

Panduan Pemasangan dan Penggunaan R dan ggplot2

Wihelmus Wedo

2019-06-17

Daftar Isi

Kata Pengantar	5
1 Bahasa Pemrograman R	7
1.1 Apa itu R?	7
1.2 Pemasangan R	7
1.3 Penggunaan R	11
2 Paket ggplot2	13
2.1 Apa itu ggplot2?	13
2.2 Pemasangan ggplot2	13
2.3 Penggunaan ggplot2	13
3 Tidyverse	17
3.1 Apa itu tidyverse?	17
3.2 Install tidyverse	17
3.3 Paket inti Tidyverse	17
3.4 Sumber ilmu lainnya	20
4 Project	23

Kata Pengantar

Selama ini penulis mengamati bahwa grafik dalam publikasi BPS biasanya dibuat dalam perangkat lunak *Microsoft Office Excel* (MS Excel). Aplikasi ini telah banyak digunakan oleh banyak orang dan berbasis *Graphical User Interface*. Penulis merasa bahwa grafik yang dibuat dapat ditingkatkan efektivitas dan efisiensinya. Penulis – seseorang yang telah belajar tentang R sewaktu kuliah – ingin mengaplikasikan R dan *ggplot2* pada pekerjaannya. Penulis menyadari bahwa untuk membuat grafik yang efektif, seseorang bisa saja menggunakan program seperti *Microsoft Office Excel*. Akan tetapi terdapat beberapa hambatan ketika menggunakan 3, diantaranya

1. Menghabiskan waktu relatif lama untuk membuat grafik yang efektif
2. Sejatinya, seseorang harus membayar untuk menggunakan MS Excel.
3. Proses yang sama perlu dilakukan berkali-kali seiring dengan pemutakhiran data.
4. Untuk membuka file Excel diperlukan MS Excel.

Penulis merasa bahwa hal-hal tersebut dapat diminimalisasi dengan menggunakan R dan *ggplot2*. R sebagai sebuah *Free and Open Source Software* (FOSS) bersifat *gratis* dan *bebas* sehingga siapa pun dapat dengan mudah mengunduh, menggunakan, dan mengembangkan perangkat lunak tersebut. Selain itu, *Script R* hanyalah berbentuk sebuah *text file*. Hal tersebut memudahkan pengguna untuk melakukan *copy-paste* dan isinya dapat dibaca oleh *text editor* seperti Notepad, Neovim, Gedit, dll. Kemudahan untuk membuka dan membaca *script R* juga memiliki manfaat agar orang lain dapat mengetahui proses pengerjaan suatu data dari awal dan akhir. Hal ini mendorong terbentuknya *reproducible analysis*.

Penulis dalam dokumen ini hanya sekadar membagi informasi mengenai dasar-dasar penggunaan R, dan *ggplot2*. Dokumen ini **tidak menjelaskan semua** hal mengenai R dan *ggplot2* kepada pembaca. Akhir kata, penulis ingin mengucapkan terima kasih kepada pembaca yang membaca dokumen ini. Selamat membaca!

Bab 1

Bahasa Pemrograman R

1.1 Apa itu R?

R adalah sebuah bahasa pemrograman yang digunakan untuk melakukan *data analysis* atau *data science*. R diprakarsai oleh Ross Ihaka dan Robert Gentleman sebagai bentuk *open source* dari S, salah satu bahasa pemrograman statistik juga. Di masa kini, R banyak digunakan dalam hal *data science* atau *data analysis* sebab sifatnya yang *Free* dan *Open Source* memudahkan siapapun untuk membuat *package* untuk R. File R berbentuk *script* atau *text*. Bentuk ini memudahkan pengguna R untuk mereproduksi atau replikasi dengan cara yang sederhana (read: *copy-paste*). Hadley Wickham – salah satu orang paling populer di dunia R – membuat proses dilakukannya *data analysis* atau *data science*. Salah satu proses tersebut adalah visualisasi. Bentuk file R yang merupakan teks akan mempermudah proses ini.

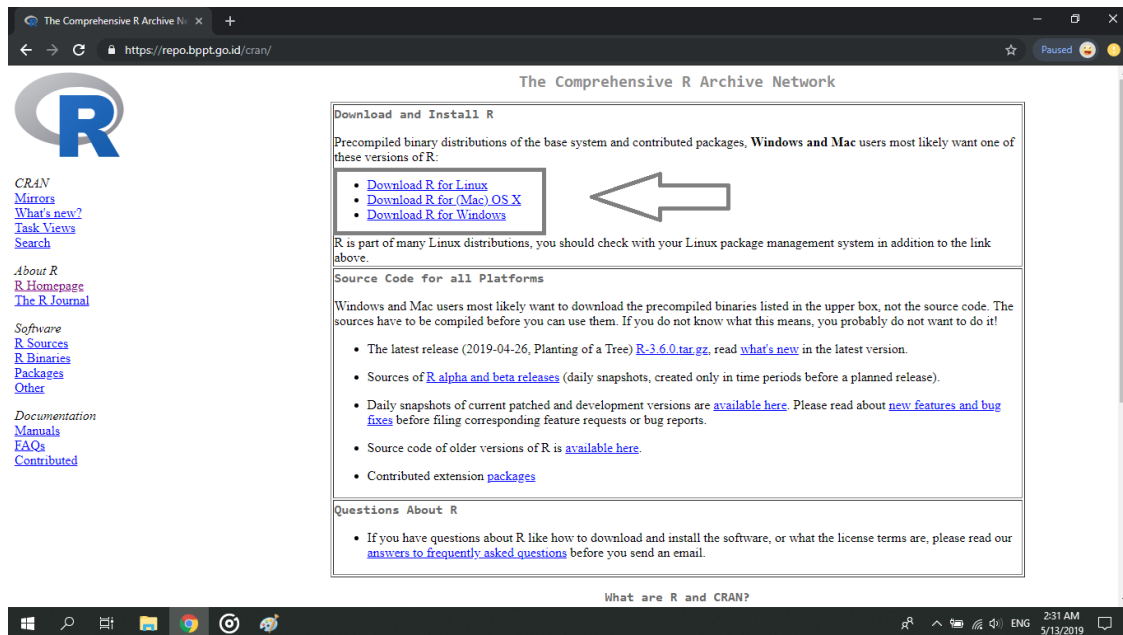
1.2 Pemasangan R

Windows 10

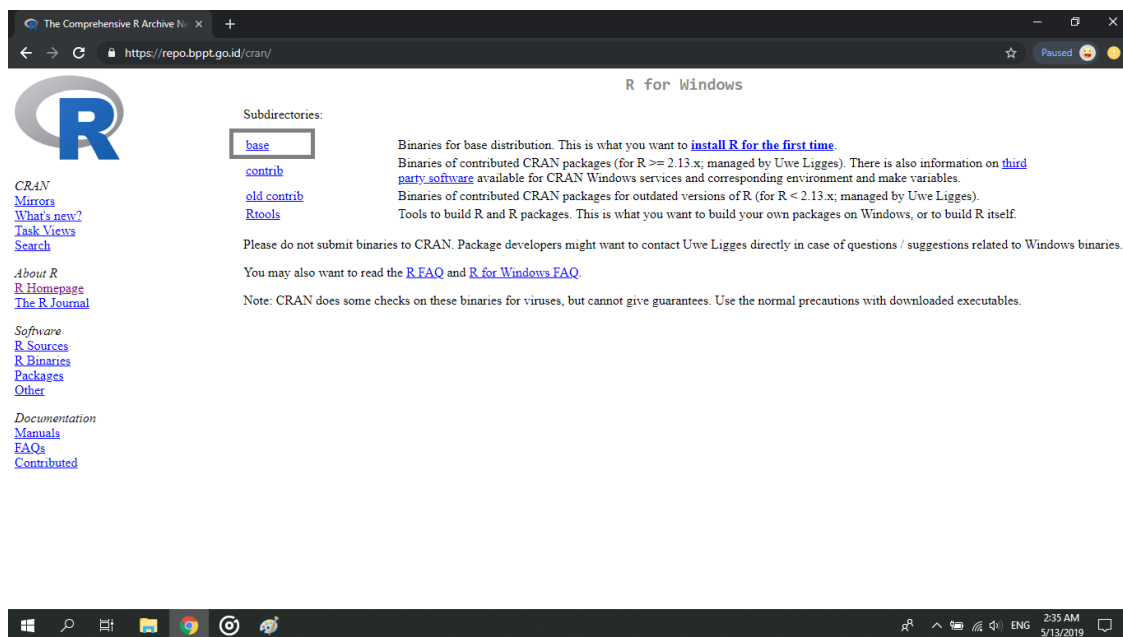
Cara terbaik untuk mengunduh R adalah dengan menggunakan situs resminya yaitu <https://www.r-project.org/>. Dari situs tersebut, akan diarahkan menuju *mirror* terdekat. Indonesia memiliki *mirror* terdekat yaitu milik Badan Pengkajian dan Penerapan Teknologi (BPPT).

1. Untuk mempermudah, silahkan menuju tautan <https://repo.bppt.go.id/cran/>.
2. Setelah halaman terbuka, klik pada bagian *Download R for Windows*
3. Kemudian akan muncul halaman baru, klik pada tautan *base*
4. Setelah halaman baru muncul, klik pada tautan *Download R 3.6.0 for Windows*.
5. Setelah selesai mengunduh R, double pada file *executable (.exe)*.

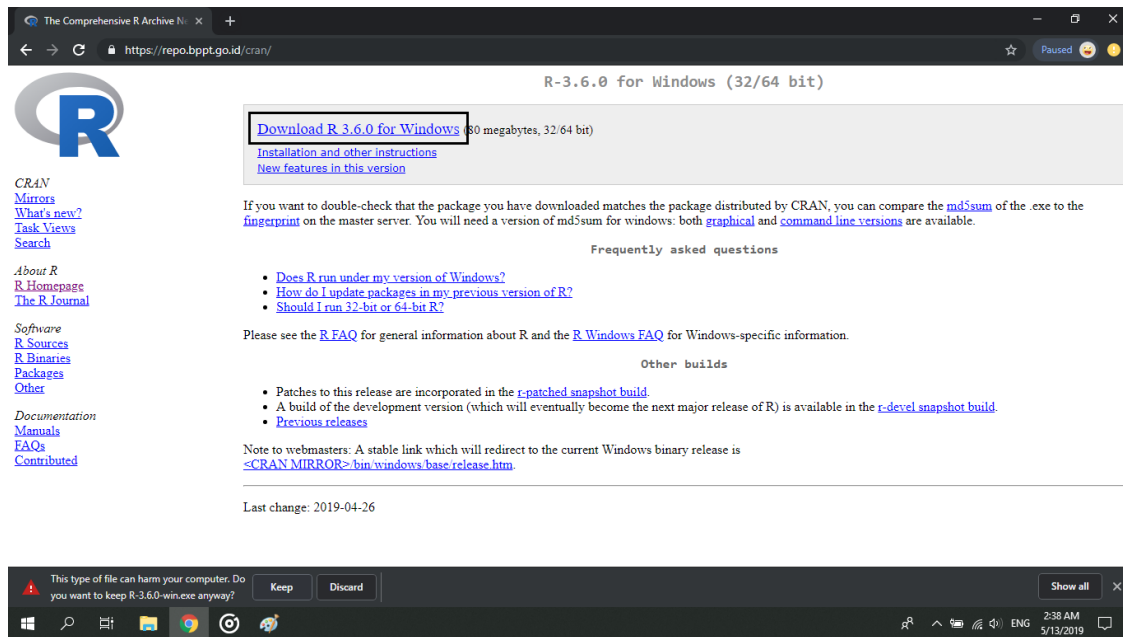
Proses pemasangan R merupakan proses mudah, pembaca hanya perlu klik tombol *Next* atau *OK* pada setiap jendela dialog.



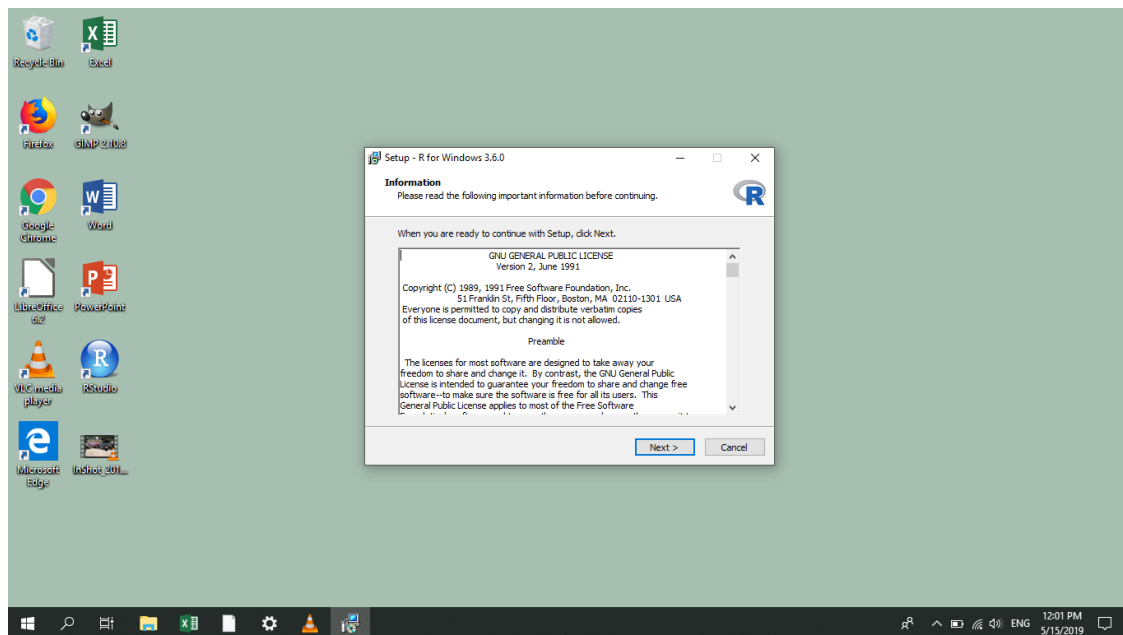
Gambar 1.1: Repository R dari BPPT



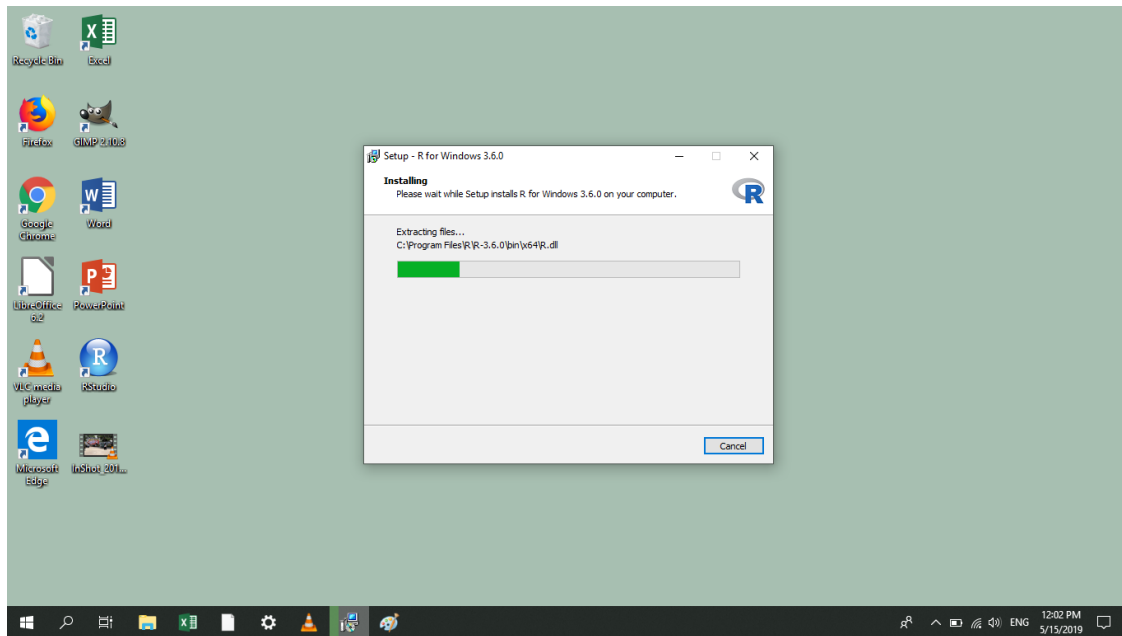
Gambar 1.2: Klik pada tautan base



Gambar 1.3: Klik pada Download R 3.6.0 for Windows



Gambar 1.4: Kotak dialog untuk R



Gambar 1.5: Kotak dialog terakhir setelah tombol *Next* dan *Ok* dipilih

Untuk mempermudah pengerjaan menggunakan R, digunakan program Rstudio IDE. Berikut adalah cara menggunakan Rstudio

1. Pergi ke tautan <https://www.rstudio.com/products/rstudio/download/>
2. Cari tulisan “Installers”
3. klik pada Rstudio 1.2.1335-Windows 7+ (64-bit)

Memasang Rstudio IDE juga sama mudahnya dengan memasang program R.

Ubuntu 18.04

Pengguna Ubuntu 18.04 dan Linux secara umum memiliki kelebihan yaitu dapat menginstall program menggunakan terminal. Oleh karena itu, untuk menginstall R dan Rstudio, pembaca hanya perlu melakukan Copy-Paste masing-masing script berikut secara baris per baris di terminal.

1. Unduh dan pasang R

```
sudo add-apt-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/'
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAE
sudo apt update
sudo apt install r-base r-base-dev
```

2. Unduh dan pasang Rstudio

```
sudo apt install gdebi
wget https://download1.rstudio.org/desktop/bionic/amd64/rstudio-1.2.1335-amd64.deb
sudo gdebi -i rstudio-1.2.1335-amd64.deb
```

1.3 Penggunaan R

1.3.1 Kalkulator

R pada bentuk sederhananya dapat dianggap sebagai sebuah kalkulator. Kita dapat melakukan operasi dasar seperti $2 + 2 = 4$.

```
2 + 2 * 10 / 5 - 10
```

```
## [1] -4
```

1.3.2 Menyimpan objek dengan <-

Dalam R, hasil dari suatu perhitungan dapat disimpan dalam sebuah objek. Kita dapat menggunakan tanda <- untuk memasukkan sesuatu ke dalam sebuah objek. Berikut adalah contoh.

```
x <- 2 + 2 * 10 / 5 - 10
y <- 9
z <- x*y
```

```
print(z)
```

```
## [1] -36
```

1.3.3 string

Selain angka, karakter atau *string* juga dapat disimpan ke dalam sebuah objek. String dalam R diawali dan diakhiri dengan tanda petik (“

```
nama <- "Wilhelmus Wedo"
```

```
nama
```

```
## [1] "Wilhelmus Wedo"
```

```
class(nama)
```

```
## [1] "character"
```

1.3.4 Vektor

Vektor dalam R dapat dipahami sebagai sekumpulan angka atau string. Vektor memiliki sebuah kelas atau *class*. Maksudnya, vektor dengan kelas *numeric* memiliki isi angka, dan vektor dengan kelas *character* memiliki isi string. Jika terdapat angka dan string dalam suatu vektor, maka vektor tersebut menjadi vektor string. Vektor dapat dibuat dengan menggunakan `c()`.

```
angka <- c(1, 2, 3, 4, 5, 6)
karakter <- c("a", "b", "c")
coba_ini <- c(1, 2, "tiga")
```

Selain *numeric* dan *character*, terdapat juga kelas lain yang merupakan pengembangan. Dari kelas *numeric*, pengembangannya berupa

1. *integer*, yaitu vektor dengan bilangan bulat.
2. *double*, yaitu vektor dengan bilangan riil.

Sedangkan dari kelas *character*, pengembangannya berupa

1. *factor*, yaitu vektor dengan adanya urutan pada elemen.
2. *date*, yaitu vektor dengan pengembangan untuk penanggalan kalender.

Secara umum, pengguna R disarankan untuk menggunakan kelas vektor yang tepat. Sebagai contoh, angka seperti 1, 2, dan 3 disimpan dalam vektor *integer* dan tanggal di dalam vektor *date*. Penggunaan yang keliru seperti memasukkan sebuah tanggal ke dalam vektor angka atau memasukkan bilangan riil ke dalam vektor *factor* dan lain-lain, dapat memberikan hasil yang membingungkan.

```
class(angka)
```

```
## [1] "numeric"
```

```
class(karakter)
```

```
## [1] "character"
```

```
class(coba_ini)
```

```
## [1] "character"
```

1.3.5 Tabel atau *data frame*

Pekerjaan di BPS biasanya dilakukan dalam sebuah tabel. Di dalam R, tabel lebih dikenal dengan nama *data frame*. Tabel atau *data frame* merupakan satu tingkat di atas vektor. Tabel memiliki baris dan kolom. Kolom dalam sebuah tabel merupakan sebuah vektor. Dapat dikatakan bahwa tabel merupakan kumpulan vektor yang memiliki panjang yang sama. Berikut adalah salah satu *data frame* bawaan dari R.

```
head(beaver1)
```

```
##   day time temp activ
## 1 346  840 36.33     0
## 2 346  850 36.34     0
## 3 346  900 36.35     0
## 4 346  910 36.42     0
## 5 346  920 36.55     0
## 6 346  930 36.69     0
```

Bab 2

Paket ggplot2

2.1 Apa itu ggplot2?

ggplot2 adalah paket yang berfokus pada visualisasi statis. ggplot2 adalah paket R yang ditulis oleh Hadley Wickham. Bagian gg dari ggplot2 merupakan singkatan untuk *Grammar of Graphics*, buku oleh Leland Wilkinson. Paket ggplot2 banyak digunakan pada bagian visualisasi data.

ggplot2 mengaplikasikan suatu sistem pembuatan grafik yang disebut dengan *The Grammar of Graphics*. Pada sistem ini, gambar dilihat dan dibentuk lapisan demi lapisan. Ada 3 hal yang penting dari ggplot2, antara lain:

1. *data* dalam bentuk *tidy*,
2. *aesthetics* seperti koordinat x, dan y, *shape*, *color*, *fill*, *shape*, dan
3. *geoms* yaitu objek geometrik seperti *point*, *bar*, dan *line*.

Pengguna memberikan data, *aesthetics*, dan jenis *geoms* dan ggplot2 akan mengatur sisanya.

2.2 Pemasangan ggplot2

ggplot2 adalah salah satu bagian dari *tidyverse*, paket R yang berfokus pada analisis data. Memasang *tidyverse* secara otomatis juga memasang ggplot2.

```
install.packages("tidyverse")
```

Namun, Jika pembaca ingin hanya memasang paket ggplot2, berikut adalah cara untuk memasang paket ggplot2

```
install.packages("ggplot2")
```

2.3 Penggunaan ggplot2

Dalam ggplot2, kita melihat plot atau grafik sebagai kumpulan lapisan atau *layer*. Banyak grafik yang dapat dihasilkan menggunakan ide atau teori ini; *scatter plot*, *bar plot*, *lollipop chart*, *line chart*, *heatmap*, *map plot* dan masih

banyak lagi. Selain itu, ggplot2 adalah paket R, sehingga banyak orang membuat paket-paket pendamping ggplot2 untuk menghasilkan lebih banyak jenis plot; *Treemap*, *animation*, *waffle chart*, dan masih banyak lagi.

```
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
```

```
##   method      from
##   [.quosures  rlang
##   c.quosures  rlang
##   print.quosures rlang
```

```
#diamonds <- diamonds[1:1000,]
```

```
diamonds
```

```
## # A tibble: 53,940 x 10
```

```
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23 Ideal    E      SI2     61.5   55   326   3.95   3.98   2.43
## 2 0.21 Premium E      SI1     59.8   61   326   3.89   3.84   2.31
## 3 0.23 Good    E      VS1     56.9   65   327   4.05   4.07   2.31
## 4 0.290 Premium I      VS2     62.4   58   334   4.2    4.23   2.63
## 5 0.31 Good    J      SI2     63.3   58   335   4.34   4.35   2.75
## 6 0.24 Very Good J      VVS2    62.8   57   336   3.94   3.96   2.48
## 7 0.24 Very Good I      VVS1    62.3   57   336   3.95   3.98   2.47
## 8 0.26 Very Good H      SI1     61.9   55   337   4.07   4.11   2.53
## 9 0.22 Fair    E      VS2     65.1   61   337   3.87   3.78   2.49
## 10 0.23 Very Good H      VS1     59.4   61   338   4      4.05   2.39
## # ... with 53,930 more rows
```

Data diamonds – data *built in* dari paket ggplot2 – digunakan sebagai contoh. Terdapat beberapa variabel dalam data diamond, diantaranya

- *carat* : berat dari berlian
- *cut* : kualitas potongan
- *color* : Warna berlian, dari J (terburuk) ke D (terbaik)
- *clarity* : Ukuran seberapa jernih suatu berlian
- *depth* : persentase kedalaman
- *table* : lebar dari ujung atas berlian relatif terhadap titik terlebar
- *price* : harga berlian dalam USD.
- *x* : panjang dalam milimeter
- *y* : lebar dalam milimeter
- *z* : kedalaman dalam milimeter

Informasi lainnya mengenai data diamonds dapat dilihat dengan mengetik `?diamonds` pada *console*. Misalkan kita ingin mengetahui sebaran dari harga berlian. Dengan menggunakan informasi yang tersedia, kita dapat mengidentifikasi lapisan *data*, *aesthetics*, dan *geom* untuk gambar yang kita butuhkan.

1. data : diamonds

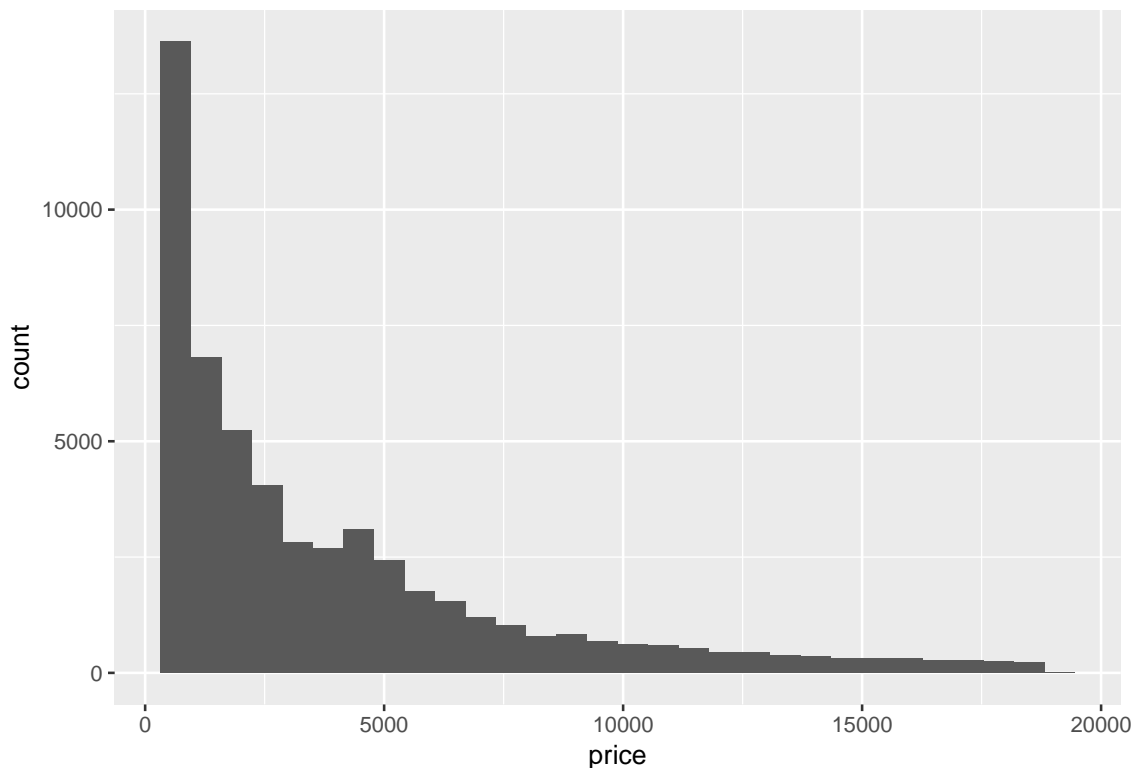
2. aes : price
3. geom : histogram

Dengan begitu, kita hanya perlu menyuruh R untuk membuat gambar.

```
ggdiamonds <- ggplot(data = diamonds, aes(x = price)) +  
  geom_histogram()
```

```
ggdiamonds
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



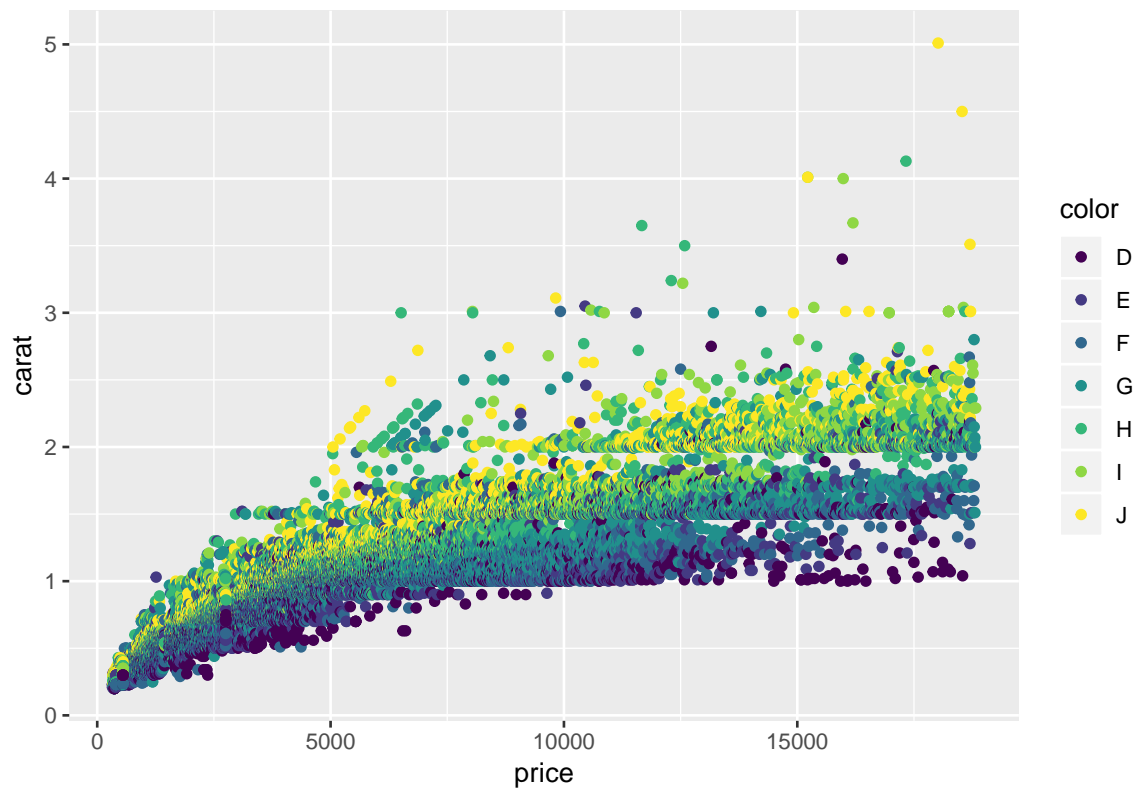
Pada fungsi `aes()`, kita memasukkan sumbu x sebab kita hanya memerlukan 1 sumbu. Contoh berikutnya adalah menggunakan 2 sumbu. Misalkan kita ingin melihat pola antara harga (*price*) dan kualitas potongan dari berlian (*cut*). Arinyta kita akan membuat *scatter plot* antara *price* dan *cut*. Selama kita dapat mengidentifikasi lapisan yang diperlukan:

1. data : diamonds
2. aes : (x = price, y = cut)
3. geom : point

Akan tetapi, kita juga ingin melihat pola antara harga har dan kualitas potongan, dan juga warna. Oleh karena itu, kita membuatuhkan satu *aesthetic* lagi pada goem point.

```
ggprice <- ggplot(data = diamonds, aes(x = price, y = carat)) +  
  geom_point(aes(colour = color))
```

ggprice



Bab 3

Tidyverse

3.1 Apa itu tidyverse?

Ketika bekerja dengan data di dunia nyata seperti di BPS, paket `ggplot2` saja tidak cukup. Paket `ggplot2` membutuhkan *tidy data* sebagai masukan, namun data yang dikumpulkan BPS ataupun data yang berada di dunia nyata tidak selalu berada dalam bentuk tersebut. Selain itu, terkadang dibutuhkan proses awal sebelum data divisualisasikan. Proses ini dapat berupa agregasi berdasarkan variabel tertentu, penyaringan terhadap kategori tertentu, ataupun menambahkan variabel baru yang berperan dalam proses visualisasi. Untuk itu, diperlukan paket-paket lainnya.

Hadley Wickham dan komunitas R kemudian membuat paket-paket R lainnya yang kemudian disebut sebagai *tidyverse*. Tidyverse merupakan koleksi dari paket yang bekerja kriteria *tidy data*. Tidyverse dibuat untuk memudahkan proses analisis data, mulai dari memasukkan data (*import data*), melakukan proses perapihan data (*tidy*), melakukan transformasi dan agregasi (*transform*), visualisasi data (*visualization*), melakukan pemodelan (*modelling*), hingga menulis laporan (*reporting*).

3.2 Install tidyverse

Untuk menginstall paket `tidyverse` caranya sama dengan menginstall paket `ggplot2`.

```
install.packages("tidyverse")
```

Memasang `tidyverse` juga akan memasang paket-paket lain yang tergabung dalam ekosistem tidyverse, sehingga waktu unduh dan pasang akan lebih lama dibandingkan dengan waktu `ggplot2`

3.3 Paket inti Tidyverse

Pada panduan ini penulis akan menjelaskan sedikit informasi mengenai paket-paket inti dari tidyverse. Paket-paket ini disebut paket inti atau *core* sebab paket ini juga di *attach* ketika `tidyverse` di *attach*. Paket `ggplot2` telah

dibahas pada bab sebelumnya sehingga tidak dibahas pada bagian ini.

```
library(tidyverse)
```

```
## Registered S3 method overwritten by 'rvest':
##   method          from
##   read_xml.response xml2

## -- Attaching packages ----- tidyverse 1.2.1 --

## v tibble  2.1.1      v purrr   0.3.2
## v tidyr   0.8.3      v dplyr  0.8.0.1
## v readr   1.3.1      v stringr 1.4.0
## v tibble  2.1.1      v forcats 0.4.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

3.3.1 readr dan tibble

Paket `readr` digunakan dalam proses memasukkan data ke dalam lingkungan R. `readr` menyediakan cara yang mudah untuk memasukkan data yang *rectangular* seperti *csv* (*comma separated values*) dan *tsv* (*tab separated values*).

Paket `tibble` mungkin kurang populer dibandingkan paket inti lainnya. *Tibble* merupakan pengembangan dari *data frame*. Terdapat beberapa penambahan dan pengurangan fitur dari *data.frame* yang dimiliki oleh sebuah `tbl_df` yang dapat membuat pekerjaan lebih mudah. Penambahan fitur seperti memberitahu tipe atau *class* dari semua kolom tabel serta pengurangan fitur seperti meniadakan *partial matching* diyakini oleh para pembuat paket untuk membuat pekerjaan analisis data menjadi lebih mudah. *Tibble* juga dapat melakukan hal lain namun hal tersebut jarang penulis gali lebih dalam.

Data yang diimport menggunakan `readr` akan memiliki kelas `tbl_df`.

3.3.2 tidyr

Paket `tidyr` digunakan dalam proses perapihan data. Data rapi atau *tidy data* adalah bentuk data yang memenuhi semua kriteria berikut.

1. 1(satu) variabel menempati 1(satu) kolom
2. 1(satu) observasi atau individu menempati 1(satu) baris
3. 1(satu) nilai atau *value* menempati 1(satu) sel.

Data rapi menunjukkan cara yang baku atau standar dalam menyimpan data. Standar ini digunakan oleh semua paket dalam ekosistem Tidyverse. Terdapat 2 fungsi yang sering digunakan dari paket `tidyr`, yaitu `gather()` dan `spread()`.

3.3.3 dplyr dan %>%

Paket dplyr adalah salah satu paket terpopuler (selain ggplot2) dalam ekosistem tidyverse.

1. `select()` memilih variabel/ kolom.
2. `filter()` menyaring tabel berdasarkan suatu kriteria.
3. `mutate()` menambah variabel/kolom yang merupakan fungsi dari variabel/kolom yang sudah ada.
4. `summarise()` atau `summarize()` meringkas tabel.
5. `arrange()` mengurut tabel berdasarkan variabel/kolom.

Semua fungsi di atas digabungkan dengan `group_by()` membuat pengguna dapat menjalankan fungsi di atas *berdasarkan kelompok*. Lima fungsi di atas dibuat menggunakan kata kerja atau *verbs* untuk memudahkan pengguna memahami arti dari fungsi-fungsi tersebut. Selain lima fungsi di atas ditambah `group_by()`, dplyr memiliki operator yang disebut sebagai *forward pipe operator* atau biasanya disingkat menjadi *pipe*. Operator ini memiliki lambang `%>%` dan dibaca *kemudian* atau *lalu*.

Pada umumnya, kode dibaca dari dalam ke luar (*inside out*). Namun, dengan menggunakan `%>%`, kode dapat dibaca dari kiri ke kanan. Hal ini akan mempermudah seseorang untuk mengerti apa yang sedang terjadi sehingga lebih mudah dimengerti.

Sebagai ilustrasi, perhatikan cerita pendek di bawah ini.

Hatori adalah seorang ninja.

Ia mendaki gunung, kemudian menuruni lembah, dan akhirnya mengarungi samudra.

Berdasarkan cerita pendek di atas, pada umumnya *script* akan ditulis dalam satu baris, seperti berikut

```
hatori <- as_ninja()

mengarungi(menuruni(mendaki(hatori, gunung), lembah), samudra)
```

atau dalam bentuk yang agak condong ke dalam jika menggunakan *indent*.

```
hatori <- as_ninja()

mengarungi(
  menuruni(
    mendaki(hatori, gunung),
    lembah
  ),
  samudra
)
```

Untuk membaca *script* di atas, diperlukan waktu yang relatif lebih banyak dibandingkan ketika penulisan *script* menggunakan `%>%`. Berikut adalah *script* yang sama jika ditulis menggunakan `%>%`.

```
hatori <- as_ninja()

hatori %>%
```

```
mendaki(gunung) %>%  
menuruni(lembah) %>%  
mengarungi(samudra)
```

3.3.4 stringr dan forcats

Paket `stringr` – sesuai namanya – digunakan ketika berurusan dengan *string* dalam proses analisis data. Di dalam dunia R, *string* juga dikenal dengan istilah *character*. Fungsi-fungsi dari paket `stringr` memiliki awalan `str_` pada nama.

forcats adalah sebuah anagram untuk *factors*. Sesuai dengan namanya, `forcats` digunakan ketika berurusan *factors*. Mirip dengan `stringr`, fungsi-fungsi dalam paket ini memiliki awalan `fct_`.

3.3.5 purrr

Paket `purrr` dapat dikatakan memiliki tingkat kesulitan yang lebih tinggi dibandingkan dengan paket inti lainnya. `purrr` digunakan untuk mengimplementasikan *functional programming*.

3.4 Sumber ilmu lainnya

Tidyverse merupakan paket R yang luas sehingga panduan ini tidak cukup menjelaskan semua paket di dalam ekosistem. Panduan ini juga hanya menjelaskan sedikit informasi dari paket inti dari tidyverse. Masih banyak informasi dari paket inti tidyverse yang penulis tidak bahas dalam dokumen ini, terlebih mengenai paket-paket lain yang bukan paket inti. Oleh karena itu, penulis berharap pembaca dapat mencari dan membaca sendiri sumber-sumber lain mengenai tidyverse dan analisis data menggunakan R.

Tidyverse.org

Tidyverse.org adalah situs yang khusus menjelaskan paket inti dari tidyverse. Ini adalah sumber nomor 1 bagi penulis ketika mengalami masalah dengan paket inti tidyverse.

stackoverflow.com

StackOverflow adalah sebuah forum dunia maya yang pada umumnya dihuni oleh *developer*. Dapat dibilang StackOverflow ada forum tanya jawab seputar hal-hal *programming*. Banyak pertanyaan mengenai R dan Rstudio yang ditanyakan di stackoverflow.

community.rstudio.com,

Mirip seperti StackOverflow, ini ada tempat bertanya khusus mengenai R dan Rstudio. Forum ini dibuat oleh pihak Rstudio.

youtube.co.id

YouTube adalah situs berbagi video terbesar saat ini. Terdapat banyak video tutorial maupun presentasi mengenai paket-paket dalam tidyverse.

#rstat

Pengguna R memiliki komunitas di twitter. Komunitas ini biasanya menggunakan tagar #rstat memposting sesuatu tentang R.

r/rstats

Seperti di twitter, komunitas R juga terdapat di reddit. Komunitas R di reddit berkumpul di *subreddit* r/rstats.

Google

Google adalah *search engine* yang paling besar. Mencari hal-hal mengenai paket tidyverse merupakan hal mudah dengan menggunakan google.

Bab 4

Project

Aktualisasi yang penulis lakukan bertujuan untuk meningkatkan efektivitas dan efisiensi grafik dalam publikasi. Grafik yang efektif adalah grafik yang mampu menyampaikan informasi. Konsumen dari grafik yang efektif dapat dengan mudah mengerti tentang apa yang ditunjukkan oleh suatu grafik. Grafik yang efisien adalah grafik yang dapat dibuat dengan waktu yang lebih singkat serta dengan biaya yang sedikit mungkin.

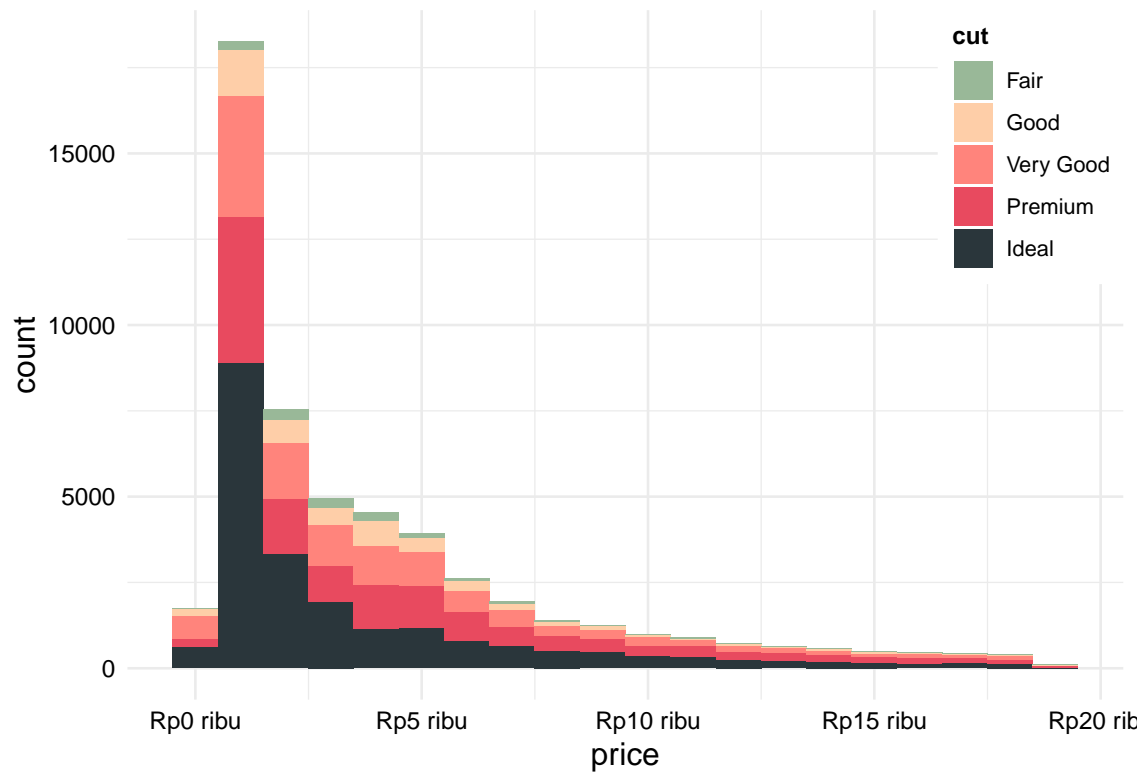
Untuk meningkatkan efektivitas dan efisiensi grafik dalam publikasi, penulis menuliskan 3 file.

1. `ggplot_theme.R` berisikan tema untuk objek `ggplot2`.
2. `ggplot_skala.R` berisikan skala untuk sumbu x dan y
3. `ggplot_pallette.R` berisikan *palette* untuk *aesthetic color* dan *fill*.

```
source(here("Script/ggplot/ggplot_tema.R"))
source(here("Script/ggplot/ggplot_palette.R"))
source(here("Script/ggplot/ggplot_skala.R"))

gg_diamonds <- ggplot(diamonds, aes(x = price)) +
  geom_histogram(aes(fill = cut), binwidth = 1000) +
  scale_x_continuous(labels = skl_rupiah(dari = "rupiah", "ribu")) +
  scale_fill_manual(values = plt5_04) +
  theme_sbd()

gg_diamonds
```



Lumbung atau *repository* aktualisasi terdapat pada <https://git.bps.go.id/eus.wedo/sbd-grafik>. Cara mengunduh repo tersebut terdapat pada README