda mengubah analisis dengan cara yang sangat bagus.

```
> # PCA with function PCA
> library(FactoMineR)
> # apply PCA
> pca3 = PCA(enviro2, graph = FALSE)
> pca3$eig
       eigenvalue percentage of variance cumulative percentage of variance
comp 1  8.885592972             68.3507152                          68.35072
comp 2  1.987787494             15.2906730                          83.64139
comp 3  0.788580312              6.0660024                          89.70739
comp 4  0.553127212              4.2548247                          93.96222
comp 5  0.300298027              2.3099848                          96.27220
comp 6  0.215030608              1.6540816                          97.92628
comp 7  0.148592378              1.1430183                          99.06930
comp 8  0.074026203              0.5694323                          99.63873
comp 9  0.040722544              0.3132503                          99.95198
comp 10 0.006242249              0.0480173                         100.00000
> # correlations between variables and PCs
> pca3$var$coord
                Dim.1       Dim.2       Dim.3        Dim.4       Dim.5
elevation    0.9132833  0.17939448  0.15890262 -0.023960955  0.25983782
slope        0.8570158 -0.44161309  0.11478421  0.134022303 -0.16003826
temperature -0.8135908  0.47978330  0.19242796  0.080406145  0.04847895
C.organic    0.9198036 -0.17689784  0.12590694  0.045741551 -0.21756528
N.total      0.6519426 -0.24895742 -0.61606581  0.287330325  0.19908329
P2O5         0.8146260  0.44574713  0.22538739  0.228008378  0.13695712
pH.H2O       0.6266060 -0.72033657  0.13015466 -0.134893274  0.09785316
kadar.air    0.8784351  0.12613403 -0.18398144 -0.324783992  0.05075238
Kdd          0.9733329 -0.07632035  0.08000402  0.007683779  0.05231233
```

```
Cadd         0.7820436  0.56883557  0.08547368  0.092913288  0.03901170
Mgdd         0.7969338  0.41165274 -0.23588009  0.226432695 -0.27741610
Nadd         0.9295834 -0.10879890  0.31485115 -0.012787219 -0.01747719
KTK          0.7056459  0.43924865 -0.21833633 -0.455718222 -0.07361481
```

Opsi 4: menggunakan dudi.pca ()

Pilihan lain adalah menggunakan fungsi dudi.pca () dari paket "ade4" yang memiliki banyak metode lain serta beberapa grafik yang menarik.Opsi 4: menggunakan dudi.pca ()

Pilihan lain adalah menggunakan fungsi dudi.pca () dari paket "ade4" yang memiliki banyak metode lain serta beberapa grafik yang menarik.

```
> # PCA with function dudi.pca
> library(ade4)
Attaching package: 'ade4'
The following object is masked from 'package:FactoMineR':
    reconst

> # apply PCA
> pca4 = dudi.pca(enviro2, nf = 5, scannf = FALSE)
> # eigenvalues
> pca4$eig
[1] 8.885592972 1.987787494 0.788580312 0.553127212 0.300298027
0.215030608
 [7] 0.148592378 0.074026203 0.040722544 0.006242249

> # loadings
> pca4$c1
                  CS1         CS2         CS3         CS4         CS5
elevation   -0.3063813 -0.12724013 -0.17894027  0.03221748  0.47416133
slope       -0.2875052  0.31322539 -0.12925852 -0.18020406 -0.29204353
temperature  0.2729373 -0.34029859 -0.21669317 -0.10811270  0.08846613
C.organic   -0.3085687  0.12546933 -0.14178383 -0.06150329 -0.39702088
N.total     -0.2187087  0.17657943  0.69375183 -0.38633936  0.36329429
P2O5        -0.2732845 -0.31615756 -0.25380878 -0.30657610  0.24992423
pH.H2O      -0.2102090  0.51091716 -0.14656719  0.18137515  0.17856593
kadar.air   -0.2946907 -0.08946379  0.20718154  0.43669891  0.09261475
Kdd         -0.3265263  0.05413216 -0.09009254 -0.01033148  0.09546140
Cadd        -0.2623540 -0.40346119 -0.09625193 -0.12492959  0.07118994
Mgdd        -0.2673493 -0.29197525  0.26562462 -0.30445746 -0.50623878
Nadd        -0.3118496  0.07716841 -0.35455394  0.01719347 -0.03189300
KTK         -0.2367247 -0.31154835  0.24586859  0.61275079 -0.13433493
> # correlations between variables and PCs
> pca4$co
                Comp1       Comp2       Comp3        Comp4       Comp5
elevation   -0.9132833 -0.17939448 -0.15890262  0.023960955  0.25983782
slope       -0.8570158  0.44161309 -0.11478421 -0.134022303 -0.16003826
temperature  0.8135908 -0.47978330 -0.19242796 -0.080406145  0.04847895
C.organic   -0.9198036  0.17689784 -0.12590694 -0.045741551 -0.21756528
N.total     -0.6519426  0.24895742  0.61606581 -0.287330325  0.19908329
P2O5        -0.8146260 -0.44574713 -0.22538739 -0.228008378  0.13695712
pH.H2O      -0.6266060  0.72033657 -0.13015466  0.134893274  0.09785316
kadar.air   -0.8784351 -0.12613403  0.18398144  0.324783992  0.05075238
Kdd         -0.9733329  0.07632035 -0.08000402 -0.007683779  0.05231233
Cadd        -0.7820436 -0.56883557 -0.08547368 -0.092913288  0.03901170
```

```
Mgdd        -0.7969338 -0.41165274  0.23588009 -0.226432695 -0.27741610
Nadd        -0.9295834  0.10879890 -0.31485115  0.012787219 -0.01747719
KTK         -0.7056459 -0.43924865  0.21833633  0.455718222 -0.07361481
> # PCs
> head(pca4$li)
                   Axis1      Axis2      Axis3       Axis4       Axis5
Bangsri        -5.25577133 -0.2489482  1.3739440 -0.35559906 -0.60060595
Puri.Semanding -4.05049242  0.7207119  0.4554448  0.42995199 -0.23513255
Gebang.Bunder  -2.61956774  1.3968942 -0.1007409  0.01513205  1.09448014
Mangunan       -0.21754686  0.8157378 -1.3342038  1.00028908 -0.04584856
Kabuh           0.03976812  1.6374485 -0.9244216 -0.46744240  0.14902492
Karangpakis     0.32817792 -0.3283564 -1.1973541 -1.21157664 -0.81195516
```

Opsi 5: menggunakan acp ()

Kemungkinan kelima adalah fungsi acp () dari paket "amap".

```
> # PCA with function acp
> library(amap)
Attaching package: 'amap'
The following object is masked _by_ '.GlobalEnv':
> # apply PCA
> pca5 = acp(enviro2)
> # sqrt of eigenvalues
> pca5$sdev
     Comp 1       Comp 2       Comp 3       Comp 4       Comp 5       Comp 6
2.980871e+00 1.409889e+00 8.880204e-01 7.437252e-01 5.479946e-01 4.637139e-01
     Comp 7       Comp 8       Comp 9      Comp 10      Comp 11      Comp 12
3.854768e-01 2.720776e-01 2.017983e-01 7.900791e-02 7.193229e-16 7.193229e-16
     Comp 13
6.571536e-16
> # loadings
> pca5$loadings
               Comp 1      Comp 2      Comp 3      Comp 4      Comp 5
elevation   -0.3063813 -0.12724013  0.17894027  0.03221748 -0.47416133
slope       -0.2875052  0.31322539  0.12925852 -0.18020406  0.29204353
temperature  0.2729373 -0.34029859  0.21669317 -0.10811270 -0.08846613
C.organic   -0.3085687  0.12546933  0.14178383 -0.06150329  0.39702088
N.total     -0.2187087  0.17657943 -0.69375183 -0.38633936 -0.36329429
P2O5        -0.2732845 -0.31615756  0.25380878 -0.30657610 -0.24992423
pH.H2O      -0.2102090  0.51091716  0.14656719  0.18137515 -0.17856593
kadar.air   -0.2946907 -0.08946379 -0.20718154  0.43669891 -0.09261475
Kdd         -0.3265263  0.05413216  0.09009254 -0.01033148 -0.09546140
Cadd        -0.2623540 -0.40346119  0.09625193 -0.12492959 -0.07118994
Mgdd        -0.2673493 -0.29197525 -0.26562462 -0.30445746  0.50623878
Nadd        -0.3118496  0.07716841  0.35455394  0.01719347  0.03189300
KTK         -0.2367247 -0.31154835 -0.24586859  0.61275079  0.13433493
                Comp 6      Comp 7      Comp 8       Comp 9     Comp 10
elevation    0.14133664  0.31995222  0.27659107  0.610938674  0.05924415
slope        0.03258273 -0.26032390 -0.08091774  0.153878059  0.56674221
temperature -0.40825892  0.39582569 -0.19519324 -0.007011539  0.12024750
C.organic    0.19629981  0.56280347  0.12788152 -0.117852955  0.20432186
N.total     -0.20582930  0.11221924 -0.02712222 -0.046791502  0.14778441
P2O5         0.01695202  0.03770765 -0.17637454 -0.572209123  0.19611280
pH.H2O      -0.38703526  0.02990749 -0.38896585  0.026219998 -0.17764113
kadar.air    0.48501662  0.15512975 -0.45398272 -0.154624073 -0.15084491
Kdd         -0.04957972 -0.17315115  0.60305603 -0.348419946 -0.34737896
Cadd         0.08036484 -0.51979166 -0.21568111  0.241732639  0.04782007
Mgdd        -0.15560580  0.12275696 -0.11376119  0.221015310 -0.43379284
Nadd        -0.32513085 -0.01631682 -0.12379491  0.049718933 -0.26297053
```

```
KTK         -0.45521751 -0.04428777  0.18653573 -0.061459867  0.35240156
                  Comp 11       Comp 12      Comp 13
elevation     0.117172342  0.117172342 -0.09033071
slope        -0.259383942 -0.259383942 -0.02030271
temperature  -0.509547726 -0.509547726 -0.07170962
C.organic     0.009091081  0.009091081 -0.02621458
N.total      -0.138034688 -0.138034688  0.21025455
P2O5          0.357415813  0.357415813 -0.11718442
pH.H2O        0.181659581  0.181659581 -0.50052933
kadar.air    -0.309307209 -0.309307209  0.09203889
Kdd          -0.354268486 -0.354268486 -0.17731166
Cadd         -0.118358507 -0.118358507 -0.10888160
Mgdd          0.138177471  0.138177471 -0.22343078
Nadd         -0.123417081 -0.123417081  0.75837778
KTK           0.129041931  0.129041931 -0.01749191
```

```
> # scores
> head(pca5$scores)
                   Comp 1     Comp 2      Comp 3      Comp 4      Comp 5
Bangsri        -5.01118134 -0.2373628 -1.31000424 -0.33905040  0.57265530
Puri.Semanding -3.86199299  0.6871719 -0.43424954  0.40994314  0.22419009
Gebang.Bunder  -2.49765984  1.3318864  0.09605264  0.01442784 -1.04354587
Mangunan       -0.20742279  0.7777755  1.27211339  0.95373821  0.04371489
Kabuh           0.03791742  1.5612459  0.88140143 -0.44568884 -0.14208968
Karangpakis     0.31290537 -0.3130755  1.14163231 -1.15519300  0.77416887
                   Comp 6     Comp 7     Comp 8      Comp 9     Comp 10
Bangsri        -0.3389827  0.1393525 -0.1955473  0.14583630 -0.08120780
Puri.Semanding  0.6322102 -0.3472085  0.4200570 -0.10262349  0.08480774
Gebang.Bunder   0.1053847  0.4676475 -0.3122381 -0.18468206  0.03352180
Mangunan        0.2217403 -0.1359093 -0.1817175  0.41581331 -0.01454251
Kabuh          -0.6057007 -0.4827838  0.2111776 -0.16683656 -0.10868498
Karangpakis    -0.1142516  0.1572012 -0.1489883 -0.05510395  0.12157962
                     Comp 11       Comp 12       Comp 13
Bangsri         1.393132e-15  1.393132e-15  5.464633e-16
Puri.Semanding -8.241902e-16 -8.241902e-16  3.350286e-16
Gebang.Bunder   3.367642e-16  3.367642e-16 -8.691793e-16
Mangunan        1.593023e-15  1.593023e-15  6.285196e-18
Kabuh          -2.071978e-16 -2.071978e-16  3.068351e-16
Karangpakis    -3.679579e-16 -3.679579e-16 -7.541549e-16
```

c. Plot PCA

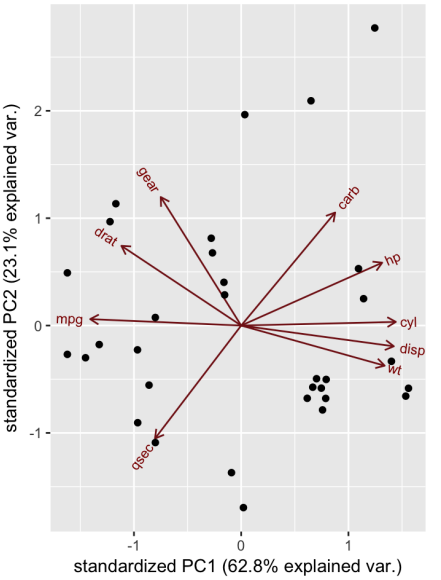
Sekarang saatnya merencanakan PCA, akan membuat biplot, yang mencakup posisi setiap sampel dalam kaitannya dengan PC1 dan PC2 dan juga akan menunjukkan bagaimana variabel awal memetakannya. Penggunaan paket ggbiplot, menawarkan fungsi yang ramah pengguna dan cantik untuk memplot biplot. Biplot adalah jenis plot yang memungkinkan untuk memvisualisasikan bagaimana sampel berhubungan satu sama lain di PCA (sampel mana yang serupa dan mana yang berbeda) dan secara bersamaan akan mengungkapkan bagaimana setiap variabel berkontribusi pada setiap komponen utama.

Sebelum dimemulai, jangan lupa untuk menginstal **ggbiplot** terlebih dahulu!

```
> library(devtools)
Loading required package: usethis
Attaching package: 'devtools'
The following object is masked from 'package:permute':
Check
```

Selanjutnya, dapat memanggil ggbiplot di PCA Anda:

```
> library(ggbiplot)
> ggbiplot(mtcars.pca)
```

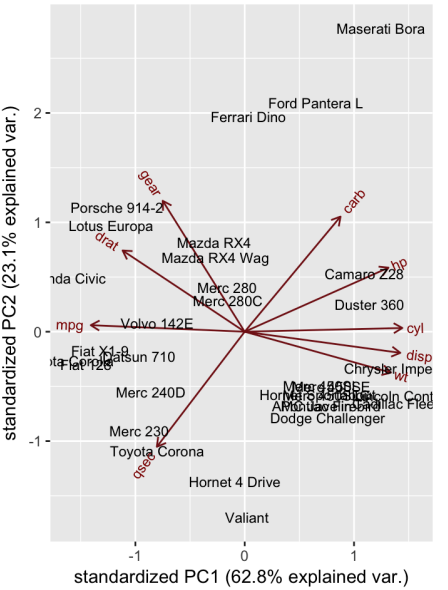


Gambar 178. Grafik ggbiplot PCA

Sumbu terlihat seperti anak panah yang berasal dari titik tengah. Di sini, melihat bahwa variabel `hp`, `cyl`, dan `disp` semuanya berkontribusi pada PC1, dengan nilai yang lebih tinggi pada variabel tersebut memindahkan sampel ke kanan pada plot ini. Ini memungkinkan melihat bagaimana titik data berhubungan dengan sumbu, tetapi ini tidak terlalu informatif tanpa mengetahui titik mana yang sesuai dengan sampel mana (car).

Selanjutnya berikan argumen ke ggbiplot: mari kita berikan nama belakang mtcars sebagai label. Ini akan memberi nama setiap titik dengan nama mobil yang dimaksud:

```
> ggbiplot(mtcars.pca, labels=rownames(mtcars))
```



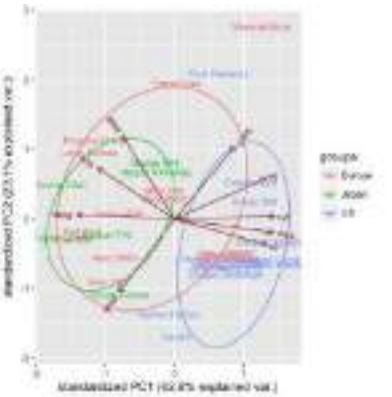
Gambar 179. Grafik ggbiplot PCA terstandar

Sekarang dapat melihat mobil mana yang mirip satu sama lain. Misalnya, Maserati Bora, Ferrari Dino, dan Ford Pantera L semuanya berkumpul di atas. Ini masuk akal, karena semuanya adalah mobil sport.

10.5 Menafsirkan hasil

Mungkin jika melihat asal muasal masing-masing mobil tersebut. Anda akan memasukkannya ke dalam salah satu dari tiga kategori (kategori?), Masing-masing untuk mobil AS, Jepang, dan Eropa. Anda membuat daftar untuk info ini, lalu meneruskannya ke argumen group`ggbiplot`. Anda juga akan menyetel argumen elipse menjadi TRUE, yang akan menggambar elips di sekitar setiap grup.

```
>mtcars.country <- c(rep("Japan", 3), rep("US",4), rep("Europe",
7),rep("US",3), "Europe", rep("Japan", 3), rep("US",4),
rep("Europe", 3), "US", rep("Europe", 3))
> ggbiplot(mtcars.pca,ellipse=TRUE,  labels=rownames(mtcars),
groups=mtcars.country)
```



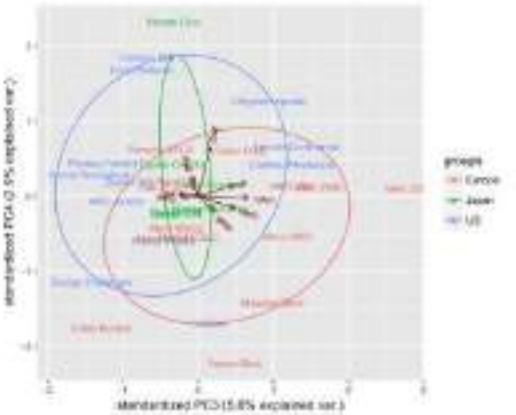
Gambar 180. Grafik ggbiplot dengan pengelompokkan

Sekarang melihat sesuatu yang menarik: mobil-mobil Amerika membentuk kelompok yang berbeda di sebelah kanan. Melihat sumbu, melihat bahwa mobil-mobil Amerika dicirikan oleh nilai tinggi untuk `cyl, disp`, dan `wt`. Mobil Jepang, sebaliknya, memiliki karakter mpg yang tinggi. Mobil-mobil Eropa agak berada di tengah dan tidak terlalu padat daripada kelompok mana pun.

Tentu saja, memiliki banyak komponen utama yang tersedia, yang masing-masing dipetakan secara berbeda ke variabel aslinya, juga dapat meminta ggbiplot untuk memplot komponen lain ini, dengan menggunakan argumen options.

Mari kita lihat PC3 dan PC4:

```
>
ggbiplot(mtcars.pca,ellipse=TRUE,choices=c(3,4),labels=rownames(
mtcars), groups=mtcars.country)
```



Gambar 181. Grafik ggbiplot dengan pengelompokan berbentuk ellips

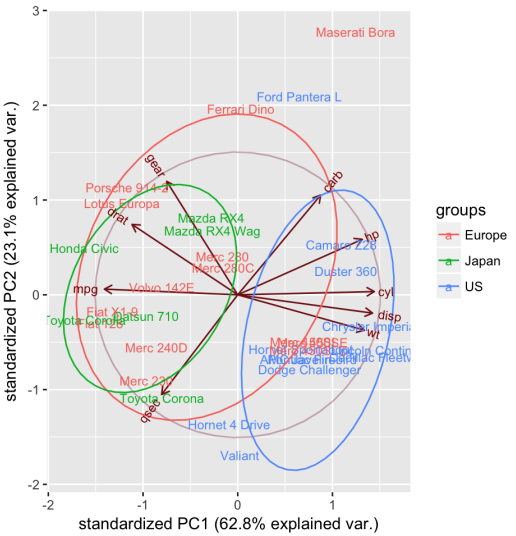
Sepertinya tidak melihat banyak hal di sini, tetapi ini tidak terlalu mengejutkan. PC3 dan PC4 menjelaskan persentase yang sangat kecil dari total variasi, jadi akan mengejutkan jika menemukan bahwa mereka sangat informatif dan memisahkan kelompok atau mengungkapkan pola yang terlihat.

Mari luangkan waktu sejenak untuk merekap: setelah melakukan PCA menggunakan set data mtcars, kita dapat melihat pemisahan yang jelas antara mobil Amerika dan Jepang di sepanjang komponen utama yang berkorelasi erat dengan `cyl, disp, wt`, dan `mpg.` Ini memberi kami beberapa petunjuk untuk analisis di masa mendatang; jika dicoba membuat model klasifikasi untuk mengidentifikasi asal mobil, variabel ini mungkin berguna.

10.6 Parameter grafis dengan ggbiplot

Ada juga beberapa variabel lain yang dapat mainkan untuk mengubah biplot. Disini dapat ditambahkan lingkaran ke tengah kumpulan data (*circle argument*):

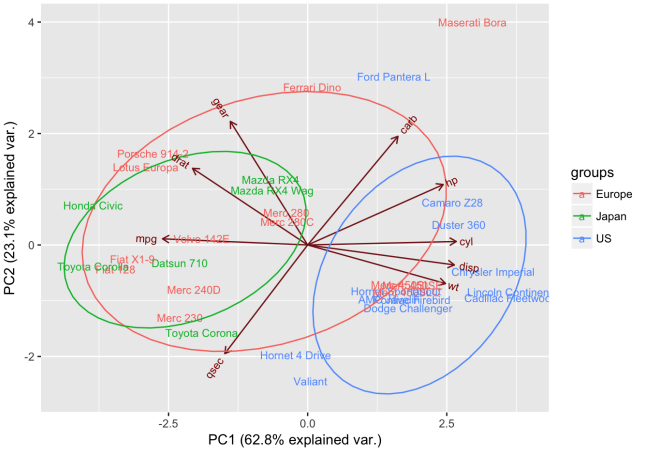
```
> ggbiplot(mtcars.pca,ellipse=TRUE,circle=TRUE,
labels=rownames(mtcars), groups=mtcars.country)
```



Gambar 182. Grafik ggbiplot dengan kategori kelompok

Juga dapat menskalakan sampel `(obs.scale)` dan variabel `(var.scale):`

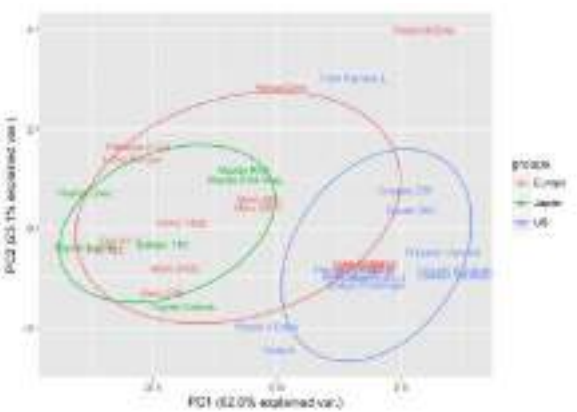
```
>ggbiplot(mtcars.pca,ellipse=TRUE,obs.scale = 1, var.scale = 1,
labels=rownames(mtcars), groups=mtcars.country)
```



Gambar 183. Grafik ggbiplot dengan penskalaan

Selain itu juga dapat menghapus panah sama sekali, menggunakan: `var.axes.`

```
> ggbiplot(mtcars.pca,ellipse=TRUE,obs.scale = 1, var.scale =
1,var.axes=FALSE,   labels=rownames(mtcars),
groups=mtcars.country)
```



Gambar 184. Grafik ggbiplot tanpa skala

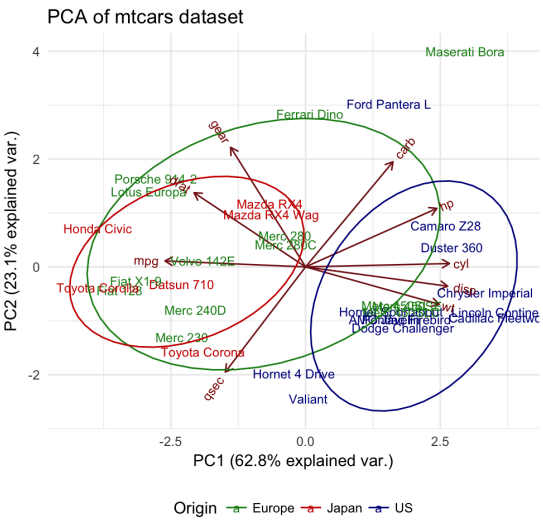
10.7 Sesuaikan ggbiplot

Karena ggbiplot didasarkan pada fungsi ggplot, dapat pula digunakan set parameter grafis yang sama untuk mengubah biplot Seperti yang dilakukan untuk ggplot mana pun. Di sini, bisa dilakukan:

1. Tentukan warna yang akan digunakan untuk grup dengan scale_colour_manual ()

2. Tambahkan judul dengan ggtitle ()
3. Tentukan tema minimal ()
4. Pindahkan legenda dengan theme ()

```
> ggbiplot(mtcars.pca,ellipse=TRUE,obs.scale = 1, var.scale = 1,
labels=rownames(mtcars), groups=mtcars.country) +
+    scale_colour_manual(name="Origin", values= c("forest green",
"red3", "dark blue"))+
+   ggtitle("PCA of mtcars dataset")+
+   theme_minimal()+
+   theme(legend.position = "bottom")
```



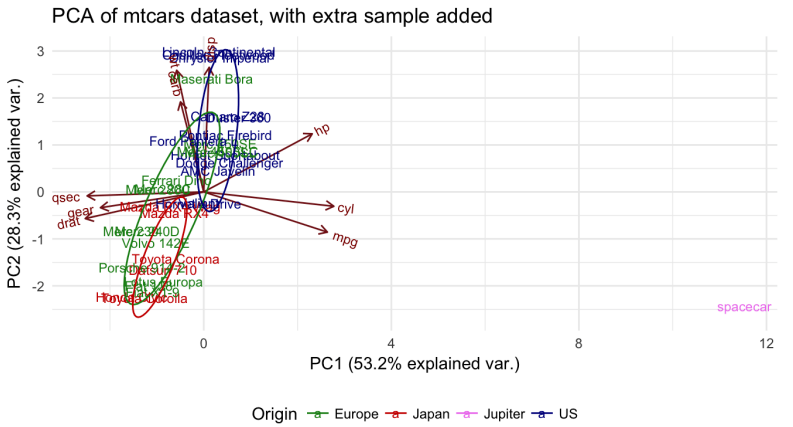
Gambar 185. Grafik ggbiplot dengan penyesuaian warna

10.8 Menambahkan sampel baru

Untuk menambahkan sampel baru ke kumpulan data, yaitu mobil yang sangat istimewa, dengan statistik tidak seperti yang lain. Ini sangat bertenaga, memiliki mesin 60 silinder, penghematan bahan bakar yang luar biasa, tanpa gigi dan sangat ringan. Ini adalah "mobil antariksa", dari Jupiter. Maka perlu menambahkannya ke mtcars, membuat mtcarsplus, lalu ulangi analisis. Hal ini mungkin berharap bisa melihat mobil daerah mana yang paling disukai.

```
> spacecar <- c(1000,60,50,500,0,0.5,2.5,0,1,0,0)
> mtcarsplus <- rbind(mtcars, spacecar)
> mtcars.countryplus <- c(mtcars.country, "Jupiter")
> mtcarsplus.pca <- prcomp(mtcarsplus[,c(1:7,10,11)], center =
TRUE,scale. = TRUE)
> ggbiplot(mtcarsplus.pca, obs.scale = 1, var.scale = 1, ellipse
= TRUE, circle = FALSE, var.axes=TRUE,
```

```
labels=c(rownames(mtcars), "spacecar"),
groups=mtcars.countryplus)+
+   scale_colour_manual(name="Origin", values= c("forest green",
"red3", "violet", "dark blue"))+
+   ggtitle("PCA of mtcars dataset, with extra sample added")+
+   theme_minimal()+
+   theme(legend.position = "bottom")
```



Gambar 186. Grafik ggbiplot dengan taambahan sampel baru

Tapi itu asumsi yang naif! Bentuk PCA telah berubah secara drastis, dengan penambahan sampel ini. Ketika mempertimbangkan hasil ini dengan sedikit lebih detail, itu sebenarnya masuk akal. Dalam kumpulan data asli, memiliki korelasi yang kuat antara variabel tertentu (misalnya, `cyl dan mpg`), yang berkontribusi pada PC1, memisahkan grup satu sama lain di sepanjang sumbu ini. Namun, saat menjalankan PCA dengan sampel ekstra, korelasi yang sama tidak ada, yang membengkokkan seluruh kumpulan data. Dalam kasus ini, pengaruhnya sangat kuat karena sampel ekstra merupakan pencilan ekstrem dalam banyak hal. Jika ingin melihat bagaimana sampel baru dibandingkan dengan grup yang dihasilkan oleh PCA awal, perlu memproyeksikannya ke PCA tersebut.

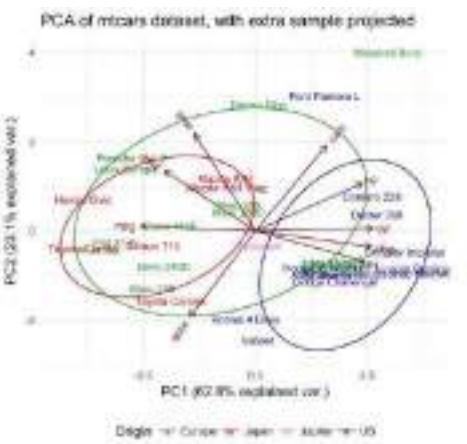
10.9 Proyeksikan sampel baru ke PCA asli

Artinya, komponen utama ditentukan tanpa kaitannya dengan sampel mobil ruang angkasa, lalu menghitung di mana mobil ruang

angkasa ditempatkan dalam kaitannya dengan sampel lain dengan menerapkan transformasi yang telah dihasilkan PCA., dapat menganggap ini sebagai, alih-alih mendapatkan mean dari semua sampel dan membiarkan spacecar membelokkan mean ini, mendapatkan mean dari sisa sampel dan melihat spacecar dalam hubungannya dengan ini.

Artinya, cukup menskalakan nilai untuk spacecar dalam hubungannya dengan pusat PCA (`mtcars.pca $ center`). Kemudian menerapkan rotasi matriks PCA ke sampel spacecar. Kemudian dapat rbind () nilai yang diproyeksikan untuk spacecar ke sisa matriks $ x pca dan meneruskannya ke `ggbiplot` seperti sebelumnya:

```
> s.sc <- scale(t(spacecar[c(1:7,10,11)]), center=
mtcars.pca$center)
> s.pred <- s.sc %*% mtcars.pca$rotation
> mtcars.plusproj.pca <- mtcars.pca
> mtcars.plusproj.pca$x <- rbind(mtcars.plusproj.pca$x, s.pred)
> ggbiplot(mtcars.plusproj.pca, obs.scale = 1, var.scale = 1,
ellipse = TRUE, circle = FALSE, var.axes=TRUE,
labels=c(rownames(mtcars), "spacecar"),
groups=mtcars.countryplus)+
+   scale_colour_manual(name="Origin", values= c("forest green",
"red3", "violet", "dark blue"))+
+   ggtitle("PCA of mtcars dataset, with extra sample
projected")+
+   theme_minimal()+
+   theme(legend.position = "bottom")
```



Gambar 187. Proyek grafik ggbiplot dengan sampel baru

Hasil ini sangat berbeda. Perhatikan bahwa semua sampel lainnya kembali ke posisi awal mereka, sementara spacecar ditempatkan agak

dekat tengah. Sampel ekstra tidak lagi mendistorsi keseluruhan distribusi, tetapi tidak dapat ditetapkan ke grup tertentu. Tapi mana yang lebih baik, proyeksi atau penghitungan ulang PCA?

Itu agak tergantung pada pertanyaan yang coba jawab; perhitungan ulang menunjukkan bahwa spacecar adalah pencilan, proyeksi memberi tahu bahwa tidak dapat menempatkannya di salah satu grup yang ada. Melakukan kedua pendekatan ini sering kali berguna saat melakukan analisis data eksplorasi dengan PCA. Jenis analisis eksplorasi ini sering kali menjadi titik awal yang baik sebelum mendalami kumpulan data lebih dalam. PCA memberi tahu variabel mana yang memisahkan mobil Amerika dari yang lain dan mobil angkasa itu merupakan pencilan dalam kumpulan data kami. Langkah selanjutnya yang mungkin adalah melihat apakah hubungan ini berlaku untuk mobil lain atau untuk melihat bagaimana mobil dikelompokkan berdasarkan marque atau tipe (mobil sport, 4WD, dll).

BAB XI
REGRESI NONPARAMETRIK

11.1 Kernel Smoother

Penghalus kernel (*kernel smoother*) adalah teknik statistic untuk estimasi sebuah nilai riil $f(X)\ (X \in \mathbb{R}^p)$ oleh penggunaan pengamatan yang kasar, ketika model tidak parametric untuk fungsi ini diketahui. Fungsi estimasi ini adalah penghalus, dan tingkat penghalus dibentuk oleh satu parameter.

Sedikit atau tidak ada latihan yang disyaratkan untuk operasi penghalus kernel. Teknik ini hampir mendekati pada dimensi rendah (p <3) untuk maksud visual data. Kenyataannya, penghalus kernel menghadirkan sekumpulan titik-titik data tak beraturan sebagai sebuah permukaan dan garis penghalus.

1. Definisi

Ambillah $K_{h_\lambda}(X_0, X)$sebagai kernel yang didefinisikan oleh:

$$K_{h_\lambda}(X_0, X) = D\left(\frac{\|X - X_0\|}{h_\lambda(X_0)}\right)$$

Dimana:

a. $X, X_0 \in \mathbb{R}^p$
b. $\|\cdot\|$adalah Euclidean normal
c. $h_\lambda(X_0)$ adalah sebuah parameter (radius kernel)
d. $D(t)$ secara tipikal adalah fungsi nilai real positif, dimana nilai ini menurun saat bertambahnya jarak antara x dan x_o

Kernel popular digunakan untuk penghalus termasuk:

a. Epanechnikov
b. Tri-cube
c. Gaussian

Ambillah $\hat{Y}(X) : \mathbb{R}^p \rightarrow \mathbb{R}$sebagai fungsi kontinu pada x. untuk tiap-tiap $X_0 \in \mathbb{R}^p$, rata-rata terbobot kernel Nadaraya-Watson (estimasi penghalus Y(X)) didefinisikan oleh:

$$\hat{Y}(X_0) = \frac{\sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i) Y(X_i)}{\sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i)}$$

Dimana:

a. N adalah jumlah titik pengamatan
b. $Y(X_i)$ adalah pengamatan pada titik X_i

Pada bagian ini, akan dijelaskan beberapa kasus utama penghalus kernel.

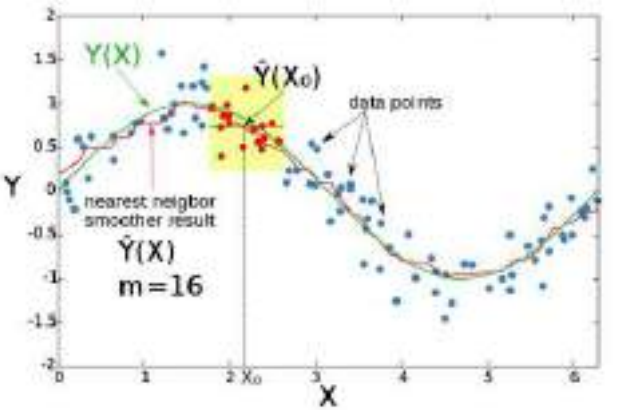
b. Nearest Neighbor Smoother

Gagasan pada penghalus Nearest Neighbor adalah sebagai berikut: Pada tiap-tiap titik Xo, ambil m nearest neighbors dan estimasi nilai pada Y(Xo) oleh rata-rata nilai pada neighbors.

Secara formal, $h_m(X_0) = \left\| X_0 - X_{[m]} \right\|$, dimana $X_{[m]}$ adalah nilai m tertutup pada Xo neighbors, dan

$$D(t) = \begin{cases} 1/2 & \text{if } |t| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Contoh:



Gambar 188. Grafik penghalus Nearest-neigbors
Sumber: R. Tibshirani & L. Wasserman, 2015.

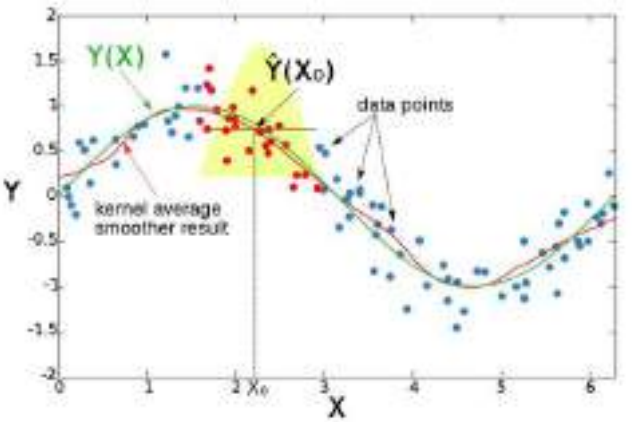
Pada contoh ini, X adalah dimensi satu, untuk setiap Xo, $\hat{Y}(X_0)$ adalah nilai rata-rata 16 tertutup pada titik Xo (ditandai warna merah). Hasilnya tidak cukup smooth.

c. Penghalus Rata-rata Kernel

Gagasan penghalus rata-rata kernel adalah sebagai berikut. Pada setiap titik data Xo, pilih ukuran jarak konstans λ (radius kernel, atau window width untuk dimensi p = 1), dan hitung rata-rata terbobot pada semua titik data yang tertutup padaλ untuk Xo (tertutup untuk titik Xo mendapatkan bobot tertinggi).

Formalnya, $h_\lambda(X_0) = \lambda$ = konstan dan $D(t)$adalah satu pada kernel popular.

Contoh:



Gambar 189. Grafik Penghalus rata-rata Kernel
Sumber: R. Tibshirani & L. Wasserman, 2015.

Pada tiap-tiap X_0 lebar jendela adalah constant, dan pembobot tiap-tiap titik dalam jendela adalah secara skematis ditandai oleh gambar kuning dalam grafik. Ini dapat terlihat bahwa estimasi adalah smooth, tetapi titik-titik batasnya bias. Alasan untuk itu adalah jumlah tidak sama pada titik-titik (dari kanan ke kiri pada Xo) dalam jendela, ketika Xo adalah cukup tertutup pada batas.

d. Regresi Linier Lokal

Diasumsikan bahwa dibawah fungsi Y(X) adalah Locally constant, maka kita bisa menggunakan rata-rata terbobot untuk estimasi. Ide ini pada regresi linier local adalah untuk menffitkan locally sebagai garis lurus (atau hyperplane untuk dimensi tinggi), dan tidak konstan (garis horizontal). Setelah fitting garis, estimasi $\hat{Y}(X_0)$ disediakan oleh nilai pada garis ini pada titik Xo. Dengan pengulangan prosedur ini untuk setiap Xo, maka mendapatkan fungsi estimasi $\hat{Y}(X)$. Seperti pembahasan sebelumnya, lebar jendela adalah constant $h_\lambda(X_0) = \lambda$ = constant. Umumnya, regresi linier local adalah dihitung oleh solusi pada permasalahan rata-rata kuadrat terbobot.

Untuk satu dimensi (p = 1):

$$\min_{\alpha(X_0),\beta(X_0)} \sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i)\,(Y(X_i) - \alpha(X_0) - \beta(X_0)X_i)^2$$

$$\Downarrow$$

$$\hat{Y}(X_0) = \alpha(X_0) + \beta(X_0)X_0$$

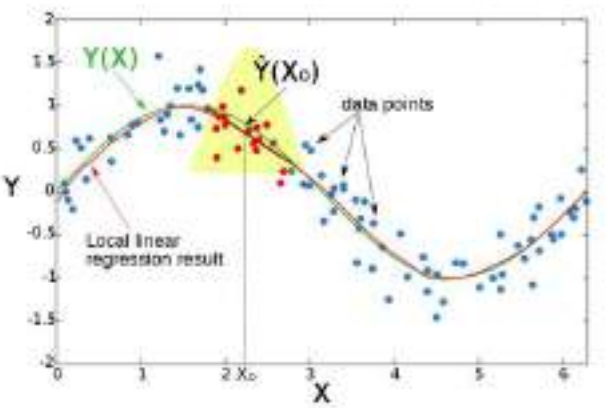
Solusi bentuk tertutup diberikan oleh:

$$\hat{Y}(X_0) = (1, X_0)\left(B^T W(X_0) B\right)^{-1} B^T W(X_0) y$$

Dimana:

a. $y = (Y(X_1), \ldots, Y(X_N))^T$
b. $W(X_0) = \text{diag}\,(K_{h_\lambda}(X_0, X_i))_{N\times N}$
c. $B^T = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ X_1 & X_2 & \ldots & X_N \end{pmatrix}$

Contoh:



Gambar 190. Grafik Regresi linier lokal
Sumber: R. Tibshirani & L. Wasserman, 2015.

Hasil fungsi adalah smooth, dan masalah dengan titik batas bias telah diatasi.

e. Regresi Polinomial Lokal

Selain menfitkan fungsi linier local, fungsi polynomial bisa fit.

Untuk p = 1, regresi harus diminimalisir.

$$\min_{\alpha(X_0),\beta_j(X_0),j=1,\ldots,d} \sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i)\left(Y(X_i) - \alpha(X_0) - \sum_{j=1}^{d} \beta_j(X_0) X_i^j\right)^2$$

Dengan:

$$\hat{Y}(X_0) = \alpha(X_0) + \sum_{j=1}^{d} \beta_j(X_0) X_0^j$$

Dalam kasus umum (p > 1), fungsi harus diminimalisir.

$$\hat{\beta}(X_0) = \underset{\beta(X_0)}{\arg\min} \sum_{i=1}^{N} K_{h_\lambda}(X_0, X_i) \left(Y(X_i) - b(X_i)^T \beta(X_0)\right)^2$$

$$b(X) = (1, \quad X_1, \quad X_2, \ldots \quad X_1^2, \quad X_2^2, \ldots \quad X_1 X_2 \ \ldots)$$

$$\hat{Y}(X_0) = b(X_0)^T \hat{\beta}(X_0)$$

f. **Regresi Kernel**

Regresi kernel adalah teknik nonparametrik dalam statistic untuk estimasi ekspektasi kondisional pada variabel acak. Tujuannya adalah untuk mendapatkan hubungan yang tidak linier antara pasangan pada variabel acak X dan Y.

Dalam beberapa regresi nonparametrik, ekspektasi kondisional pada variabel Y relative pada variabel X yang bisa ditulis:

$$Y = m(X),$$

Dimana m adalah fungsi yang tidak diketahui.

11.2 Kode R dalam Regresi Kernel

Kode R dalam regersi kernel bisa melalui input data simulasi dan data penelitian. Untuk data penelitian digunakan data pengukuran biomassa pohon kelengkeng dan mangga yang tumbuh dilahan kritis. Data penelitian di ambil pada 4 lokasi di Kabupaten Jombang

a. Kode R dengan data Simulasi

Sebelum kita melakukan analisis data dengan regresi kernel kita persiapkan data penelitiannya. Data x merupakan data Area dan y adalah data RiverFlow, seperti ditunjukkan pada tabel berikut:

Tabel 5. Data Penelitian dari variabel Area(x) dan RiverFlow (y)

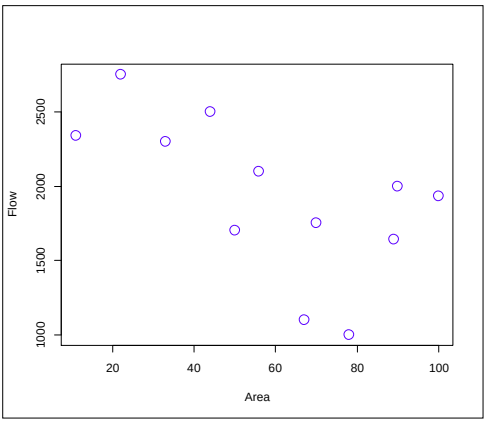
X	11	22	33	44	50	56	67	70	78	89	90	100
Y	2337	2750	2301	2500	1700	2100	1100	1750	1000	1642	2000	1932

Selanjutnya masukkan data tersebut dalam R, seperti berikut ini;

```
data <- data.frame(Area =
c(11,22,33,44,50,56,67,70,78,89,90,100),        RiverFlow =
c(2337,2750,2301,2500,1700,2100,1100,1750,1000,1642, 2000,1932))
x <- data$Area
```

```
y <- data$RiverFlow
Kemudian buat coding untuk fungsi kernel Gaussian:
#function to calculate Gaussian kernel
gausinKernel <- function(x,b){
+   K <- (1/((sqrt(2*pi))))*exp(-0.5 *(x/b)^2)
+   return(K)
+ }
b <- 10 #bandwidth
kdeEstimateyX <- seq(5,110,1)
ykernel <- NULL
for(xesti in kdeEstimateyX){
+   xx <-  xesti - x
+   K <-gausinKernel(xx,b)
+   Ksum <- sum(K)
+   weight <- K/Ksum
+   yk <- sum(weight*y)
+   xkyk <- c(xesti,yk)
+   ykernel <- rbind(ykernel,xkyk)
+ }
plot(x,y,xlab = "Area", ylab = "Flow", col = 'blue', cex = 2)
```

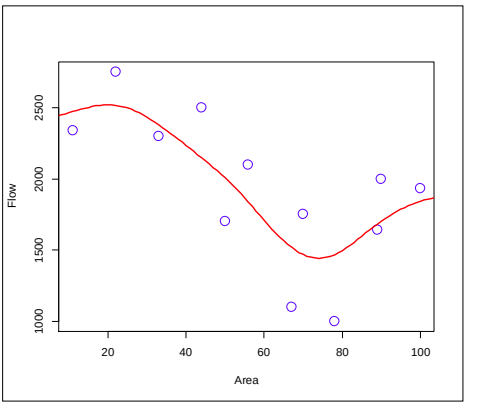
Diperoleh output scaterplot seperti gambar berikut:



Gambar 191. Grafik Scatter Plot antara x dan y

Selanjutnya ditambahkan grafik garis dari fungsi kernelnya, dengan kode R seperti berikut:

```
lines(ykernel[,1],ykernel[,2], col = 'red', lwd = 2)
```



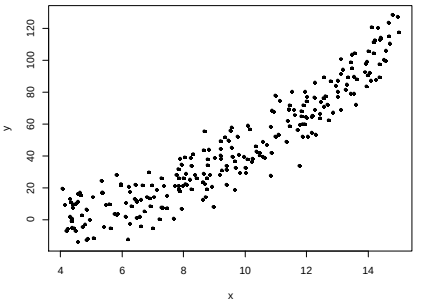
Gambar 192. Fungsi Kernel dalam bentuk grafik garis

b. Kode R dengan Data Penelitian

Sebagai comtoh dari fungsi regresi, berikut coding dalam R. Console, sebelumnya dibangun algoritma kode yang diinputkan pada jendela R. Console. Coding yang dibangun menggunakan kombinasi persamaan alometrik pohon dengan kerapatan jenis kayu (kelengkeng(ρ) = 0,91).

```
> x = runif(250,min=4.0, max=15.0)
> y = 0.1001*x^2.62 + rnorm(250, sd=10.5)
> plot(x,y, pch=20)
```

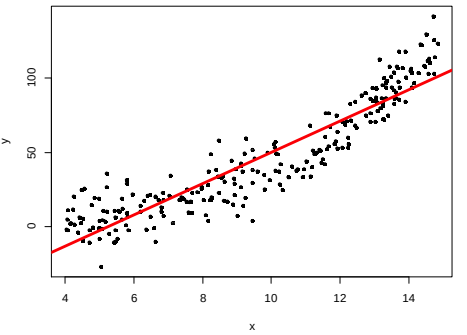
Output tampilan scatter plotnya adalah sebagai berikut:



Gambar 193. Scatter plot pola hubungan dari variabel X dan Y

Grafik scatter plot tersebut menunjukkan secara jelas pola antara X dan Y. Apa yang terjadi jika diterapkan pada ordinasi regresi linier?

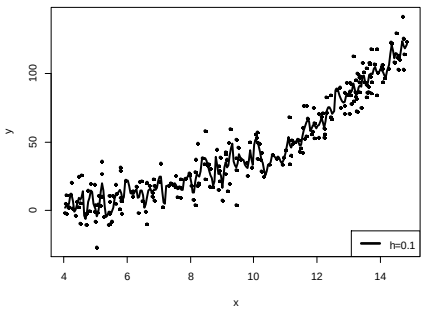
```
> fit = lm(y~x)
> plot(x,y, pch=20)
> abline(fit, lwd=4, col="red")
```



Gambar 194. Scatter Plot dengan garis linier

Regresi linier yang bagus (garis merah) tidak mengatasi struktur hubungan yang dibangun. Hal ini faktanya benar-benar regresi linier hanya mampu mendeteksi hubungan linier antara dua variabel. Dalam dataset ini, kita dapat mengamati secara jelas hubungan antara X dan Y tetapi hubungan tersebut tidak linier, menhasilkan luaran yang buruk pada regresi linier. Apa yang dapat kita lakukan dalam kasus ini? Maka perlu pendekatan yang baik yang disebut dengan regresi kernel, yang merupakan metode regresi nonparametrik yang memperkenankan kita untuk mengatasi strustuk yang underlying.

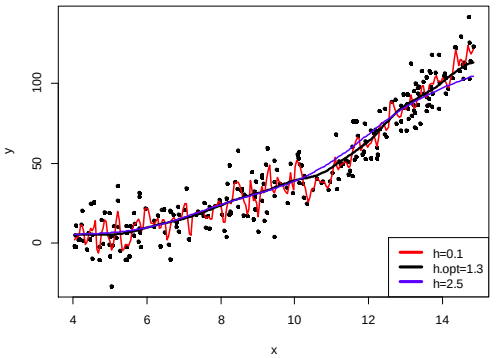
```
> Kreg = ksmooth(x=x,y=y,kernel = "normal",bandwidth = 0.1)
> plot(x,y, pch=20)
> lines(Kreg, lwd=3.0, col="black")
> legend("bottomright", c("h=0.1"), lwd=4, col=c("black"))
```



Gambar 195. Grafik fungsi kernel dengan ukuran Bandwidth h = 0,1

```
> Kreg1 = ksmooth(x=x,y=y,kernel = "normal",bandwidth = 0.1)
> Kreg2 = ksmooth(x=x,y=y,kernel = "normal",bandwidth = 1.3)
> Kreg3 = ksmooth(x=x,y=y,kernel = "normal",bandwidth = 2.5)
```

```
> plot(x,y,pch=20)
> lines(Kreg1, lwd=2.5, col="red")
> lines(Kreg2, lwd=3.5, col="black")
> lines(Kreg3, lwd=2.5, col="blue")
>  legend("bottomright",  c("h=0.1","h.opt=1.3","h=2.5"),  lwd=4,
col=c("red","black","blue"))
```



Gambar 196. Grafik fungsi kernel dengan berbagai ukuran bandwidth

```
> n = length(x)
> # n: sample size
> CV_err = rep(NA, n)
> for(i in 1:n){
+ x_val = x[i]
+ y_val = y[i]
+ # validation set
+ x_tr = x[-i]
+ y_tr = y[-i]
+ # training set
+     y_val_predict     =     ksmooth(x=x_tr,y=y_tr,kernel     =
"normal",bandwidth=0.5,
+ x.points = x_val)
+ CV_err[i] = (y_val - y_val_predict$y)^2
+ # we measure the error in terms of difference sqaure
+ }
> mean(CV_err)
[1] 136.2747
```

11.3 Beberapa Heuristik tentang Localy Regression dan Kernel Smoothing

Dalam model linier standar, diasumsi bahwa :

$$\mathrm{E}\left(Y \middle| X = x\right) = \beta_0 + \beta_1 x$$

Alternatif bisa dipertimbangkan, ketika asumsi kinier begitu kuat.

a. Regresi Polinomial

Diluar kealamiahan mungkin bisa diasumsikan beberapa fungsi polynomial:

$$\mathrm{E}\left(Y \middle| X = x\right) = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_k x^k$$

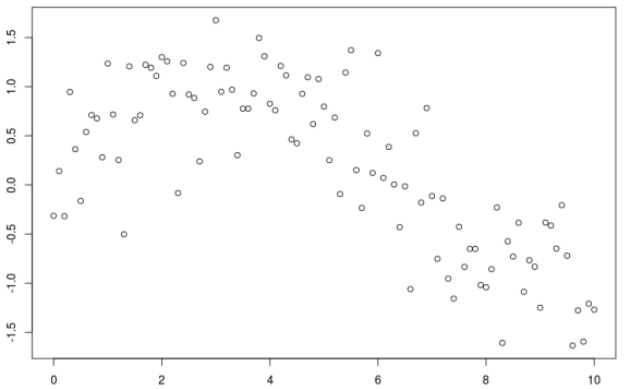
Sekali lagi dalam pendekatan model linier standar (dengan kondisi distribusi normal menggunakan terminology GLM), parameter $\beta = \beta_0, \beta_1, \ldots, \beta_k$.

Bisa diperoleh dengan menggunakan least squares, dimana regresi pada Y dalam $X = \left(1, X, X^2, \ldots, X^k\right)$ dipertimbangkan.

Sama jika model polynomial ini tidak riil, maka masih perlu dijadikan pendekatan yang bagus untuk $\mathrm{E}\left(Y \middle| X = x\right) = h(x)$. Aktualnya, dari teorema Stone-Weierstrass, jika h(.) adalah kontinu dalam beberapa interval, maka adalah pendekatan uniform pada h(.) oleh fungsi polynomial.

Hanya sebagai ilustrasi, pertimbangkan dataset (simulasi) berikut:

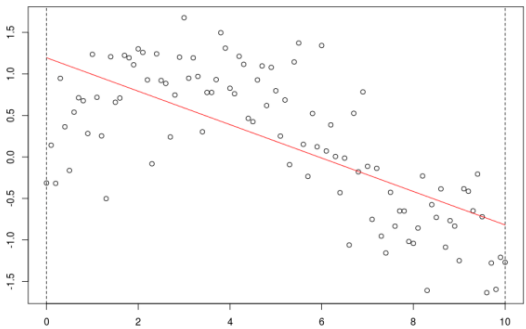
```
> set.seed(1)
> xr = seq(0,n,by=.1)
> yr = sin(xr/2)+rnorm(length(xr))/2
> db = data.frame(x=xr,y=yr)
> plot(db)
```



Gambar 197. Scatter Plot

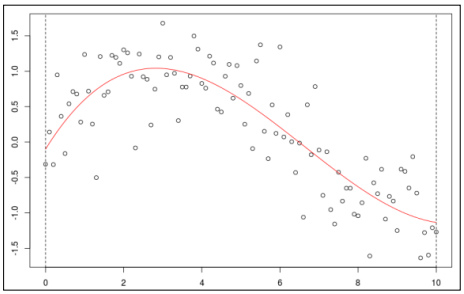
Dengan garis regresi standar:

```
> reg = lm(y ~ x,data=db)
> abline(reg,col="red")
```



Gambar 198. Fungsi linier

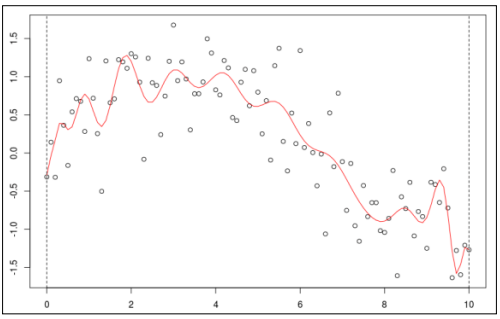
Pertimbangkan beberapa regresi polynomial, jika derajat pada fungsi polynomial cukup besar, ada beberapa jenis pola yang bisa didapatkan.



Gambar 199. Grafik polynomial derajat 1

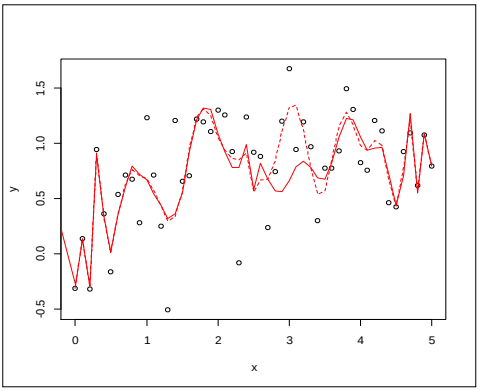
Tetapi jika derajat terlalu besar dan terlalu banyak oscilasi yang dioeroleh:

```
reg=lm(y~poly(x,25),data=db)
```



Gambar 200. Grafik polynomial derajat 2

Dan estimasi bisa dilihat tidak robust: jika mengubah satu titik, dimana bisa menjadi perubahan yang penting (local).



Gambar 201. Grafik dengan perubahan titik

b. Regresi Lokal

Kenyataaannya, jika tertarik untuk memiliki secara local sebagai pendekatan yang baik pada h(.), mengapa tidak menggunakan regresi local?. Ini dapat dikerjakan secara mudah menggunakan regresi terbobot, dimana dalam formulasi least square, dengan pertimbangkan:

$$\min\left\{\sum_{i=1}^{n}\omega_i\left[Y_i-\left(\beta_0+\beta_1X_i\right)\right]^2\right\}$$

(ini dimungkinkan untuk mempertimbangkan pembobot dalam framework GLM, tetapi marilah dijaga bahwa untuk post yang lain). Dua komentar disini:

a. Disini pertimbangkan model linier, tetapi beberapa model polynomial dapat dipertimbangkan. Sama sebagai satu konstanta. Dalam kasus ini, problem optimalisasi adalah:

$$\min\left\{\sum_{i=1}^{n}\omega_i\left[Y_i-\beta_0\right]^2\right\}$$

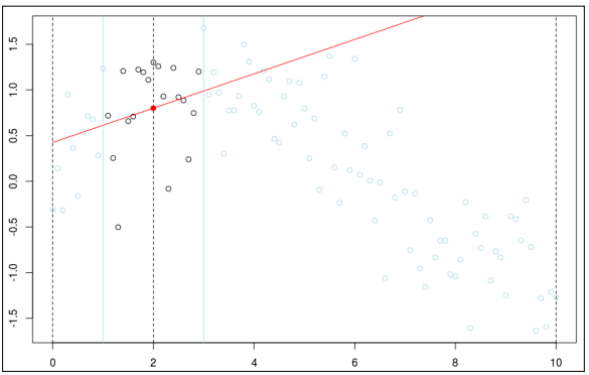
Dimana dapat diselesaikan secara eksplisit, ketika:

$$\hat{\beta}_0=\frac{\sum\omega_iY_i}{\sum\omega_i}$$

b. Sejauh ini, tidak ada yang dijelaskan tentang pembobot. Idenya sederhana, jika kita bisa memprediksi secara baik pada titik x_i ketika ω_i bisa menjadi proposional pada beberapa jarak antara

X_i dan x_0 jika X_i terlalu jauh dari x_0 ketika ini seharusnya tidak memiliki pengaruh dalam pendugaan.

Untuk diperhatikan, jika kita ingin mendapatkan pendugaan pada sebuah titik x_0, pertimbangkan $\omega_i \propto \mathbf{1}(|X_i - x_0| < 1)$. Dengan model ini, kita mengubah pengamatan terlalu jauh,



Gambar 202. Grafik simulasi fungsi regresi local

Nyatakan, disini adalah sama sebagai:

```
reg=lm(yr~xr,subset=which(abs(xr-x0)<1)
```

Sebagai sebuah gagasan umum adalah perlu mempertimbangkan sebuah fungsi kerne K(.), yang memberikan bentuk fungsi pembobot, dan sebuah bandwidth (biasanya ditandai dengan h) yang membentuk panjang disekitar data, sehingga:

$$\omega_i = K\left(\frac{x_0 - X_i}{b}\right)$$

Ini adalah secara actual disebut estimator Nadaraya-Watson pada fungsi h(.).

Pada kasus sebelumnya, kita telah mempertimbangkan sebuah kernel uniform:

$$K(x) = 1\left(x \in \left[-1/2, +1/2\right]\right)$$

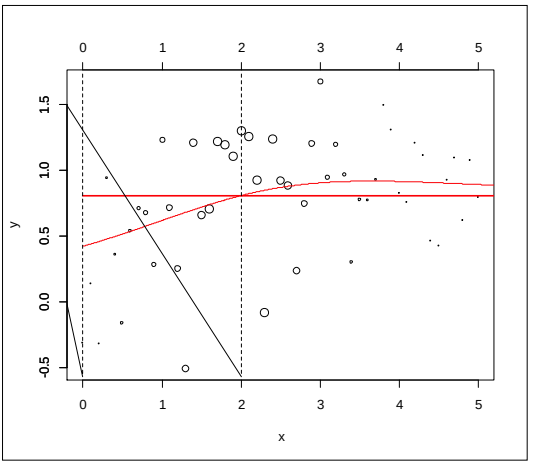
Dengan bandwidth 2.

Tetapi penggunaan fungsi pembobot ini, dengan diskontinuitas yang kuat bisa bukan sebuah ide yang baik, mengapa tidak kernel Gaussian.

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

Ini bisa menjadi berguna.

```
> fitloc0 = function(x0){
+ w=dnorm((xr-x0))
+ reg=lm(y~1,data=db,weights=w)
+ return(predict(reg,newdata=data.frame(x=x0)))}
      Pada dataset kita, kita bisa plotkan:
> ul=seq(0,10,by=.01)
> vl0=Vectorize(fitloc0)(ul)
> u0=seq(-2,7,by=.01)
> linearlocalconst=function(x0){
+ w=dnorm((xr-x0))
+ plot(db,cex=abs(w)*4)
+ lines(ul,vl0,col="red")
+ axis(3)
+ axis(2)
+ reg=lm(y~1,data=db,weights=w)
+ u=seq(0,10,by=.02)
+ v=predict(reg,newdata=data.frame(x=u))
+ lines(u,v,col="red",lwd=2)
+ abline(v=c(0,x0,10),lty=2)
+ }
> linearlocalconst(2)
```



Gambar 203. Grafik simulasi linier local pada pergerakkan titik secara horizontal.

Disini, kita ingin sebuah regresi local pada titik 2. Garis horizontal berikut adalah regresi (ukuran pada titik adalah proposional pada pembobot). Kurva, warna merah adalah evoluasi pada regresi local.Marilah kita gunakan sebuah animasi untuk menvisualkan konstruksi pada kurva. Animasi ini dapat menggunakan:

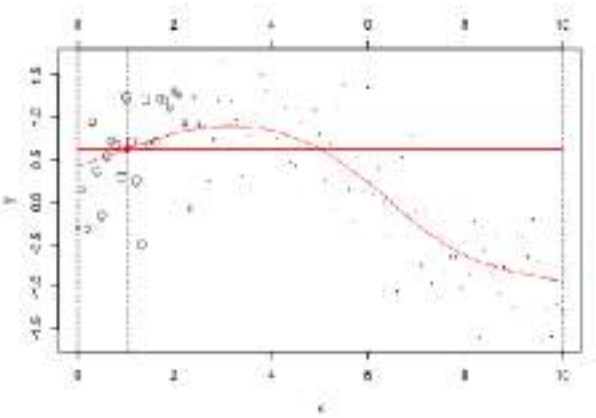
```
>library(animation)
```

Tetapi untuk beberapa alasan, kita tidak bisa menginstall paket dengan mudah dalam Linux. Dan ini bukan ide besar, kita masih dapat menggunakan sebuah loop untuk membangkitkan sebuah grafik.

```
> vx0=seq(1,9,by=.1)
> vx0=c(vx0,rev(vx0))
> graphloc=function(i){
+ name=paste("local-reg-",100+i,".png",sep="")
+ png(name,600,400)
+ linearlocalconst(vx0[i])
+ dev.off()}
> for(i in 1:length(vx0)) graphloc(i)
```

Dan kemudian, dalam sebuah terminal, mudahnya menggunakan:

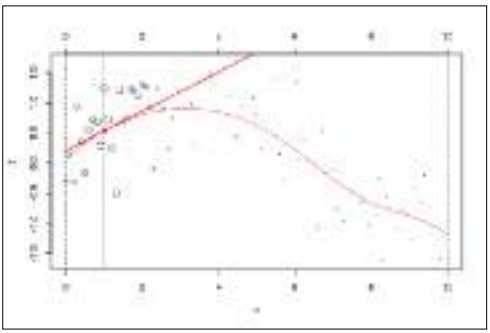
```
> convert -delay 25 /home/freak/local-reg-1*.png
/home/freak/local-reg.gif
```



Gambar 204. Grafik simulasi bentuk animasi

Tentunya, animasi ini mungkin untuk mempertimbangkan model linier, locally,

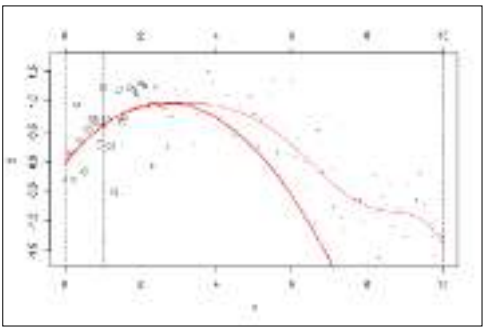
```
> fitloc1 = function(x0){
+ w=dnorm((xr-x0))
+ reg=lm(y~poly(x,degree=1),data=db,weights=w)
+ return(predict(reg,newdata=data.frame(x=x0)))}
```



Gambar 205. Grafik simulasi linier local

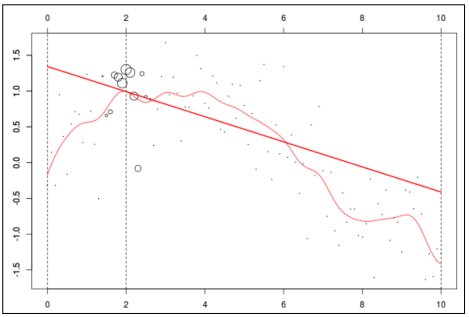
Atau sama regresi (local) kuadratik

```
> fitloc2 = function(x0){
+ w=dnorm((xr-x0))
+ reg=lm(y~poly(x,degree=2),data=db,weights=w)
+ return(predict(reg,newdata=data.frame(x=x0)))}
```



Gambar 206. Grafik fungsi regresi local kuadratik

Tentunya, bisa mengubah bandwidth



Gambar 207Grafik fungsi regersi dengan peubah bandwidth

Untuk memasukkan bagian teknikal post ini, pengamatan bahwa praktisnya, memiliki untuk memilih bentuk pada fungsi pembobot (Sehingga disebut kernel). Tetapi ini adalah teknik (mudah) untuk memilih bandwidth h optimal. Gagasan pada Cross validasi adalah dipertimbangkan.

$$\min\left\{\sum_{i=1}^{n}\left[Y_i - \hat{Y}_i(b)\right]^2\right\}$$

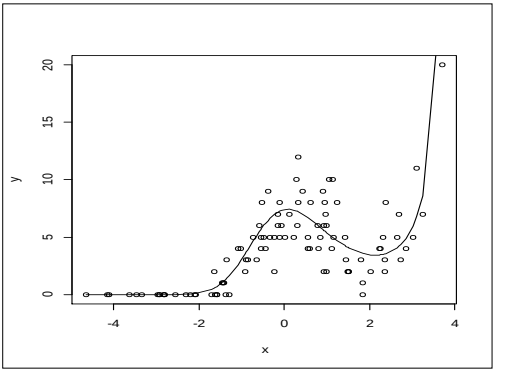
Dimana $\hat{Y}_i(b)$ adalah pendugaan yang diperoleh menggunakan teknik regresi local, dengan bandwidth h. dan untuk mendapatkan bandwidth lebih akurat (dan optimal) $\hat{Y}_i(b)$ diperoleh menggunakan model estimasi pada sampel dimana pengamatan ke-i telah diubah. Tetapi sekali lagi, bahwa hal ini bukan titik utama dalam post ini, sehingga marilah kita jaga untuk yang lain. Barangkali kita bisa mencoba pada beberapa data riil, inspirasi dari post terbesar dalam http://f.briatte.org/teaching/ida/092_smoothing.html, by François Briatte.

11.4 Splines

a. Kiat GLM: dapatkan non-linear dengan splines

Tip ini bagus untuk tes non-linear cepat, sebelum Anda melanjutkan dengan GAM atau model non-linear parametrik. Anda akan memerlukan pustaka splines, yang dikirimkan bersama R pula. Pertama, mari kita membuat sedikit data hitungan. Model 'benar' yang mendasarinya adalah poisson (think count data) dengan tautan log (sehingga estimasi kemiringan berlipat ganda dari mean poisson). Tetapi kami akan memperkenalkan sedikit non-linearitas.

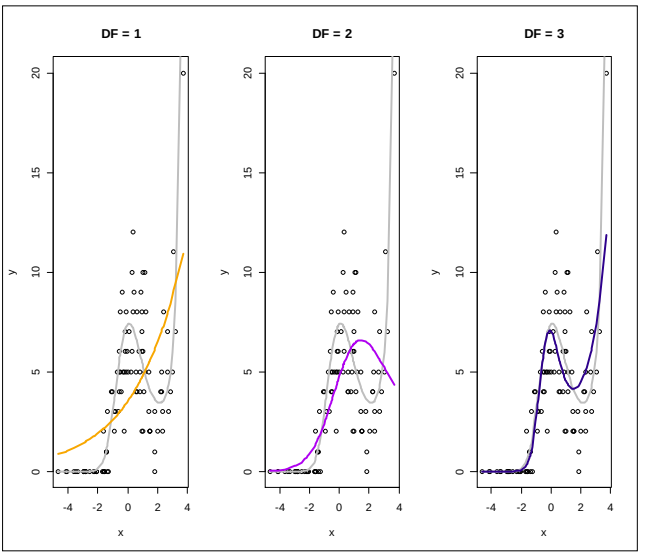
```
n <- 100
set.seed(101)
x <- sort(rnorm(n, sd = 2))
mu <- 2 + 0.1*x - 0.6*x^2 + 0.18*x^3#linear predictor
y <- rpois(n, exp(mu))
plot(x, y)
lines(x, exp(mu))
```



Gambar 208. Grafik Parametrik Nonlinier

Sekarang hanya bisa cocok dengan polinomial, tetapi untuk data nyata tidak akan tahu struktur rata-rata dihasilkan sebagai polinomial. Jadi mungkin ingin menggunakan sesuatu yang sedikit lebih fleksibel, seperti spline kubik. Jadi, inilah cara membuat spline kubik. Ini hanya perlu memilih derajat kebebasan Df dari 1 akan memberi kecocokan linier, Df yang lebih tinggi memungkinkan lebih banyak tikungan ('simpul'), yang akan mencocokkan model log-linear, model dengan df = 2 dan model dengan df = 3 Mengingat untuk menghasilkan data dengan polinomial kubik, diharapkan model 3 df akan melakukan yang terbaik.

```
library(splines)
#log linear model
m1 <- glm(y ~ x, family = "poisson")
m1pred <- predict(m1, type = "response")
#non-linear models
m2 <- glm(y ~ ns(x,2), family = "poisson")
m2pred <- predict(m2, type = "response")

m3 <- glm(y ~ ns(x,3), family = "poisson")
m3pred <- predict(m3, type = "response")
par(mfrow = c(1,3))
plot(x, y, main = "DF = 1")
lines(x, exp(mu), lwd = 2, col = "grey")
lines(x, m1pred, col = "orange", lwd = 2)
plot(x, y, main = "DF = 2")
lines(x, exp(mu), lwd = 2, col = "grey")
lines(x, m2pred, col = "purple", lwd = 2)
plot(x, y, main = "DF = 3")
lines(x, exp(mu), lwd = 2, col = "grey")
lines(x, m3pred, col = "darkblue", lwd = 2)
```



Gambar 209. Tampilan Grafik log-linier dan nonlinier dari DF1, D32 dan DF3

Garis oranye adalah fit linear naif, pada dasarnya tidak menunjukkan tren.Garis ungu (df = 2) lebih baik, tetapi melewatkan tendangan di akhir. Garis biru terlihat paling dekat dengan fungsi rata 'benar' (garis abu-abu).Garis ungu dan biru adalah spline masing-masing dengan 2 dan 3 knot. Keduanya jelas menangkap non-linearitas. Garis abu-abu adalah struktur rata 'benar' yang buat di atas. Jadi splines hanya memuncak agak terlalu keras, tetapi mendapatkan bentuk yang benar.

Supaya dapat meyakinkan bahwa model 3 df splines lebih baik dengan AIC

```
AIC(m1)
## [1] 492.1094
AIC(m2)
## [1] 434.6807
AIC(m3)
## [1] 362.3653
```

Berdasarkan nilai AIC, model 3 spline memiliki AIC jauh lebih rendah meskipun menggunakan lebih banyak model D.f., jadi lebih baik.

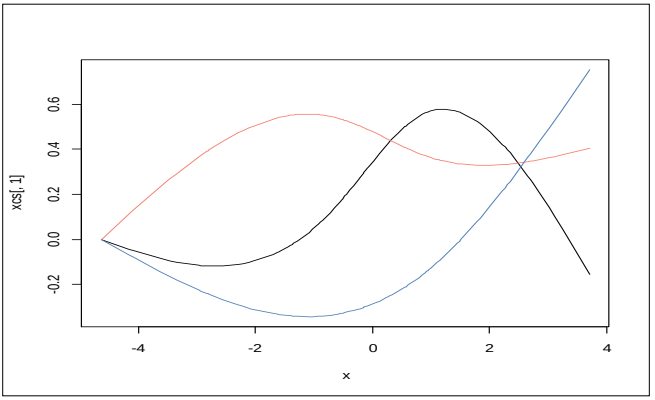
b. Aplikasi Spline.

Fitur bagus dari trik spline kubik ini adalah dapat menggunakannya di mana saja yang menggunakan matriks model sebagai input. Jadi ini akan bekerja dengan glm, glmer, lmer, dan metode Bayesian GLM apa pun yang

ingin digunakan. Cukup terapkan kriteria pemilihan model normal untuk menemukan jumlah simpul 'terbaik'.

Pertama, atur simpul menggunakan x:

```
library(splines)
xcs <- ns(x, 3) #3 knots!
head(xcs, 3)
##                 1         2           3
## [1,]  0.00000000 0.0000000  0.00000000
## [2,] -0.04388799 0.1157184 -0.07117461
## [3,] -0.04775128 0.1262731 -0.07766646
plot(x, xcs[,1], type = 'l', ylim = c(min(xcs),
max(xcs)))
lines(x, xcs[,2], col = "salmon")
lines(x, xcs[,3], col = "steelblue")
```



Gambar 210. Grafik tiga simpul

Ini membagi x menjadi tiga kovariat (perhatikan matriks xcs baru memiliki tiga kolom), yang telah diplot di atas. Algoritma spline kubik menempatkan tikungan dalam kovariat baru sesuai dengan kepadatan data. Dapat digunakan kovariat baru ini dalam model dan GLM akan memperkirakan koefisien untuk masing-masing. Karena *xcs* adalah fungsi non-linear dari x, cocok dengan model melawan berarti hal ini dapat mencampur kurva untuk mendapatkan kecocokan non-linear. Tentunya akan kehilangan beberapa derajat kebebasan, karena sekarang x adalah tiga kovariat, bukan hanya satu.yang bisa menggunakan xcs dalam rumus model sebagai kovariat. Tapi lebih tepat jika menempatkan perintah *ns* langsung ke formula model, dengan cara itu mudah untuk mengubah simpul, seperti yang dilakukan di atas.

c. Spline dengan GAM

Jika telah benar-benar ingin masuk ke fitting tren non-linear, harus digunakannya model aditif umum (GAM=*General Aditive Model*), seperti dari paket mgcv. Faktanya, GAM juga bisa cocok dengan splines semacam ini untuk (dan lebih banyak lagi), itu hanya menggunakan metode yang berbeda untuk memilih jumlah simpul. Tapi trik splines masih berguna. Misalnya, metode ini dapat dengan cepat memodifikasi model linier yang ada untuk memiliki spline non-linear. Atau gunakan itudalam model linier Bayesian yang tidak memiliki padanan GAM (Anda juga dapat memuntahkan matriks model dari mgcv dan menggunakannya dalam model Bayesian jika diinginkan jenis splines lainnya). Jadi silahkan menikmati splining.

References

R. Tibshirani and L. Wasserman, 2015. Nonparametric Regression, Statistical Machine Learning, lecture note of Carnegie Mellon University

BAB XII
APLIKASI REGERSI KERNEL PADA RISET TERAPAN

12.1 Judul Riset Sain Terapan

KOMBINASI ESTIMATOR KERNEL ORDER TINGGI UNTUK PENDUGAAN BIOMASSA POHON BUAH PADA LAHAN KRITIS

Oleh:
Zulfikar[1], Munawarah[2], Ambar Susanti[3]
[1.] Prodi Informatika, Fakultas Teknologi Informasi
[2]. Prodi Sistem Informasi, Fakultas Teknologi Informasi
[3.] Prodi Agroekoteknologi, Fakultas Pertanian
Universitas KH. A. Wahab Hasbullah, Tambakberas Jombang
Email: zulfikardia@gmail.com

12.2 Ringkasan

Penelitian ini bertujuan untuk menduga biomassa pohon buah dengan estimator kernel Gaussian order tinggi. Sumber data diperoleh dengan menduga hubungan diameter pohon (x) dan biomassa pohon (y) yang diperoleh melalui persamaan allometrik $Y = 0.11\rho D^{2,62}$ untuk pohon bercabang. Sampel data diambil sebanyak 250 diperoleh dari 5 lokasi kecamatan yang ada di lahan kritis pada vegetasi kelengkeng dan mangga. Hubungan antara x dan y selanjutnya diestimasi dalam bentuk persamaan $y_i = m(x_i) + \varepsilon_i$ dimana, i = 1, 2, ..., n. Hasil analisis dengan estimator kernel menunjukkan bahwa estimasi biomassa pohon menghasilkan kurva regresi yang smooth pada skor bandwidth (h).optimum 1.3 (GCV = 122.92) pada kelengkeng dan skor 2.6 (GCV = 3523.25) pada mangga. Hubungan antara MSE dan bandwidth menunjukkan korelasi negative yang sangat kuat dengan skor r = -0.959 (p.value = 0.04) pada estimasi biomassa kelengkeng dan r = -0.964 (p.value = 0.03) pada estimasi biomassa mangga dan signifikan ($\alpha = 0.05$). Hal ini berarti bahwa peningkatan secara parallel skor bandwidth akan menurunkan skor MSE ketika GCV terkontrol.

Kata kunci: Estimator kernel; Persamaan Allometrik; Biomassa pohon;

12.3 Latar Belakang

Indonesia telah ditetapkan sebagai negara dengan keragaman vegetasi terbesar kedua di dunia (Whitten et al. 1984). Sebagai daerah tropis, Indonesia merupakan sumber potensial spesies baru yang memiliki lebih dari setengah keanekaragaman flora dan fauna di dunia, sehingga diduga sebagai sumber terkaya penemuan jenis makro

organisme baru (Gandjar et al., 2006). Pada saat ini kondisi sumber daya lahan dan lingkungan di Indonesia semakin memprihatinkan ditunjukkan oleh meluasnya luas lahan kritis sehingga berdampak pada penurunan keragaman vegetasi. Lahan kritis umumnya lebih rapuh, mudah mengalami erosi, kurang produktif dan tidak mudah dikelola sehingga semakin menimbulkan penurunan kualitas lahan (Naidu et al., 2014). Kementerian LHK (2018) melaporkan bahwa luas lahan kritis dan sangat kritis di Indonesia pada tahun 2015 tanpa DKI Jakarta seluas lebih kurang 24.303.294 ha terdiri dari kritis sebesar 19.564.911 ha dan sangat kritis 4.738.384 ha[. Meluasnya lahan kritis tersebut disebabkan oleh beberapa hal antara lain: kerusakan hutan, perluasan areal pertanian yang tidak sesuai daya dukung lingkungan, tekanan jumlah penduduk yang terus meningkat dan kebakaran yang tidak terkendali (Muharam, 2011).

Pengembangan model keberhasilan konservasi harus mampu melakukan pendekatan penyelamatan lingkungan dari kerusakan ekologis yang salah satunya dengan estimasi biomassa pohon. Konsep pengembangan model keberhasilan konservasi yang mengkombinasikan karakteristik arsitektur pohon terhadap potensi biomassa dari tiap-tiap individu pohon sehingga dengan adanya metode pengukuran estimasi biomassa yang praktis maka potensi lahan bisa diketahui secara cepat sehingga upaya penyelamatan lingkungan segera teratasi. Pendugaan biomassa pohon bisa dilakukan dengan metode analisis dari kombinasi estimator Kernel Order Tinggi dengan *dendrometric analysis*. Metode analisis dendrometrik merupakan konsep pengukuran biomassa pohon dengan menggunakan pengukuran sampel pohon pada batang, cabang dan kanopi (Fernadez-Puratich et. al. 2013). Metode ini diharapkan dapat memberikan hasil estimasi biomassa pohon lebih akurat dan praktis dan tidak meninggalkan kerusakan pada pohon dan hutan. Model keberhasilan konservasi di lahan kritis yang dibangun adalah membentuk hubungan antara produktivitas pohon buah dengan faktor pembatas pertumbuhan dalam lingkungan faktor dan abiotik serta sosial ekonomi dan budaya yang ada di wilayah penelitian sehingga diharapkan terbentuknya model keberhasilan konservasi untuk menjadi rekomendasi bagi *stakeholder* untuk membuat kebijakan bagi program konservasi ke depannya.

Pengukuran biomassa pohon terutama untuk mengukur komponen biomassa pohon seperti daun, cabang, dan akar agar tidak memakan waktu lama dan mengurangi biaya hubungan empiris dapat digunakan untuk memperkirakan biomassa total variabel biometrik seperti diameter setinggi dada atau tinggi pohon (Pilli R, 2006; Cordero & Kanninen, 2003), dimana hubungan empiris antara komponen biomassa pohon membentuk persamaan alometrik. Karena pengukuran biomassa pohon di lapangan membutuhkan waktu yang lama terutama untuk mengukur komponen biomassa pohon seperti daun, cabang, dan

akar serta biaya yang tidak sedikit maka hubungan empiris dapat digunakan untuk mengestimasi biomassa total biometrik. variabel seperti diameter setinggi dada atau tinggi pohon (Pilli R, 2006; Cordero & Kanninen, 2003). Oleh karena itu, wajar jika pendugaan biomassa pohon dan hutan telah menjadi topik penelitian jangka panjang seperti yang pernah dilakukan oleh Kunze pada tahun 1873 dan Burger pada tahun 1929 (Fehrmann & Kleinn, 2006). Krisnawati (2012) telah menyusun 807 model alometrik biomassa dan model alometrik volume pohon pada beberapa tipe ekosistem hutan, dimana sebanyak 437 model alometrik untuk pendugaan komponen biomassa pohon dan 370 model alometrik untuk pendugaan beberapa jenis volume pohon. Hampir semua tipe ekosistem hutan utama di Indonesia tersedia model alometrik biomassa dan/atau volume pohon meskipun sebarannya tidak merata di pulau-pulau besar di Indonesia. Dengan demikian penelitian penyusunan model persamaan alometrik untuk pendugaan biomassa suatu jenis pohon pada suatu tipe ekosistem masih diperlukan untuk memperkaya data yang ada di seluruh nusantara.

Hampir semua estimasi studi biomassa difokuskan pada penerapan model regresi linier dan nonlinier (Kasischke et al., 1995), (Polatin et al., 1994), (Rignot et al., 1994). Tetapi pemetaan antara parameter permukaan tanah dan Citra SAR selalu sangat kompleks karena nonlinier yang kuat. Model regresi berdasarkan pengukuran data nyata tidak dapat memberikan hubungan yang cukup jelas. Metode alometrik tradisional berupa persamaan Schumacher-Hall memberikan pendugaan biomassa yang kurang akurat, dimana transformasi logaritmik memiliki kelemahan yang tidak dapat meningkatkan validitas pendugaan (Sangietta, C.R., et al, 2015). Oleh karena itu, diperlukan metode estimasi yang mampu memberikan hasil yang lebih akurat dengan jumlah data yang besar, selain itu teknik estimasi harus fleksibel dan tidak memerlukan asumsi regresi. Penduga kernel adalah teknik aproksimasi regresi yang tidak memerlukan asumsi normalitas. Zulfikar (2010) menyatakan bahwa High Order Kernel Estimator memiliki nilai MSE yang lebih kecil dan merupakan indikator untuk memilih estimator terbaik.

Faktor pertumbuhan lebih sederhana daripada membangun dan menerapkan, tetapi model alometrik lebih disukai karena peningkatan fleksibilitas untuk menggambarkan variasi arsitektur pohon dan kompartementalisasi biomassa (Petrokofsky, G. et al., 2012). Aspek penting dalam kuantifikasi biomassa individu pohon adalah variabilitas alami yang besar dalam data, terutama untuk spesies asli dari daerah tropis dan subtropis. Formulasi matematis tunggal mungkin tidak dapat mereproduksi seperti variasi alami yang hebat. Faktor ini mempengaruhi kualitas model yang cocok dan dapat memberikan perkiraan yang salah. Ciri lain dari model alometrik adalah ketika menggunakan teknik regresi beberapa asumsi harus dicapai. Asumsi-

asumsi tersebut adalah sebagai berikut: aditif dan linieritas, independensi residual, homoskedastisitas, dan normalitas residual (Osborne J, Waters E, 2002). Terkait dengan hal tersebut, maka penelitian ini bertujuan untuk mengestimasi biomassa pohon buah pada lahan kritis menggunakan estimator kernel order tinggi berdasarkan diameter batang terhadap besarnya biomassa tanaman untuk memperoleh penduga kurva regresi yang diperoleh dengan menaksir parameter tersebut (Hardle, W., 1990).

12.4 Metodelogi

12.4.1 Sumber Data dan Variabel Penelitian

Data yang digunakan dalam penelitian ini adalah data primer di Kabupaten Jombang. Model regresi yang dibangun adalah untuk mendapatkan bentuk hubungan antara diameter batang (X) dengan biomassa pohon (Y). Parameter tanaman yang diukur adalah diameter batang pada ketinggian 1,3 meter di atas permukaan tanah, dan diameter batang dalam sentimeter. Pohon buah yang diteliti adalah kelengkeng dan mangga yang masing-masing diambil 250 sampel data. Pengukuran biomassa pohon menggunakan rumus persamaan alometrik pohon bercabang:

$$Y = 0.11\rho D^{2,62} \quad (1)$$

Dimana:
Y = biomassa di atas permukaan (Kg / pohon)
ρ = kerapatan jenis kayu (g cm^{-2})
D = Diameter pohon (cm)
Sumber: Hairiah, et. al. (2007), dan Watson (2009).
Pohon buah yang diukur biomassanya adalah kelengkeng dan mangga dengan kerapatan jenis kayu masing-masing 0.91 dan 0.58 (Baruna et.al.).

12.4.2 Metode Analisis

Langkah-langkah yang dilakukan dalam penelitian ini adalah dengan melakukan tahapan estimasi, selanjutnya dilakukan analisis model regresi kernel dengan menggunakan program R tipe 3.61. Estimasi biomassa dengan model regresi nonparametrik berdasarkan estimator kernel polinomial lokal dengan tahapan estimasi sebagai berikut:

a. Didapatkan data observasi (y_i, x_i) yang memenuhi regresi nonparametrik:

$$Y_i = m(x_i) + \varepsilon_i, \quad i = 1,2,3,...,n \quad (2)$$

b. membuat plot data berpasangan: $(y_i, x_i), \quad i = 1,2,...,n$
c. menentukan jenis dan fungsi tertimbang dari Kernel Gaussian
d. tentukan matriks A (h) berukuran N x N.

e. memilih orde polinomial p dan meminimalkan nilai bandwidth yang optimal

$$GCV = \frac{n^{-1}\sum_{i=1}^{n}\left[y-\hat{y}_i\right]^2}{\left(n^{-1}tr\left[1-A(h)\right]\right)^2} \qquad (3)$$

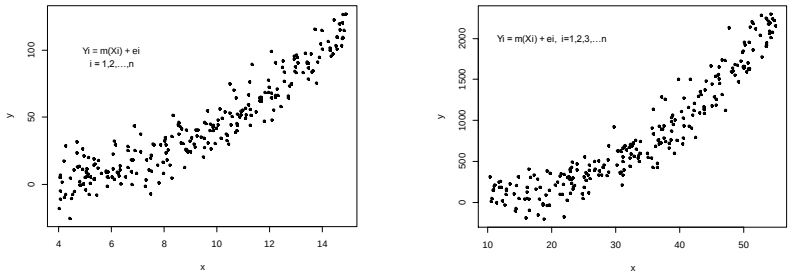
f. memodelkan orde polinomial lokal p dan nilai bandwidth optimal dari langkah 4) secara bersamaan
g. hitung nilai rata-rata kuadrat error :

$$MSE(h) = n^{-1}\sum_{i=1}^{n}\left(y_i-\hat{y}_i\right) \qquad (4)$$

h. mendapatkan model estimasi biomassa dengan estimator kernel polinomial lokal.

12.5 Hasil dan Pembahasan

Untuk mengaplikasikan estimator kernel Gaussian digunakan data penelitian dari hasil pengukuran diameter batang pohon kelengkeng dan mangga masing-masing sebanyak 250 data sampel. Akan dilihat hubungan antara diameter batang (*x*) dengan besarnya biomassa pohon (*y*) berdasarkan persamaan (2) dengan *m* kurva regresi. Plot antara x dan y diberikan oleh Gambar.12.1.



(a) (b)

Gambar 211. Plot data x dan y pada biomassa kelengkeng (a), mangga (b)

Dari Gambar 1 terlihat bahwa belum adanya pola yang jelas mengenai hubungan antara *x* dan *y*. Selanjutnya digunakan regresi nonparametrik khususnya estimator Kernel order 2 untuk mengestimasi *m*. Estimasi kernel diberikan oleh:

$$\hat{m}(x,0,h) = \frac{\sum_{i=1}^{n} K_h(x_i - x) y_i}{\sum_{i=1}^{n} K_h(x_i - x)} \qquad (5)$$

dengan kernel Gaussian. Bentuk kernel Gaussian order 2 diperoleh dengan mensubstitusikan fungsi kernel:

$$K(z) = 1/\sqrt{2\pi}\exp(-z^2/2) \qquad (6)$$

ke dalam persamaan (5).

Pertama diberikan pemilihan bandwidth optimal untuk masing-masing kernel dengan menggunakan metode GCV. Fungsi GCV diberikan oleh:

$$\text{GCV(h)} = n^{-1}\sum_{i=1}^{n} \frac{\{y_i - \hat{m}(x_i)\}^2}{\{1 - n^{-1}tr(A(h)\}^2}, \qquad (7)$$

Dengan A(h) diperoleh dari persamaan :

$$\hat{m}(x, p = 0, h) = A(h)y$$

a. Fungsi regresi kernel pada estimasi biomassa kelengkeng dan Mangga

Regresi kernel order 2 yang digunakan untuk estimasi biomassa kelengkeng dibangung dengan pendekatan meminimumkan fungsi GCV(h) untuk diperoleh nilai bandwidth optimum. Untuk beberapa bandwidth nilai GCV(*h*) yang tertuang dalam tabel 1 dan 2 berikut:

Tabel 6. Nilai Bandwidth Estimator kernel order 2 pada pengukuran biomassa pohon kelengkeng

No.	GCV(h)	Bandwidth(h)	No.	GCV(h)	Bandwidth(h)
1.	126.4935	0.5	10.	122.9487	1.4
2.	125.8920	0.6	11.	123.0502	1.5
3.	125.1507	0.7	12.	123.2061	1.6
4.	124.4403	0.8	13.	123.4125	1.7
5.	123.8527	0.9	14.	123.6715	1.8
6.	123.4140	1.0	15.	123.9822	1.9
7.	123.1241	1.1	16.	124.3464	2.0
8.	122.9667	1.2	17.	124.7719	2.1
9.	**122.9177**	**1.3**	18.	125.2594	2.2

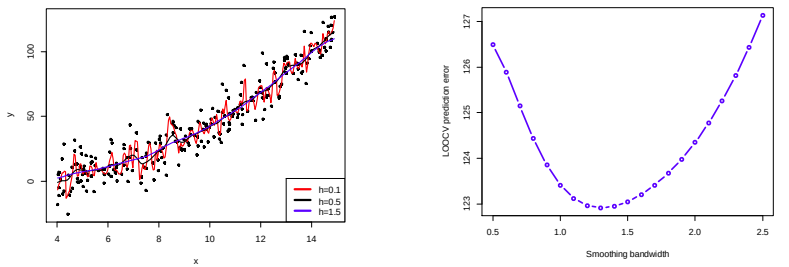
Pada tabel 1 terlihat bahwa nilai bandwith optimum pada nilai 1.3 dengan nilai GCV minimum sebesar 122.9177 pada estimasi biomassa

pohon kelengkeng. Hal yang sama juga ditunjukkan pada tabel 2, dimana nilai bandwidth optimum pada nilai 2.6 dengan nilai GCV minimum sebesar 3523.248 pada estimasi biomassa pohon mangga seperti ditunjukkan pada tabel 2. Penggunaan h optimum pada regresi kernel digunakan untuk pendugaan kurva regresi yang diperoleh dengan menaksir parameter tersebut.

Tabel 7. Nilai Bandwidth Estimator kernel order 2 pada pengukuran biomassa pohon mangga

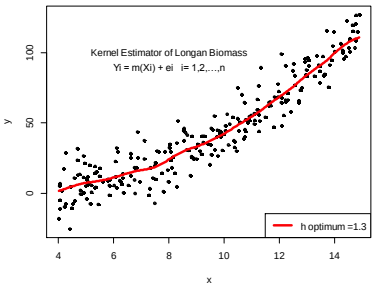
No.	GCV(h)	Bandwidth(h)	No.	GCV(h)	Bandwidth(h)
1.	3631.274	1.4	10.	3527.719	2.3
2.	3611.305	1.5	11.	3525.106	2.4
3.	3593.866	1.6	12.	3523.632	2.5
4.	3578.582	1.7	13.	**3523.248**	**2.6**
5.	3565.415	1.8	14.	3523.755	2.7
6.	3554.208	1.9	15.	3525.064	2.8
7.	3544.951	2.0	16.	3527.008	2.9
8.	3537.562	2.1	17.	3529.537	3.0
9.	3531.791	2.2	18.	3532.424	3.1

Fungsi $GCV(h)$ diberikan dalam Gambar 212 terlihat bahwa fungsi GCV(h) membentuk kurva pada nilai minimum untuk didapatkan nilai bandwidth optimum. Gambar kurva ini mempertegas meminimumkan skor dari fungsi GCV(h) akan didapatkan nilai bandwidth (h) optimum baik yang ditunjukkan pada estimasi biomassa kelengkeng maupun mangga.



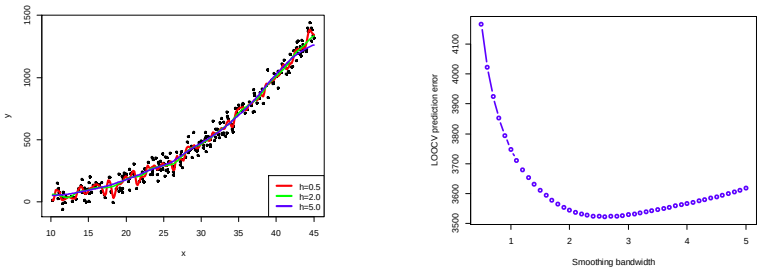
(a) (b)

Gambar 212. Grafik fungsi kernel pada berbagai ukuran bandwidth (a), dan penetapan GCV optimum (b)



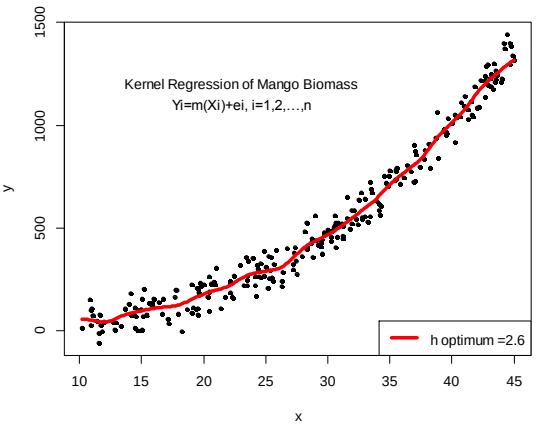
Gambar 213. Fungsi regresi kernel order 2 dengan bandwidth (h) optimum

Pada gambar 212 terlihat bahwa pembentukkan kurva regresi kernel untuk estimasi biomassa pohon kelengkeng terlihat bahwa pada h = 0,1 membentuk kurva yang kasar, sedangkan pada h = 1,5 kurva regresi mulai terlihat smooth. Agar didapatkan kurva yang smooth dengan nilai h optimum maka perlu dilakukan dengan meminimumkan nilai GCV seperti terlihat pada gambar 212, dimana nilai GCV minimum pada estimasi kernel ini didapatkan nilai h optimum = 1.3. Selanjutnya didapatkan kurva regresi kernel dengan h optimum seperti terlihat pada gambar 12.3, dan menjadi fungsi regresi kernel yang ideal untuk estimasi besaran biomassa pohon kelengkeng. Hal yang sama juga ditunjukkan pada gambar 12.4, yang merupakan pembentukan kurva regresi kernel yang smooth pada bandwidth optimum sebesar 2.6.



(a) (b)

Gambar 214. Grafik fungsi regresi kernel dengan berbagai nilai bandwidth (a) dan penetapan bandwidth optimum dengan GCV (b)



Gambar 215. Fungsi regresi kernel order 2 dengan bandwidth (h) optimum

b. Uji Validasi Estimator kernel

Fungsi GCV(h) yang minimum untuk penentuan bandwidth optimum akan didapatkan fungsi regresi kernel yang smooth, selanjutnya akan membentuk hubungan antara MSE dan bandwidth. Pada gambar 215 terlihat bahwa semakin besar skor bandwidth akan memperhalus kurva dimana batas nilai h tidak bisa ditentukan sejauh mana kurva regresi kernel optimal. Sebagai bentuk validasi hasil penggunaan regresi kernel order tinggi diperlukan skor MSE dan membendingkannya dengan nilai bandwidth dan GCV.

Tabel 8. Nilai *MSE* dan GCV dan bandwidth (h) Estimator Kernelpada Estimasi Biomassa Kelengkeng dan Mangga

Estimasi Biomassa	**GCV(h)**	**Bandwidth (h)**	***MSE***
a. Kelengkeng	151.6033	0.1	88.83703
	126.4935	0.5	48.88685
	122.9177	**1.3***	**43.09379**
	123.0502	1.5	41.66774
	131.7924	3.0	28.41932
b. Mangga	4166.082	0.5	8448.196
	3523.632	2.5	7753.965
	3523.248	**2.6***	**7670.941**
	3673.627	5.0	7591.417
	4273.334	7.5	6988.180

Keterangan: * h optimum

Hubungan antara MSE dan bandwidth untuk melihat validasi regresi kernel yang digunakan selanjunya dilakukan uji korelasi parsial dengan metode Pearson dimana GCV sebagai variabel control seperti yang ditunjukkan pada tabel 12.4 dan 12.5.

Tabel 9. Korelasi parsial antara MSE dan Bnadwidth dengan GCV sebagai control pada estimasi biomassa kelengkeng dengan metode Pearson

	Bandwidth	GCV	MSE
Bandwidth	1.000	0.893	-0.960*
p-value	1.000	0.107	0.040
GCV	0.893	1.000	0.961
p-value	0.107	1.000	0.038
MSE	-0.960*	0.961	1.000
p-value	0.040	0.038	1.000

*) Signifikansi korelasi parsial pada level P-value < ($\alpha = 0.05$)

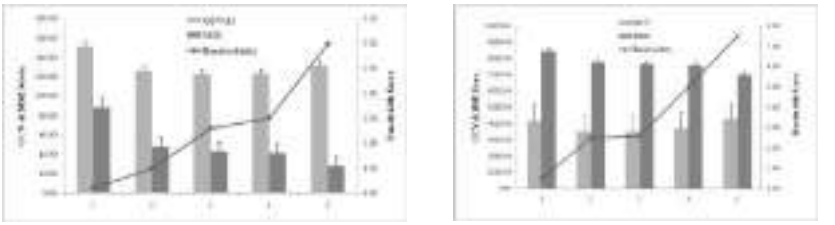
Korelasi parsial antara bandwidth dan MSE pada skor -0.9598964 merupakan korelasi negative yang kuat. Hal ini menunjukkan bahwa peningkatan secara parallel nilai bandwidth akan menurunkan skore MSE ketika GCV dikontrol. Besaran p-value pada korelasi parsial ini adalah 0.04010363 yang mana secara statistik adalah signifikan pada $\alpha = 0.05$.

Tabel 10. Korelasi parsial antara MSE dan Bnadwidth dengan GCV sebagai control pada estimasi biomassa Mangga dengan metode Pearson

	Bandwidth(h)	GCV(h)	MSE
Bandwidth(h)	1.000	0.651	-0.964*
p-value	1.000	0.349	0.035
GCV(h)	0.651	1.000	0.613
p-value	0.349	1.000	0.387
MSE	-0.964*	0.613	1.000
p-value	0.036	0.387	1.000

*) Signifikansi korelasi parsial pada level P-value < ($\alpha = 0.05$)

Hal yang sama ditunjukkan dari hasil uji korelasi parsial antara bandwidth dan MSE pada skor -0.9642895 merupakan korelasi negative yang kuat. Peningkatan skor secara parallel nilai bandwidth akan menurunkan skore MSE ketika GCV terkontrol dengan p-value sebesar 0.03571046 yang secara statistik adalah signifikan pada $\alpha = 0.05$. Tingkat perbedaan nilai MSE, GCV dan bandwidth dapat dilihat pada gambar 12.6.



(a) (b)

Gambar 216. Hubungan antara MSE, bandwidth dan GCV, dimana MSE dan bandwidth menunjukkan korelasi negative yang kuat (r = -0.82) untuk estimasi biomassa kelengkeng dan (r = -0.94) untuk mangga. Sedangkan nilai GCV minimum untuk menunjukkan nilai bandwidth optimum

Pada gambar 12.6 terlihat bahwa grafik batang yang dibentuk antara MSE, GCV dan bandwidth membentuk pola yang sama baik pada estimasi biomassa kelengkeng dan mangga. Pola skor MSE menunjukkan npenurunan seiring dengan naiknya nilai bandwidth, sedangkan skor GCV menunjukkan pola menurun lalu naik pada kenaikan skor bandwidth dan penurunan skor MSE. Penetapan bandwidth optimum pada nilai GCV minimum akhirnya penentapkan nilai MSE sebagai skor nilai yang valid untuk model regresi kernel yang dibentuk.

Pembahasan

Kendala utama dalam estimasi biomassa pohon adalah penggunakan estimator yang kurang cermat dalam memilih metode analisis, dimana sering ditemukan data estimasi dengan pendekatan regresi klasik. Apalagi banyak para ahli ekologi dalam melakukan studi hampir semua estimasi biomassa difokuskan pada penerapan model regresi linier dan nonlinier (Kasischke et al., 1995), (Polatin et al., 1994), (Rignot et al., 1994). Ditambahkan pula bahwa hasil penelitian terdahulu menunjukkan bahwa estimasi biomassa memiliki parameter yang berhadapan dengan lingkungan yang komplek dimana pemetaan antara parameter permukaan tanah dan Citra SAR selalu sangat kompleks karena nonlinier yang kuat. Model regresi berdasarkan pengukuran data nyata tidak dapat memberikan hubungan yang cukup

jelas. Metode alometrik tradisional berupa persamaan Schumacher-Hall memberikan pendugaan biomassa yang kurang akurat, dimana transformasi logaritmik memiliki kelemahan yang tidak dapat meningkatkan validitas pendugaan (Sangietta, C.R., et al, 2015).

Metode estimasi biomassa mengggunakan estimator kernel order tinggi sangat membantu mengatasi kendala-kendala tersebut, karena model hubungan variabel yang dibentuk fleksibel tanpa perlu memprediksi bentuk hubungan dari kedua variabel tersebut. Hasil penelitian ini menunjukkan bahwa estimator kernel mampu menghasilkan estimasi biomassa pohon baik pada kelengkeng maupun mangga dengan melakukan pendugaan bentuk regresi dengan meminimumkan skor GCV dan berhasil mendapatkan bandwidth optimum dengan 1,3 untuk estimasi biomassa kelengkeng dan 2,6 untuk mangga. Penggunaan h optimum pada regresi kernel digunakan untuk pendugaan kurva regresi yang diperoleh dengan menaksir parameter tersebut (Hardle, W., 1990). Pendekatan pemilihan bandwidth telah didapatkan bahwa metode GCV secara asimtotik optimal nilai rata-rata pada kasus data independensi (Hart& Vie 1990). Hall et al. (1995) menambahkan bahwa studi berikutnya tentang sifat-sifat asimtotik pada bandwidth optimal dibawah taraf yang berbeda pada dependensi.

Validasi metode regresi yang dibangun dengan mendekatan ini digunakan dengan melakukan mengukuran MSE, dimana bisa menunjukkan skor MSE yang rendah, dimana penggunakan estimator Kernel Estimator memiliki nilai MSE yang lebih kecil dan merupakan indikator untuk memilih estimator terbaik. Hubungan antara Bandwidth dan MSE secara parsial dengan GCV terkontrol menunjukkan bahwa didapatkan korelasi negative yang kuat, dimana semakin besar bandwidth akan memperkecil skor MSE. Peran bandwidth sebagai penghalus kurva dengan skor yang semakin naik akan dikendalikan dengan menurunnya nilai MSE, dengan demikian peran GCV akan menunjukkan batas nilai bandwidth optimumnya. Pada penelitian ini telah menunjukkan bahwa hubungan antara bandwidth dan MSE berkorelasi negative yang kuat dan signifikan pada taraf (α) 0.05, dengan nilai r sebesar -0.960 pada estimasi biomassa kelengkeng dan r sebesar -0.964 pada mangga. Kim et al. 2016 menambahkan bahwa penggunnan MSE dengan skor rendah untuk validasi model regresi lebih diusulkan daripada metode standar. Keuntungan lain dari penggunaan estimator kernel order tinggi adalah tahapan metode lebih ringkas, karena tanpa perlu menggunakan persyaratan klasik seperti pada penggunaan regresi parametrik. Pada estimasi regresi nonparametrik ini regresi yang dibentuk lebih ditunjukkan pada pembentukkan kurva regresi dari bentuk yang kasar sampai batas smooth yang terkendali dengan GCV.

12.6 Kesimpulan

Penggunnan estimator kernel order tinggi mampu mengatasi kesulitan pada estimasi biomassa pohon baik pada kelengkeng maupun mangga. Hasil estimasi biomassa menunjukkan bahwa kurva regresi yang dihasilkan membentuk kurva yang smooth pada bandwidth optimal dengan skor 1,3 pada kelengkeng dan 2,6 pada mangga dari hasil meminimumkan fungsi GCV(h). Validasi estimator terbaik digunakan pemilihan skor MSE dan hubungannya dengan Bandwidth pada nilai GCV terkontrol. Hasil penelitian menunjukkan bahwa terdapat korelasi negative yang kuat dimana semakin besar nilai bandwidth akan memperkecil MSE.

12.7 Ucapan terima kasih.

Peneliti mengucapkan terima kasih atas bantuan hibah Penelitian Terapan multi years tahun 2021-2022, dari Riset dan Pengabdian Masyarakat kepada Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset, Teknologi, dan Pendidikan Tinggi. Ucapan terima kasih juga disampaikan kepada Dinas Lingkungan Hidup Kabupaten Jombang atas jalinan kerja sama penelitian yang dilakukan selama ini, Dinas Pertanian dan segenap Civitas akademika Universitas KH. A. Wahab Hasbullah.

12.8 Referensi

Cordero, L.D.P & M. Kanninen. 2003. Aboveground biomass of Tectona grandis plantation in Costa Rica. *Journal of Tropical Forest Science* 15 (1): 199-213.

Fehrmann, L & C. Kleinn. 2006. General considerations about the use of allometric equations for biomass estimation on the example of Norway spruce in central Europe. *Forest Ecology and Managemen*t 236: 412-421.

Fernadez-Puratich, Jose V. Oliver-Villanueva, David Alfonso-Solar and Elisa Penalvo-Lopez. 2013. Quantification of Potential Lignocellulosic Biomass in Fruit Trees Grown in Mediterranean Regions. *BioResources* 8(1), 88-103.

Hardle, W. 1991. Applied nonparametric regression. Gambridge: Cambridge University Press

Hart, J. D. & Vieu, P. (1990). Data-driven bandwidth choice for density estimation based on dependent data. The Annals of Statistics 18, 873–890.

Hall, P., Lahiri, S. N. & Truong, Y. K. (1995). On bandwidth choice for density estimation with dependent data. The Annals of Statistics 23, 2241–2263.

Kasischke, E. S., Christensen, N. L. and Bourgeau-Chavez, L. L., (1995). Correlating radar backscatter with components of biomass in loblolly pine forest. IEEE Trans. Geosci. *Remost Sensing* 32, pp. 643–659.

Kementerian LHK. 2018. Rencana Strategis 2015-2019. Kementerian Lingkungan Hidup dan Kehutanan: Jakarta.

Krisnawati,H.,W.C. Adinugroho & R. Imanuddin. 2012. Monograf: Model-model allometrik untuk pendugaan biomassa pohon pada berbagai tipe ekosistem hutan di Indonesia. Pusat Penelitian dan Pengembangan Konservasi dan Rehabilitasi. Badan Penelitian dan Pengembangan Kehutanan,Bogor. Indonesia.

Muharam (2011). Pengembangan Model konservasi Lahan dan Sumberdaya Air dalam rangka Pengentasan Kemiskinan. *Solusi Unsika*. 10(20)- Ed. Sept-Nop.

Naidu, LGK, S. Dharumaraja, M. Lalitha, S. Srinivas, V. Ramamurthy and SK. Singh. 2014.Categorization and delineation of prime and marginal lands of Andhra Pradesh for different uses. *Agropedology*, 24 (02), 253-261

Osborne J, and Waters E. 2002. Four assumptions of multiple regression that researchers should always test. *Pract Assessment Res Evaluation*;8(2):1–8.

Petrokofsky G, Kanamaru H, Achard F, Goetz SJ, Joosten H, Holmgren P, et al. 2012. Comparison of methods for measuring and assessing carbon stocks and carbon stock changes in terrestrial carbon pools. How do the accuracy and precision of current methods compare? A systematic review protocol. *Environmental Evidence*.1:1–22.

Polatin, P. F., Sarabandi, K. and Ulaby, F. T., (1994). An iterative inversion algorithm with application to the polarimetric radar response of vegetation canopies. *IEEE Trans. Geosci. Remost Sensing 32*, pp. 62–71.

Pilli R., Anfodillo T., and Carrer M., 2006. Towards a functional and simplified allometrry for estimating forest biomass. *For. Ecol. Manage. 237: 583–593.*

Pretzscha, H., Peter Bibera, Enno Uhla, Jens Dahlhausena, Thomas Rötzera, Juan Caldenteyb, Takayoshi Koikec, Tran van Cond, Aurélia Chavannee, Thomas Seifert f., Ben du Toit f. Craig Farndeng, Stephan Pauleith. 2015. Crown size and growing space requirement of common tree species in urban centres, parks, and forests, *Urban Forestry & Urban Greening* 14 (2015) 466–479

Rignot, E., Way, J. B., Williams, C. and Viereck, L., (1994). Radar estimates of aboveground biomass in boreal forests of interior alaska. IEEE Trans. *Geosci. Remost Sensing 32*, pp. 1117–1124.

Sanquetta C. R. Jaime Wojciechowski, Ana P. Dalla Corte†, Alexandre Behling†, Sylvio Péllico Netto†, Aurélio L. Rodrigues*†and Mateus N. I. Sanquetta†. 2015. Comparison of data mining and allometric model in estimation of tree biomass. *BMC Bioinformatics*, 16:247.

Whitten, A., Anwar, J., Damanik, S., and Hisyam, N. 1984. The Ecology of Sumatra**,** Oxford University Press, 512 pp.

Zulfikar. 2010. Kernel Order Tinggi untuk Estimasi *Value at Risk* (VaR) Manajemen Resiko Tenaga Kerja, *Proseding Seminar Nasional Manajemen Teknologi XII*, MMT ITS, Surabaya. Hal ; C .1 – 6.

Kim, H. J. Steven N., MacEachern & Yoosuh Jung. 2016. Bandwidth selection for kernel density estimation with a Markov Chain Monte Carlo sample. arXiv:1607.08274v1[stat.ME] 27 Ju 2016.

DAFTAR PUSTAKA

Alain F. Zur., 2009. A Beginner's Guide to R, Springer.

Annette J. Dobson , 1990. An Introduction to Generalized Linear Models, Chapman and Hall, London.

Budiharto, W dan Ro'fah N. R. 2013. Pengantar Praktis pemrograman R untuk Ilmu Komputer. Jakarta: Halaman Moeka Publishing.

Cordero, L.D.P & M. Kanninen. 2003. Aboveground biomass of Tectona grandis plantation in Costa Rica. *Journal of Tropical Forest Science* 15 (1): 199-213.

Emanuel Paradis, 2005. R for the Beginner, Institut des Sciences de l' _ Evolution, Paris.

Fehrmann, L & C. Kleinn. 2006. General considerations about the use of allometric equations for biomass estimation on the example of Norway spruce in central Europe. *Forest Ecology and Managemen*t 236: 412-421.

Fernadez-Puratich, Jose V. Oliver-Villanueva, David Alfonso-Solar and Elisa Penalvo-Lopez. 2013. Quantification of Potential Lignocellulosic Biomass in Fruit Trees Grown in Mediterranean Regions. *BioResources* 8(1), 88-103.

Gio, P.U. dan E. Rosmaini, 2015. Belajar Olah Data dengan SPSS, Minitab, R, Microsoft Excel, EViews, LISREL, AMOS, dan SmartPLS. USUpress.

Hall, P., Lahiri, S. N. & Truong, Y. K. (1995). On bandwidth choice for density estimation with dependent data. The Annals of Statistics 23, 2241–2263.

Hardle, W. 1991. Applied nonparametric regression. Gambridge: Cambridge University Press

Hart, J. D. & Vieu, P. (1990). Data-driven bandwidth choice for density estimation based on dependent data. The Annals of Statistics 18, 873–890.

Hornik, K. 2016. R FAQ. Retrieved March 16, 2016, from https://cran.r-project.org/doc/FAQ/R-FAQ.html#Why-is-R-named-R_003f

http://cran.r-project.org/doc/manuals/r-release/R-intro.html

John A. Rice, 1995. Mathematical Statistics and Data Analysis. Second edition. Duxbury Press, Belmont, CA, 1995.

Kasischke, E. S., Christensen, N. L. and Bourgeau-Chavez, L. L., 1995. Correlating radar backscatter with components of biomass in loblolly pine forest. IEEE Trans. Geosci. *Remost Sensing* 32, pp. 643–659.

Kementerian LHK. 2018. Rencana Strategis 2015-2019. Kementerian Lingkungan Hidup dan Kehutanan: Jakarta.

Kim, H. J. Steven N., MacEachern & Yoosuh Jung. 2016. Bandwidth selection for kernel density estimation with a Markov Chain Monte Carlo sample. arXiv:1607.08274v1[stat.ME] 27 Ju 2016.

Krisnawati,H.,W.C. Adinugroho & R. Imanuddin. 2012. Monograf: Model-model allometrik untuk pendugaan biomassa pohon pada berbagai tipe ekosistem hutan di Indonesia. Pusat Penelitian dan Pengembangan Konservasi dan Rehabilitasi. Badan Penelitian dan Pengembangan Kehutanan,Bogor. Indonesia.

Muharam, 2011. Pengembangan Model konservasi Lahan dan Sumberdaya Air dalam rangka Pengentasan Kemiskinan. *Solusi Unsika*. 10(20)- Ed. Sept-Nop.

Naidu, LGK, S. Dharumaraja, M. Lalitha, S. Srinivas, V. Ramamurthy and SK. Singh. 2014.Categorization and delineation of prime and marginal lands of Andhra Pradesh for different uses. *Agropedology*, 24 (02), 253-261

Nicholas Walliman, 2011. Research Method Basics, Rouledge Publisher.

Osborne J, and Waters E. 2002. Four assumptions of multiple regression that researchers should always test. *Pract Assessment Res Evaluation*;8(2):1–8.

Peter McCullagh and John A. Nelder, Generalized Linear Models. Second edition, Chapman and Hall, London, 1989

Petrokofsky G, Kanamaru H, Achard F, Goetz SJ, Joosten H, Holmgren P, et al. 2012. Comparison of methods for measuring and assessing carbon stocks and carbon stock changes in terrestrial carbon pools. How do the accuracy and precision of current methods compare? A systematic review protocol. *Environmental Evidence*.1:1–22.

Polatin, P. F., Sarabandi, K. and Ulaby, F. T., 1994. An iterative inversion algorithm with application to the polarimetric radar response of vegetation canopies. *IEEE Trans. Geosci. Remost Sensing 32*, pp. 62–71.

Pilli R., Anfodillo T., and Carrer M., 2006. Towards a functional and simplified allometrry for estimating forest biomass*. For. Ecol. Manage. 237: 583–593.*

Pretzscha, H., Peter Bibera, Enno Uhla, Jens Dahlhausena, Thomas Rötzera, Juan Caldenteyb, Takayoshi Koikec, Tran van Cond, Aurélia Chavannee, Thomas Seifert f., Ben du Toit f. Craig Farndeng, Stephan Pauleith. 2015. Crown size and growing space requirement of common tree species in urban centres, parks, and forests, *Urban Forestry & Urban Greening* 14, 466–479

Rignot, E., Way, J. B., Williams, C. and Viereck, L., 1994. Radar estimates of aboveground biomass in boreal forests of interior alaska. IEEE Trans. *Geosci. Remost Sensing 32*, pp. 1117–1124.

Sanquetta C. R. Jaime Wojciechowski, Ana P. Dalla Corte†, Alexandre Behling†, Sylvio Péllico Netto†, Aurélio L. Rodrigues*†and Mateus N. I. Sanquetta†. 2015. Comparison of data mining and allometric model in estimation of tree biomass. *BMC Bioinformatics*, 16:247.

Suhartono, 2010. Analisis Data Statistik dengan R, Lab Statistika.

Ulrich, J. 2010, December 14). Why Use R?. Retrieved from http://www.r-bloggers.com/why-use-r/

Usman, H., & Sobari, N. (2013). *Aplikasi Teknik Multivariate Untuk Riset Pemasaran.* Jakarta: PT Grafindo Persada.

Whitten, A., Anwar, J., Damanik, S., and Hisyam, N. 1984. The Ecology of Sumatra**,** Oxford University Press, 512 pp.

Zulfikar. 2010. Kernel Order Tinggi untuk Estimasi *Value at Risk* (VaR) Manajemen Resiko Tenaga Kerja, *Proseding Seminar Nasional Manajemen Teknologi XII*, MMT ITS, Surabaya. Hal ; C .1 – 6.

GLOSARIUM

Algoritma	:	proses atau serangkaian aturan yang harus diikuti dalam perhitungan atau operasi pemecahan masalah lainnya, terutama oleh komputer. Dengan kata lain, semua susunan logis yang diurutkan berdasarkan sistematika tertentu dan digunakan untuk memecahkan suatu masalah.
Allometrik	:	Suatu fungsi atau persamaan matematika yang menunjukkan hubungan antara bagian tertentu dari makhluk hidup dengan bagian lain atau fungsi teretntu dari makhluk hidup tersebut.
Assimtotik	:	metode untuk menggambarkan perilaku yang membatasi dan memiliki aplikasi lintas ilmu mulai dari matematika terapan hingga mekanika statistik hingga ilmu komputer. Istilah asimtotik itu sendiri mengacu pada pendekatan nilai atau kurva secara sewenang-wenang saat beberapa batasan diambil.
Bandwidth	:	merupakan lingkaran dengan radius (b) dari titik pusat lokasi yang digunakan sebgai dasar penentuan bobot setiap pengamatan terhadap model regresi pada lokasi tersebut.
Biomassa	:	massa organisme biologis hidup di suatu area atau ekosistem pada suatu ketika tertentu. Biomassa pada ekologi mampu mengacu pada biomassa spesies, yang adalah massa dari satu atau semakin spesies, atau biomassa komunitas yang adalah massa dari seluruh spesies pada suatu komunitas.
Clustering	:	metode pengelompokan data. Clustering merupakan proses partisi satu set objek data ke dalam himpunan bagian yang disebut dengan cluster. Objek yang di dalam cluster memiliki kemiripan karakteristik antar satu sama lainnya dan berbeda dengan cluster yang lain.
Dendrometrik	:	cabang botani yang berkaitan dengan

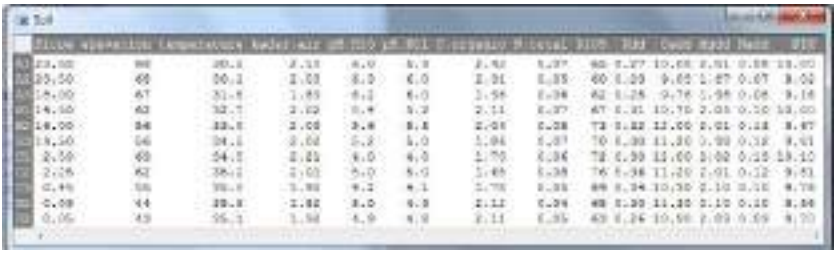
		pengukuran berbagai dimensi pohon, seperti diameter, ukuran, bentuk, usia, volume keseluruhan, ketebalan kulit kayu, dll.,
Estimator Kernel	:	Pengembangan dari estimator histogram. Estimator ini merupakan estimator linier yang mirip dengan estimator regresi nonparametrik yang lain, perbedaannya hanya karena estimator kernel lebih khusus dalam penggunaan metode bandwidth (Eubank, 1999).
Kerapatan jenis	:	disebut juga dengan istilah rapat massa adalah perbandingan antara massa suatu zat dengan vulumenya
Open access	:	salah satu cara untuk memastikan hilangnya hambatan dalam mendapatkan informasi ilmiah secara digital.
Polinomial	:	merupakan bentuk aljabar yang terdiri dari variabel, konstanta, dan eksponen (pangkat). Pangkat tertinggi suku banyak ini disebut dengan derajat dan hanya ada pada satu variabel tersebut.
Program R	:	bahasa pemrograman untuk analisis statistik yang paling banyak digunakan, karena dengan menggunakan R mampu melakukan import data dari berbagai sumber database dan format yang berbeda. R juga memiliki 7000+ packages yang gratis untuk digunakan.
R. Commander	:	antarmuka pengguna grafis (GUI) untuk Bahasa Pemrograman R yang bisa dipakai untuk analisis statistika dengan jenis lisensi GNU.
R. Console	:	adalah jendela untuk mengeksekusi perintah dari script R yang dibuat.
Regresi kernel	:	teknik statistik nonparametrik untuk mengestimasi nilai E(Y|X) = m(X) atau dalam suatu variabel.
Regresi nonparametrik	:	merupakan pendekatan metode regresi dimana bentuk kurva dari fungsi regresinya tidak diketahui. Dalam regresi nonparametrik kurva regresi hanya

diasumsikan mulus (smooth) dalam arti termuat dalam suatu ruang fungsi tertentu sehingga mempunyai sifat fleksibilitas yang tinggi (Winarti & Sony, 2010).

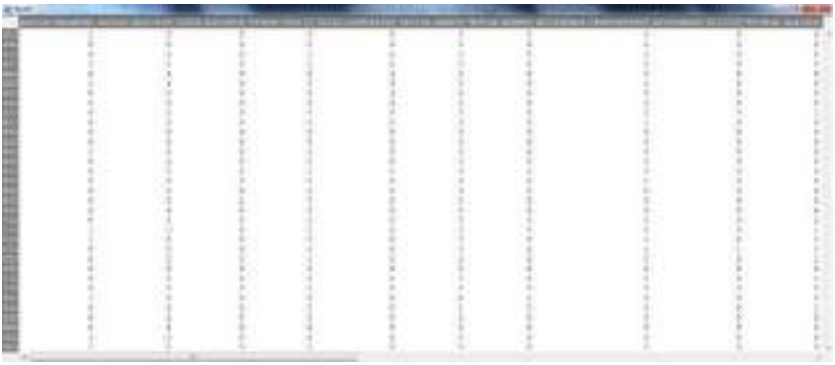
INDEKS

LAMPIRAN

Lampiran 1. Data Hasil Analisis Tanah

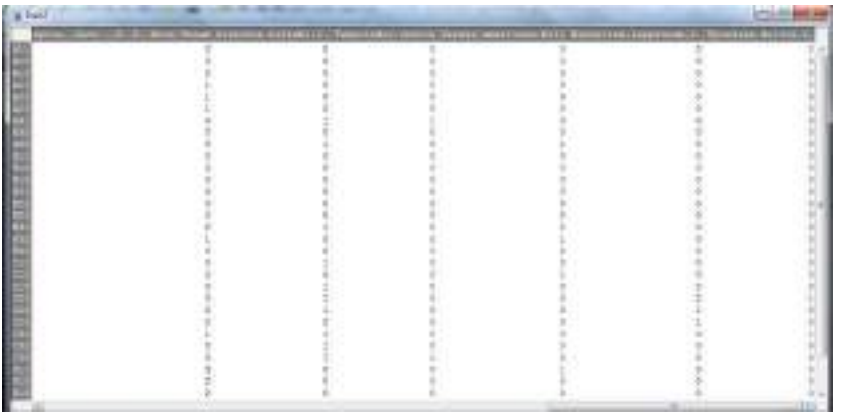


Lampiran 2. Data Spesies Tanaman Buah (a)

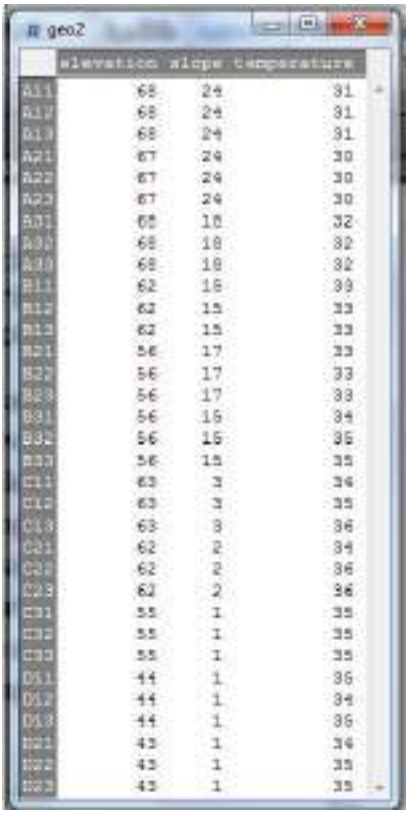


Lampiran 3. Data Spesies Tanaman Buah (b)

Lampiran 4. Data Spesies Tanaman Buah ©



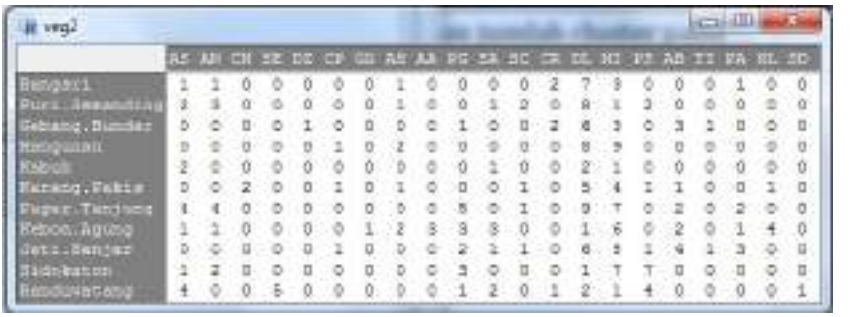
Lampiran 5. Data Pengamatan Geografis



geo2

	elevation	slope	temperature
A11	68	24	31
A12	68	24	31
A13	68	24	31
A21	67	24	30
A22	67	24	30
A23	67	24	30
A31	68	18	32
A32	68	18	32
A33	68	18	32
B11	62	16	33
B12	62	15	33
B13	62	15	33
B21	56	17	33
B22	56	17	33
B23	56	17	33
B31	56	16	34
B32	56	16	35
B33	56	15	35
C11	63	3	34
C12	63	3	35
C13	63	3	36
C21	62	2	34
C22	62	2	36
C23	62	2	36
C31	55	1	35
C32	55	1	35
C33	55	1	35
D11	44	1	36
D12	44	1	34
D13	44	1	36
D21	43	1	34
D22	43	1	35
D23	43	1	35

Lampiran 6. Spesies Tanaman Buah



Lampiran 7. Data Lingkungan Geografis, dan karakteristik tanah.

BIOGRAFI PENULIS

Zulfikar, SP. M.Si adalah dosen tetap pada Universitas KH. A. Wahab Hasbullah (Unwaha) Jombang. Lahir di Sidoarjo, Jawa Timur 24 Nopember 1968, Sarjana Agronomi dioperoleh dari Universitas Syiah Kuala, Banda Aceh, Nanggro Aceh Darussalam (1995) dan Magister Statistika dari Institut Teknologi Sepuluh Nopember (ITS) Surabaya (2005). Program Doktor Biologi, konsertrasi pada Biomodeling di Universitas Brawijaya, Malang (2022).

Pengalaman kerja dimulai dari dosen STMIK Bahrul 'Ulum tahun 2005 sampai tahun 2013 dan dosen STAI Bahrul 'Ulum tahun 2006 sampai tahun 2012. Sebagai dosen tetap pada Fakultas Teknologi Informasi Universitas KH. A. Wahab Hasbullah mulai tahun 2013 sampai sekarang. Pernah dipercaya sebagai ketua LPPM STMIK Bahrul 'Ulum (2005-2009), pemimpin redaksi jurnal ilmiah SAINTEKBU tahun 2009-2013. Menjabat sebagai pembantu ketua bidang akademik STMIK Bahrul 'Ulum (2007-2009) dan Pembantu Ketua bidang kemahasiswaan dan kerja sama tahun 2009-2010. Menjabat sebagai Dekan Fakultas Pertanian, Unwaha 2014 sampai sekarang.

Pengalaman sebagai fasilisator pada workshop Komputasi Statistik dan Manufaktur di LPPM STMIK Bahrul 'Ulum tahun 2006, fasilisator pelatihan penulisan karya tulis ilmiah untuk guru dan dosen sewilayah Jombang tahun 2007. Aktif dalam berbagai workshop dan pelatihan di antaranya workshop Pemrograman Computer Statistic di ITS tahun 2003, pelatihan Penyusunan Proposal Pengabdian Masyarakat Bagi Dosen Swasta Kopertis wilayah VII Jawa Timur (2008), workshop Penyusunan Bahan Ajar Dan Penelitian Tindakan Kelas (PTK) Depag Kabupaten Mojokerto (2009), Workshop Pengembangan Insfrastruktur PTNU Dirjen Dikti dan Dirjen Pendis di Unipdu Jombang (2009), Kegiatan *Capacity Building* Peningkatan Mutu Perguruan Tinggi di Jawa Timur (2009) oleh Dinas Pendidikan Pemerintahan Provinsi Jawa Timur, pelatihan *Applied Approach* (AA) angkatan V Kopertis VII Jawa Timur (2010), *Sort Course* Metodologi Penelitian Kuantitatif angkatan II Kemenag RI (2010), Workshop Penyusunan Buku Ajar bagi dosen PTS oleh Kopertis VII Jawa Timur (2011). Mengikuti program SSSV tahun 2018, kolaborasi Riset bidang Bioteknologi dengan Agriculture Faculty of Yamaghuci University, Jepang.

Mendapatkan hibah penelitian Kompetitif Individual bidang Eksakta dari Diktis Kemenag RI tahun 2010, penelitian Kompetitif Kolektif bidang Penelitian Sosial Keagamaan tahun 2011. Penelitian Dosen Pemula dari Ristek Dikti tahun 2012. Mendapatkan Hibah penulisan buku ajar dari Ristek Dikti tahun 2013. Hibah penulisan

Penelitian Non Disertasi dan Non Thesis (PPNDT) Kemenag RI tahun 2015. Tahun 2020 mendapatkan Hibah Penelitian Terapan (Multi Years 2021-2022) dari Ristek Dikti.

Memiliki publikasi artikel riset terindeks Scopus, dan jurnal bereputasi bidang pengabdian kepada masyarakat. Menulis buku diantaranya: Modul Komputerisasi Statistik (2008), modul Simulasi dan Pemodelan (2009), modul Statistika Dasar (2009) modul Statistika II (2010) dan modul Metodologi Penelitian (2011), juga memulis beberapa makalah ilmiah pada jurnal ilmiah Saintekbu, serta pada prosiding seminar nasional MMT ITS tahun 2010 dan tahun 2012.

Menulis buku Manajemen Riset dengan Pendekatan Komputasi Statistika (2014), Pengantar Pasar Modal dengan Pendekatan Statistika (2016), dan Pemanfaatan Limbah Jerami dengan Sistem Bioteknologi Probiotik sebagai Upaya Peningkatan Pendapatan Petani Cabai Merah (2019), Komputasi Statistika Pendidikan (2020), Budidaya Kangkung Darat (2020), Krupuk Tape Singkong (2020), Kripik Pisang (*Musa paradisiacal*, L.) Produksi Industri Rumah Tangga Binaan CSR PT. Petrokimia Gresik (2020).

Munawarah, S.Kom. M.Si, adalah dosen tetap pada Universitas KH. A. Wahab Hasbullah (Unwaha) Jombang. Lahir di Jakarta, 20 Oktober 1978, Sarjana Komputer dioperoleh dari Sekolah Tinggi Manajemen Informatika dan Komputer Jakarta (STMI&K)(1999) dan Magister Sains Program Pascasarjana Universitas Darul Ulum Jombang (2013).

Pengalaman kerja dimulai dari dosen STMIK Bahrul 'Ulum tahun 2005 sampai tahun 2013 dan dosen tetap di Fakultas teknologi Informasi Universitas KH. A. Wahab hasbullah mulai tahun 2013 sampai sekarang. Telah mempublikasikan hasil penelitian dalam jurnal terakreditasi dengan judul Mobilisasi penyebaran Informasi kampus berbasis Firebase Cloud Message (FCM) pada jurnal SAINTEKBU tahun 2019, 11, 1, 2541-1942. Judul artikel Penerapan game Edukasi 'Speak English' pada Sekolah Dasar Menggunakan Teknologi Speech Recognition pada jurnal SAINTEKBU 2019, 11,2, 2541-192.

Ambar Susanti, S.P. M.P, adalah dosen tetap pada Universitas KH. A. Wahab Hasbullah (Unwaha) Jombang. Lahir di Jakarta, 14 Oktober 1975, Sarjana Pertanian dioperoleh dari Universitas Negeri Jember (Unej)(1999) dan Magister Pertanian Program Pascasarjana Universitas Negeri Jember (2013).

Pengalaman kerja dimulai dari dosen tetap di Fakultas Pertanian Universitas KH. A. Wahab hasbullah mulai tahun 2013 sampai sekarang. Dipercaya sebagai Ketua program Studi Agribisnis mulai tahun 2013 sampai sekarang. Telah menerbitkan buku dengan judul, Manajemen Tanaman Perkebunan Menuju Komditi Sehat (2018), Pemanfaatan dan Teknis Praktis Perbanyakan Agens Hayati (Manajemen Pengelolaan Hama dan Penyakit Terpadu)(2019), dan Peranan Mikoriza untuk pengendalian Penyakit Karat daun Kedelai (2020). Memiliki 3 sertifikat HaKi dari penulisan buku yang diterbitkan oleh penerbit Fakultas Pertanian Unwaha.

Telah mempublikasikan hasil penelitian dalam jurnal terakreditasi dengan judul Mobilisasi penyebaran Informasi kampus berbasis Firebase Cloud Message (FCM) pada jurnal SAINTEKBU tahun 2019, 11, 1, 2541-1942. Judul artikel Penerapan game Edukasi 'Speak English' pada Sekolah Dasar Menggunakan Teknologi Speech Recognition pada jurnal SAINTEKBU 2019, 11,2, 2541-192.

STATISTICAL COMPUTING DENGAN PROGRAM R

Seringkali periset mengalami kesulitan dalam melakukan komputasi data risetnya, sehingga mereka terkadang mengandalkan aplikasi-aplikasi statistika yang lisensinya cukup mahal untuk dibeli. Bahkan jalan pintas sering dipakai dengan menggunakan software-software bajakan yang akibatnya kualitas risetnya menjadi rendah sehingga publikasi risetnya sulit diterima di jurnal internasional bereputasi. Sebagai upaya untuk mengatasi mahalnya software statistika maka alternatifnya adalah menggunakan software statistika open access, yaitu software yang free tanpa membeli lisensi. Salah satu software statistika yang terkenal adalah program R, dimana software ini bisa didownload bebas. Keunggulan software statistika ini memiliki paket aplikasi yang komplit, memuat berbagai model analisis dari analisis statistika sederhana sampai pada tingkat analisis statistika untuk data-data yang memiliki banyak variabel (multivariate).Keunggulan software statistika ini terkoneksi Cloud sehingga mampu mengupdate paket program sewaktu-waktu terhadap paket-paket analisis statistika yang ingin digunakan periset. Namun masih banyak periset beranggapan aplikasi program R cukup rumit, karena menggunakan coding yang harus dibaca dalam R.Program R memiliki sistem coding, sehingga pengguna bisa bebas berkreasi baik untuk fungsi statistikanya maupun tampilan visual yang dihasilkan dari analisis data. Hal ini menjadikan output visual yang dihasilkan program R sangat menarik dan cukup beragam, bahkan mampu menampilan sisi lain fungsi grafik secara detail. Sebagai upaya untuk memudahkan periset dalam mempelajari aplikasi sortware R ini makan disusunkan buku ini, dimana materi yang dikandung dalam buku ini mampu menjelaskan secara mendetail tahapan operasi program R.

Penerbit Fakultas Pertanian
Universitas KH. A. Wahab Hasbullah
Jln. Garuda 09 Tambakberas Jombang