

MAKALAH
APLIKASI R
MATA KULIAH STATISTIKA DAN PROBABILITAS



DISUSUN OLEH KELOMPOK 8

Miskun	240401010368
Mochamad Noor Syamsu	240401010302
Nuri Dwi Wulandari	240401010334
Muhammad Fahri Mujahid	240401010312
Salman Muhammad Farras Al Farid	240401010356
Aldyarisnandar	240401010362

PJJ INFORMATIKA - IF202
UNIVERSITAS SIBER ASIA
2024 / 2025

DAFTAR ISI

BAB I.....	3
R LANGUAGE.....	3
A. Pendahuluan.....	3
1. Apa itu R.....	3
2. Kelebihan R.....	3
3. Kekurangan RStudio.....	4
4. Fungsi RStudio.....	4
5. Sejarah RStudio.....	5
B. SYNTAX.....	5
1. Variable.....	5
2. Aritmatika Dasar.....	6
3. Data.....	8
4. Fungsi.....	11
BAB II.....	15
APLIKASI R.....	15
A. PENGAPLIKASIAN R.....	15
BAB III.....	21
KESIMPULAN.....	21
DAFTAR PUSTAKA.....	22
LAMPIRAN I.....	23

BAB I

R LANGUAGE

A. Pendahuluan

1. Apa itu R

R adalah bahasa untuk komputasi statistik dan grafis yang merupakan proyek GNU yang mirip dengan bahasa dan lingkungan S yang dikembangkan oleh Bell Laboratories (sebelumnya AT&T, sekarang Lucent Technologies) John Chambers dan koleganya. R memiliki beberapa perbedaan dengan S tetapi ada banyak kode pada S yang bisa berjalan di R tanpa harus dirubah.

R menyediakan berbagai macam statistik (pemodelan linier dan non-linier, tes statistik klasik, analisis deret waktu, klasifikasi, pengelompokan), serta teknik grafis.

R termasuk free software environment untuk komputasi statistik dan grafis yang mencakup bahasa pemrograman dan domain-specific untuk manipulasi data analisis statistik dan visualisasi. Terdapat lingkungan *runtime* dengan *debugger* interaktif untuk grafik. Selain itu, ekosistem paket (package) yang sangat luas di CRAN (Comprehensive R Archive Network).

R dirancang sebagai bahasa pemrograman tingkat tinggi yang mendukung bentuk analisis data yang kompleks serta pengembangan metodologi analisis data baru. Dalam beberapa tahun terakhir R telah menjadi kerangka kerja utama untuk menyediakan akses ke metodologi statistika baru dan ribuan paket ekstensi tersedia saat ini. Saat ini R banyak digunakan dalam pelatihan statistik untuk tingkat apapun, dan juga digunakan dalam mata pelajaran teoritis dan terapan. Selain itu, R juga menjadi alat yang penting untuk setiap penelitian yang melibatkan segala jenis analisis atau visualisasi data.

2. Kelebihan RStudio

Antarmuka yang ramah untuk pengguna baru dan pemula. RStudio dirancang untuk membuat bahasa R jauh lebih mudah diakses dan digunakan oleh pengguna baru. Hal ini tentu sangat baik karena pengguna, terutama yang baru pertama kali menggunakan aplikasi ini, akan lebih dimudahkan dalam memahami cara menggunakan R. Hal ini dapat membuat waktu pengerjaan menjadi lebih efisien karena pengguna baru tidak membutuhkan waktu lama untuk mempelajari dasar-dasar penggunaan aplikasi R ini.

Lingkungan pengembangan yang terintegrasi, RStudio menggunakan editor skrip, konsol, manajemen paket, panel riwayat, dan penampil grafik dalam satu jendela, sehingga pengguna dapat menulis, menjalankan, dan melakukan debugging kode.

RStudio sangat mendukung reproduibilitas penelitian melalui integrasi dengan R Markdown, vignette, dan proyek RStudio, pengguna dapat membuat dokumen analisis yang dapat dijalankan ulang dan dikemas sebagai paket R untuk membangun workflow penelitian yang terstruktur dan transparan.

Dalam RStudio juga tersedia fasilitas analisis yang interaktif mulai dari eksplorasi data interaktif, pembuatan grafik, hingga otomatisasi pembuatan laporan (pdf/word/HTML) dapat dilakukan dalam satu program, ini mempercepat siklus pengembangan dan publikasi dan juga terdapat fasilitas untuk dokumentasi.

3. Kekurangan RStudio

Menurut Bertin & BAumer (2020) Meski RStudio menawarkan banyak fitur untuk reproducible research ada ‘significant behavioral and technical challenges’ yang menghambat implementasi workflow yang konsisten, misalnya, kesalahan konfigurasi proyek, ketergantungan paket yang tidak sistematis, dan perbedaan setup antar pengguna.

Kesalahan directori kerja (working directory misconfiguration) dalam studi reproducibility as a service (2023) peneliti mengidentifikasi bahwa incorrect working directory adalah salah satu kesalahan paling umum kedua setelah missing library. Saat script R mengasumsikan path tertentu tetapi working directory tidak diatur dengan benar, misal karena proyek dibuka tanpa Rproj maka akan muncul error “file not found” mengganggu reproducibility dan workflow otomatisasi.

Kebutuhan sumberdaya komputasi yang tinggi untuk integrasi AI, pada paper terbitan tahun ini mencatat bahwa penggunaan RStudio untuk beban analisis besar misalnya training model atau memproses dataset skala besar menuntut sumber daya komputasi yang signifikan dan sering kali menimbulkan lag atau bahkan crash jika RAM atau CPU tidak memadai.

Kurva belajar yang curam bagi pemula meskipun bukan kelemahan unik RStudio sejumlah artikel menegaskan bahwa pengguna yang belum terbiasa dengan paradigma statistik dan struktur proyek di RStudio akan mengalami steep learning curve terutama dalam memahami environment workspace dan integrasi paket sehingga onboarding bisa terasa menantang.

4. Fungsi RStudio

Digunakan dalam lingkungan belajar statistik yang interaktif. RStudio, terutama lewat RStudio cloud digunakan untuk menyusun buku ajar dan materi daring interaktif yang menggabungkan teks kode R dan visualisasinya. Study oleh tierney et al melaporkan bahwa integrasi RStudio cloud dalam pengajaran statistik meningkatkan pemahaman konsep dan keterlibatan mahasiswa karena semua komponen (skrip, output, dan dokumentasi) dapat diakses dari browser tanpa hambatan instalasi.

Dukungan penuh untuk Reproducible Research melalui integrasi R Markdown, Git, dan renv, RStudio memfasilitasi pembuatan dokumen dan proyek R yang dapat dijalankan ulang secara konsisten. Sebuah tutorial MDPI (2023) Menekankan bagaimana paket Repro dijalankan dalam RStudio, membimbing peneliti dalam menyiapkan alur kerja reproducible, mulai dari pengelolaan dependency hingga mengeksekusi skrip yang terotomasi.

Pembuatan tutorial interaktif dengan learnr, paket learnr yang terintegrasi erat dengan RStudio IDE, memungkinkan konversi dokumen R Markdown menjadi modul pembelajaran interaktif. Menurut artikel di The R Journal (2022), learnr menyediakan sandbox code yang dapat diedit dan dijalankan langsung oleh pengguna, sehingga metode literasi programming menjadi lebih menarik dan efektif untuk self-paced learning.

Skalabilitas pengajaran dan kolaborasi massal, RStudio dirancang untuk lingkungan kelas besar maupun workshop, memperbolehkan instruktur mengelola workspace memberikan akses berbeda untuk instruktur, asisten dan peserta. Serta memantau kemajuan secara real-time. Posit(2022) melaporkan bahwa platform ini menyediakan set up khusus data science skala besar, mengurangi kebutuhan dukungan TI, dan mempercepat proses distribusi materi.

5. Sejarah RStudio

RStudio bermula sebagai proyek sumber terbuka yang diprakarsai oleh J. J. Allaire—sosok yang sebelumnya dikenal lewat ColdFusion—pada tahun 2009. Visi Allaire adalah menyatukan berbagai komponen R (console, editor, grafik, help) ke dalam satu antarmuka yang mulus dan produktif. Seiring waktu, ia merekrut Joe Cheng, yang kemudian mengembangkan kerangka kerja Shiny untuk aplikasi web interaktif di R, memperkaya ekosistem RStudio dengan kemampuan pembuatan aplikasi langsung dari kode statistik.

Pada Desember 2010, kode dasar ide mulai ditulis, dan pada 28 Februari 2011 RStudio merilis versi beta publik pertamanya (v0.92). Setelah menyempurnakan fitur-fitur inti selama beberapa tahun, tim memperkenalkan RStudio 1.0 pada 1 November 2016, diikuti oleh pembaruan besar 1.1 pada 9 Oktober 2017. Setiap rilis tak hanya menambah stabilitas, tetapi juga memajukan integrasi alat seperti Git, R Markdown, dan manajemen paket.

Sejak 2012, RStudio kian tumbuh menjadi pusat inovasi: selain Shiny, tokoh-tokoh seperti Hadley Wickham, Winston Chang, Garrett Grolemund, dan Yihui Xie bergabung, membawa paket populer (ggplot2, lubridate, knitr) dan memperkuat fondasi RStudio sebagai IDE sekaligus platform pengembangan paket. Hadirnya tim inti ini mendorong RStudio menjadi lebih dari sekadar editor—melainkan ekosistem bagi literate programming dan reproducible research.

Memasuki era baru pada Juli 2022, RStudio Inc. berganti nama menjadi Posit PBC, menandai perluasan fokus mereka melampaui R—misalnya ke Python dan format dokumen Quarto. Langkah ini memperlihatkan ambisi untuk menyediakan alat data science yang lebih inklusif, sambil tetap menghormati akar mereka sebagai pionir IDE untuk R.

B. SYNTAX

1. Variable

Distribusi Probabilitas (Probability distributions): Menyediakan daftar fungsi distribusi statistik di R yang diawali dengan prefix 'd', 'p', 'q', 'r' untuk density, cumulative, quantile, dan simulasi. Berisi penjelasan tentang penggunaannya, contoh penggunaan, dan distribusi umum seperti normal, binomial, poisson, gamma, dan distribusi lain.

Analisis Distribusi Data dan Uji Statistik (Examining the distribution of a set of data): Membahas cara mengeksplorasi distribusi data univariat dengan fungsi summary(), fivenum(), serta membuat histogram, plot density, fungsi distribusi kumulatif empiris (ecdf()), dan Q-Q plot. Selain itu, dilengkapi dengan uji normalitas seperti Shapiro-Wilk dan Kolmogorov-Smirnov.

Pengujian dan Perbandingan Dua Sampel (One- and two-sample tests): Penggunaan uji t, uji Wilcoxon, dan Kolmogorov-Smirnov untuk membandingkan dua sampel, termasuk pengujian varians dan analisis visual seperti boxplot dan diagram CDF empiris.

Pengelompokan, Loop, dan Kondisional (Grouping, loops and conditional execution): Menjelaskan tentang struktur kontrol di R seperti if, else, for, while, dan repeat, serta penggunaan fungsi seperti split(), abline(), dan plot() untuk analisis dan visualisasi.

Penulisan Fungsi Pengguna (Writing your own functions): Memaparkan cara mendefinisikan fungsi baru dengan function(), termasuk contoh seperti

fungsi t-statistik dan operator binary baru. Dijelaskan juga tentang argumen default, penggunaan ... untuk argumen lanjutan, serta scope variabel dan lexical scoping yang berpengaruh pada fungsi di R.

Model Statistik di R (Statistical models in R): Mengulas pembuatan model regresi linier dengan `lm()`, analisis varians (`aov()`), dan fungsi-fungsi untuk ekstraksi dan penilaian model seperti `summary()`, `anova()`, `predict()`. Termasuk juga model regresi umum lewat `glm()` dan distribusi statistik terkait.

Grafik dan Prosedur Visualisasi (Graphical procedures): Meliputi fungsi grafis tinggi dan rendah seperti `plot()`, `hist()`, serta fitur-fitur grafis rendah seperti penambahan titik, garis, teks, dan pengaturan parameter grafis melalui `par()`. Juga penjelasan tentang grafis interaktif dan pengaturan posisi figure dan layout halaman.

Penggunaan Driver Grafik dan Device Drivers: Menjelaskan tentang memulai perangkat grafis seperti `X11()`, `windows()`, `quartz()`, serta menyimpan gambar ke file menggunakan `postscript()`, `png()`, `jpeg()`, dan pengelolaan banyak perangkat grafis secara bersamaan.

Fasilitas Sistem dan OT (OS facilities): Menjawab manipulasi file dan direktori dengan fungsi seperti `file.create()`, `file.remove()`, `list.files()`, serta mengelola jalur file dan izin akses dengan fungsi terkait. Tambahan fitur menjalankan perintah sistem melalui `system()`, `system2()`, dan menjalankan file skrip R secara otomatis dengan `Rscript`.

2. Aritmatika Dasar

a. Vektor

Vektor disimpan seperti array dalam C, secara berurutan, dan dengan demikian Anda tidak dapat memasukkan atau menghapus elemen, yaitu sesuatu yang mungkin biasa Anda lakukan jika Anda seorang programmer Python. Ukuran vektor ditentukan pada saat pembuatannya, oleh karena itu, jadi jika Anda ingin menambahkan atau menghapus elemen, Anda harus menetapkan ulang vektor tersebut. Misalnya, mari tambahkan elemen di tengah vektor empat elemen: `> x <- c(88,5,12,13) > x <- c(x[1:3],168,x[4])` # masukkan 168 sebelum 13 `> x [1] 88 5 12 168 13` Di sini, kami membuat vektor empat elemen dan menetakannya ke x. Untuk memasukkan angka baru 168 di antara elemen ketiga dan keempat, kami merangkai tiga elemen pertama x, lalu 168, lalu elemen keempat x. Ini menciptakan vektor lima elemen baru, yang membiarkan x tetap utuh untuk sementara waktu. Kami kemudian menetapkan vektor baru itu ke x. Hasilnya, tampak seolah-olah kami benar-benar telah mengubah vektor yang disimpan di x, tetapi sebenarnya kami membuat vektor baru dan menyimpan vektor itu di x. Perbedaan ini mungkin tampak halus, tetapi memiliki implikasi. Misalnya, dalam beberapa kasus, panjang vektor dengan menggunakan fungsi `length()`: `> x <- c(1,2,4) > length(x) [1] 3`.

Dalam contoh ini, kita sudah mengetahui panjang x, jadi tidak perlu menanyakannya. Namun, dalam penulisan kode fungsi umum, Perlu mengetahui panjang argumen vektor. Misalnya, anggaplah kita ingin memiliki fungsi yang menentukan indeks 1 nilai pertama dalam argumen vektor fungsi (dengan asumsi kita yakin ada nilai tersebut). Berikut ini adalah salah satu cara (yang tidak selalu efisien) untuk menulis kode: `first1 <- function(x) {for (i in 1:length(x)) {if (x[i] == 1) break # break out of loop return(i)`

Tanpa fungsi `length()`, kita perlu menambahkan argumen kedua ke `first1()`, misalnya dengan menamakannya `n`, untuk menentukan panjang `x`. Dalam kasus ini, penulisan loop sebagai berikut tidak akan berhasil: `for (n in x)`.

Pendekatan ini bermasalah karena tidak memungkinkan kita untuk mengambil indeks elemen yang diinginkan. Jadi, kita memerlukan loop eksplisit, yang pada gilirannya memerlukan penghitungan panjang `x`.

Satu hal lagi tentang loop tersebut: Untuk pengodean yang cermat, kita harus khawatir bahwa `length(x)` mungkin bernilai 0. Dalam kasus seperti itu, lihat apa yang terjadi pada ekspresi `1:length(x)` dalam loop `for` kita: `> x <- c()
> x NUL askssssss > length(x) [1] 0 > 1:length(x)`

Matrix (matriks) merupakan kumpulan data yang memiliki baris dan kolom (`i, j`), dimana `i` adalah baris dan `j` adalah kolom. Isi di dalam suatu matriks disebut elemen atau anggota. Dalam suatu matriks hanya dapat memuat 1 jenis tipe data. Untuk membuat suatu matriks dapat dibuat dengan perintah `matrix()`.

Selain secara manual dengan ilmu matematika atau perhitungan, matriks dapat juga diselesaikan dengan program R. Dengan memanfaatkan program R, seseorang dapat menyelesaikan suatu kasus matematika khususnya matriks, dengan lebih mudah dan cepat. Banyak kasus matematika ataupun ilmu lainnya yang membutuhkan penyelesaian melalui perhitungan matriks.

b. Membuat Matriks

Kita juga dapat melakukan operasi matematika pada matriks. Pada operasi matematika pada matriks proses yang terjadi bisa lebih kompleks dibanding pada vektor, dimana kita dapat melakukan operasi untuk memperoleh gambaran data pada tiap kolom atau baris.

Berikut adalah contoh operasi matematika sederhana pada matriks :

```
# mengalikan masing-masing elemen matriks dengan 2
my_data*2
##      col1 col2 col3
## row1  10   4  14
## row2  12   8   6
## row3  14  10   8
## row4  16  18  16
## row5  18  16  14
# memperoleh nilai log basis 2 pada masing-masing elemen matriks
log2(my_data)

##      col1 col2 col3
## row1 2.322 1.000 2.807
## row2 2.585 2.000 1.585
## row3 2.807 2.322 2.000
## row4 3.000 3.170 3.000
## row5 3.170 3.000 2.807
```

Seperti yang telah penulis jelaskan sebelumnya, kita juga dapat melakukan operasi matematika penjumlahan elemen pada tiap baris atau kolom dengan menggunakan fungsi `rowSums()` atau `colSums()`.

```
# Total pada tiap kolom
colSums(my_data)
## col1 col2 col3
## 35 28 29
# Total pada tiap baris
rowSums(my_data)
## row1 row2 row3 row4 row5
## 14 13 16 25 24
```

jika kita tertarik untuk mencari nilai rata-rata tiap baris atau kolom kita juga dapat menggunakan fungsi `rowMeans()` atau `colMeans()`. Berikut adalah contoh penerapannya :

```
# Rata-rata tiap baris
rowMeans(my_data)
## row1 row2 row3 row4 row5
## 4.667 4.333 5.333 8.333 8.000
# Rata-rata tiap kolom
colMeans(my_data)
## col1 col2 col3
## 7.0 5.6 5.8
```

c. **Logic Factor**

Dalam bahasa pemrograman R, logic operator digunakan untuk melakukan operasi logika antara nilai-nilai boolean (TRUE atau FALSE). Berikut adalah operator logika utama dalam R:

```
# Contoh operator !
!TRUE # Hasil: FALSE
# Contoh operator &
TRUE & FALSE # Hasil: FALSE
TRUE & TRUE # Hasil: TRUE
# Contoh operator |
TRUE | FALSE # Hasil: TRUE
FALSE | FALSE # Hasil: FALSE
# Contoh && dan ||
x <- 5
(x > 0) && (x < 10) # Hasil: TRUE
(x < 0) || (x > 100) # Hasil: FALSE
```

3. Data

R secara umum mengoperasikan tiga tipe data utama: **vector**, **list**, **matrices**, **factor** dan **data frame**. Melalui operasi-operasi yang diterapkan pada tipe data ini, kita dapat melakukan **transformasi**, **kombinasi**, dan **ekstraksi** data.

a. **Vector**

Vector adalah jenis data di R dengan struktur yang menyimpan deretan nilai (lebih dari satu nilai) dengan tipe data sama. Jadi, jika tipe datanya teks maka seluruh data harus bertipe teks. Demikian juga jika tipenya angka maka seluruh data angka semua.

Vector didefinisikan dengan nama function yang memiliki satu huruf saja: c. Sebagai contoh untuk membuat vector yang isinya angka dengan nilai 2, 5, dan 7, maka perintahnya adalah

```
> c(2, 5, 7).
```

Output :

```
[1] 2 5 7
```

Pembentukan vector yang memiliki deretan nilai sangat banyak dapat dipermudah dengan operator, yang digunakan untuk mempersingkat penulisan dari nilai-nilai vector yang berurutan. Sebagai contoh, perintah `c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)` yang membentuk vector dengan angka 1 sampai dengan 10 dapat dipersingkat dengan `c(1:10)`. Contoh *command* :

```
>c(1:10)
```

Output:

```
[1] 1 2 3 4 5 6 7 8 9 10
```

Tidak hanya angka, vector juga dapat memuat teks. Tapi perlu diperhatikan dalam **R**, ketika menggabungkan elemen dengan tipe berbeda (seperti angka dan text) menggunakan fungsi `c()`, **R akan memaksa semua elemen menjadi tipe yang sama**, mengikuti hierarki tipe data. Dalam kasus ini, keberadaan string karakter ("abdul", "apel") menyebabkan **semua elemen diubah menjadi tipe charat**. contoh:

```
>c(1 , "Abdul" , "Apel" , 4)
```

Output :

```
[1] "1" "Abdul" "Apel" "4"
```

Mengakses data pada vector:

- Untuk mengambil isi vector, kita perlu menggunakan posisinya. Posisi ini diwakili oleh angka urutan yang disebut sebagai **index**.
- Cara penulisan index di variable adalah angka yang diapit dengan kurung siku tunggal atau ganda. Penulisan ini selanjutnya disebut sebagai **accessor**.
- Jika index yang ingin diambil lebih dari satu, maka hanya boleh menggunakan kurung siku tunggal.

Contoh:

```
>Num <- c(11,22,33,44)
```

```
[1] 11 22 33 44
```

Untuk mengambil angka 22 maka menggunakan command

`>print(num[2])` karena angka 22 merupakan data ke-2 pada deret nilai vector "Num"

Output:

```
[1] 22
```

Selain dengan angka dan teks, indeks pada vector juga dapat dilengkapi dengan nama untuk tiap elemennya dengan menggunakan format penulisan name=value

Contoh:

`Skor <- c(tesA = 83, tesB= 92, tesC= 77)` vector ini mewakili nilai-nilai untuk skor pada tes "tesA", "tesB" dan "tesC".

Command yang digunakan `>print(skor["tesC"])`

Output :

```
fisika
```

```
95
```

b. Matrices

Matriks banyak digunakan dalam statistik, sehingga memainkan peran penting dalam R. Untuk membuat matriks, gunakan fungsi `>matrix()`, dengan menentukan elemen-elemen berdasarkan kolom terlebih dahulu.

Contoh :

```
>matrix(1 : 12, nrow = 3 , ncol = 4
```

```
      [1,] [2,] [3,] [4,]  
[1,]  1    4    7   10  
[2,]  2    5    8   11  
[3,]  3    6    9   12
```

c. List

List adalah jenis data di R yang mirip dengan vector, **perbedaannya adalah list dapat menyimpan lebih dari satu jenis data**. Untuk memasukkan isi ke dalam struktur data ini kita gunakan function list.

Contoh :

```
>list(1:10 , "Abdul")
```

Output:

```
[[1]]  
[1] 1 2 3 4 5 6 7 8 9 10  
[[2]]  
[1] "abdul"
```

Terlihat outputnya agak berbeda dengan vector - dimana tiap output terdapat dua tampilan accessor

d. Data Frame

Data frame adalah jenis struktur data yang dirancang untuk representasi table, yang terdiri dari banyak kolom dengan tiap kolom berisi list ataupun vector dengan jumlah data yang sama. Untuk membuat data frame kita bisa gunakan function `>data.frame`.

```
> buku <- data.frame(penulis=c("Salman", "Deni", "Sulthan", "Chandra"),
tahun=c(1980, 1979, 1999, 2006), publisher=c("Tirta Nusa", "Suryalaya",
"Cerdas Bangsa", "suryalaya"))
>Buku
```

	<i>Penulis</i>	<i>Tahun</i>	<i>Publisher</i>
<i>1</i>	<i>Salman</i>	<i>1980</i>	<i>Tirta Nusa</i>
<i>2</i>	<i>Deni</i>	<i>1979</i>	<i>Suryalaya</i>
<i>3</i>	<i>Sulthan</i>	<i>1999</i>	<i>Cerdas Bangsa</i>
<i>4</i>	<i>Chandra</i>	<i>2006</i>	<i>Suryalaya</i>

e. Factor

Berbeda dengan Vector, list, matrix, dan data frame yang bersifat numerikal. Factor merupakan tipe data yang digunakan untuk merepresentasikan data kategorikal, yaitu data yang terdiri dari satu set nilai terbatas dan tetap (seperti jenis kelamin, status pernikahan, atau tingkat pendidikan). Dalam R, factors sangat berguna untuk analisis statistik karena mereka menyimpan informasi tidak hanya tentang nilai data, tetapi juga tentang kategori (disebut "levels") yang mungkin.

Factors dapat bersifat ordinal (berurutan, seperti "rendah", "sedang", "tinggi") atau nominal (tidak berurutan, seperti "merah", "hijau", "biru"). R memperlakukan factors secara khusus dalam model statistik dan visualisasi data. Kita bisa membuat factor dengan fungsi `factor()`, dan menentukan urutan level jika perlu. Contoh:

```
pendidikan <- factor(c("SMA", "S1", "SMA", "S2", "S1"),
                     levels = c("SMA", "S1", "S2"),
                     ordered = TRUE)
print(pendidikan)
[1] SMA S1 SMA S2 S1
Levels: SMA < S1 < S2
```

4. Fungsi

Segala sesuatu yang dilakukan dalam R dilakukan melalui *functions*. Sebuah *function* dalam bahasa pemrograman, serupa dengan padanannya dalam matematika, memiliki sejumlah *input* yang disebut *arguments* serta satu keluaran yang disebut return value. Dalam R, hanya satu objek yang dapat dikembalikan oleh sebuah *function*. Ketika nama suatu function diketikkan di *console*, strukturnya akan ditampilkan.

Contoh kita menampilkan fungsi `setdiff(x,y)` □ `setdiff` adalah salah satu operasi set bawaan R yang mengembalikan elemen yang ada dalam satu vektor tetapi tidak ada di vektor lain

```
>Setdiff
```

```
function (x, y)
{
  if (is.null(x))
    return(NULL)
  if (.set_ops_need_as_vector(x, y)) {
    x <- as.vector(x)
    y <- as.vector(y)
  }
  x <- unique(x)
  names(x) <- NULL
  y <- unique(y)
  names(y) <- NULL
  x[match(x, y, 0L) == 0L]
}
<bytecode: 0x00000118ea404300>
<environment: namespace:base>
```

Kita dapat melihat bagaimana fungsi *setdiff* terbentuk , dari menerima argumen x dan y diproses hingga memberikan output berupa data yang ada pada x(args ke – 1) dan tidak ada pada y (args ke-2)

Args

Arguments sedikit rumit dalam R. Anda akan melihat bahwa *arguments* memiliki nama: *arguments* dari fungsi *setdiff*() disebut x dan y. Namun, Anda biasanya tidak perlu menyebutkan nama *arguments* secara eksplisit, karena *arguments* juga memiliki posisi

```
> args(setdiff)

function (x, y)

NULL
```

Mendeklarasikan *functions*

Untuk mendeklarasikan *functions* sendiri dapat menggunakan format yang sama yaitu

```
> (nama fungsi) <- functions (args1,args2,args3,...,Args-...){ fungsi yang
memberikan return value}
```

Contoh:

Membuat fungsi yang memberikan y kali dari rata-rata suatu vektor dalam args x

```
Deret1 <- c(1 : 10)
Fungsi1 <- function(x,y){
+   n <- mean(x) * y
+   print(n) }
```

```
Fungsi1 <- functions( Deret1 , 9 )  
[1] 49,5
```

akan mengembalikan rata-rata dari vector deret 1-10 dikalikan 9 menjadi $5,5 * 9 = 49,5$

Looping

Looping dalam pemrograman adalah teknik untuk menjalankan blok kode secara berulang hingga kondisi tertentu terpenuhi. Dalam bahasa pemrograman R, looping dapat dilakukan melalui 2 metode yaitu While() Loop dan For()Loop. Berikut syntax dari pendeklarasian functions menggunakan looping:

For()Loop

```
Deret <- c(1:5)  
Cth1 <- function(x) {  
  for (i in 1:length(vec)) {  
    vec[i] <- vec[i] + 3  
  }  
  return(vec)}
```

```
Cth1(Deret)
```

```
[1] 4  
[1] 5  
[1] 6  
[1] 7  
[1] 8
```

While() loop

```
Cth2 <- function(x) {  
  i <- 1  
  while (x[i] <= 4) {  
    print(x[i])  
    i <- i + 1  
  }  
}
```

```
deret <- c(1:10000)  
Cth2(deret)
```

```
[1] 1  
[1] 2  
[1] 3  
[1] 4
```

Conditional (IF)

Conditional statements dalam pemrograman digunakan untuk mengontrol alur eksekusi berdasarkan kondisi tertentu. Dalam bahasa pemrograman R Conditional Statement dilakukan dengan syntax sebagai berikut:

```
x <- 3
if (x > 5) {
  print("x lebih besar dari 5")
} else {
  print("x kurang dari atau sama dengan 5")
}
```

Fungsi tersebut membuat suatu conditional statements dimana console akan mengeluarkan output “x lebih besar dari 5” jika x memiliki nilai lebih besar dari 5, **else** merupakan command yang dieksekusi jika kondisi tidak memenuhi syarat. Dalam hal ini, console akan mengeluarkan output “x kurang dari atau sama dengan 5” karena $x = 3$.

BAB II APLIKASI R

A. Penggunaan Analisis Statistik Menggunakan Bahasa R

R seperti yang sudah dibahas pada chapter sebelumnya R adalah bahasa yang ditujukan untuk keperluan komputasi statistik sehingga secara bawaan paket modulnya sudah tersedia banyak paket-paket yang diperlukan untuk kebutuhan statistik antara lain sebagai berikut.

*Matrix lattice mgcv survival KernSmooth MASS base boot class cluster
codetools compiler datasets foreign grDevices graphics grid methods nlme nnet
parallel rpart spatial splines stats stats4 tcltk tools utils*

Paket-paket tersebut cukup untuk melakukan fungsi-fungsi statistik dasar, seperti linear models, classical testst, fungsi plotting atau tools survival analysis, selain itu juga banyak paket-paket lain yang dibuat oleh komunitas yang bisa dipasang secara langsung dalam R prompt, fungsi untuk menginstalnya adalah **install.packages** sebagai contoh kita menginstall package sandwich estimator.

```
R> install.packages("sandwich")
```

Paket-paket lainnya bisa ditemukan di sini, sebelum kita membahas lebih jauh tentang pengaplikasian statistik menggunakan R. Kita perlu membahas bagaimana R menangani dan memanipulasi data seperti yang kita singgung pada chapter sebelumnya tidak seperti bahasa pemrograman lain seperti C yang menyimpan variabel berdasarkan data-typenya R menyimpan variabel tersebut kedalam sebuah objek alih-alih data typenya, ini membuat R sangat fleksibel untuk analisis data dan eksplorasi cepat ada 5 tipe objek dasar pada R; **Vectors, Lists, Matrices, Factors dan Data Frames**

Makalah ini menggunakan kumpulan data dari 2000 perusahaan terkemuka di dunia, daftar Forbes 2000 untuk tahun 2004 yang dikumpulkan oleh 'Majalah Forbes'. Dan sebagai bagian dari sampel data R, data-data tersebut merupakan bagian dari HSAUR paket (Source: From Forbes.com, New York, New York, 2004. With permission.).

Langkah pertama kita sediakan datanya dulu yaitu ("Forbes2000") yang berada dalam paket HSAUR berikut caranya :

- 1) install terlebih dahulu paket HSAURnya

```
R> install.packages("HSAUR")
```
- 2) kemudian siapkan datanya dengan perintah

```
R> data("Forbes2000", package = "HSAUR")
```
- 3) dan check apakah data tersebut sudah tersedia dengan perintah

```
R> ls()
```

 hasilnya adalah list nama semua objek yang saat ini berada dalam global environment.

Sekarang data tersebut bisa kita gunakan kedepannya kita juga bisa mencetak dan melihat isi samples datanya dengan perintah

```
R> print(Forbes2000)
```

rank	name	country	category	sales	profits
assets					
1 1	Citigroup	United States	Banking	94.71	
17.85 1264.03					
2 2	General Electric	United States	Conglomerates		
134.19 15.59 626.93					
3 3	American Intl Group	United States	Insurance	76.66	
6.46 647.66					

4	4	ExxonMobil	United States	Oil & gas operations	222.88	20.96	166.99
5	5	BP	United Kingdom	Oil & gas operations	10.27	177.57	232.57
6	6	Bank of America	United States	Banking	10.81	736.45	49.01
7	7	HSBC Group	United Kingdom	Banking	44.33	6.66	757.60
8	8	Toyota Motor	Japan	Consumer durables	7.99	171.71	135.82

Hasilnya seperti contoh di atas tidak terlalu membantu dalam pengumpulan informasi awal tentang data tersebut; kita juga melihat deskripsi dari setiap strukturnya

```
R>str(Forbes2000)
```

```
'data.frame': 2000 obs. of 8 variables:
```

```
$ rank : int 1 2 3 4 5 6 7 8 9 10 ...
```

```
$ name : chr "Citigroup" "General Electric" "American Intl Group"
"ExxonMobil" ...
```

```
$ country : Factor w/ 61 levels "Africa","Australia",...: 60 60 60 60 56 60 56 28 60
60 ...
```

```
$ category : Factor w/ 27 levels "Aerospace & defense",...: 2 6 16 19 19 2 2 8 9 20 ...
```

```
$ sales : num 94.7 134.2 76.7 222.9 232.6 ...
```

```
$ profits : num 17.85 15.59 6.46 20.96 10.27 ...
```

```
$ assets : num 1264 627 648 167 178 ...
```

```
$ marketvalue: num 255 329 195 277 174 ...
```

Output dari perintahnya bisa dilihat seperti contoh di atas berisi penjelasan tentang objek Forbes2000 yang merupakan bagian dari tipe data data.frames salah data struktur yang menangani tabular statistical data di R, seperti yang terlihat berisi informasi tentang 2000 observasi perusahaan yang tersedia dalam objek ini, selain perintah str ada perintah2 lainnya

```
R> help("Forbes2000")
```

```
or
```

```
R> ?Forbes2000
```

Karena R adalah bahasa pemrograman berorientasi objek kita juga bisa melihat class nya dengan perintah

```
R> class(Forbes2000)
```

data.frames merepresentasikan data dengan cara tabel tradisional setiap barisnya berasosiasi dengan satu observasi dan setiap kolomnya responden terhadap satu variabel dimensi dari tabel tersebut juga bisa diekstrak dengan fungsi

```
R> dim(Forbes2000)
```

```
[1] 2000 8
```

Penjelasannya adalah bagian kiri adalah rowsnya,dan bagian kanan adalah kolomnya hasilnya adalah tabel forbes mempunyai 2000 rows dan 8 kolom, untuk melihat 8 variabel kolom tersebut kita bisa mengaksesnya menggunakan perintah

```
R> names(Forbes2000)
```

```
[1] "rank" "name" "country" "category" "sales" "profits" "assets"
"marketvalue"
```


Selanjutnya kita akan membahas tentang Vector yang merupakan struktur data dasar yaitu sekumpulan elemen sederhana yang semua objeknya mempunyai kelas yang sama, mirip dengan array jika di bahasa lain untuk membuat vector bisa dengan perintah berikut

```
R> c(1,2,3) [1] 1 2 3
[1] 1 2 3
```

Dalam contoh Forbes2000 sebenarnya data nama perusahaannya disimpan sebagai vector dengan tipe character bisa check dengan cara

```
R> class(Forbes2000[, "name"])
[1] "character"
```

Sementara untuk kategorikal atau kita sebut saja array asosiatif direpresentasikan dengan type factor di R seperti contohnya kolom category

```
R> class(Forbes2000[, "category"])
[1] "factor"
```

Perbedaan keduanya antara vector character dan factor adalah pada bagaimana mereka menyimpan nilainya, pada vector character disimpan sebagai character variabel sementara, pada factor variabel disimpan dalam level factor seperti berikut

```
R> nlevels(Forbes2000[, "category"])
[1] 27
```

Untuk melihat nilainya perlu kita ekstrak dengan perintah

```
R> levels(Forbes2000[, "category"])
[1] "Aerospace & defense"
[2] "Banking"
[3] "Business services & supplies"
...
```

Kita bisa melihatnya dengan format ringkasan statistik dengan perintah

```
R> table(Forbes2000[, "category"])
      Aerospace & defense      Banking      Business services & supplies
              19              313              70
      Capital goods      Chemicals      Conglomerates
              53              50              31
      Construction      Consumer durables      Diversified financials
              79              74              158
```

Sementara sisanya seperti sales, assets, profits dan nilai pasar variabel bertipe numeric, yaitu natural tipe data untuk penilaian numerical seperti contoh :

```
R> class(Forbes2000[, "sales"])
[1] "numeric"
```

Kita juga bisa melakukan statistik sederhana seperti mean, median dan range dengan perintah berikut :

```
R> mean(Forbes2000[, "sales"])
[1] 9.69701
```

Ringkasan statistik sederhana juga bisa diterapkan pada numeric vector untuk memberikan nilai ringkasan statistik yang berguna seperti max, mean, median dan kuartil 25% & 75% seperti berikut :

```
R> summary(Forbes2000[, "sales"])
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
0.010  2.018   4.365   9.697   9.547 256.330
```

Setelah melakukannya mari kita analisis, ringkasan statistik sederhana dari tabel Forbes2000 dengan perintah:

```
R> summary(Forbes2000)
```

Hasilnya adalah :

```

rank      name      country      category      sales
Min.   : 1.0 Length:2000   United States :751 Banking           : 313 Min.   :
0.010
1st Qu.: 500.8 Class :character Japan           :316 Diversified financials: 158 1st
Qu.: 2.018
Median :1000.5 Mode  :character United Kingdom:137 Insurance           : 112
Median : 4.365
Mean   :1000.5          Germany      : 65 Utilities           : 110 Mean   :
9.697
3rd Qu.:1500.2          France       : 63 Materials           : 97 3rd Qu.: 9.547
Max.   :2000.0          Canada       : 56 Oil & gas operations : 90 Max.
:256.330
              (Other)   :612 (Other)           :1120
profits      assets      marketvalue
Min.   : -25.8300 Min.   : 0.270 Min.   : 0.02
1st Qu.: 0.0800 1st Qu.: 4.025 1st Qu.: 2.72
Median : 0.2000 Median : 9.345 Median : 5.15
Mean   : 0.3811 Mean   : 34.042 Mean   : 11.88
3rd Qu.: 0.4400 3rd Qu.: 22.793 3rd Qu.: 10.60
Max.   : 20.9600 Max.   :1264.030 Max.   :328.54
NA's   :5
```

Dari output di atas kita bisa melihat bahwa sebagian besar perusahaan berada di U.S dan sebagian besar perusahaan bergerak di sektor perbankan serta keuntungan negatif atau kerugian hingga 26 miliar USD terjadi. Karena method summary merupakan fungsi umum yang bisa digunakan oleh banyak kelas, summary juga bisa diterapkan ke objek dari kelas yang berbeda dan akan tetap menghasilkan hasil yang baik sebagai contoh karena Forbes2000 adalah objek data.frames yang terdiri dari banyak variabel kita bisa menerapkan summarynya ke setiap variabel yang ada dalam data.frames tersebut sebagai contoh kita bisa gunakan perintah berikut:

```
R> lapply(Forbes2000, summary)
$rank
  Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
 1.0   500.8  1000.5  1000.5  1500.2  2000.0
```

```
$name
 Length Class  Mode
2000 character character
```

Perintah lapply membantu kita menyelesaikan tugas berulang pada setiap data.frame, matrix, list ataupun setiap tingkat faktor. Selanjutnya kita bisa membandingkan keuntungan masing-masing dari 27 kategori untuk melakukannya kita harus menghitung keuntungan rata-rata untuk setiap kategorinya dengan cara

```
R> mprofits <- tapply(Forbes2000$profits, Forbes2000$category, median, na.rm = TRUE)
```

Penjelasannya adalah untuk setiap level kategori faktor, tentukan elemen-elemen yang sesuai dari vector numerik profits dan berikan fungsi median dengan argumen tambahan na.rm = True, kenapa perlu ditambahkan na.rm = True karena jika tidak nilai yang kosong atau hilang akan menyebabkan hasil yang tidak sesuai pada fungsi median.

Selanjutnya kita bisa melihat kategori teratas dengan rata-rata keuntungan tertinggi dari hasil komputasi vector yang telah diurutkan dengan perintah:

```
R> rev(sort(mprofits))
```

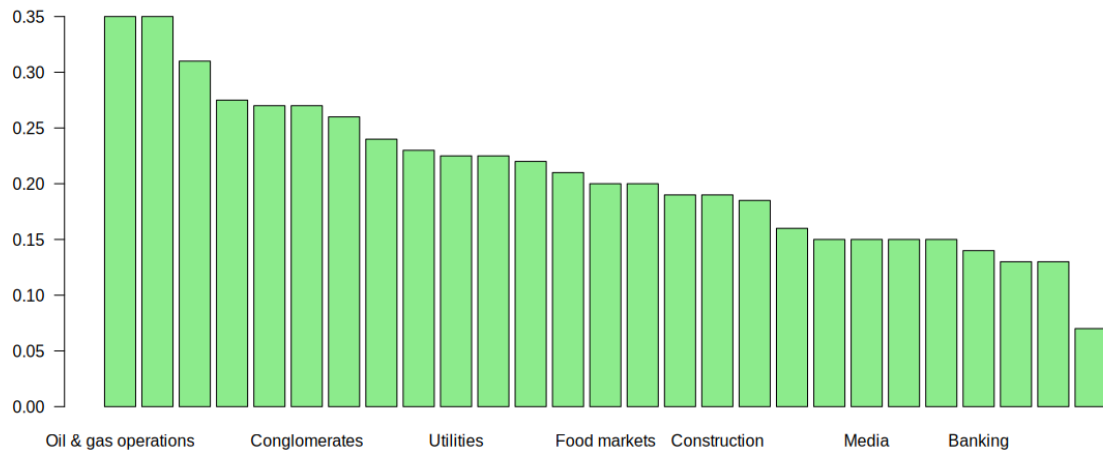
```
> rev(sort(mprofits))
```

Oil & gas operations	Drugs & biotechnology
0.350	0.350
Household & personal products	Telecommunications services
0.310	0.275
Food drink & tobacco	Conglomerates
0.270	0.270
Aerospace & defense	Software & services
0.260	0.240
Health care equipment & services	Utilities
0.230	0.225
Consumer durables	Insurance
0.225	0.220
Hotels restaurants & leisure	Food markets
0.210	0.200
Diversified financials	Retailing
0.200	0.190
Construction	Chemicals
0.190	0.185
Business services & supplies	Semiconductors
0.160	0.150
Media	Materials
0.150	0.150
Capital goods	Banking
0.150	0.140
Transportation Technology hardware & equipment	
0.130	0.130
Trading companies	
0.070	

Dimana fungsi rev berguna untuk mengurutkan vector median profetis dari yang terkecil sampai yang terbesar. Kita juga bisa mengganti median dengan apapun seperti fungsi mean dan sebagainya.

Selanjutnya agar visualisasinya lebih mudah dipahami, kita bisa membuat plot datanya berupa barchart dengan perintah :

```
> barplot(rev(sort(mprofits)), horiz = FALSE, las = 1, col = "lightgreen")
```



BAB III

KESIMPULAN

Berdasarkan pembahasan dalam makalah ini, dapat disimpulkan bahwa Bahasa R, khususnya ketika diintegrasikan dengan lingkungan pengembangan RStudio, merupakan instrumen yang sangat ampuh dan fleksibel untuk keperluan komputasi statistik, analisis data, dan visualisasi grafis. RStudio secara signifikan menurunkan hambatan awal bagi pengguna baru melalui antarmuka yang intuitif dan fitur-fitur terintegrasi yang mendukung alur kerja analisis data yang efisien, termasuk kemampuan *reproducible research* yang krusial dalam penelitian ilmiah. Meskipun makalah mengakui adanya tantangan, seperti kurva belajar yang relatif curam dan potensi kebutuhan sumber daya komputasi yang besar untuk tugas-tugas spesifik, keunggulan R dalam menyediakan beragam fungsi statistik bawaan dan akses ke ribuan paket ekstensi dari CRAN menempatkannya sebagai pilihan utama bagi para statistisi dan analis data.

Aspek fundamental dari kekuatan R terletak pada struktur sintaksis dan tipe data yang dirancangnya. Makalah ini dengan jelas menguraikan tipe data dasar seperti vector, list, matrix, factor, dan data frame, menjelaskan bagaimana data diorganisasi dan dimanipulasi di R. Pemahaman terhadap struktur data ini sangat penting karena operasi analisis data pada dasarnya melibatkan transformasi, kombinasi, dan ekstraksi informasi dari objek-objek data ini. Lebih lanjut, penjelasan mengenai dasar-dasar pemrograman seperti penggunaan fungsi (`function()`), operator logika, kontrol alur dengan *looping* (`for()`, `while()`), dan *conditional statement* (`if(){} else{}`) menunjukkan bagaimana R memungkinkan pengguna untuk membangun skrip analisis yang kompleks, mengotomasi tugas, dan menerapkan logika pengambilan keputusan dalam proses analisis data.

Bagian aplikasi R secara efektif menjembatani teori dengan praktik, menggunakan dataset Forbes2000 sebagai studi kasus. Demonstrasi langkah demi langkah, mulai dari memuat data, mengeksplorasi strukturnya, hingga melakukan analisis statistik deskriptif dasar (seperti mean, median, dan ringkasan data) serta visualisasi hasilnya dalam *barplot*, membuktikan kapabilitas R dalam menangani data nyata. Analisis ini menunjukkan bagaimana R dapat digunakan untuk mendapatkan wawasan awal dari data, mengidentifikasi pola (misalnya, kategori industri dengan keuntungan median tertinggi), dan menyajikan temuan secara visual. Oleh karena itu, kombinasi fitur statistik yang komprehensif, kemampuan manipulasi data yang kuat, dan dukungan lingkungan kerja yang efisien melalui RStudio menjadikan Bahasa R alat yang tak tergantikan dalam ekosistem data science saat ini, baik untuk tujuan pendidikan maupun profesional.

DAFTAR PUSTAKA

R: Apa itu R? <https://www.r-project.org/abbout.html>

M. Isabel Parra, Eva L. Sanjuán, M. Carmen Robustillo, Mario M. Pizarro, Menggunakan R untuk pengajaran dan penelitian, arXiv:2306.12200 (statistik)

Amy E. Pomeroy , Andrea Bixler , Stefanie H. Chen , Jennifer E. Kerr , Todd D. Levine and Elizabeth F. Ryder *; “R” U ready?: a case study using R to analyze changes in gene expression during evolution, Curriculum, Instruction, and Pedagogy PUBLISHED 22 March 2024 DOI 10.3389/feduc.2024.1379910

Joseph F. Hair Jr., G. Tomas M. Hult, Christian M. Ringle, Marko Sarstedt, Nicholas P. Danks , Soumya Ray Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R

Matti Vuorre, Matthew J C Crump, Sharing and organizing research products as R packages, 2020 Sep 1;53(2):792–802. doi: 10.3758/s13428-020-01436-x

Branimir K. Hackenberger, R software: unfriendly but probably the best, February 2020, Croatian Medical Journal 61(1):66-68 , DOI:10.3325/cmj.2020.61.66, License CC BY 2.5

Audrey M. Bertin, Benjamin S. Baumer; Creating optimal conditions for reproducible data analysis in R with 'fertile'; arXiv:2008.12098 (cs) [Submitted on 18 Aug 2020]

Mahesh Minnakanti, Vita Global Science, Waltham, US Karthik Perala, Presbyterian Health Care, Albuquerque, US; Integration of AI with RStudio: Advantages and Disadvantages; Paper MH32 25 Maret 2025.

Branimir K Hackenberger ; R software: unfriendly but probably the best PMCID: PMC7063554 PMID: 32118381 2020 Feb;61(1):66–68. doi: 10.3325/cmj.2020.61.66
Tilman M.Davies The Book of R : A first Course in Programming and Statistics,
Robin Evans, R Programming , University of Oxford

Y Asdi, Modul R , Universitas Sebelas Maret

Torsten Hothorn, Brian Everitt, Handbook of Statistical Analyses Using R

Dr. Widodo Budiharto, S.Si., M.Kom Ro’fah Nur Rachmawati, S.Si., M.Si, Pengantar Praktis Pemrograman R untuk Ilmu Komputer

Norman Matloff, The Art Of R Programming, 2011, ISBN-10: 1-59327-384-3 ISBN-13: 978-1-59327-384-2

W. N. Venables, D. M. Smith, An Introduction to R, Version 4.5.0 (2025-04-11)

LAMPIRAN I

1. **MISKUN (240401010368)**
Presentasi karena ada kepentingan sehingga tidak bisa ikut serta dalam mengerjakan kerangka makalah ini.
2. **MOCHAMAD NOOR SYAMSU (240401010302)**
Kesimpulan dan daftar pustaka, merapikan seluruh kerangka makalah.
3. **NURI DWI WULANDARI (240401010334)**
Pengenalan R Bab I poin A.
Menyatukan semua hasil pengerjaan anggota ke dalam satu dokumen Ms. Word.
Menyatukan semua daftar pustaka.
4. **M. FAHRI MUJAHID (240401010312)**
BAB II Contoh pengaplikasian R.
5. **SALMAN M. FARAS A. (240401010356)**
Syntax data R Bab I Poin B sub poin 3 dan 4.
6. **ALDYARISNANDAR (240401010362)**
Syntax pada R Bab I poin B sub poin 1 dan 2.