Coding for Social Scientist

Disampaikan dalam Workshop *Big Data* yang diselenggarakan oleh CDC, FISIPOL, UGM, 2018

Ujang Fahmi and Canggih Puspo Wibowo

2018-10-31

Contents

Selamat Datang **5**

Pengantar **7**

1 Pengenalan dasar **9**
1.1 Menyiapkan R . . . 10
1.2 Membuat skrip R . . . 12
1.3 Rangkuman . . . 14

2 Proses Ekstraksi Informasi **15**
2.1 Impor dan Ekspor Data . . . 15
2.2 Pre-processing . . . 16
2.3 Eksplorasi . . . 17
2.4 Visualisasi . . . 19

3 Latihan **21**
3.1 Mendapatkan data . . . 21
3.2 Eksplorasi data . . . 22

4 Contoh hasil kerja **25**

5 Sumber Belajar Mandiri **27**
5.1 Komunitas . . . 27
5.2 Free Course . . . 27

Selamat Datang

Pengantar

Saat ini sumber data yang dapat digunakan baik untuk tujuan penelitian maupun bisnis banyak tersedia di internet. Sayangnya, tidak semua orang bisa memanfaatkannya. Terdapat beberapa kendala mengapa tidak semua orang bisa mengekstrak pengetahuan dari sumber data yang cenderung lebih murah dan sebenarnya mudah untuk di dapatkan tersebut. Salah satu sebab utamanya kurangnya keterampilan untuk membuat alat untuk mengambilnya. Dalam konteks ini adalah keterampilan untuk memanfaatkan open source, salah satunya adalah R.

R merupakan salah satu open source yang saat ini cukup populer dan banyak digunakan oleh berbagai organisasi dengan skala besar hingga kecil. Pengguna R tersebar mulai dari perusahaan seperti Google dan Facebook, pemerintahan, hingga usaha kecil menengah. Berdasarkan definisi di laman resminya, R merupakan bahasa pemrograman untuk mengolah data secara statistik. Dalam praktinya R juga banyak digunakan untuk mengolah data tidak terstruktur, termasuk data dari media sosial.

Sayangnya, *coding* diidentikan hanya dilakukan oleh anak teknik. Hanya sedikit akademisi sosial yang memiliki kemampuan tersebut. Padahal, akademisi sosial memiliki salah satu modal utama untuk bisa membuat data menjadi lebih berarti, yaitu *domain knowledge*. Sebaliknya, hanya sedikit yang bisa *coding* memiliki *domain knowledge* untuk bisa memanfaatkan informasi yang diekstrak dari data dalam jumlah banyak. Dalam konteks ini, kolaborasi lintas disiplin ilmu dapat menjadi salah satu solusi. Tapi, masing-masing pihak minimal memiliki pengetahuan dan pemahaman dasar tentang cara kerja masing-masing. Selain itu, akademisi sosial juga bisa belajar sendiri dengan memanfaatkan berbagai sumber baik yang gratis maupun berbayar yang saat ini banyak tersedia di Interent.

Melalui workshop ini, kami bertujuan untuk mengenalkan beberapa dasar pengelolaan big data dengan menggunakan bahasa pemrograman R. Setelah mengikuti workhshop, peserta diharapkan memiliki:

1. Pengetahuan tentang bahasa pemrograman;
2. Pemahaman alur pengolahan big data; dan
3. Kemampuan untuk membuat skrip/menjalankan skrip untuk mendapatkan data dari internet

Workshop ini terdiri dari tiga kegiatan. *Pertama*, penjelasan tentang R dan Rstudio. *Kedua*, memahami proses ekstraksi informasi dari big data. *Ketiga*, praktik mendapatkan dan mengeksplorasi data dari twitter.

Chapter 1

Pengenalan dasar

Sebelum melangkah lebih jauh, mungkin kita terlebih dahulu perlu mengetahui apa itu bahasa pemrograman? apakah ia merupakan bahasa yang berfungsi sama dengan bahasa yang kita gunakan sehari-hari? Menurut Wikipedia, bahasa pemrograman adalah:

> … a formal language, which comprises a set of instructions used to produce various kinds of output. Programming languages are used to create programs that implement specific algorithms.

Berdasarkan definisi di atas, maka fungsi bahasa pemrograman kurang lebih sama dengan bahasa yang kita gunakan sehari-hari dalam membuat buku petunjuk atau resep masakan. Perbedaannya, bahasa yang kita gunakan ditujukan agar dapat dipahami oleh manusia, sedangkan bahasa pemrograman agar dapat dipahami oleh komputer, di mana `R` merupakan salah satu di antara bahasa pemrograman yang saat ini ada dan berkembang dengan pesat.

Sementara `Rstudio` adalah alat untuk mempermudah penggunaan `R`. Di sini `RStudio` sering disebut sebagai *integrated development environment* (IDE) untuk R. Sederhananya, RStudio digunakan sebagai tampilan dari R. Oleh karena itu, untuk menggunakannya pun kita terlebih dahulu harus menginstall `R`. Gambar `1.1` menunjukkan tampilan antar muka `Rstudio` yang perlu diperhatikan.

Seperti dapat dilihat pada gambar `1.1`. Rstudio memiliki empat bagian yang memiliki fungsi masing-masing. Bagian pertama (1): digunakan untuk menulis script dan memiliki beberapa tombol. Untuk menjalankan script bisa klik run pada bagian kanan atas. Bagian kedua (2) : merupakan bagian `console` di mana kita bisa melihat script yang dijalankan. Bagian ketiga (3): merupakan bagian environment, dimana pada bagian tersebut terdapat beberapa bagian yang bisa pilih. Misalnya, bagian `Environment` untuk menampilkan data yang dimasukkan (diimport), bagian `Hystory` untuk melihat aktivitas yang sudah dilakukan dalam satu sesi R, dan bagian `Connection` untuk melihat dan mengatur koneksi R dengan database seperti `SQL` atau `SPARK`.

Bagian terakhir (4): berfungsi untuk menampilkan hasil visualiasi (`plot`). Selain itu, pada bagian ini kita juga bisa melihat repository dan file-file yang ada di dalamnya. Lebih penting lagi, di sini kita juga dapat menemukan bantuan ketika kita lupa instruksi yang dibutuhkan. Untuk menemukan bantun atau penjelasan kita dapat menggunakan fungsi `?` diikuti dengan objek yang ingin dilihat. Contohnya adalah sebagai berikut.

```
?read.csv
```

Dengan menuliskan code di atas pada bagian `console` dan menekan `Enter` pada bagian 4 akan menampilkan hasilnya. Di mana pada tampilan tersebuk kita bisa mendapatkan definisi fungsi sekaligus contoh penggunaannya. Selain itu, anda juga perlu mencoba untuk menggunakan fungsi bantuan lainnya, yaitu `help()` pada console dan lihat apa yang dihasilkan.

Sebagai rangkuman, pada bagian ini kita sudah mengetahui dan mengenal beberapa bagian. Pertama untuk menggunakan `RStudio` kita terlebih dahulu perlu menginstall `R`. Untuk mendapatkan bantuan penjelasan kita bisa menggunakan fungsi `help()` atau `?` diikuti objek yang ingin dilihat. Rstudi terdiri dari beberapa.

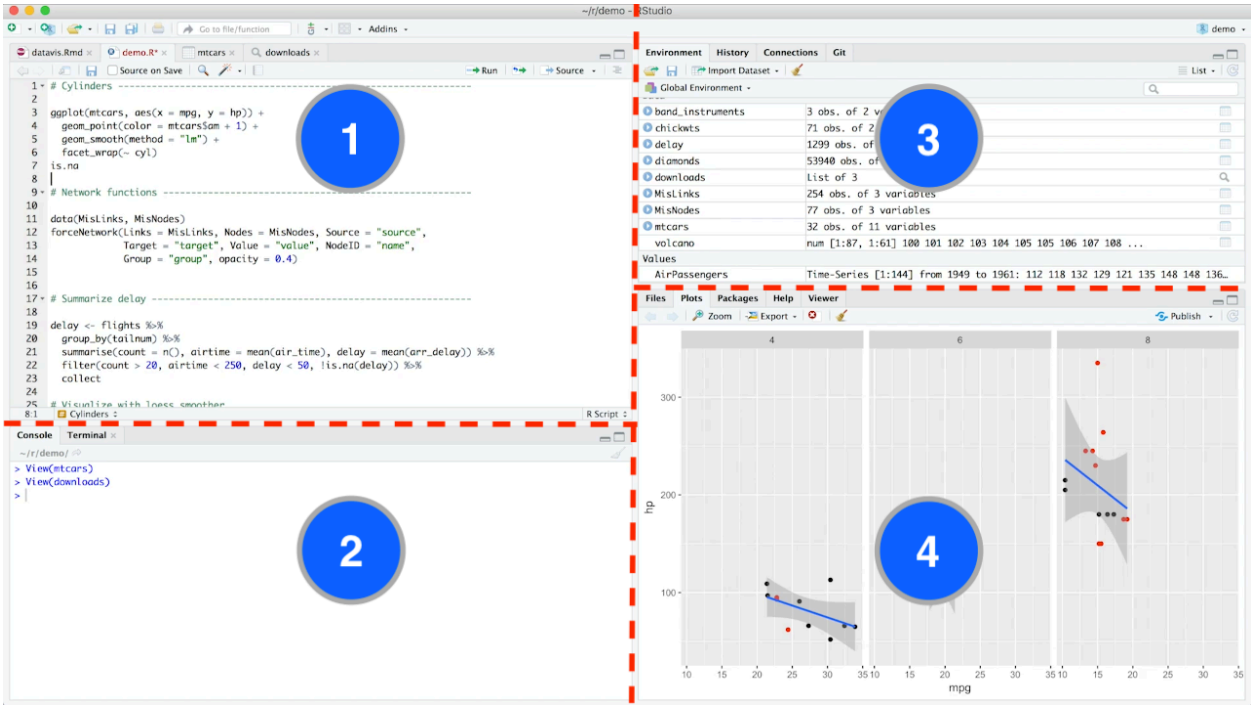

Figure 1.1: RStudio Interface terdiri empat bagian utama.

Untuk ini saya sarankan untuk mengeksplorasinya lebih jauh, karena pada bagian selanjutnya kita akan mulai belajar untuk menulis dan menjalankan script.

1.1 Menyiapkan R

Agar dapat menggunakan R secara lokal atau PC/laptop masing-masing, kita perlu mengunduh installernya terlebih dahulu di laman berikut:

1. Dari Indonesia, R bisa didapat secara gratis di laman: https://repo.bppt.go.id/cran/
2. Rstudio dapat didownload melalui laman: https://www.rstudio.com/products/rstudio/download/#download

Kita bisa menyesuaikan file installer yang sesuai dengan machine laptop/PC masing, misalnya untuk `MAC`, `WINDOWS` atau `LINUX`. Setelah dua file tersebut terunduh silahkan install `R` terlebih dahulu kemudian `RStudio`. Setelah keduanya berhasil di install saat ini mesin pc/latop, jalankan Rstudio. Selanjutnya coba tuliskan script dibawah ini pada bagian `console` lalu tekan `enter`.

```
?help
?base
```

Script di atas akan mengarahkan anda pada sebuah halaman baru yang berisi keterangan tentang package dasar (R Core Team, 2018) pada bagian kolom 1 dan keterangan fungsi-fungsi dasar pada kolom 4. Pada bagian kolom 4 ada tulisan `[Package base version 3.5.1 Index]`, klik bagian `index` yang berwarna biru dan anda akan diarahkan pada dokumentasi fungsi-fungsi dasar R. Klik salah satu fungsi untuk membaca penjelasn dan melihat contoh penggunaannya.

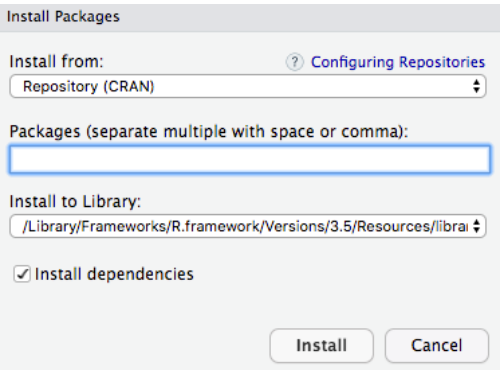

Figure 1.2: RStudio Interface terdiri empat bagian utama.

1.1.1 Install Package

Seperti telah dijelaskan dalam pengantar, pengolahan data dengan R didukung oleh berbagaik packages atau library yang dokumentasinya dapat dilihat di laman ini: https://cran.r-project.org/web/packages/available_packages_by_name.html. Untuk dapat menggunakannya anda perlu menginstallnya terlebih dahulu. Hal ini bisa dilakukan dengan menggunakan salah satu fungsi dasar R, yaitu `install.packages()`, di dalam kurung di isi dengan nama package/library. Misalnya, nama package yang akan diinstall adalah `tidyverse` (Wickham, 2017), maka pada bagian `console` kita bisa menuliskan script berikut lalu klik enter.

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

Agar dapat menginstall package, pc/laptop kita harus terhubung dengan internet karena R akan mendownload dari laman `CRAN` di atas. Selam proses menginstall sebaiknya kita tidak menyimpan script apapun. Barus setelah selasai menginstall kita bisa melanjutkan aktivitas di RStudio. Kita dapat mengetahui instalasi sudah selese dengan memperhatikan bagian console yang akan mengeprint aktivitas yang sedang berlangsung. Selain itu, pada bagian `console` sebelah kanan atas juga ada tanda merah jika masih ada proses yang berlangsung.

Selain dengan menggunakan fungsing `install.packages()`, packaages atau library juga diinstall dengan menggunakan fitur yang sematkan di RStudi, yaitu fitur `Tools`. Fitur `Tools` terletak dibagian atas antar muka, dan kita hanya perlu memilih menu `install packages` yang ada di dalamnya dan akan muncul kota dialog baru seperti di terlihat pada gambar berikut.

Dialog box seperti pada gambar 1.2 menunjukkan ada tiga isian yang perlu diperhatikan. Pertama, bagian sumber. Secara *default* sumber package yang akan diinstall diambil dari CRAN. Kedua, kolom nama package yang harus diisi secara manual (ketik). Ketiga, kolom yang menunjukkan *repository* atau folder tempat package akan diinstall. Keemapat, dependency yang ketika dicentang berarti kita akan menginstal package atau library lain yang dibutuhkan oleh package yang akan diinstall. Untuk sementara, dua kolom yang disebut terakhir disarankan untuk biarkan pengaturan semula atau *default*.

1.1.2 Latihan 1

> Untuk mengelola teks, terdapat salah satu library dengan nama `tidytext`. Installah library tersebut dengan salah satu metode install package yang sudah dijelaskan.

1.1.3 Simbol-simbol utama

Secara lengkapnya, simbol dan fungsi-fungsi dasar dapat dipelajari di sini. Namun pada umumnya, dalam `R` ada dua simbol utama yang sering digunakan yaitu `"xyz"` (tanda petik) dan `#` (tanda pagar).

1. Tanda petik (`"xyz"`) digunakan untuk menunjukkan bahwa apapun yang ada diantara dua tanda petik tersebut dianggap sebagai character (`chr`). Sebagai contoh, jalankan script di bawah ini.

```
# vector
huruf_a <- "1"
# cek apakah karakter atau bukan
is.character(huruf_a)
```

```
## [1] TRUE
```

```
# vector
huruf_b <- 1
# cek apakah karakter atau bukan
is.character(huruf_b)
```

```
## [1] FALSE
```

2. Tanda pagar (`#`) digunakan untuk memberikan komentar yang berfungsi untuk memberikan keterangan terhadap script yang kita buat. Sebagai contoh, pada script di atas term vector yang didahului oleh tanda pagar saya gunakan untuk mengingat bahwa kode yang ada dibahwanya merupakan sebuah vector. Sementara komentar berikutnya saya gunakan untuk memberikan keterangan bahwa kode yang ada dibawahnya digunkan untuk mengecek apakah benar `huruf_a` dan `huruf_b` merupakan karakter.
3. Tanda `<-` atau `=` dibaca sama dengan.

1.2 Membuat skrip R

Dengan menggunakan `RStudio` kita bisa membuat beberja jenis script untuk menuliskan bahasa R. Untuk sementara saya hanya akan mengenalkan dua jenis dulu, yaitu `.R`. Caranya adalah dengan klik menu `File -> New file -> R Script` atau `Command+Shift+N` jika menggunakan Macbook. Sama seperti saat kita menulis atau membuat file pada umumnya, pada saat menulis script R untuk mengorganisasinya dengan mudah kita perlu meletakannya dalam sebuah folder atau *repository* khusus. Di dalam RStudio, kita bisa membuatnya dengan memilih `New Project...` pada menu file, lalu pilih `New Directory` dan pilih `New Project`. Setelah itu, kita bisa menentukan hardisk yang akan digunakan untuk menyimpan file *project* yang sedang dikerjakan, menamai folder, dan lain sebagainya. Dengan cara seperti itu, pekerjaan kita bisa menjadi lebih terorganisasi.

```
# Untuk memastikan di mana kita menyimpan file atau project
getwd()
```

1.2.1 Latihan 2

> Buatlah sebuah proyek baru dengan nama yang anda pilih sendiri. Setelah itu, buat sebuah R script dan simpan dalam project tersebut.

1.2.2 Menulis skrip

Untuk memulai menulis skrip, terdapat dua hal pertama yang ingin saya anggap sebagai dasar, yaitu cara membuat variabel dan mengenal jenis data. Terkait dengan jenis data, terdapat beberapa data yang bisa digunakan seperti vector, data frame, lists, dan matrix. Di sini kita akan lebih fokus ke vector dan data frame saja. Namun jika tertarik untuk mempelajarinya lebih lanjut bisa memulainya di sini.

- **Nama file**

Table 1.1: Contoh Isi sebuah data frame

name	genus	vore
Cheetah	Acinonyx	carni
Owl monkey	Aotus	omni
Mountain beaver	Aplodontia	herbi
Greater short-tailed shrew	Blarina	omni
Cow	Bos	herbi

Dalam membuat variabel atau nama file di R tidak boleh ada spasi. Selain itu R juga case sensitve. Sebagi contoh kita ingin menamai file dengan nama **`Data Kita`** akan menyebabkan error karena ada spasi. Sebaiknya jika ingin menggunakan dua term atau elemen menyambungnya dengan `_` seperti: **`Data_Kita`**. Berikut adalah contohnya:

```
Data_Kita <- mtcars
```

Keterangan: skrip di atas mengambil data dengan nama mtcars, yang merupakan data bawaan dari R, sebagai data dengan nama Data_Kita.

- **Data frame**

Sebuah data frame pada prinsipnya adalah sebuah tabel di mana setiap kolom memiliki satu jenis variable dan setiap baris (*row*) mengandung satu nilai untuk masing-masing variabel/kolom. Berikut ini adalah karakteristik dari suatu frame data yang contohnya dapat dilihat pada tabel 1.1.

1. Memiliki nama kolom
2. Nama baris harus unik
3. Data yang disimpan dalam data frame dapat berupa angka, faktor atau karakter
4. Setiap kolom memiliki item data yang sama

Untuk melihat rangkuman dan jenis sebuah data kita bisa menggunakan `summary()` dan `class()`. Untuk melihat struktur sebuah data bisa menggunakan fungsi `str()`.

- **Menjalankan skrip**

Untuk menjalankan skrip R yang sudah kita tulis, kita bisa menggunakan *shortcut*. Jika menggunakan mac, kita blok bari skrip yang akan dijalankan dan tekan command+shift+enter. Selain itu kita bisa menggunakan tombol run dibagian kanan atas.

Lathian: Coba jalankan skrip di bawah ini dan lihat hasil dari tiga fungsi di atas, yaitu `summary()`, `class()`, dan `str()` pada data tentang pola tidur yang sudah ada dalam Rstudio.

```
data <- msleep # data

str(data)
class(data)
summary(data)
```

1.2.3 Tidyverse

Tidyverse merupakan kumpulan packages yang terdiri dari beberapa package lain, seperty `readr`, `dplyr`, `ggplot` dan beberapa package lainnya. Ketika menggunakan R untuk mengolah data atau mengeksplorasi data kita akan banyak menggunakan package-package ini. Secara keseluruhan, dalam modul ini saya akan mengenalakan beberapa fungsi dari tiga package yang disebutkan di atas dalam beberapa bagian yang berebeda sesuai dengan fungsinya.

Pertama, readr akan lebih banyak digunakan untuk mengimpor data. Kedua, dplyr akan lebih banyak digunakan dalam melakukan pre-processing. Sementara ggplot akan sangat membantu dalam melakukan eksplorasi dengan visualisasi. Di sini saya akan lebih fokus untuk menjelaskan `dplyr` terlebih dahulu.

`dplyr` memiliki fungsi pipe yang tandangnya adalah `%>%` yang berfungsi untuk mengaplikasikan baris berikutnya pada baris sebelumnya. Sebagai contoh dapat dilihat pada skrip di bawah ini.

```
library(dplyr)

data <- msleep %>%
  select(1:3)
```

Skrip di atas kita mengambil `msleep` yang kita sebuat sebagai `data`, lalu kita hanya memilih kolom nomor 1 sampai tiga saja dari data. Fungsi `select()` di sini berfungsi untuk memilih. Berikutnya adalah beberapa fungsi lain yang mungkin akan sering digunakan dalam proses belajar kita.

`select()`: pick columns by name

`filter()`: keep rows matching criteria

`arrange()`: reorder rows

`mutate()`: add new variables

`summarise()`: reduce variables to values

Untuk belejar lebih lanjut untuk fungsi yang dimiliki oleh masing-masing package kita bisa menggunakan skrip di bawah ini. Misalnya kita ingin belajar tentang dplyr. Untuk package lain kita hanya perlu mengganti `dplyr` dan `readr` dengan nama package lainnya.

```
browseVignettes(package = "dplyr")
browseVignettes(package = "readr")
```

1.3 Rangkuman

Beberapa hal dasar yang telah sampaikan pada bagian ini adalah:

1. R bisa digunakan dengan RStudio yang bisa di dapat secara gratis
2. Dalam mengolah data di R kita terbantu dengan package dan library yang ada yang dapat dinstall dengan menggunakan fungsi `install.package()`
3. Agar skrip lebih mudah dibaca dan dipahami kita bisa menambahkan komentar dengan didahului tanda pagar `#`
4. Untuk memulai menulis skrip kita sebaiknya terlebih dahulu membuat project dalam folder yang spesifik yang semuanya bisa diakses melalui menu `File`
5. Menggunakan library tidyverse dalam pengolahan data
6. Mengetahui beberapa fungsi dasar dan dari tidyverse yang cenderung akan lebih banyak digunakan seperti `class()`, `str()`, `summary()` dan beberapa fungsi lainnya yang bisa dipelajari lebih lanjut melalui dokumentasi yang disediakan pembuat package dan dapat diakses baik dengan menggunakan fungsi `help()` maupun `browseVignettes(package = "dplyr")`.

Chapter 2

Proses Ekstraksi Informasi

Pada bagian ini kita akan mempelajari bagaimana cara untuk mengekstrak informasi dari teks sosial media, terutama Twitter. Ada empat langkah utama dalam tahap ini, yaitu: impor/ekspor data, preprocessing, eksplorasi, dan visualisasi. Kita akan membahas secara singkat masing-masing langkah.

2.1 Impor dan Ekspor Data

Pada pengolahan data menggunakan R, direkomendasikan menggunakan tipe data file CSV (`Comma Separated Value`). Secara sederhana, file CSV merupakan file tabel yang serupa dengan XLS namun dengan variasi delimiter atau pemisah nilai. File CSV dapat diolah sebagaimana XLS dalam aplikasi Microsoft Excel.

2.1.1 Membaca data (Import)

Untuk membaca file CSV dalam lingkungan R, ada banyak cara yang bisa dilakukan. Cara pertama yaitu dengan menggunakan fungsi yang sudah tersedia pada R, yaitu `read.csv()` dengan contoh perintah sebagai berikut

```
df <- read.csv(nama_file_csv)
```

Cara tersebut dapat digunakan untuk membaca CSV berukuran kecil. Sedangkan bila kita ingin membaca file CSV berukuran relatif besar, direkomendasikan menggunakan library `readr` seperti yang sudah disinggung pada bab sebelumnya. Penggunaan library ini memungkinkan file dibaca jauh lebih cepat dibandingkan metode sebelumnya. Perintah yang digunakan adalah

```
df <- readr::read_csv(nama_file_csv)
```

Hasil pembacaan yang disimpan dalam `df` merupakan sebuah data tabel.

2.1.2 Menyimpan data (Ekspor)

Kita dapat menggunakan fungsi dari `readr` yaitu `write_csv()` untuk menyimpan data tabel ke dalam file CSV. Perintah yang digunakan adalah sebagai berikut:

```
readr::write_csv(nama_file_csv)
```

2.1.3 Menampilkan data

Untuk menampilkan data, disediakan perintah `print()`. Contoh penggunaannya sebagai berikut:

```
print(df)
```

Perintah di atas akan menampilkan isi dari `df`. Namun jika jumlah barisnya banyak, maka data hanya akan ditampilkan beberapa saja. Untuk menampilkan kolom tertentu dalam suatu data tabel, kita perlu menambahkan `$<namakolom>` seperti contoh berikut:

```
print(df$text)
```

Perintah tersebut akan menampilkan kolom `text` pada data tabel `df`.

2.2 Pre-processing

Preprocessing merupakan sebuah langkah yang perlu dilakukan sebelum data siap untuk diproses atau dianalisis. Ada banyak jenis langkah preprocessing yang dapat dilakukan, namun langkah ini harus disesuaikan dengan data yang kita miliki. Pada kesempatan ini, kita akan melakukan preprocessing untuk teks yang berasal dari media sosial.

2.2.1 Menghapus karakter non-ASCII

Karakter teks yang digunakan di Indonesia menggunakan standar ASCII, namun pada banyak negara terdapat karakter-karakter yang tidak terdapat dalam standar ASCII. Pada kasus tersebut, digunakan standar yang lain, yaitu Unicode yang memiliki variasi karakter lebih banyak. ASCII pada dasarnya adalah bagian dari Unicode. Pada teks sosial media, seringkali karakter non-ASCII tertulis dan itu cukup membuat sulit dalam pengolahan teks, karena kita menggunakan ASCII. Oleh karena itu, pada langkah pertama preprocessing, kita harus menghapus karakter non-ASCII tersebut.

Ciri karakter non-ASCII yang terlihat adalah adanya format seperti `<U+...>` pada teks yang kita miliki. Untuk menghapusnya, kita akan menggunakan perintah `global substitution` sebagai berikut:

```
text <- gsub("[<].*[>]", "", text)
```

Perintah di atas bermakna mengganti semua teks dengan format `<..>` dalam variabel `text` dengan kosong, atau dengan kata lain, menghapusnya kemudian menyimpannya kembali ke dalam variabel `text`. Ini salah satu strategi dalam menghilangkan karakter non-ASCII

2.2.2 Menghapus alamat URL

Sering kita temui pada teks sosial media, alamat URL sebuah website atau sejenisnya. Tentu kita tidak memerlukan alamat URL ini pada analisis selanjutnya. Untuk menghapusnya, bisa digunakan perintah:

```
text <- gsub('http\\S+\\s*',"", text)
```

Dengan menggunakan perintah tersebut, semua teks dengan format `http...` (format alamat URL) akan dihapus. ### Menghapus tanda baca Tanda baca menjadi hal selanjutnya yang akan kita hapus untuk mendapatkan teks yang siap untuk dianalisis. Metode yang bisa digunakan adalah dengan memanfaatkan perintah `gsub` sama seperti sebelumnya dengan pola yang berbeda.

```
text <- gsub("[^[:alnum:][:space:]#@]", "", text)
```

Penjelasan mengenai perintah tersebut * Tanda `[^ ]` bermakna ambil selain pola yang ada di dalam kurung siku * Teks `[:alnum:]` bermakna semua huruf dan angka (alfanumerik) * Teks `[:space:]` bermakna spasi

* Karakter `#@` bermakna literal Sehingga perintah itu secara umum bermakna menghapus semua karakter selain huruf, angka, spasi, `#` dan `@`. Dua yang terakhir sengaja tidak kita hapus karena akan digunakan untuk analisis akun dan tagar pada tahap selanjutnya.

2.2.3 Menghapus tanda ganti baris

Pada teks sosial media maupun teks pada umumnya, kita akan sering menjumpai karakter `\n` yang merupakan karakter pindah baris (ada ketika kita menekan `enter` pada keyboard). Karakter ini normalnya tidak akan terlihat, namun ketika kita ambil teks dalam format ASCII, karakter ini akan diterjemahkan menjadi `\n`. Untuk menghapusnya, kita dapat menggunakan perintah berikut:

```
text <- gsub("\n"," ",text)
```

Jika diperlukan kita bisa menggunakan cara yang sama untuk menghapus karakter sejenis lain, misalnya `\t` yang berarti `tab`. Namun karakter tersebut sangat jarang ada di sosial media twitter.

2.2.4 Mengubah ke huruf kecil

Menyeragamkan huruf kapital menjadi hal yang sangat penting dalam pengolahan teks. Hal ini dikarenakan komputer akan menganggap `teks` berbeda dari `Teks`. Oleh sebab itu, dengan perintah bawaan R, `tolower()`, kita akan membuat semua karakter ke dalam huruf kecil.

```
text <- tolower(text)
```

2.3 Eksplorasi

Pada eksplorasi ini, kita akan mencoba mencari tahu tentang akun paling banyak disebut, tagar paling sering digunakan, serta kata yang paling sering ditulis. Namun sebelumnya satu langkah yang harus dilakukan adalah proses tokenisasi.

2.3.1 Tokenisasi

Tokenisasi pada dasarnya adalah proses membagi teks yang berupa kalimat atau paragraf menjadi bagian-bagian tertentu. Dalam konteks ini, kita akan membagi kumpulan kalimat ke dalam kumpulan kata-kata. Untuk melakukan tokenisasi, kita bisa menggunakan metode dari library `tidytext` yaitu `unnest_tokens()`. Contoh perintahnya adalah sebagai berikut:

```
df_new <- tidytext::unnest_tokens(df,word,text,token='regex',pattern="[:space:]")
```

Penjelasan dari kode tersebut adalah sebagai berikut:

- `df_new` merupakan data frame tempat menyimpan hasil tokenisasi
- `tidytext::unnest_tokens` perintah untuk memanfaatkan fungsi `unnest_tokens` yang berasal dari library `tidytext`
- `df` adalah data frame awal yang belum diproses
- `word` adalah nama kolom yang kita buat untuk menampung hasil dari tokenisasi
- `text` adalah nama kolom dari data frame `df` dimana berisi teks yang akan ditokenisasi
- `token='regex'` berarti kita menggunakan metode tokenisasi dengan memanfaatkan pola regex. Regex merupakan sebuah pola karakter yang lazim digunakan untuk pencarian teks tertentu.
- `pattern="[:space:]"` berarti bahwa pola regex yang akan kita gunakan adalah spasi. Perintah ini juga bermakna bahwa teks akan kita pisah-pisah berdasarkan spasi, sehingga akan menghasilkan kata-kata.

Table 2.1: Hasil tokenisasi

word
saya
mereka
makan

Hasil dari perintah di atas adalah sebuah data frame `df_new` dengan satu kolom bernama `word` yang berisi satu kata per baris. Contohnya dapat dilihat pada Tabel 2.1.

2.3.2 Akun paling banyak disebut

Untuk mencari siapa akun-akun yang paling banyak disebut dalam menulis twit, pada dasarnya kita cukup melakukan filter dari data frame hasil tokenisasi sebelumnya, kemudian menghitung frekuensi munculnya tiap akun. Pada langkah ini, kita akan menggunakan bantuan fungsi `filter()` dari library `dplyr` dan fungsi `str_detect()` dari library `stringr`. Contoh penggunaannya dapat dilihat pada perintah berikut:

```
df_akun <- dplyr::filter(df_new,stringr::str_detect(word, "@"))
```

- `dplyr::filter()` merupakan fungsi yang digunakan untuk melakukan filtering pada data frame sesuai dengan kondisi tertentu
- `stringr::str_detect()` digunakan untuk mencari baris tertentu pada suatu kolom yang memiliki pola tertentu

Perintah di atas bermakna kita hanya akan mengambil baris-baris pada data frame `df_new` kolom `word` yang mengandung karakter `@`, dengan kata lain berupa akun, kemudian hasilnya disimpan pada `df_akun`.

Setelah semua akun terambil, kita dapat menghitung frekuensi masing-masing dengan mengubah data frame `df_akun` ke dalam format tabel. Secara otomatis R akan menghitung frekuensi dari masing-masing baris yang sama. Kemudian kita ubah `df_akun` kembali ke data frame untuk bisa ditampilkan dan diolah lebih lanjut.

```
df_akun <- table(df_akun)
df_akun <- as.data.frame(df_akun)
```

Selanjutnya untuk mengurutkan data berdasar frekuensi, kita dapat menggunakan fungsi `arrange()` dari library `dplyr` sebagai berikut.

```
df_akun <- dplyr::arrange(df_akun,desc(Freq))
```

Kode di atas akan menghasilkan data frame `df_akun` yang berisi daftar akun dan frekuensinya yang telah diurutkan secara descending.

2.3.3 Tagar paling sering digunakan

Untuk mencari frekuensi tagar yang digunakan, kita dapat menggunakan cara yang mirip dengan langkah mencari akun di atas. Hanya saja untuk filter yang digunakan, karakter `@` diganti dengan `#` seperti berikut.

```
df_tagar <- dplyr::filter(df_new,stringr::str_detect(word, "#"))
```

Langkah selanjutnya sama dengan sebelumnya.

2.3.4 Kata paling sering ditulis

Untuk mencari frekuensi kata selain nama akun dan tagar, kita butuh mengubah pola pencarian yang digunakan. Contoh yang bisa digunakan adalah sebagai berikut.

```
df_kata <- dplyr::filter(df_new,stringr::str_detect(word, "^((?!@|#).)*$"))
```

Perintah di atas akan mencari kata selain yang berawalan `@` dan `#` dalam data frame `df_new` kemudian menyimpannya ke dalam data frame `df_kata`.

2.4 Visualisasi

Untuk visualisasi dari data yang sudah didapat, kita dapat menggunakan bar chart. Implementasi bar chart dalam R didapat dengan memanfaatkan library `ggplot` dengan contoh kode sebagai berikut:

```
ggplot(head(df_akun,n=10),aes(x=reorder(col_akun,Freq),y=Freq)) +
  geom_col() +
  coord_flip() +
  labs(title='10 akun paling banyak disebut',x='nama akun',y='jumlah')
```

- `head(df_akun,n=10)` maksudnya ialah mengambil 10 baris pertama dari data frame `df_akun`
- `aes()` merupakan perintah untuk mendeskripsikan variabel mana yang akan ditampilkan dalam sumbu grafik. Perintah `reorder(col_akun,Freq)` bermakna data pada kolom `col_akun` akan diurutkan berdasarkan kolom `Freq`
- `geom_col()` merupakan perintah yang bermakna grafik yang kita buat merupakan bar chart
- `coord_flip()` merupakan perintah untuk menukar sumbu pada grafik. Pada kasus ini, kita menggunakan perintah ini untuk membuat horizontal bar chart
- `labs()` digunakan untuk memberi label

Chapter 3

Latihan

Dalam latihan kali ini terdapat dua hal utama yang akan kita coba. Pertama, mendapatkan data. Kedua, mengeksplorasi data. Data yang akan coba didapatkan berupa twit dengan memanfaatkan API basic. Menggunakan API basic tersebut kita bisa mendapatkan tweet sejauh 7 hari ke belakang dengan jumlah paling banyak sekitar 40 ribu.

3.1 Mendapatkan data

Untuk mendapatkan data dari Twitter kita bisa menggunakan dua cara. Pertama dengan menggunakan API. Kedua dengan melakukan scrapping. Pada kesempatan ini kita akan lebih dahulu mencoba mendapatkan data dengan menggunakan API basik. Untuk bisa menggunakan API basic kita harus terlebih dahulu mendaftar di: https://apps.twitter.com/. Setelah mendafatar dengan mengikuti prosedurnya, masuklah pada aplikasi yang telah anda dan dapatkan `api key`, `api secret`, `access token` dan `Access Token Secret` pada bagian `Keys and Access Token` seperti tampak pada gambar berikut.

Selanjutnya, anda bisa menggunakan script dibawah ini untuk mengatur penggunaan API dalam R dengan menggunakan library `twitteR`. Untuk itu langkah-langkah yang diperlukan adalah:

1. Memanggil library

```
library(twitteR)
```

2. Setting API

```
api_key <- "2OKUwihqyw3PCm4tSfGZHDXum"
api_secret <- "6ykMH9XdMOQnduj4cAI6lyGRRKG1abZU4TdQFdE5HZ4rKq1M4g"
token <- "73705532-WlCKXW7Cjd2U2fcSUflTOnoLEONrk26gy6xddFzeM"
token_secret <- "JrUQjxTSx3QSTAGQnL1lnsO2ua8g4LKDV6xzZ4iJW3Rwh"
```

3. Setting permission access

```
setup_twitter_oauth(api_key, api_secret, token, token_secret)
```

Ketika anda menjalankan script di atas pada bagian `console` akan ada perintah untuk mengonfirmasi. `Tekan 1 dan enter untuk menlanjutkan`.

4. Proses ambil data

Dalam proses ini kita membutuhkan akses internet yang akan menentukan lama atau cepatnya pengambilan data. Di dalam librar `twitteR` terdapat beberapa fungsi yang bisa digunakan untuk mengumpulkan data seperti untuk mendapatkan teks/twit yang berasal dari letak geografis dan jam tertentu atau untuk

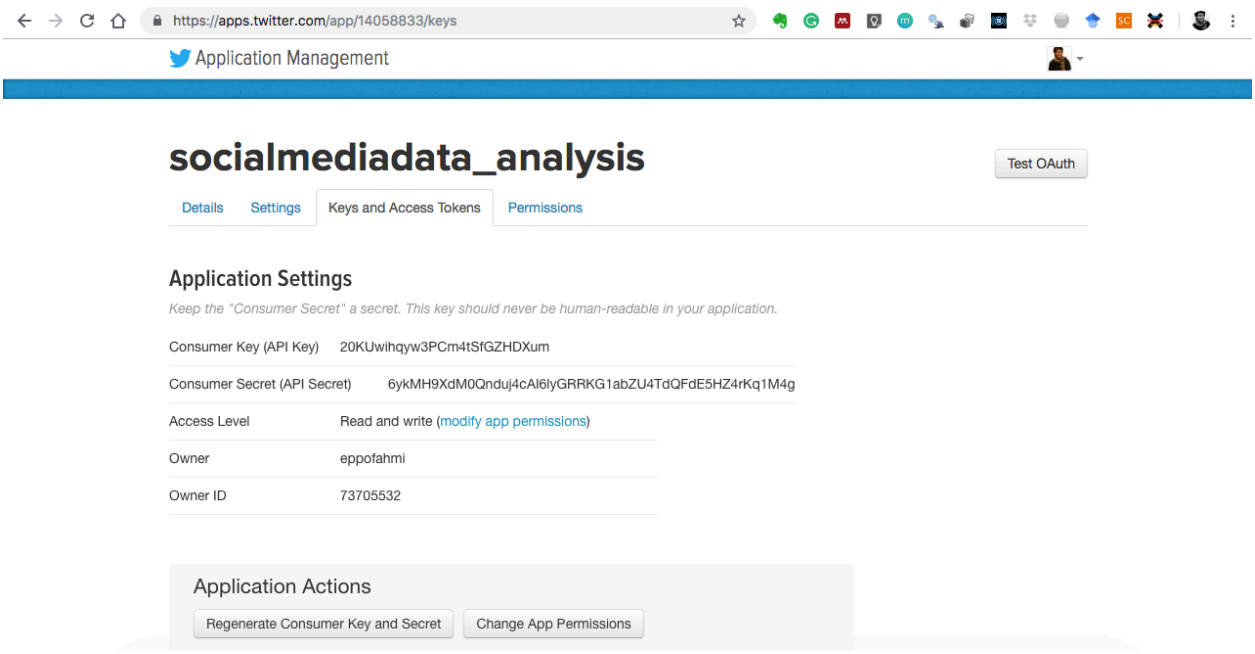

Figure 3.1: Keys and Access Token Twitter.

mendapatkan timeline sebuah nama akun (username). Pada kesempatan ini kita akan menggunakan fungsi `searchTwitter` untuk mendapatkan twit sebanyak 1000 (`n = 1000`) dengan kata kunci tagar **#bubarkanbanser**.

```
banser <- searchTwitter("#bubarkanbanser", n = 1000) # collect tweets
banser <- twListToDF(banser) # mengubah format data menjadi data frame
write.csv(banser, "contoh_data.csv") # menyimpan data
```

Data yang didapat berupa list, untuk itu pada script bari kedua kita akan mengubahnya menjaid data frame agar lebih mudah dieksplorasi. Selanjutnya data yang didapat kita simpan dalam directory yang sudah kita tentukan sebelumnya saat membuat `project`.

3.2 Eksplorasi data

Di dalam proses mengeksporasi data, hal pertama yang harus diketahui adalah datanya itu sendiri. Misalnya berapa jumlah observasi, variabel, jenis variabel dan lain sebagainya. Pada konteks data yang kita dapatkan dengan metode di atas, terdapat 1000 observasi dan 16 variabel, seperti dapat dilihat pada bagian `Environment` (sebelah kanan atas). Untuk mendapatkan tilikan dengan cepat kita bisa menggunakan fungsi `summary` seperti berikut.

```
summary(banser)
```

Fungsi di atas akan memberikan rangkuman semua variabel atau kolom yang ada. Di mana dengan melihat rangkuman tersebut di antarnya kita akan segera mengetahui kapan twit pertama dan terakhir di kirim dalam data. Selain itu, kita juga bisa mengetahui jenis data pada masing-masing kolom serta beberapa statistik dasar. Sehingga berdasarkan rangkuman tersebut kita bisa memutuskan untuk dapat menentukan hal apa yang akan di eksplorasi terlebih dahulu.

Dengan menggunakan materi sebelumnya (lihat Chapter 2.3), kita selanjutnya dapat mengeksplorasi akun paling banyak disebut, tagar paling sering digunakan dan kata paling banyak ditulis kolom text. Di mana

dalam kolom tersebut selain teks twit juga terdapat nama akun yang disebut oleh pengirimnya, tagar, dan konten lainnya.

Chapter 4

Contoh hasil kerja

Berikut ini adalah beberapa hasil penelitian yang memanfaatkan data dari media sosial di berbagai negara dalam berbagai kasus.

Jika dilihat dari sumber data yang digunakan dalam penelitian-penelitian di atas (4.1), maka kita bisa mengetahui bahwa data yang digunakan cukup banyak jika dikerjakan secara manual. Selain itu, beberapa penelitian memang tidak menyebutkan jumlahnya secara spesifik, tapi umumnya mereka menggunakan data secara menyeluruh atau dengan sampel yang memadai. Oleh karena itu, mereka banyak memanfaatkan bahwa pemrogaraman dan bahkan membuat aplikasi khusus untuk mengumpulkan data seperti dengan judul "Mapping the Public Agenda with Topic Modeling: The Case of the Russian LiveJournal" (Koltsova and Koltcov, 2013).

Table 4.1: Contoh jurnal yang memanfaatkan data dari media sosial

Judul	
Mapping the Public Agenda with Topic Modeling: The Case of the Russian LiveJournal	
Topic Modelling and Event Identification from Twitter Textual Data	
Politicians and the Policy Agenda: Does Use of Twitter by the U.S. Congress Direct New York Times Content?	2
Soft Data and Public Policy: Can Social Media Offer Alternatives to Official Statistics in Urban Policymaking?	C
Increasing the reach of government social media: A case study in modeling government-citizen interaction on Facebook	
Citizen-government collaboration on social media: The case of Twitter in the 2011 riots in England	2
Three dimensions of the public sphere on Facebook	
The "Social Side" of Public Policy: Monitoring Online Public Opinion and Its Mobilization During the Policy Cycle	A
Freedom to hate: social media, algorithmic enclaves, and the rise of tribal nationalism in Indonesia	3

Chapter 5

Sumber Belajar Mandiri

5.1 Komunitas

1. Kaggle

Kaggle dapat menjadi salah satu sumber utama bagi siapa saja yang ingin belajar atau menjadi data scientist. Kaggle menyediakan berbagai jenis data untuk latihan. Selain itu, Kaggle kita juga bisa belajar dari script yang dibuat oleh orang lain dalam mengelola data dan mengekstrak informasi dari sebuah data.

2. Stackoverflow

Stackoverflow adalah sebuah tempat bagi orang yang baru belajar hingga sudah mahir untuk bertanya dan menjawab pertanyaan terkait dengan script dari berbagai bahasa pemerograman yang ada di dunia. Di sini kita bisa mencari jawab atau langsung bertanya tentang kendala yang dihadapi dalam menulis script untuk mendapatkan suatu hasil yang dituju.

3. Github

Walaupun GitHub sebenarnya memiliki fungsi lain yang dapat dimanfaatkan dalam proses scripting. Namun di sini saya ingin menekankan bahwa GitHub banyak digunakan oleh para develovers untuk meletakan scriptnya. Terdapat dua versi, yaitu versi private dan publik. Untuk versi publik kita bisa menggunakannya sebagai sumber belajar dengan mengopi foldernya atau satu persatu.

5.2 Free Course

Saat ini diinternet banyak kursus dan tutorial daring yang dapat digunakan sebagai salah satu tempat untuk belajar. Misalnya kita bisa mencari di youtube, atau juga mengikuti kursus gratis di datacamp atau di Udemi.

Bibliography

Koltsova, O. and Koltcov, S. (2013). Mapping the public agenda with topic modeling: The case of the russian livejournal. *Policy & Internet*, 5(2):207–227.

R Core Team (2018). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'*. R package version 1.2.1.