3 LANGKAH KILAT MAHIR ANALISIS DATA

Pribadi Pramudya - pripramudya.com

2019-01-14

Pendahuluan

Selamat Datang

Terima kasih sudah bersedia menyempatkan sedikit waktu untuk mengunjungi website dan membaca buku ini.

Berikut materi yang akan Anda dapatkan:

- **Kenapa Anda perlu menguasai** keterampilan analisis data
- **Konsep dasar** analisis data yang tidak diketahui banyak orang
- Satu ***tool* pengolah data *powerful*** yang perlu Anda kuasai
- **Teknik dasar** analisis data yang perlu Anda pelajari
- **Panduan praktis dan rinci** cara menganalisis data dari nol sama sekali

Versi pdf buku elektronik pendek 3 Langkah Kilat Mahir Analisis Data bisa diunduh dengan mendaftarkan email Anda terlebih dulu di https://pripramudya.com.

Kenapa Anda Perlu Menguasai Keterampilan Analisis Data

Data ada di mana-mana.

Hampir setiap hari Anda berhadapan dengan data, apapun profesi Anda.

Apalagi di zaman *now*, data berkembang begitu pesat dalam hal jumlah *(volume)*, variasi *(variety)*, dan kecepatan *(velocity)*.

Ya, sekarang kita berada di era **big data**.

Begitu melimpah ruahnya, sehingga tidak banyak orang yang memiliki keterampilan mumpuni untuk mengambil *insight* (wawasan baru) dari banyaknya data tersebut.

Tidak perlu jauh mencari sumber data dari luar, data yang tersedia di dalam bidang keprofesian Anda pun sepertinya masih belum dapat digali dengan optimal.

Di tempat kerja saya, pegawai yang menguasai keterampilan analisis data memiliki peluang lebih baik dalam menanjak karir. Apalagi jika hasil analisis data ini digunakan sebagai bahan presentasi oleh kepala divisi. Karirnya meroket.

Ini salah satu contoh nyata. Manajemen begitu membutuhkan *insight* yang dapat memberikan suatu perubahan lebih baik dalam mengelola roda aktivitas perusahaan sehingga berujung pada peningkatan profitabilitas perusahaan. Manajemen tingkat atas pasti tidak akan ragu memberikan promosi bagi pegawai yang memberikan *insight* tersebut. Peluang *(opportunity)* penanjakan karir pegawai yang memiliki keterampilan analisis yang mumpuni lebih baik dari pegawai yang memiliki keterampilan ala kadarnya saja.

Saya akan meringkas bab ini cukup dengan dua kata saja:

INSIGHT & OPPORTUNITY

Ingin mendapatkan *insight* yang dapat menjawab permasalahan di bidang keprofesian Anda dan membuka *opportunity* karir terbaik Anda?

Kuasai keterampilan analisis data.

Konsep Dasar Analisis Data

“Saya sudah tahu apa itu analisis data, tidak perlu dijelaskan lagi.”

Mungkin hal tersebut yang ada di benak Anda ketika membaca judul di bab ini.

Memang, membaca teori tidaklah begitu menarik. Anda ingin langsung meloncat ke bab langkah kilat.

Silakan saja. Gunakan menu di daftar isi lalu klik ke bab yang ingin Anda baca.

Namun, tetap saya sarankan untuk membaca bab ini sampai akhir.

Anda perlu tahu konsep dasar analisis data agar memudahkan Anda agar bisa cepat menguasai keterampilan ini.

Apa itu Analisis Data

Jadi, apa sebenarnya analisis data itu?

Analisis data adalah proses mengetahui, memahami, memilah, memecah, merinci, mengurai data dengan berbagai cara atau teknik tertentu seperti manipulasi, visualisasi, pengelompokkan, perbandingan, atau teknik lainnya sehingga didapatkan informasi berguna yang dapat menjawab permasalahan serta membantu dalam pengambilan keputusan.

Ya, **tujuan dari analisis data adalah pengambilan keputusan yang lebih baik**. Keputusan yang didasarkan pada informasi hasil analisis data bukan berdasarkan naluri semata. Keputusan luar biasa krusialnya yang bisa saja membuat sebuah perusahaan sebelumnya berkibar berujung menjadi gulung tikar ataupun sebaliknya.

Ini membuat analisis data menjadi salah satu keterampilan yang sangat penting penerapannya di hampir seluruh bidang.

Beberapa contoh penerapan dalam bidang bisnis sebagai berikut:

Perkenalan produk kopi terbaru Starbucks

Starbucks memperkenalkan produk kopi terbaru dengan memperhatikan apakah pelanggan menyukai produk baru tersebut. Pada pagi hari produk tersebut mulai dijual, Starbucks memantau berbagai blog, Twitter, forum diskusi dan grup tentang kopi untuk mengetahui bagaimana reaksi pelanggan. Siang harinya, Starbucks menemukan bahwa walaupun pelanggan menyukai rasa kopinya, namun mereka berpikir harganya terlalu mahal. Starbucks langsung menurunkan harga. Pada sore harinya, semua persepsi negatif terkait produk kopi baru tersebut hilang.

Pendekatan dengan respons cepat seperti ini tentu lebih baik dibandingkan pendekatan tradisional dengan menunggu laporan penjualan harian masuk lebih dulu yang pada akhirnya ditemukan bahwa penjualannya mengecewakan. Langkah *jadul* berikutnya mencari tahu alasan kenapa penjualan mengecewakan dengan melakukan diskusi grup terpusat *(focus group discussion)*. Selanjutnya, beberapa minggu kemudian Starbucks baru menemukan bahwa harganya terlalu mahal lalu menurunkan harganya.

Sistem rekomendasi Amazon

Amazon menggunakan data penjualan, data produk yang sering dilihat, dan perilaku serta preferensi pelanggan lainnya untuk membuat sebuah sistem rekomendasi. Sistem ini membantu Amazon meningkatkan 29% penjualannya. Jika Anda mengunjungi situs Amazon, di bagian bawah Anda akan melihat beberapa produk rekomendasi, produk yang Anda lihat sebelumnya, sampai dengan produk terlaris serta tawaran produk menarik lainnya.

Efisiensi Chevron dalam aktivitas pengeboran

Biaya setiap pengeboran di selat Meksiko diperkirakan sampai dengan $100 juta. Rugi sekali jika hasil pengeboran tersebut tidak didapat sumber minyak bumi. Untuk meningkatkan peluang keberhasilan menemukan sumber minyak, Chevron menganalisis 50 terabyte data seismik. Dengan dukungan kecanggihan komputer, kapasitas penyimpanan, serta tim analis yang mumpuni berhasil meningkatkan keberhasilan menemukan sumber minyak dari yang sebelumnya 1 dari 5 pengeboran menjadi 1 berbanding 3.

Proses dan Tahapan Analisis Data

Untuk menghasilkan informasi dan *insight* yang dapat memberikan keputusan lebih baik, Anda harus tahu proses dan tahapan dari analisis data. Tahapan analisis data terdiri dari 6 tahapan.

1. Menentukan Tujuan
2. Menentukan Metode Pengukuran yang Digunakan

3. Pengumpulan Data
4. Pembersihan dan Pembentukan Data
5. Analisis Data
6. Interpretasi dan Komunikasi Hasil Analisis Data

1. Menentukan Tujuan

Ini adalah tahap pertama dari proses analisis data. Di bidang keprofesian Anda, Anda harus menentukan tujuan dari permasalahan yang akan diselesaikan dimulai dengan pertanyaan yang tepat. Tahap ini cukup vital mengingat pertanyaan yang salah akan menghasilkan informasi atau *insight* yang tidak tepat pula.

“Kenapa total penjualan bulan ini turun dibandingkan bulan lalu?”

“Kenapa 7 dari 10 proyek yang sedang berjalan saat ini terlambat?”

“Bagaimana cara mencegah pelanggan kita pindah ke kompetitor?”

Itu adalah contoh permasalahan yang akan disolusikan.

Dengan menentukan pertanyaan yang tepat, Anda bisa lebih fokus dalam mencapai tujuan serta lebih efektif untuk menyelesaikan permasalahan yang ada.

Sumber data yang akan digunakan nantinya juga tergantung dari penentuan tujuan dan pertanyaan tepat dari tahap ini.

2. Menentukan Metode Pengukuran yang Digunakan

Anda sudah meneentukan tujuan di tahap pertama, selanjutnya adalah apa yang perlu dijadikan tolak ukur dan bagaimana cara Anda mengukurnya.

Misalnya Anda mengambil contoh:

“Kenapa total penjualan bulan ini turun dibandingkan bulan lalu?”

Dari permasalahan ini, Anda bisa tentukan bahwa yang harus diukur adalah **penjualan**. Lalu, Anda tentukan lagi metrik dari penjualan, **jumlah barang** atau **jumlah uang** atau keduanya.

- Metrik jumlah barang: buah, lusin, kodi, dus, porsi, dan lainnya.
- Metrik jumlah uang: ratusan ribu rupiah, jutaan rupiah, ribuan dollar, dan lainnya.

Pengukuran juga bisa dibandingkan berdasarkan periodenya. Terhadap bulan sebelumnya atau terhadap tahun sebelumnya di bulan yang sama.

Selain itu, Anda pikirkan juga kira-kira faktor penyebab permasalahan yang bisa dijadikan sebagai tambahan pengukuran. Contoh di bidang penjualan, faktor seperti jumlah pembeli, promosi, diskon, lokasi, harga, bisa dimasukkan sebagai tambahan metode pengukuran.

Pastikan pengukuran ini dilakukan secara tepat karena nantinya akan berpengaruh terhadap kualitas hasil analisis dan pengambilan keputusan.

3. Pengumpulan Data

Anda sudah menentukan tujuan dan juga metode pengukuran di tahapan sebelumnya. Selanjutnya adalah mencari dan mengumpulkan data yang relevan. Sumber data bisa dari mana saja. Internet, basis data internal, survey, wawancara, dan lainnya.

Contoh permasalahan penjualan di tahapan sebelumnya. Data penjualan, promosi, jumlah pelanggan, bisa didapat dari internal. Untuk survey kepuasan pelanggan bisa dilakukan dengan metode pengisian kuesioner ataupun dengan wawancara langsung.

Yang perlu diingat adalah Anda harus mengumpulkan berbagai macam data tersebut secara terstruktur sehingga bisa langsung dapat diolah.

4. Pembersihan dan Pembentukan Data

Dari seluruh tahapan proses analisis data, tahap ini adalah tahap yang paling banyak menyita waktu Anda. Ketika Anda membuka data yang Anda kumpulkan di tahap 3, Anda menemukan struktur data yang tidak siap olah. Berantakan. Duplikasi, salah ketik, karakter spesial, huruf kapital-kecil, data kosong atau NA, format tanggal/waktu tidak sama, dan lainnya merupakan beberapa hal yang sering ditemui pada sebuah basis data.

Untuk mempermudah pembersihan dan pembentukan data ini Anda bisa menggunakan *tools* sejuta umat, Microsoft Excel. Anda juga bisa menggunakan software pengolah data lain yang Anda kuasai, misalnya SPSS, SAS, STATA, Python, R atau lainnya.

Jangan terlalu menganggap remeh tahapan ini. Buat dan bentuk data sebaik mungkin sehingga data bisa langsung dianalisis. Jika data tidak bersih, semahir apapun Anda dalam menganalisis data, hasilnya tidak akan optimal. Ingat, jika masukannya sampah, keluarannya juga sampah.

5. Analisis Data

Setelah data dibersihkan dan dibentuk, tahapan penting berikutnya adalah tahap analisis data. Tujuan dari tahapan ini adalah untuk mengerti lebih dalam tentang data beserta variabel-variabelnya. Ada beberapa teknik dasar analisis data yang dapat Anda gunakan.

- Teknik Eksplorasi
- Teknik Visualisasi

Teknik Eksplorasi

Sesuai dengan namanya, teknik menjelajah untuk mengerti tentang data. Anda memulainya dengan mencari tahu berapa jumlah baris dan kolom dari basis data, lalu melihat jenis variabelnya: karakter, numerik, atau kategorikal. Dilanjut dengan meringkas data tersebut sehingga dapat menampilkan informasi penting seperti variabel atau kategori dengan frekuensi terbanyak, nilai tertinggi, rerata, dan informasi lainnya.

Intinya Anda menggunakan fungsi-fungsi dari *tools* analisis untuk mengerti tentang data. Terkadang Anda perlu kembali melakukan proses di tahapan sebelumnya, yakni pembersihan data lalu balik lagi ke tahap ini. Ya, tahap 4 dan 5 merupakan proses yang iteratif. Hal ini dilakukan agar Anda bisa lebih cepat mengerti tentang informasi yang tersimpan dalam data.

Teknik Visualisasi

> “Sebuah gambar bernilai ribuan kata.”

Salah satu pepatah bahasa inggris ini benar adanya. Anda lebih cepat mengerti data dengan mereprentasikannya dalam bentuk grafik seperti diagram batang, diagram garis, diagram tebar, histogram, serta info grafik lainnya. Teknik visualisasi merupakan teknik termudah, tercepat, dan terefektif dalam menampilkan informasi tentang data.

Selain itu, hasil dari teknik visualisasi ini bisa digunakan sebagai alat bantu dalam mengkomunikasikan informasi dan insight.

Jika ada satu teknik analisis data yang saya rekomendasikan untuk dikuasai pertama kali, maka teknik visualisasi inilah yang perlu Anda pelajari secepatnya.

6. Interpretasi dan Komunikasi Hasil Analisis Data

Hasil analisis yang sudah Anda selesaikan harus diinterpretasikan. Maksud dari interpretasi ini adalah mengubah hasil analisis yang masih dalam bentuk teknikal menjadi temuan, informasi, ataupun *insight* yang bisa dimengerti oleh orang awam sekalipun.

Selanjutnya, apakah hasil interpretasi ini sudah menjawab pertanyaan permasalahan di proses tahap 1.

Buat kesimpulan, akan lebih baik lagi jika disertai rekomendasi dan langkah-langkah yang diperlukan untuk menyelesaikan permasalahan yang ada.

Sebaik apapun Anda dalam menganalisis data itu semua akan percuma jika Anda tidak mengkomunikasikan rekomendasi, informasi, dan insight ke atasan, manajemen, klien, pelanggan, masyarakat, ataupun *stakeholder* Anda. Bukan berarti teknik analisis data tidak penting sama sekali, namun dengan interpretasi dan mengkomunikasikan hasil analisis maka dampak efektivitas dalam mensolusikan permasalahan akan lebih terasa.

Media dalam mengkomunikasikan pun terdapat banyak alternatif. Anda bisa membuat hasil analisis data dalam bentuk infografik (file gambar), tulisan dalam format pdf, slide presentasi, *spreadsheet*, dahsboard, sampai dengan artikel di blog.

Dengan mengkomunikasikan hasil analisis data, maka pintu *opportunites* terbuka lebih lebar, mengundang karir terbaik Anda.

3 Langkah Kilat Mahir Analisis Data

Anda sudah mengetahui tentang konsep dasar data analisis yang menjadi landasan dan pondasi Anda dalam mempelajari keterampilan analisis data pada bab sebelumnya. Konsep tersebut merupakan hal yang penting agar Anda dapat cepat mahir dan menguasai keterampilan analisis data.

Di bab selanjutnya akan dibahas bagaimana mempelajari dan mengaplikasikan teknik analisis data. Anda akan dipandu secara rinci bagaimana agar Anda dapat dengan cepat menguasai keterampilan analisis data dengan 3 langkah kilat berikut.

Langkah Kilat 1 Instal dan Pelajari Satu *Tool* Analisis Data yang *Powerful* ini

Salah satu yang mempengaruhi proses dan kualitas hasil analisis adalah dari *tool* yang dipakai. *Tool* atau perangkat lunak ini memudahkan Anda sebagai analis untuk melakukan proses analisis data secara efektif dan efisien. Selain itu, jika Anda menguasai dan mahir salah satu *tool* analisis data maka peluang dalam membuka karir terbaik Anda akan lebih tinggi. Begitu pentingnya pemilihan *tool* ini, sehingga saya akan merekomendasikan terlebih dahulu beberapa alternatif *tool* analisis data yang dipakai banyak orang.

1.1 *Tool* atau Perangkat Lunak Analisis Data Populer yang Banyak Orang Pakai

Ada 3 *tool* yang akan dibahas di bagian ini:

- Microsoft Excel
- Python
- R

1.1.1 Microsoft Excel

Software nomor satu yang paling banyak dipakai orang di dunia.

Microsoft pertama kali membuat dan memasarkan sebuah *spreadsheet program* bernama Multiplan di tahun 1982 yang sangat populer di CP/M sistem (sistem operasi mikrokontroler), namun kalah bersaing dengan Lotus 1-2-3 di MS-DOS. Hal ini membuat Microsoft mengembangkan versi terbaru dari *spreadsheet program* bernama Excel yang dimaksudkan untuk menyaingi seluruh fitur Lotus 1-2-3 dan membuatnya lebih baik.

Versi pertama Excel dipasarkan untuk Mac di tahun 1985 dan pada Nopember 1987 untuk Windows. Di tahun 1988, Excel berhasil memimpin pangsa pasar *spreadsheet program* menyingkirkan Lotus 1-2-3 dan menjadi *software* yang paling banyak dipakai di seluruh dunia.

Yang pasti Anda sudah familiar dan mahir menggunakan *sofware* ini. Jika Anda sedang mencari pekerjaan kantoran, maka keterampilan dasar yang dijadikan sebagai syarat di sebagian besar perusahaan adalah bisa menggunakan *sofware* Excel. Saking mudahnya mengoperasikan Excel, pengguna pemula tidak perlu belajar khusus untuk menguasainya. Tidak perlu tahu tentang *programming*. Cukup bermodalkan *point, click,* dan *drag*, Anda sudah bisa melakukan analisis data dengan *software* ini.

1.1.2 Python

Python adalah salah satu bahasa pemrograman untuk membuat berbagai macam aplikasi *(general purpose programming)*. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, Python menjadi salah satu bahasa pemrograman yang banyak digunakan di dunia. Kemudahan *syntax*, lisensi *open source*, serta dukungan komunitas merupakan beberapa faktor yang membuat popularitas Python melesat di beberapa tahun belakangan ini.

Python juga menjadi salah satu *tool* yang sering digunakan untuk analisis data. Dukungan dari banyaknya paket membuat Python sangat fleksibel. Ada NumPy dan Pandas untuk manipulasi data, SciPy untuk perhitungan *science*, Matplotlib untuk data visualisasi, serta paket-paket lainnya.

IDE *(Integrated Development Environment)* Python juga ada banyak pilihan. Mulai dari Spyder, Jupyter Notebook, Atom, PyCharm, dan Rodeo. Banyaknya pilihan dan fleksibilitas serta kemudahan dalam programming membuat Python menjadi *tool* nomor satu untuk data analisis saat ini.

1.1.3 R

R adalah perangkat lunak gratis untuk komputasi dan analisis data statistik yang juga dapat menampilkan visualisasi grafik. R didukung oleh ribuan *packages* yang dibuat oleh banyak kontributor aktif sehingga memudahkan pengguna dalam pengembangan aplikasi.

R pertama kali dibuat oleh Ross Ihaka and Robert Gentleman di University of Auckland, New Zealand pada tahun 1992. Saat ini dikembangkan oleh R Development Core Team. Dinamakan R diambil dari abjad pertama kedua pembuat program ini. Versi pertama dirilis pada tahun 1995 dan versi beta yang stabil pada tahun 2000.

Seperti halnya Python, bahasa pemrograman memiliki lisensi *open source*, dan dapat diinstal di seluruh sistem operasi: Windows, Linux, ataupun Mac. Komunitasnya pun banyak ditemukan di website seperti Stackoverflow ataupun di Github.

Terkait popularitas, R masih kalah dibandingkan dengan Python. R nomor dua dalam hal *tool* analisis data paling banyak digunakan dan paling populer setelah Python.

Anda masih ingat judul bab ini? Tentang *tool* analisis data *powerful*.

Ya, **R adalah *tool* analisis data *powerful* yang perlu Anda pelajari dan kuasai.**

1.2 Kenapa harus R

Anda saat ini pasti bertanya-tanya kenapa harus belajar R? Kenapa tidak Excel saja atau Python?

Yang pertama, kenapa tidak Excel saja? Karena Anda pasti sudah cukup mahir menggunakannya. Apakah hanya Anda yang mahir Excel? Tentu saja, tidak hanya Anda yang mahir, tetapi juga banyak orang. Menurut saya, orang yang minimal berpendidikan SMA pasti mengenal Excel.

Jika Anda dalam posisi bersaing untuk mendapatkan karir yang lebih baik di suatu perusahaan atau mencari pekerjaan baru di perusahaan lain, maka Anda perlu memiliki *skill* yang berbeda dibandingkan kebanyakan orang.

R atau Python dapat memberikan perbedaan itu.

Yang kedua, kenapa tidak Python?

Karena saya belum pernah pakai Python, he he :D

Pada saat saya bertekad untuk belajar analisis data, saya memilih R sebagai *tool* utama. Python dan R sama-sama powerful, tinggal preferensi masing-masing saja. Jika Anda memiliki latar belakang *software developer*, mungkin Anda lebih cocok dengan Python, begitu pun sebaliknya. Saya memilih R karena saya bukan *software developer*. R sangat membantu meningkatkan produktifitas dalam menganalisis data di tempat kerja saya. R mungkin cocok buat Anda yang tidak memiliki latar belakang di bidang komputer atau *software*.

Setidaknya ada 3 alasan kenapa Anda harus mencoba belajar R:

1.2.1 Dukungan Paket R

Jumlah paket R yang bisa didownload sampai dengan sekarang ini sebanyak 13.409 di CRAN (Comprehensive R Archive Network), *repository* R. Paket tersedia sebanyak ini belum termasuk ribuan lainnya yang ada di GitHub, juga paket komersial yang dikembangkan oleh

vendor besar seperti Microsoft dan Oracle.

Kenapa bisa begitu banyak paket?

- R memiliki banyak komunitas pengembang sangat aktif yang berkontribusi dalam membuat paket-paket baru ke CRAN setiap hari. Hal ini membuat hampir segala analisis yang ingin Anda lakukan sudah tersedia dalam bentuk paket di CRAN.
- R dibuat sejak tahun 1992, yang berarti sampai dengan saat ini sudah berusia lebih dari 20 tahun. Selama itu pula, berbagai macam paket terus dikembangkan sehingga R menjadi salah satu *tool* analisis data yang memiliki banyak pilihan paket dibandingkan *tool* analisis data lainnya.
- R banyak digunakan sebagai *tool* utama untuk penelitan statistik. Ketika metode baru dikembangkan, hasil penelitian tersebut tidak hanya dipublikasikan dalam bentuk makalah atau jurnal tetapi juga dalam bentuk paket. Ini membuat R selalu terdepan dalam mengadopsi metodologi terbaru.
- R didesain sebagai bahasa antar muka bahasa pemrograman lainnya. Banyak paket berfungsi sebagai pendukung perangkat lunak *open source* lainnya sehingga membuat R menjadi *tool* yang mempermudah kolaborasi berbagai macam metode dan algoritma pemrograman.
- Sistem CRAN merupakan platform yang sangat efektif dalam proses kolaborasi pengembangan paket R. Dengan sistem yang sudah teratur, pengguna bisa dengan mudah membuat, mengembangkan, mengetes, dan mendistribusikan paket.

1.2.2 Belajar Analisis Data lebih mudah menggunakan R

R menyediakan paket pembelajaran yang dinamakan swirl. Swirl mengajarkan Anda pemgrograman R dan analisis data secara interaktif langsung di dalam aplikasi R sesuai kecepatan pemahaman Anda.

Selain adanya paket belajar mandiri di swirl, R juga memiliki syntax atau perintah yang lebih simpel dan lebih mudah diingat sehingga dapat mempercepat proses pembelajaran Anda. Hal ini juga membuat Anda akan lebih mudah mengerti script pemrograman yang ditulis oleh orang lain tanpa perlu membaca komentar programnya.

Ditambah dengan paket-paket yang mempermudah Anda dalam melakukan analisis data seperti dplyr (untuk manipulasi data), ggplot (visualisasi), dan lainnya. Paket tersebut dibuat sedemikian rupa sehingga Anda dibantu untuk mengerti proses dan teknik, bukan syntax sehingga Anda akan lebih diarahkan pada pembelajaran konsep.

1.2.3 Fleksibilitas R

RStudio adalah salah satu IDE (Integrated Development Environment) pemrograman R yang sangat populer. Dengan RStudio, membuat dan mengembangkan aplikasi R menjadi lebih mudah dibandingkan menggunakan konsol original R.

RStudio sangat powerful dan membuat R menjadi sangat fleksibel. Anda bisa melakukan banyak hal melalui RStudio. Mulai dari membuat pemrograman script, grafik, laporan dalam bentuk pdf, ppt, excel, bahkan ebook, sampai dengan administrasi blog.

Ya, ebook ini dan situs pripramudya.com dibuat menggunakan RStudio.

Begitulah cerita tentang R. Jika Anda yakin untuk mulai belajar R, jangan ragu lagi untuk melakukan langkah kecil berikutnya.

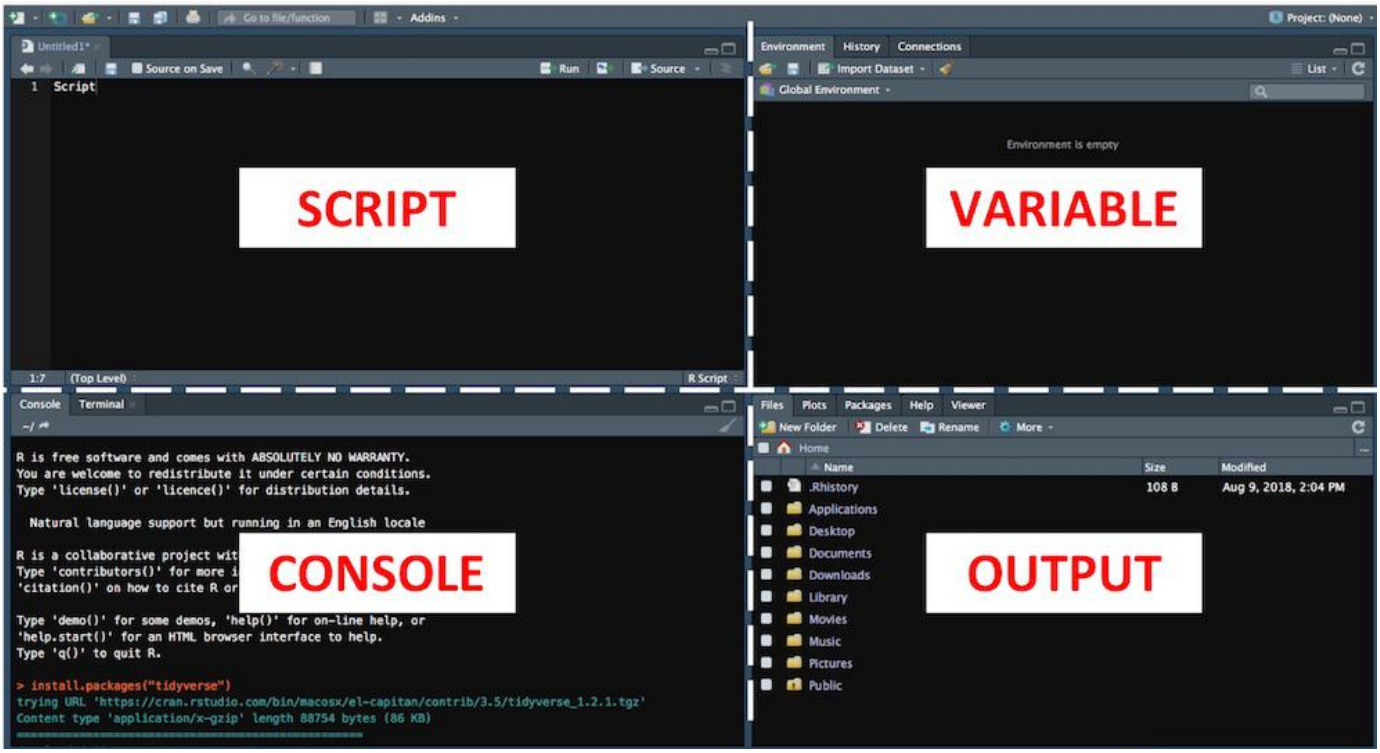
1.3 Instal R dan RStudio

Pilihlah paket instalasi sesuai sistem operasi yang Anda pakai. Untuk RStudio, pilih yang versi desktop.

- R dapat diunduh di sini.
- RStudio dapat diunduh di sini.

Buka software RStudio.

Layout RStudio terdiri dari bagian Script, Console, Variable, dan Output.



1.4 Instal Paket R

Paket adalah komponen penting yang memungkinkan pengguna untuk menggunakan program siap pakai tanpa harus membuatnya dari awal. Ada banyak ribuan paket di CRAN, tapi untuk saat ini, saya sarankan untuk instal cukup satu paket analisis data yang paling powerful, yakni **tidyverse**.

Anda dapat menginstalnya cukup dengan mengetik di CONSOLE menggunakan perintah:

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

Cukup instal paket tidyverse ini, Anda sudah bisa mulai belajar analisis data menggunakan R.

1.5 *Syntax* dan Operasi Dasar R

Di bagian ini, Anda akan diarahkan untuk mengenal *syntax* serta operasi dasar dari pemrograman R. Apa itu *syntax*?

Syntax adalah aturan menulis kata perintah di CONSOLE agar dapat dimengerti dengan benar oleh bahasa pemrograman R. Aturan *syntax* ini secara mutlak harus dipenuhi. Jika ada kesalahan penulisan *syntax*, maka akan ada umpan balik berupa pesan eror yang diterima oleh pengguna.

Jadi, bagian ini penting untuk Anda ketahui. Tutorial *syntax* dan operasi dasar ini harus Anda ketik di CONSOLE RStudio.

1.5.1 Kalkulator Interaktif

R bisa digunakan sebagai kalkulator. Ketik `2 + 5` di CONSOLE dan tekan Enter. Anda akan mendapat umpan balik berupa hasil perhitungan yang telah Anda input sebelumnya.

```
2 + 5
```

```
## [1] 7
```

Anda juga bisa menggunakan operator aritmatika lainnya seperti `*` untuk perkalian, `-` untuk pengurangan, `/` untuk bagi, ataupun `^` untuk kuadrat.

1.5.2 Variabel

Jika Anda ingin melakukan perhitungan kedua menggunakan hasil di atas, maka Anda dapat menyimpannya dalam bentuk variabel. Dengan variabel, Anda tidak perlu ketik ulang `2 + 5` setiap kali Anda memerlukannya.

Untuk membuat variabel baru, gunakan operator `panah ke kiri` , seperti ini: `<-` . Operator variabel ini merupakan gabungan karakter ‘kurang dari’ yang diikuti karakter ‘minus’.

Kenapa `panah ke kiri` ? Operator ini merupakan simbol yang Anda gunakan seolah Anda memasukkan nilai yang ada di sebelah kanan panah ke variabel yang ada di sebelah kiri panah.

Contohnya, Anda akan menyimpan hasil dari `2 + 5` ke variabel bernama `x` . Ketik `x <- 2 + 5` di CONSOLE dan tekan Enter.

```
x <- 2 + 5
```

Anda akan menyadari, R tidak langsung mengirim umpan balik hasil dari kalkulasi tersebut. Yang R lakukan hanya menyimpan hasil perhitungan tersebut ke variabel ‘x’ dan mengasumsikan Anda akan menggunakan varibel ini nantinya. Daftar nama variabel yang Anda buat akan tampil pada jendela VARIABLE di RStudio.

Jika Anda ingin menampilkan isi variabel ‘x’, Anda ketik `x` dan tekan Enter.

```
x
```

```
## [1] 7
```

Sekarang, simpan hasil dari `x * 3` pada variabel bernama ‘y’.

```
y <- x * 3
```

Tampilkan hasil dari ‘y’.

```
y
```

```
## [1] 21
```

1.5.3 Vektor

Kumpulan dari beberapa objek disebut vektor. Objek bisa berupa angka, karakter, string, dan lainnya. Vektor berupa angka merupakan bentuk simpel dari struktur data di R. Bahkan, 1 angka bisa dikatakan sebagai vektor dengan elemen sepanjang 1.

Cara membuat sebuah vektor adalah dengan menggunakan fungsi `c()` yang merupakan kepanjangan dari ‘concatenate’ atau ‘combine’.

Anda akan membuat sebuah vektor yang berisi nilai numerik 1.1, 9, dan 3.14 (pada penulisan angka, tanda titik di pengaturan standar RStudio merupakan tanda koma, begitu juga sebaliknya). Vektor ini akan Anda simpan di variabel ‘z’.

```
z <- c(1.1, 9, 3.14)
```

Lalu, Anda tampilkan variabel ‘z’.

```
z
```

```
## [1] 1.10 9.00 3.14
```

Perhatikan hasil outputnya. Tidak ada tanda koma yang memisahkan antar nilai pada vektor tersebut.

Anda juga bisa menggabungkan vektor untuk membuat vektor baru. Misalnya, vektor baru yang terdiri dari z, 777, dan z lagi. Tidak perlu menyimpan di variabel, sehingga Anda bisa melihat langsung hasilnya setelah menekan Enter.

```
c(z, 777, z)
```

```
## [1]   1.10   9.00   3.14 777.00   1.10   9.00   3.14
```

Numerik faktor bisa digunakan dengan operator aritmatika. Ketik: z * 2 + 100, dan lihat apa yang terjadi.

```
z * 2 + 100
```

```
## [1] 102.20 118.00 106.28
```

Pertama-tama, R mengalikan 2 setiap elemen vektor z, lalu menambahkannya dengan 100 sehingga hasilnya seperti di atas.

1.5.4 Urutan Angka

Cara termudah untuk membuat angka yang berurutan adalah dengan menggunakan operator `:` . Ketik `1:20` di CONSOLE dan tekan Enter untuk melihat apa yang akan terjadi.

```
1:20
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

Lalu bagaimana halnya dengan angka real? Misalnya, coba Anda ketik `pi:10` .

```
pi:10
```

```
## [1] 3.141593 4.141593 5.141593 6.141593 7.141593 8.141593 9.141593
```

Hasilnya, angka real dimulai dari pi (3.141593), angka naik (bertambah 1) sampai dengan 9. Batas atas 10 tidak tercapai karena angka selanjutnya (10.141593) lebih besar dari 10.

Selain `:`, Anda bisa menggunakan fungsi `seq()`. Kelebihan fungsi ini, Anda bisa memilih opsi penambahannya. Contoh, Anda ingin membuat urutan angka dari 1 sampai dengan 5 dengan penambahan 0.5.

```
seq(1, 5, by = 0.5)
```

```
## [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

1.5.5 Replikasi Angka

Fungsi untuk mereplika Angka adalah `rep()` yang merupakan kepanjangan dari ‘replicate’. Berikut contoh mereplikasi angka 0 sebanyak 30 kali.

```
rep(0, times = 30)
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

1.5.6 Ambil Elemen Vektor

Anda bisa mengambil elemen vektor dengan menggunakan operator `[]` dan operator urutan angka `:`, serta menggunakan fungsi `c()`.

Misalkan, Anda ambil contoh: variabel ‘a’ adalah vektor yang terdiri dari nilai 2, 4, 6, dan 8.

```
a <- c(2, 4, 6, 8)
```

```
# Ambil elemen ke-1 sampai dengan yang ke-3 dari variabel a
a[1:3]
```

```
## [1] 2 4 6
```

```
# Ambil elemen ke-2 dari variabel a
a[2]
```

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

```
# Ambil elemen ke-4 dan ke-1 dari variabel a
a[c(4, 1)]
```

```
## [1] 8 2
```

```
# Keluarkan elemen ke-3 dari variabel a
a[-3]
```

```
## [1] 2 4 8
```

```
# Keluarkan elemen ke-1 dan ke-3 dari variabel a
a[-c(1, 3)]
```

```
## [1] 4 8
```

1.5.7 *Syntax* / Fungsi Penting Lainnya

Syntax	Keterangan	Contoh Penggunaan
?	Akses ke dokumentasi dan penjelasan fungsi	?seq
getwd()	Menampilkan filepath direktori kerja	getwd()
ls()	Menampilkan daftar varibel	ls()
dir()	Menampilkan files dari sebuah direktori	dir()
rm(list=ls())	Menghapus seluruh variabel	rm(list=ls())
length()	Mendapatkan panjang elemen sebuah vektor atau variabel	length(x)

Tombol Keyboard

Beberapa tombol keyboard berfungsi untuk mempercepat Anda dalam menulis script di R:

- `tab` sangat sering digunakan untuk fitur *auto completion*. Contohnya, jika Anda memiliki varibel bernama ‘aristoteles’, Anda cukup ketik ‘a’ di CONSOLE, lalu tekan tombol `tab`, maka R akan langsung menyempurnakan penulisan variabel tersebut.
- `panah atas` dan `panah bawah` berfungsi untuk mengakses histori perintah yang sudah Anda input sebelumnya, sehingga Anda tidak perlu mengetik ulang.

Langkah Kilat 2 Kuasai Teknik Dasar Analisis Data

Analisis data merupakan subjek yang cukup luas. Anda tidak bisa belajar semuanya sekaligus. Waktu Anda terbatas dan sangat berharga. Anda perlu strategi khusus untuk menguasainya. Investasi terbaik Anda adalah waktu dan pikiran. Anda harus fokus belajar satu keterampilan yang memiliki keuntungan atau balik modal investasi yang tinggi.

Kabar baiknya, Anda sudah melakukan hal ini: Belajar R.

R merupakan tool yang memiliki keuntungan atau balik modal investasi paling tinggi.

Jadi, lanjutkanlah belajar R.

Di bagian ini, Anda akan diarahkan untuk mengetahui sebenarnya apa saja yang menjadi bagian dari teknik dasar analisis data.

Teknik dasar analisis data dibagi menjadi 3:

1. Visualisasi Data
2. Manipulasi Data
3. Interpretasi Data

Anda sebaiknya belajar teknik dasar analisis ini secara berurutan.

Pertama-tama pelajari visualisasi data.

Visualisasi data merupakan cara tercepat dan termudah untuk mendapatkan *insight* pada sebuah set data. Selain itu, keterampilan ini juga bisa dijadikan sebagai alat komunikasi ke siapa saja yang membutuhkan hasil analisis data.

2.1 Visualisasi Data

Untuk belajar visualisasi data, gunakan salah satu paket R, yakni `ggplot2` .

Huruf “gg” pada `ggplot2` merupakan kepanjangan dari *Grammar of Graphics*, sebuah buku yang ditulis oleh Leland Wilkinson, prinsip yang `ggplot2` ambil. *Grammar of Graphics* menjelaskan tentang metode membagi-bagi grafik ke berbagai elemen dan membangun

setiap elemen tersebut di beberapa lapisan untuk menampilkan representasi visual.

Singkatnya, grafik di petakan dari data ke atribut estetik, seperti warna, bentuk, atau ukuran juga ke objek geometrik, seperti titik, garis, ataupun batang. Selain itu, grafik juga dipetakan ke sistem koordinat.

Sebuah grafik ggplot terdiri dari sebuah sistem koordinat dan setidaknya sebuah objek geometrik.

Lalu, Kenapa `ggplot2` ?

`ggplot2` mengajarkan Anda bagaimana berpikir tentang visualisasi data.

Anda diajarkan tidak hanya secara *syntax* saja, tetapi juga berpikir dan memiliki konsep terkait proses memvisualisasi data.

3 Prinsip Dasar Visualisasi Menggunakan `ggplot2`

1. Pemetaan Data ke Atribut Estetik
2. Lapisan
3. Iterasi Pemetaan dan Lapisan

OK, sampai sini, mungkin Anda masih bingung, tidak apa. Ada baiknya konsep ini langsung dijelaskan melalui contoh saja.

2.1.1 Pemetaan Data ke Atribut Estetik

Pada contoh kali ini, Anda akan diarahkan untuk membuat grafik batang atau *barchart*. Grafik ini merupakan grafik yang paling sering dipakai dan paling mudah dibaca. Selain itu, akan lebih memudahkan Anda untuk memahami prinsip dan filosofi membuat grafik menggunakan `ggplot2` .

Sebenarnya ada banyak grafik jenis lainnya seperti histogram, grafik garis *(line chart)*, grafik sebar *(scatter plot)*, dan grafik lainnya. Pembahasan di bagian ini akan difokuskan pada grafik batang saja.

Cukup penjelasannya, coba langsung Anda inisiasi paket `ggplot2` menggunakan fungsi `library()` .

```
# Inisiasi paket ggplot
library(ggplot2)
```

Setiap Anda ingin menggunakan paket R yang cukup spesifik, Anda harus menginisasi paket tersebut di awal. Hal ini dilakukan agar Anda bisa mengakses seluruh fungsi yang ada di paket tersebut.

Anda akan menggunakan data *built in* dari `ggplot2` , yakni `mpg` . Untuk melihat data `mpg` , ketik `mpg` di CONSOLE dan tekan Enter.

```
mpg
```

```
## # A tibble: 234 x 11
##    manufacturer model displ  year   cyl trans drv     cty   hwy fl    cla…
##    <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <ch>
##  1 audi         a4      1.8  1999     4 auto… f        18    29 p     com…
##  2 audi         a4      1.8  1999     4 manu… f        21    29 p     com…
##  3 audi         a4      2    2008     4 manu… f        20    31 p     com…
##  4 audi         a4      2    2008     4 auto… f        21    30 p     com…
##  5 audi         a4      2.8  1999     6 auto… f        16    26 p     com…
##  6 audi         a4      2.8  1999     6 manu… f        18    26 p     com…
##  7 audi         a4      3.1  2008     6 auto… f        18    27 p     com…
##  8 audi         a4 q…   1.8  1999     4 manu… 4        18    26 p     com…
##  9 audi         a4 q…   1.8  1999     4 auto… 4        16    25 p     com…
## 10 audi         a4 q…   2    2008     4 manu… 4        20    28 p     com…
## # ... with 224 more rows
```

Seperti yang Anda lihat, `mpg` , yang merupakan kepanjangan dari *miles per gallon* adalah sebuah set data tentang efisiensi penggunaan bensin 38 mobil terpopuler di Amerika Serikat.

Anda bisa memahami lebih lanjut terkait detail set data ini melalui perintah `?mpg` .

Untuk melihat berapa jumlah baris dan kolomnya, Anda bisa menggunakan fungsi `dim` .

```
dim(mpg)
```

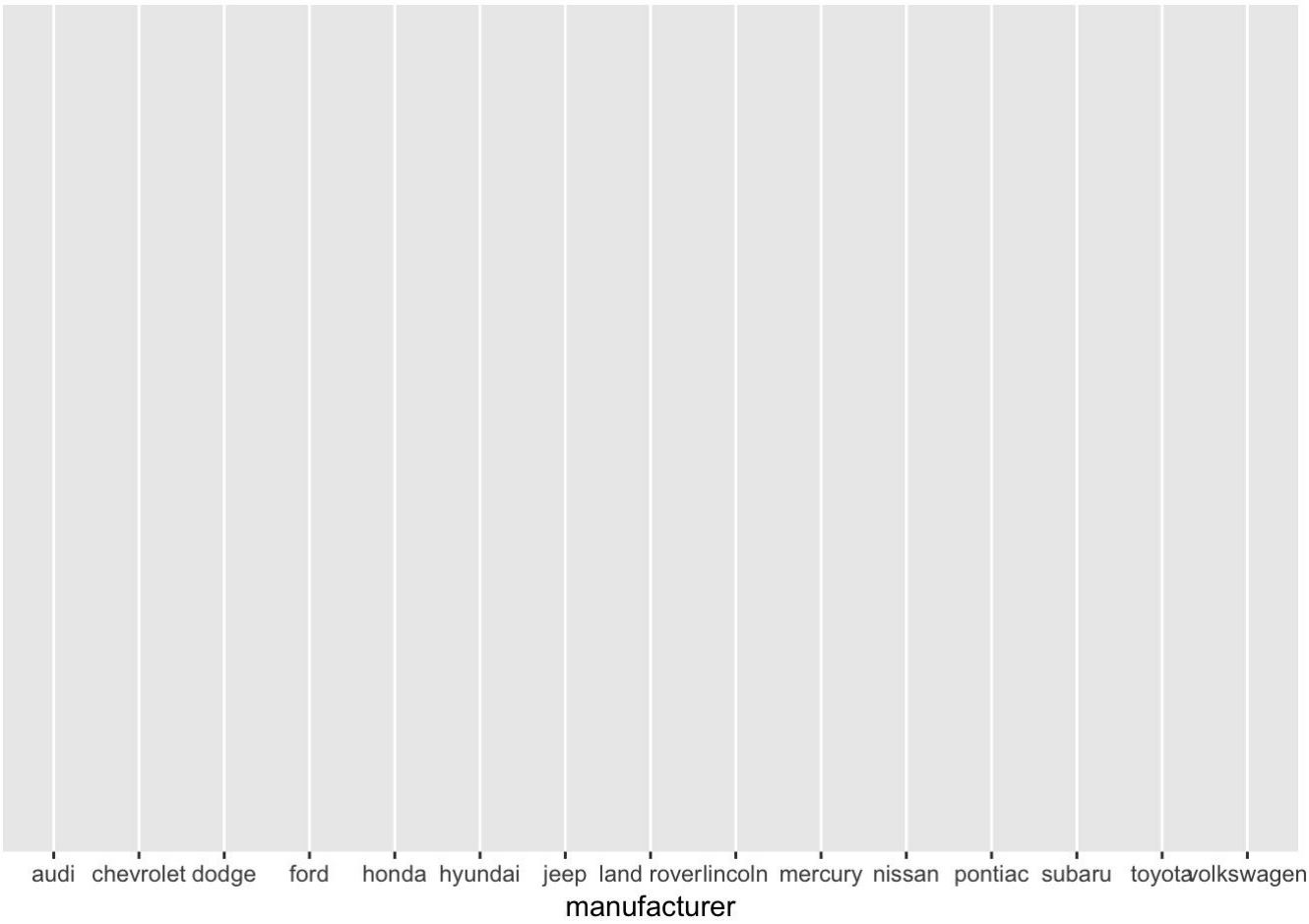
```
## [1] 234  11
```

Set data ini memiliki 234 baris dan 11 kolom atau 234 observasi dan 11 variabel. Jika Anda lihat data ini lebih teliti lagi, 234 juga merupakan jumlah total keseluruhan mobil yang menjadi objek observasi.

Sebelum melangkah lebih jauh ke teknis pembuatan grafik, Anda akan diarahkan untuk mengetahui **perbandingan jumlah mobil antar 'manufacturer'** sebagai tujuan dari proses ini.

OK, untuk mulai memetakan variabel ke grafik, gunakan fungsi `ggplot()` .

```
# Pemetaan variabel `manufacturer` ke sistem koordinat x
ggplot(data = mpg, aes(x = manufacturer))
```



Fungsi `aes()` di atas adalah memetakan atribut variabel 'manufacturer' ke sistem koordinat x. Anda bisa lihat, di garis horizontal x sudah ada nama-nama pembuat mobil yang merupakan hasil pemetaan atribut estetik.

Sampai sini Anda pasti berpikir,

"Kenapa kok jelek banget grafiknya?"

"Mana grafik batangnya?"

"Tulisannya tumpang tindih, nggak beraturan."

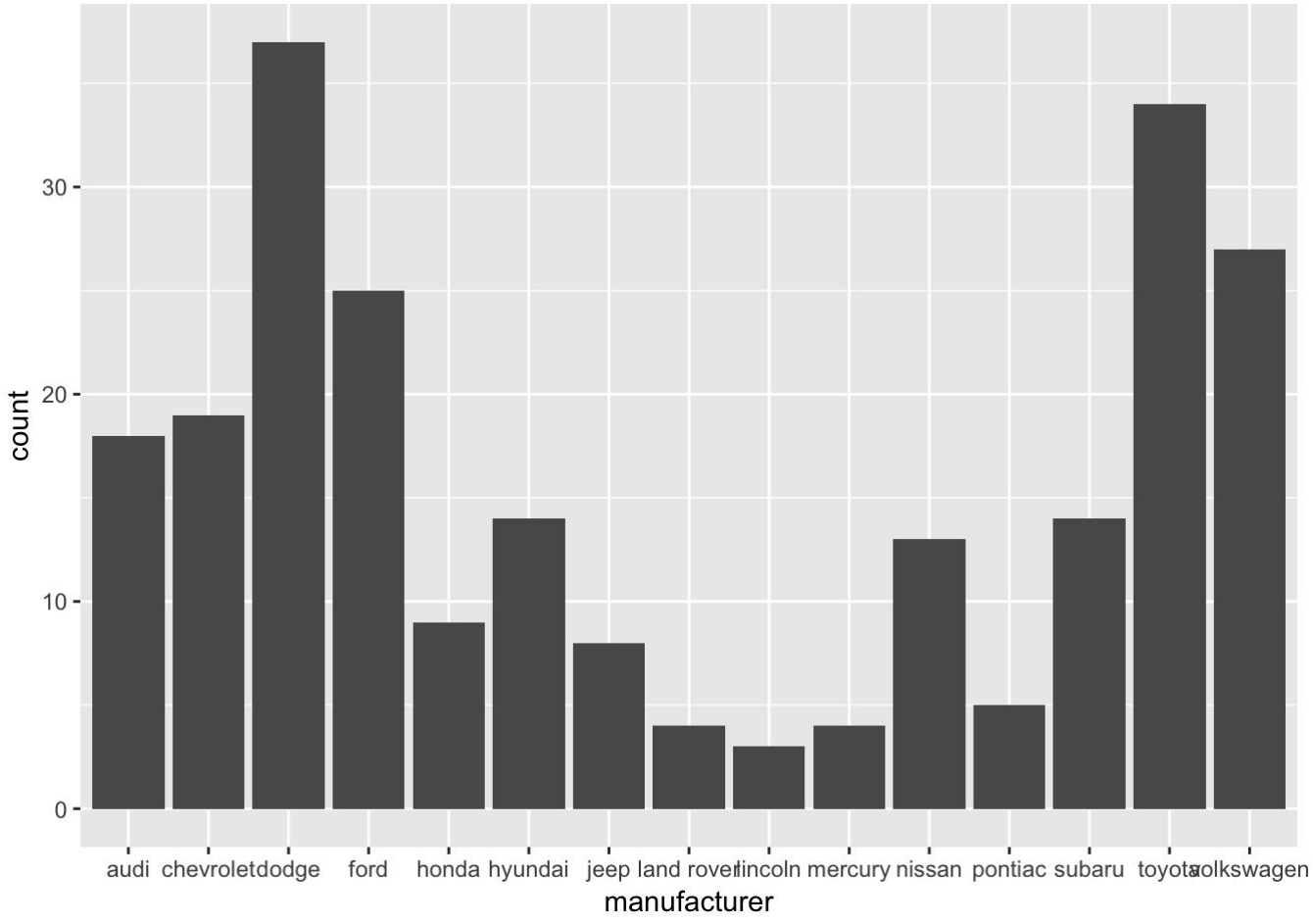
Tenang, grafik tersebut masih tahap lapisan pertama dan iterasi pertama juga prosesnya masih langkah demi langkah sehingga Anda tidak bingung dan lebih mudah mengerti konsep membuat grafik `ggplot2` .

2.1.2 Lapisan

Lapisan atau *layer* merupakan prinsip dasar visualisasi kedua menggunakan `ggplot2` . Filosofinya adalah Anda membuat grafik yang terdiri dari beberapa lapisan sehingga hasilnya merepresentasikan visualisasi yang Anda inginkan. Lapisan ini bisa berupa sistem koordinat, grafik batang, judul grafik, legenda, ataupun elemen lainnya.

Langkah selanjutnya setelah pemetaan variabel, Anda akan menambahkan elemen grafik batang pada lapisan berikutnya dengan menggunakan operator `+` dan fungsi `geom_bar()` .

```
# Penambahan elemen grafik batang pada lapisan kedua
ggplot(data = mpg, aes(x = manufacturer)) +
  geom_bar()
```



Nah, sudah mulai terlihat grafiknya.

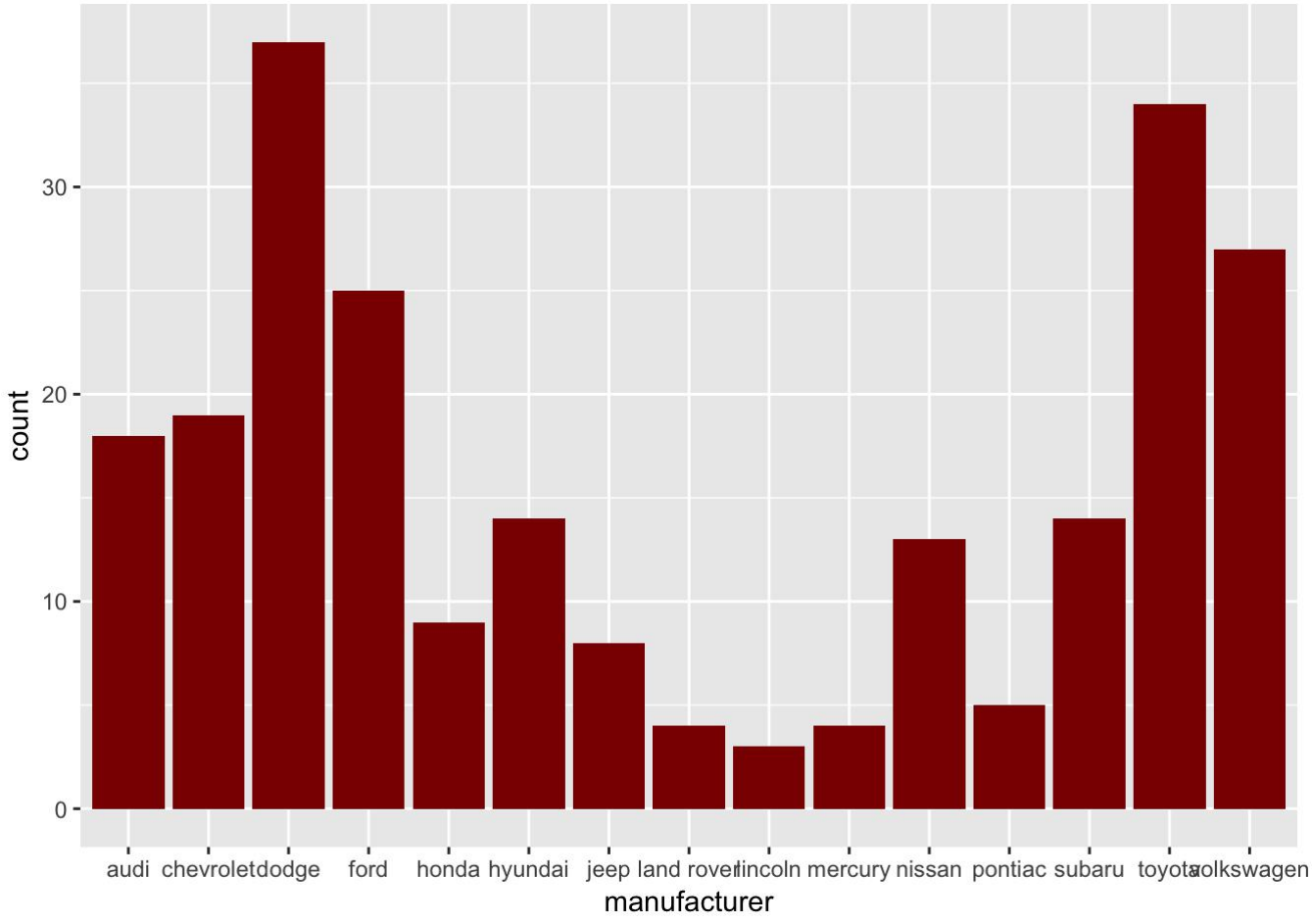
Sampai sini, sudah ada 2 lapisan grafik. Lapisan pertama diisi oleh elemen sistem koordinat, lapisan kedua diisi oleh grafik batang.

2.1.3 Iterasi Pemetaan dan Lapisan

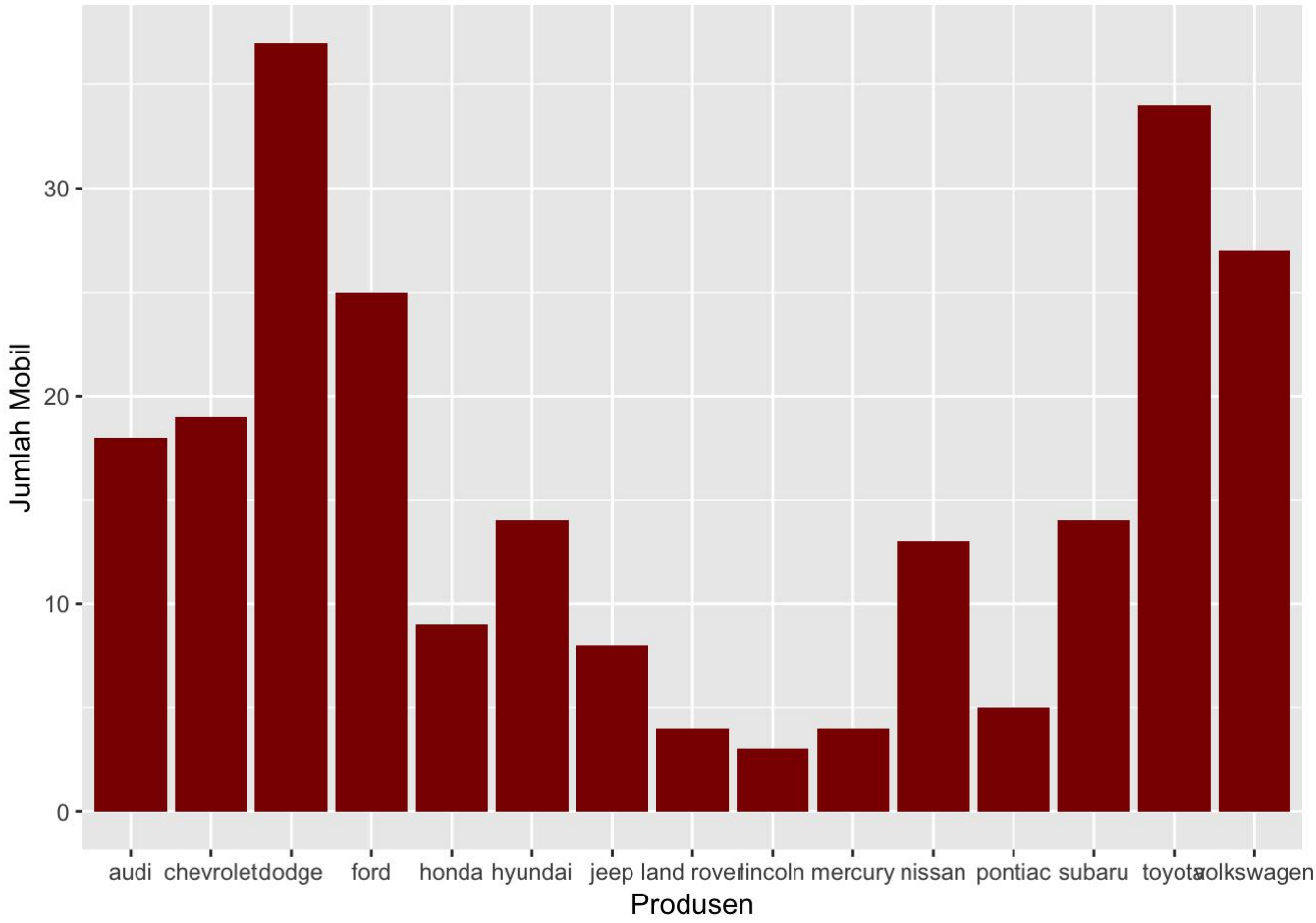
Prinsip dasar ketiga ini merupakan perulangan dari prinsip satu dan prinsip kedua. Anda terus melakukan iterasi dan modifikasi atribut estetik serta penambahan lapisan atau elemen sehingga sesuai dengan representasi visual yang ingin Anda capai.

Untuk mempermudah pemahaman, Anda akan diarahkan langsung melalui contoh penulisan kode yang disertai dengan penjelasan dari baris komentar (baris komentar diawali dengan tanda `#`).

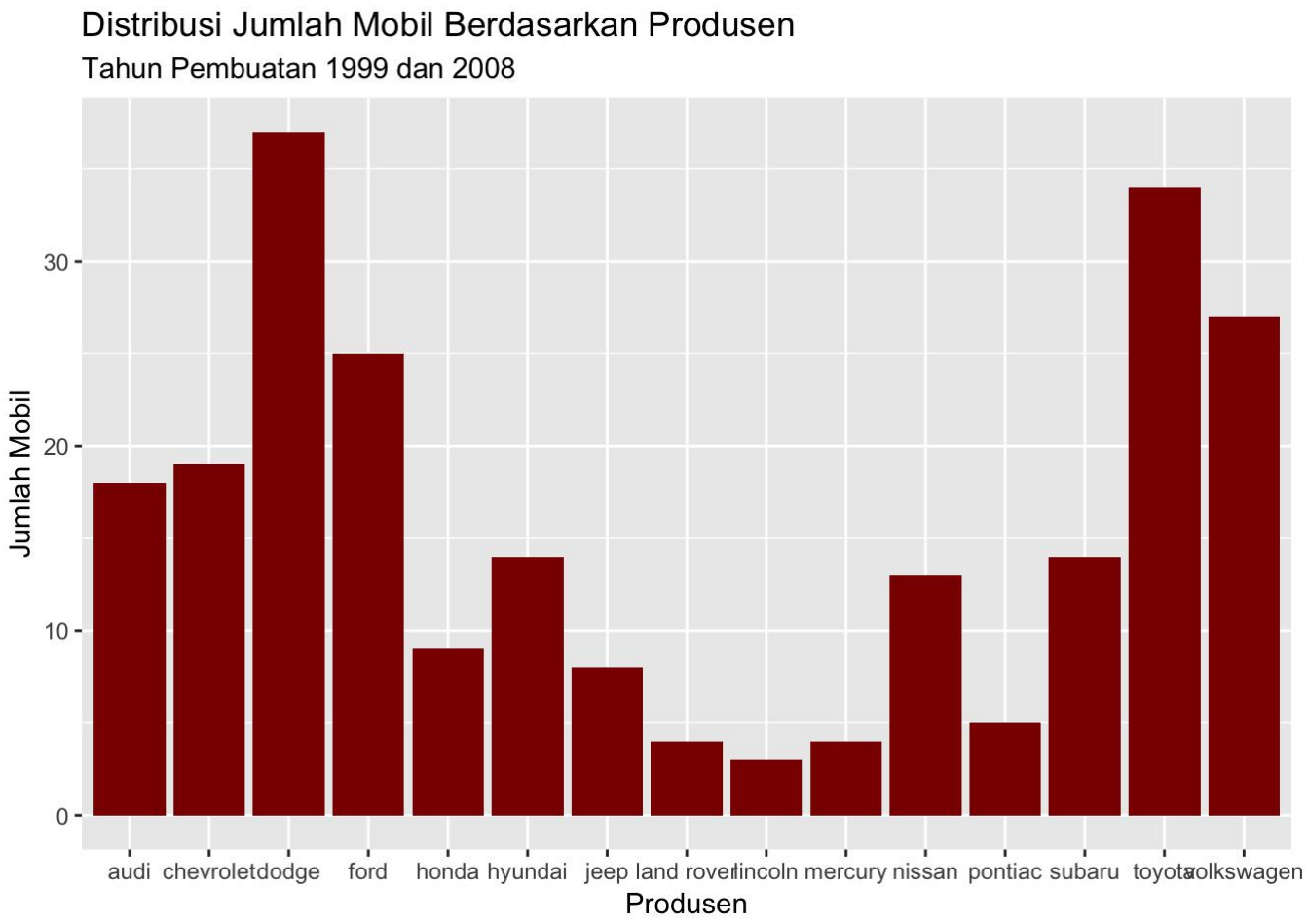
```
# Merubah warna grafik batang menjadi merah gelap
# dengan menggunakan argumen "fill" pada fungsi geom_bar()
ggplot(data = mpg, aes(x = manufacturer)) +
  geom_bar(fill = "#8B0000")
```



```
# Merubah label keterangan sistem koordinat x dan y
# dengan menggunakan fungsi labs()
ggplot(data = mpg, aes(x = manufacturer)) +
  geom_bar(fill = "#8B0000") +
  labs(x = "Produsen", y = "Jumlah Mobil")
```



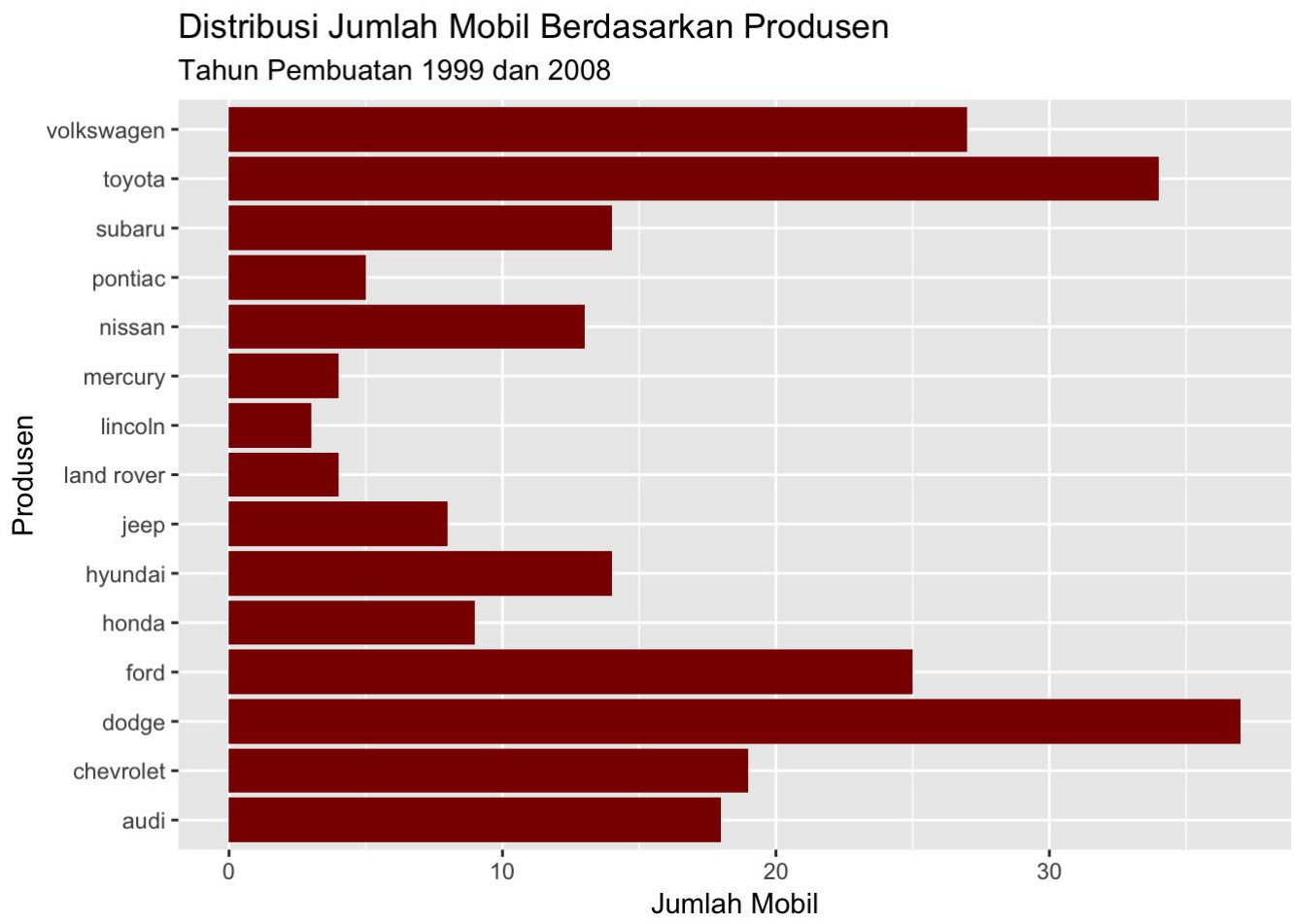

```
# Menambah judul dan subjudul
# dengan menggunakan fungsi ggtitle()
ggplot(data = mpg, aes(x = manufacturer)) +
  geom_bar(fill = "#8B0000") +
  labs(x = "Produsen", y = "Jumlah Mobil") +
  ggtitle("Distribusi Jumlah Mobil Berdasarkan Produsen",
          subtitle = "Tahun Pembuatan 1999 dan 2008")
```



Begitulah bagaimana iterasi dilakukan terus menerus sehingga representasi grafik sesuai dengan yang Anda inginkan.

Dan pada akhirnya, iterasi final, menukar sistem koordinat sehingga tulisan terlihat lebih rapih dan mudah dibaca.

```
# Menukar sistem koordinat sehingga lebih mudah dibaca
# menggunakan fungsi coord_flip()
ggplot(data = mpg, aes(x = manufacturer)) +
  geom_bar(fill = "#8B0000") +
  labs(x = "Produsen", y = "Jumlah Mobil") +
  ggtitle("Distribusi Jumlah Mobil Berdasarkan Produsen",
          subtitle = "Tahun Pembuatan 1999 dan 2008") +
  coord_flip()
```



Sekali Anda mengerti tentang prinsip dasar ini, pemahaman Anda terhadap data visualisasi akan berubah. Mungkin bagi Anda yang dulu terbiasa menggunakan grafik dari Excel akan memerlukan waktu lebih untuk menangkap ini semua. Bagaimana tidak, Excel memberikan kemudahan yang membuat Anda tidak perlu memikirkan proses bagaimana grafik dibuat. Anda hanya cukup *point, drag, and click*, sedikit konfigurasi, Excel akan membereskan semuanya.

Pelajari `ggplot2`, Anda akan paham bagaimana proses grafik dibuat tahap demi tahap. Memang membutuhkan *effort* lebih, tetapi *worth it*. Seiring Anda mengerti cara menggunakan `ggplot2`, pemahaman prinsip visualisasi data Anda akan meningkat.

2.2 Manipulasi Data

Manipulasi data berperan ketika representasi visualisasi yang Anda buat tidak cukup untuk memunculkan *insight*. Faktanya, data yang Anda peroleh hampir pasti perlu diolah terlebih dahulu. Menurut penelitian, seorang analis data kebanyakan mengerahkan usahanya berkutat di fase manipulasi atau mengolah data ini sekitar 60% - 70% dari keseluruhan proses analisis. Jadi bisa dikatakan, keterampilan manipulasi data cukup penting dan pastinya juga perlu Anda kuasai.

Manipulasi data adalah segala proses yang dilakukan pada sebuah set data dengan tujuan mempermudah melakukan analisis data dan mendapatkan *insight*. Proses tersebut diantaranya mengubah nama kolom atau variabel, mengurutkan, menyaring, sampai dengan meringkas data.

Konsep atau teknik manipulasi data yang akan dikupas diantaranya sebagai berikut:

1. Melihat dan Mengetahui Set Data
2. Mengubah Variabel, Kolom, Isi Data
3. Operator Pipa (`%>%`)
4. Memilih Variabel Data
5. Menyaring Baris Data
6. Mengurutkan Data
7. Mengelompokkan Data
8. Meringkas Data

Pada bahasan kali ini, Anda akan diarahkan untuk menggunakan kembali set data `mpg` seperti sebelumnya ditambah satu paket manipulasi data yang sangat *powerful*, `dplyr` .

Pertama-tama, inisiasi paket `dplyr` .

```
# Inisiasi paket dplyr
library(dplyr)
```

2.2.1 Melihat dan Mengetahui Set Data

Setiap Anda mendapatkan sebuah data set untuk diolah, hal yang pertama harus Anda lakukan adalah mengetahui isi dan tentang apa data tersebut. Apa format datanya? Berapa dimensinya? Apa saja variabelnya? Bagaimana tipe variabelnya? Apakah ada data yang hilang atau tidak terisi? Dan lain sebagainya.

Yang pertama adalah fungsi `class()` untuk menampilkan tipe data.

```
# Menampilkan tipe data
class(mpg)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

Apa itu “tbl_df” “tbl”?

“tbl_df” = *table data frame*

“tbl” = *table*

Intinya, “tbl_df” “tbl” “data.frame” merupakan tipe data dari `mpg` yang berupa tabel atau data frame. Tabel atau data frame adalah tipe data berbentuk “kotak” yang terdiri dari baris dan kolom.

Data yang mudah diolah atau dimanipulasi biasanya bertipe tabel atau data frame ini. Selain tabel, ada juga yang berbentuk vektor atau yang hanya terdiri dari satu baris saja atau satu kolom saja.

Lalu, bagaimana cara mengetahui jumlah baris dan kolom data `mpg` ? Ya, Anda benar, salah satunya menggunakan fungsi `dim()` .

Anda juga bisa menggunakan fungsi `nrow()` dan `ncol()` .

```
# Menampilkan jumlah baris
nrow(mpg)
```

```
## [1] 234
```

```
# Menampilkan jumlah kolom atau variabel
ncol(mpg)
```

```
## [1] 11
```

Dari fungsi di atas, Anda sudah mengetahui jumlah kolom atau variabel. Selanjutnya, Anda gali lebih jauh untuk mengetahui nama-nama kolom atau variabelnya. Fungsi `names()` bisa membantu Anda.

```
# Menampilkan seluruh variabel atau nama kolom
names(mpg)
```

```
##  [1] "manufacturer" "model"        "displ"        "year"
##  [5] "cyl"          "trans"        "drv"          "cty"
##  [9] "hwy"          "fl"           "class"
```

Anda sudah tahu variabel dari set data `mpg`, akan lebih lengkap lagi jika Anda tahu isi di dalamnya. Gunakan fungsi `head()` dan `tail()` untuk mengintip isi bagian awal dan akhir dari sebuah set data.

```
# Menampilkan 7 baris awal dari data
head(mpg, 7)
```

```
## # A tibble: 7 x 11
##   manufacturer model displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto… f        18    29 p     comp…
## 2 audi         a4      1.8  1999     4 manu… f        21    29 p     comp…
## 3 audi         a4      2    2008     4 manu… f        20    31 p     comp…
## 4 audi         a4      2    2008     4 auto… f        21    30 p     comp…
## 5 audi         a4      2.8  1999     6 auto… f        16    26 p     comp…
## 6 audi         a4      2.8  1999     6 manu… f        18    26 p     comp…
## 7 audi         a4      3.1  2008     6 auto… f        18    27 p     comp…
```

```
# Menampilkan 8 baris akhir dari data
tail(mpg, 8)
```

```
## # A tibble: 8 x 11
##   manufacturer model displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 volkswagen   new …   2.5  2008     5 auto… f        20    29 r     subc…
## 2 volkswagen   pass…   1.8  1999     4 manu… f        21    29 p     mids…
## 3 volkswagen   pass…   1.8  1999     4 auto… f        18    29 p     mids…
## 4 volkswagen   pass…   2    2008     4 auto… f        19    28 p     mids…
## 5 volkswagen   pass…   2    2008     4 manu… f        21    29 p     mids…
## 6 volkswagen   pass…   2.8  1999     6 auto… f        16    26 p     mids…
## 7 volkswagen   pass…   2.8  1999     6 manu… f        18    26 p     mids…
## 8 volkswagen   pass…   3.6  2008     6 auto… f        17    26 p     mids…
```

Selain `head()` dan `tail()` , untuk melihat isi data berformat *spreadsheet*, Anda bisa menggunakan fungsi `View()` .

```
# Menampilkan isi data berformat spreadsheet
View(mpg)
```

Hasilnya akan muncul di bagian jendela SCRIPT RStudio.

RINGKASAN DATA

Anda sudah mengetahui beberapa cara melihat dan menggali informasi dari sebuah set data. Ada satu cara untuk menampilkan ringkasan umum sehingga Anda bisa langsung melihat dan menangkap isi data secara *big picture*.

```
# Menampilkan ringkasan umum dari data
summary(mpg)
```

```
##  manufacturer          model              displ            year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl           trans               drv                 cty
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

Fungsi `summary()` memberikan Anda ringkasan umum dari sebuah set data. Jika variabelnya bertipe numerik, akan ditampilkan ringkasan seperti nilai minimal, maksimal, rata-rata, median, kuartil pertama, dan kuartil ketiga.

Selain fungsi `summary()` , ada fungsi lain yang lebih ringkas, yaitu `str()` dan `glimpse()` .

```
# Menampilkan struktur dan isi data
str(mpg)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```
# Menampilkan struktur dan isi data
dplyr::glimpse(mpg)
```

```
## Observations: 234
## Variables: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "...
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 qua...
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0,...
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1...
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 6...
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)...
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4",...
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 1...
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 2...
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p",...
## $ class        <chr> "compact", "compact", "compact", "compact", "comp...
```

Seperti yang Anda lihat, fungsi `glimpse()` sepertinya lebih rapih dalam tampilan.

Kedua fungsi ini menampilkan cukup informasi dalam sebuah tampilan yang padat dan jelas mulai dari berapa jumlah variabel, jumlah observasi, nama variabel, tipe variabel, sampai dengan isi dari sebagian data.

Ketika Anda mendapatkan sebuah set data untuk diolah, tentu fungsi ini tidak akan Anda lewatkan begitu saja.

Yang perlu Anda cermati, set data `mpg` ini tipe variabelnya masih kurang tepat. Misalnya, variabel `year` atau tahun pembuatan mobil. Di data hanya ada tahun 1999 dan 2008 saja. Seharusnya ini termasuk ke dalam tipe variabel faktor atau kategorikal, bukan numerik.

Di bagian berikutnya, akan dibahas bagaimana cara mengubah variabel.

2.2.2 Mengubah Variabel, Kolom, Isi Data

Sebelum melangkah lebih jauh, Anda harus mengetahui beberapa tipe variabel di pemrograman R.

TIPE VARIABEL

Yang sering digunakan ada 6 tipe:

- Karakter: Huruf, kata, atau kalimat seperti `b` , `apel` , `mantap jiwa` .
- Integer: Bilangan bulat, contoh `1` , `4` , `7` .
- Double: Bilangan tidak bulat atau real, contoh `2.3` , `3.5` .
- Logikal: Benar atau salah (`TRUE` / `FALSE`).
- Kompleks: `2+4i` , angka kompleks gabungan real dan imaginer.
- Faktor: Kategorikal, seperti `laki-laki` , `perempuan` (jenis kelamin).

Jika Anda lihat kembali set data `mpg` menggunakan `str()` atau `glimpse()` , maka tipe masing-masing variabelnya sebagai berikut:

- manufacturer: karakter
- model: karakter
- displ: double
- year: integer
- cyl: integer
- trans: karakter
- drv: karakter
- cty: integer
- hwy: integer
- fl: karakter
- class: karakter

Beberapa tipe variabel di atas ada yang tidak sesuai sehingga perlu diubah. Agar Anda lebih mengerti dan lebih jelas tentang variabel dan isinya serta tipe variabel yang seharusnya digunakan, maka coba Anda simak tabel berikut.

Variabel	Penjelasan	Tipe Variabel	Revisi Tipe Variabel
manufacturer	Produsen pembuat mobil	karakter	faktor
model	Nama model mobil	karakter	karakter
displ	*Engine displacement* atau kapasitas mesin dalam liter	double	double
year	Tahun pembuatan mobil	integer	faktor
cyl	Jumlah silinder mesin	integer	faktor
trans	Tipe transmisi, auto atau manual	karakter	faktor
drv	Roda penggerak, f = roda depan, r = roda belakang, 4 = 4 roda	karakter	faktor
cty	Konsumsi bensin dalam kota (mil per galon)	integer	integer
hwy	Konsumsi bensin di jalan tol (mil per galon)	integer	integer
fl	Jenis bensin, e = ethanol, d = diesel, r = reguler, p = premium, c = natural gas	karakter	faktor
class	Tipe atau kelas mobil	karakter	faktor

Beberapa variabel seperti manufacturer, year, cyl, trans, drv, fl, dan class merupakan tipe variabel berupa kategori sehingga perlu diubah menjadi tipe faktor. Di set data original, variabel-variabel tersebut dikenali sebagai tipe karakter atau integer.

MENGAKSES VARIABEL ATAU KOLOM DALAM SET DATA

Untuk mengubah variabel, Anda harus tahu cara mengakses salah satu variabel atau kolom beserta isi dari sebuah set data. Operator yang sering digunakan untuk mengakses variabel atau kolom adalah `$`. Ketik nama set data diikuti operator `$` lalu nama variabel atau kolom. Semua diketik tanpa spasi.

```
# Mengakses variabel atau kolom drv
mpg$drv
```

```
##   [1] "f" "f" "f" "f" "f" "f" "f" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4"
##  [18] "4" "r" "r" "r" "r" "r" "r" "r" "r" "r" "r" "r" "4" "4" "4" "4" "f" "f"
##  [35] "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "4" "4" "4"
##  [52] "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4"
##  [69] "4" "4" "4" "4" "4" "4" "r" "r" "r" "4" "4" "4" "4" "4" "4" "4" "4" "4"
##  [86] "4" "4" "4" "4" "4" "r" "r" "r" "r" "r" "r" "r" "r" "r" "r" "f" "f" "f"
## [103] "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
## [120] "f" "f" "f" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "r" "r"
## [137] "r" "4" "4" "4" "4" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "4" "4" "4"
## [154] "4" "f" "f" "f" "f" "f" "f" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4" "4"
## [171] "4" "4" "4" "4" "4" "4" "4" "4" "4" "f" "f" "f" "f" "f" "f" "f" "f" "f"
## [188] "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "4" "4" "4" "4" "4" "4"
## [205] "4" "4" "4" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
## [222] "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f" "f"
```

Seluruh 234 isi dari variabel drv ditampilkan. Anda juga bisa mencoba variabel yang lainnya dengan cara yang sama.

MENGUBAH TIPE VARIABEL

Sebelum melakukan manipulasi data, pertama-tama pastikan Anda menyalin set data asli ke set data yang baru.

```
# Menyalin set data mpg ke set data baru bernama df.mobil
# df merupakan singkatan dari data frame
df.mobil <- mpg
```

Mulai sekarang dan selanjutnya, segala manipulasi data dilakukan di set data `df.mobil` .

Untuk mengubah tipe variabel ke tipe faktor atau kategorikal digunakan fungsi `as.factor()` . Fungsi pengubah tipe variabel lainnya:

- `as.character()` mengubah ke tipe karakter
- `as.integer()` mengubah ke tipe integer
- `as.double()` mengubah ke tipe double

- `as.logical()` mengubah ke tipe logikal
- `as.complex()` mengubah ke tipe kompleks

```
# Mengubah tipe variabel manufacturer, year, cyl, trans, drv, fl, dan class
# ke tipe faktor
df.mobil$manufacturer <- as.factor(df.mobil$manufacturer)
df.mobil$year <- as.factor(df.mobil$year)
df.mobil$cyl <- as.factor(df.mobil$cyl)
df.mobil$trans <- as.factor(df.mobil$trans)
df.mobil$drv <- as.factor(df.mobil$drv)
df.mobil$fl <- as.factor(df.mobil$fl)
df.mobil$class <- as.factor(df.mobil$class)
```

Hasilnya bisa dicek melalui fungsi `str()` atau glimpse.

```
# Menampilkan struktur set data df.mobil
str(df.mobil)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: Factor w/ 15 levels "audi","chevrolet",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : Factor w/ 2 levels "1999","2008": 1 1 2 2 1 1 2 1 1 2 ...
##  $ cyl         : Factor w/ 4 levels "4","5","6","8": 1 1 1 1 3 3 3 1 1 1 ...
##  $ trans       : Factor w/ 10 levels "auto(av)","auto(l3)",..: 4 9 10 1 4 9 1 9 4 10
##  $ drv         : Factor w/ 3 levels "4","f","r": 2 2 2 2 2 2 2 1 1 1 ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : Factor w/ 5 levels "c","d","e","p",..: 4 4 4 4 4 4 4 4 4 4 ...
##  $ class       : Factor w/ 7 levels "2seater","compact",..: 2 2 2 2 2 2 2 2 2 2 ...
```

Variabel-variabel terkait yang sebelumnya bertipe integer atau karakter telah diubah menjadi faktor.

Perubahan lain yang bisa dilihat melalui fungsi `summary()` .

```
# Menampilkan ringkasan umum dari set data df.mobil
summary(df.mobil)
```

```
##      manufacturer    model               displ         year     cyl
##  dodge     :37    Length:234         Min.   :1.600   1999:117   4:81
##  toyota    :34    Class :character   1st Qu.:2.400   2008:117   5: 4
##  volkswagen:27    Mode  :character   Median :3.300              6:79
##  ford      :25                       Mean   :3.472              8:70
##  chevrolet :19                       3rd Qu.:4.600
##  audi      :18                       Max.   :7.000
##  (Other)   :74
##         trans    drv          cty              hwy        fl
##  auto(l4)  :83   4:103   Min.   : 9.00   Min.   :12.00   c:  1
##  manual(m5):58   f:106   1st Qu.:14.00   1st Qu.:18.00   d:  5
##  auto(l5)  :39   r: 25   Median :17.00   Median :24.00   e:  8
##  manual(m6):19           Mean   :16.86   Mean   :23.44   p: 52
##  auto(s6)  :16           3rd Qu.:19.00   3rd Qu.:27.00   r:168
##  auto(l6)  : 6           Max.   :35.00   Max.   :44.00
##  (Other)   :13
##         class
##  2seater   : 5
##  compact   :47
##  midsize   :41
##  minivan   :11
##  pickup    :33
##  subcompact:35
##  suv       :62
```

Dengan mengubah tipe variabel menjadi faktor, kategori pada masing-masing variabel akan terlihat dan dihitung frekuensinya. Tampilan ini memudahkan Anda untuk mengetahui bahwa produsen dodge dan toyota memiliki oberservasi terbanyak dibandingkan produsen mobil lainnya.

Bandingkan dengan hasil ringkasan set data aslinya yang kurang informatif:

```
# Menampilkan ringkasan umum dari set data df.mobil
summary(mpg)
```

```
##  manufacturer          model               displ            year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl            trans               drv                 cty
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

MENGUBAH NAMA VARIABEL

Nama variabel set data `mpg` menggunakan singkatan yang pendek dan berbahasa Inggris. Untuk memudahkan pemahaman, nama variabel tersebut akan diganti ke bahasa Indonesia serta tanpa singkatan.

```
# Menampilkan nama variabel set data df.mobil
names(df.mobil)
```

```
##  [1] "manufacturer" "model"        "displ"        "year"
##  [5] "cyl"          "trans"        "drv"          "cty"
##  [9] "hwy"          "fl"           "class"
```

Perhatikan urutan dari output di atas. Urutan nama variabel dimulai dari baris paling atas kiri ke kanan, lalu baris kedua mulai kiri ke kanan dan seterusnya. Angka di dalam kurung kotak `[]` adalah alat bantu untuk mempermudah identifikasi nomor urut variabel. Contohnya `[7]` adalah nomor urut variabel ‘drv’, sebelah kanannya, ‘cty’ sudah tentu nomor delapan dan seterusnya.

Mengubah nama variabel bisa dilakukan satu per satu atau sekaligus.

Berikut cara untuk mengubah salah satu nama variabel pada sebuah set data.

```
# Mengubah nama variabel displ menjadi kapasitas
names(df.mobil)[3] <- "kapasitas"
```

Penjelasan dari kode di atas adalah mengubah nama variabel yang merupakan elemen vektor ketiga set data df.mobil menjadi “kapasitas”.

Yang perlu Anda identifikasi adalah nomor urut ke berapa variabel yang akan diubah. Variabel ‘displ’ urutan ketiga. Dengan menggunakan prinsip elemen vektor, maka digunakan `[3]` sebagai akses elemen dengan urutan yang ketiga.

Pastikan bahwa nama variabel ‘displ’ sudah diubah.

```
# Menampilkan nama variabel set data df.mobil
names(df.mobil)
```

```
##  [1] "manufacturer" "model"        "kapasitas"    "year"
##  [5] "cyl"          "trans"        "drv"          "cty"
##  [9] "hwy"          "fl"           "class"
```

Variabel kapasitas sudah ada di set data df.mobil.

Lalu, bagaimana cara mengubah semua nama variabel sekaligus?

```
# Mengubah seluruh nama variabel
names(df.mobil) <- c("produsen", "model", "kapasitas", "tahun", "silinder", "transmisi",
```

Dengan menggunakan fungsi `names()` dan `c()`, kesebelas nama varibel baru disusun secara berurutan.

```
# Menampilkan nama variabel set data df.mobil
names(df.mobil)
```

```
##  [1] "produsen"     "model"        "kapasitas"    "tahun"
##  [5] "silinder"     "transmisi"    "roda.gerak"   "kota"
##  [9] "jalan.tol"    "jenis.bensin" "kelas"
```

Seluruh nama variabel telah diubah.

MENGUBAH ISI DATA

Konsep mengubah isi data sebenarnya mirip seperti halnya konsep *"find and replace"* pada aplikasi *spreadsheet*. Untuk manipulasi ini digunakan fungsi `gsub()`.

Argumen pertama huruf/kata yang akan diganti, argumen kedua huruf/kata pengganti, dan argumen terakhir adalah target variabelnya.

```
# Mengubah isi variabel drv menjadi yang sebelumnya
# satu huruf menjadi satu kata berbahasa Indonesia
df.mobil$roda.gerak <- gsub("f", "depan", df.mobil$roda.gerak)
df.mobil$roda.gerak <- gsub("r", "belakang", df.mobil$roda.gerak)
df.mobil$roda.gerak <- gsub("4", "4roda", df.mobil$roda.gerak)
```

Hasil pengubahan isi data ini bisa dilihat menggunakan fungsi `table()`.

```
# Menampilkan ringkasan variabel roda.gerak pada set data df.mobil
table(df.mobil$roda.gerak)
```

```
##
##    4roda belakang    depan
##      103       25      106
```

Fungsi `table()` menampilkan ringkasan isi berupa banyaknya frekuensi faktor yang ada di salah satu variabel.

Bandingkan dengan data aslinya.

```
# Menampilkan ringkasan variabel drv pada set data mpg
table(mpg$drv)
```

```
##
##   4   f   r
## 103 106  25
```

Yang baru saja dilakukan adalah cara mengubah isi data bertipe karakter. Bagaimana cara mengubah data numerik?

Misalnya, Anda ingin mengubah isi data variabel ‘kota’ dan ‘jalan.tol’ pada set data df.mobil yang sebelumnya menggunakan satuan miles per gallon (mpg) menjadi kilometer per liter (untuk seterusnya disingkat kpl). Satuan kpl dalam perhitungan konsumsi BBM lebih familiar di Indonesia.

1 miles per gallon (mpg standar Amerika) setara dengan 0.425144 kilometer per liter (kpl). Agar memudahkan perhitungan, dibulatkan menjadi 0.43 kpl.

Di bab sebelumnya, Anda sudah belajar bagaimana R berfungsi sebagai Kalkulator Interaktif. Ya, Anda cukup menggunakan operator `*` untuk mengalikan 0.43 variabel ‘kota’ dan ‘jalan.tol’ sehingga isi data berubah menjadi satuan kpl.

```
# Mengubah isi dari variabel kota dan jalan tol menjadi satuan kpl
# dengan mengalikan 0.43
df.mobil$kota <- df.mobil$kota * 0.43
df.mobil$jalan.tol <- df.mobil$jalan.tol * 0.43
```

Melihat hasilnya menggunakan fungsi `head()` .

```
# Melihat 5 data pertama dari variabel kota dan jalan.tol pada set data df.mobil
head(df.mobil$kota, 5)
```

```
## [1] 7.74 9.03 8.60 9.03 6.88
```

```
head(df.mobil$jalan.tol, 5)
```

```
## [1] 12.47 12.47 13.33 12.90 11.18
```

2.2.3 Operator Pipa (`%>%`)

R adalah bahasa fungsional. Artinya, *syntax* atau fungsi banyak sekali menggunakan tanda kurung `(` dan `)`. Ketika Anda melakukan manipulasi data yang kompleks, maka konsekuensinya kode juga banyak mengandung tanda kurung. Fungsi dalam fungsi yang bersarang. Ini membuat kode Anda sulit dibaca dan dimengerti. Operator pipa `%>%` bisa menyederhanakannya.

Sebagai contoh, Anda melakukan perhitungan berikut.

```
sin(exp(log(sqrt(9))))
```

```
## [1] 0.14112
```

Ada 4 fungsi yang terlibat dan banyak tanda kurung sehingga untuk mengetiknya pun Anda cukup rumit. Struktur urutan pengetikkan pun dimulai dari dalam ke luar atau dari kiri ke kanan. Mulai dari `sqrt(9)` lalu dikurung oleh `log()`, dikurung, dikurung sampai terakhir oleh `sin()`.

Sekarang bandingkan dengan menggunakan `%>%`. Hasil perhitungannya sama.

```
# Inisiasi paket magrittr agar mengenali operator pipa
library(magrittr)

# Contoh penggunaan operator pipa
9 %>% sqrt() %>% log() %>% exp() %>% sin()
```

```
## [1] 0.14112
```

Operator pipa memberikan kemudahan memahami kode pemrograman dalam hal:

- Struktur urutan fungsi dari kiri ke kanan (yang biasanya dari kanan ke kiri)
- Menghindari banyaknya tanda kurung yang bersarang dari fungsi
- Menambah fungsi sesuai urutan jika diperlukan

Contoh lainnya:

- Kode umum yang biasa digunakan untuk melihat 7 baris pertama pada set data df.mobil:
  `head(df.mobil, 7)`
- Kode menggunakan operator pipa: `df.mobil %>% head(7)`

```
# Contoh lain penggunaan operator pipa
# melihat 7 baris pertama pada set data df.mobil
df.mobil %>% head(7)
```

```
## # A tibble: 7 x 11
##   produsen model kapasitas tahun silinder transmisi roda.gerak  kota
##   <fct>    <chr>     <dbl> <fct> <fct>    <fct>     <chr>      <dbl>
## 1 audi     a4          1.8 1999  4        auto(l5)  depan       7.74
## 2 audi     a4          1.8 1999  4        manual(m… depan       9.03
## 3 audi     a4          2   2008  4        manual(m… depan       8.6
## 4 audi     a4          2   2008  4        auto(av)  depan       9.03
## 5 audi     a4          2.8 1999  6        auto(l5)  depan       6.88
## 6 audi     a4          2.8 1999  6        manual(m… depan       7.74
## 7 audi     a4          3.1 2008  6        auto(av)  depan       7.74
## # ... with 3 more variables: jalan.tol <dbl>, jenis.bensin <fct>,
## #   kelas <fct>
```

Anda harus mengenal dan memahami operator pipa ini terlebih dulu karena pembahasan ke depan akan sering digunakan.

2.2.4 Memilih Variabel Data

Untuk memilih beberapa variabel atau kolom pada sebuah set data, gunakan fungsi `select()` .

```
# Memilih variabel produsen, tahun, dan roda.gerak dari 11 variabel yang ada
# pada set data df.mobil
contoh.pilih <- df.mobil %>%
  select(produsen, tahun, roda.gerak)


# Melihat 8 baris pertama pada set data contoh.pilih
contoh.pilih %>%  head(8)
```

```
## # A tibble: 8 x 3
##   produsen tahun roda.gerak
##   <fct>    <fct> <chr>
## 1 audi     1999  depan
## 2 audi     1999  depan
## 3 audi     2008  depan
## 4 audi     2008  depan
## 5 audi     1999  depan
## 6 audi     1999  depan
## 7 audi     2008  depan
## 8 audi     1999  4roda
```

Kode pemrograman di atas dapat digabung tanpa perlu menyimpan hasilnya di sebuah variabel baru.

```
# Memilih variabel produsen, tahun, dan roda.gerak
# lalu menampilkan 8 baris pertama dari set data df.mobil
df.mobil %>%
  select(produsen, tahun, roda.gerak) %>%
  head(8)
```

```
## # A tibble: 8 x 3
##   produsen tahun roda.gerak
##   <fct>    <fct> <chr>
## 1 audi     1999  depan
## 2 audi     1999  depan
## 3 audi     2008  depan
## 4 audi     2008  depan
## 5 audi     1999  depan
## 6 audi     1999  depan
## 7 audi     2008  depan
## 8 audi     1999  4roda
```

2.2.5 Menyaring Baris Data

Fungsi `filter()` digunakan untuk menyaring baris pada sebuah set data dengan kondisi tertentu.

Misalnya, Anda ingin mengetahui model mobil yang merupakan kelas dua tempat duduk atau 2seater dari set data `df.mobil` .

```
# Memilih variabel model dan kelas
# lalu menyaring baris data dengan kondisi kelas 2seater
df.mobil %>%
  select(model, kelas) %>%
  filter(kelas == "2seater")
```

```
## # A tibble: 5 x 2
##   model    kelas
##   <chr>    <fct>
## 1 corvette 2seater
## 2 corvette 2seater
## 3 corvette 2seater
## 4 corvette 2seater
## 5 corvette 2seater
```

2.2.6 Mengurutkan Data

Anda bisa mengurutkan data dengan menggunakan fungsi `arrange()` .

Anda ingin mengetahui 10 peringkat teratas model mobil yang memiliki efisiensi konsumsi bensin (kpl) tertinggi dalam kota dari set data `df.mobil` .

```
# Memilih variabel model, transmisi, dan kota
# lalu mengurutkan kpl dari yang tertinggi ke terendah
# menggunakan fungsi arrange dan desc
# serta menampilkan hasilnya untuk 10 baris teratas
df.mobil %>%
  select(model, transmisi, kota) %>%
  arrange(desc(kota)) %>%
  head(10)
```

```
## # A tibble: 10 x 3
##    model      transmisi   kota
##    <chr>      <fct>      <dbl>
##  1 new beetle manual(m5)  15.0
##  2 jetta      manual(m5)  14.2
##  3 new beetle auto(l4)    12.5
##  4 civic      manual(m5)  12.0
##  5 corolla    manual(m5)  12.0
##  6 civic      manual(m5)  11.2

##  7 corolla    manual(m5)  11.2
##  8 corolla    auto(l4)    11.2
##  9 civic      manual(m5)  10.8
## 10 civic      auto(l5)    10.8
```

2.2.7 Menambah Variabel Data

Pada contoh kali ini, akan ditambahkan variabel atau kolom efisiensi bensin untuk penggunaan dalam kota dengan satuan kilometer per liter dari set data `mpg` . Untuk manipulasi ini, fungsi `mutate()` akan digunakan.

```
# Memilih variabel model, cty, dan hwy
# lalu menambahkan variabel baru: kota.kpl dan tol.kpl
# yang merupakan konversi satuan mpg ke kpl
# serta menampilkan hasilnya untuk 10 baris teratas
mpg %>%
  select(model, cty, hwy) %>%
  mutate(kota.kpl = cty * 0.43, tol.kpl = hwy * 0.43) %>%
  head(10)
```

```
## # A tibble: 10 x 5
##    model        cty   hwy kota.kpl tol.kpl
##    <chr>      <int> <int>    <dbl>   <dbl>
##  1 a4            18    29     7.74    12.5
##  2 a4            21    29     9.03    12.5
##  3 a4            20    31     8.6     13.3
##  4 a4            21    30     9.03    12.9
##  5 a4            16    26     6.88    11.2
##  6 a4            18    26     7.74    11.2
##  7 a4            18    27     7.74    11.6
##  8 a4 quattro    18    26     7.74    11.2
##  9 a4 quattro    16    25     6.88    10.8
## 10 a4 quattro    20    28     8.6     12.0
```

2.2.8 Mengelompokkan dan Meringkas Data

Fungsi manipulasi terakhir yang dibahas adalah fungsi `group_by()` dan `summarise()` untuk mengelompokkan dan meringkas data.

Kembali ke set data `df.mobil` . Anda ingin mengetahui berapa rata-rata efisiensi konsumsi bensin penggunaan dalam kota dan jalan tol untuk masing-masing kategori roda gerak.

```
# Mengelompokkan berdasarkan kategori roda.gerak
# lalu menampilkan rata-rata efisiensi menggunakan fungsi mean
# untuk masing-masing konsumsi kpl kota dan jalan tol
df.mobil %>%
  group_by(roda.gerak) %>%
  summarise(kota.kpl = mean(kota), tol.kpl = mean(jalan.tol))
```

```
## # A tibble: 3 x 3
##   roda.gerak kota.kpl tol.kpl
##   <chr>         <dbl>   <dbl>
## 1 4roda          6.16    8.25
## 2 belakang       6.05    9.03
## 3 depan          8.59   12.1
```

2.3 Interpretasi Data

Interpretasi data berarti menyimpulkan hasil analisis data yang sudah Anda lakukan. Anda sudah bersusah payah membuat visualisasi serta manipulasi data, namun tidak diinterpretasikan, maka hasilnya nol besar. Anda harus menarik kesimpulan. Anda harus mendapatkan cerita dari data. Inti dari analisis data yang Anda sudah lakukan adalah untuk mendapatkan *insight*.

Memang, kemampuan interpretasi setiap orang berbeda dan subjektif dalam memandang sebuah permasalahan atau dalam menarik informasi dari data. Untuk mengasah kemampuan interpretasi ini, Anda setidaknya harus memiliki 3 sikap positif: tekad/semangat *(passion)*, rasa penasaran *(curiosity)*, dan antusias.

Ketiga sikap positif ini akan sangat membantu Anda, disamping keahlian teknis dalam analisis data tentunya. Mendapatkan *insight* dari kumpulan set data merupakan hal yang cukup menantang. Jika keahlian dan keterampilan teknis sudah semaksimal mungkin dikeluarkan, namun masih belum juga optimal dalam menggali kesimpulan dari data, sikap semangat pantang menyerah, rasa penasaran tinggi, dan antusias yang tidak pernah habis akan menjadi tambahan energi luar biasa bagi Anda.

Selain faktor *soft skill* yang baru saja dibahas, berikut panduan pertanyaan-pertanyaan teknis sebagai pemicu untuk membantu Anda dalam proses interpretasi dan menggali informasi serta *insight* dalam analisis data.

- **Sebenarnya apa tujuan utama atau motivasi Anda dalam menganalisis set data tertentu? Apa inti permasalahan yang Anda atau organisasi/perusahaan hadapi?** Ambil contoh set data `mpg` yang merupakan data efisiensi penggunaan bensin pada mobil. Anggaplah dalam beberapa bulan ke depan, Anda berencana membeli mobil pertama yang merupakan hasil jerih payah kerja Anda selama ini. Anda tidak ingin salah dalam membeli mobil pertama impian Anda. **Kriteria pilihan Anda adalah mobil yang paling irit digunakan di dalam kota**. Bisa dikatakan, kriteria pemilihan mobil merupakan salah satu motivasi Anda untuk menganalisis dan menggali informasi dari set data `mpg`. Anda ingin tahu, tipe atau jenis mobil seperti apa yang memiliki efisiensi penggunaan bensin yang tinggi sehingga Anda bisa membeli mobil dengan kriteria atau karakterisitik yang sama di Indonesia.

- **Dari beberapa variabel pada suatu set data, kira-kira variabel mana saja yang paling berpengaruh terhadap pengambilan keputusan dalam menjawab permasalahan Anda?** Berkaitan dengan contoh sebelumnya, Anda mencari tahu mobil yang paling irit, maka tentu saja variabel utama yang paling berpengaruh adalah variabel ‘cty’ dan ‘hwy’.

Kedua variabel tersebut merupakan data miles per gallon penggunaan bensin di kota dan jalan tol. Selain kedua variabel tersebut, Anda juga harus mencari variabel lain yang berpengaruh terhadap efisiensi penggunaan mobil.

- **Bagaimana cara mendapatkan temuan atau *insight* pada set data yang berguna dalam mengatasi permasalahan Anda?** Seringlah latihan dan otak-atik data. Untuk mendapatkan *insight*, pertama-tama Anda harus memahami data yang Anda analisis. Set data dapat Anda pahami dengan melakukan banyak visualisasi dan manipulasi sehingga Anda bisa menemukan semacam relasi variabel juga pola-pola tertentu yang jika terus dilakukan dan diasah akan meningkatkan intuisi Anda dalam menarik *insight* dari set data.

5 TIPE HUBUNGAN ANTAR VARIABEL DALAM SET DATA

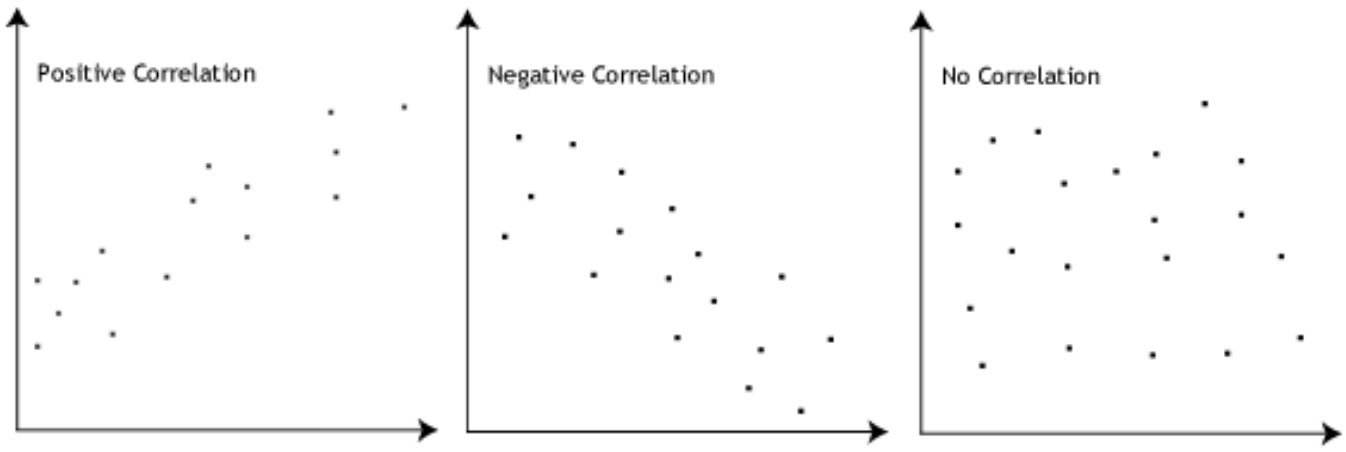
Ketika Anda bermain dengan set data, ada satu titik Anda menemukan sebuah pola atau keterkaitan antar variabel yang bisa membantu Anda mendapatkan *insight*. Anda harus mengetahui 5 tipe hubungan antar variabel dalam set data yang sering ditemukan:

1. Korelasi
2. *Trend*
3. Distribusi
4. *Outlier*
5. Perbandingan dan Peringkat

2.3.1 Korelasi

Korelasi antar dua variabel bisa diidentifikasi dengan menggunakan grafik sebar *(scatter plot)*.

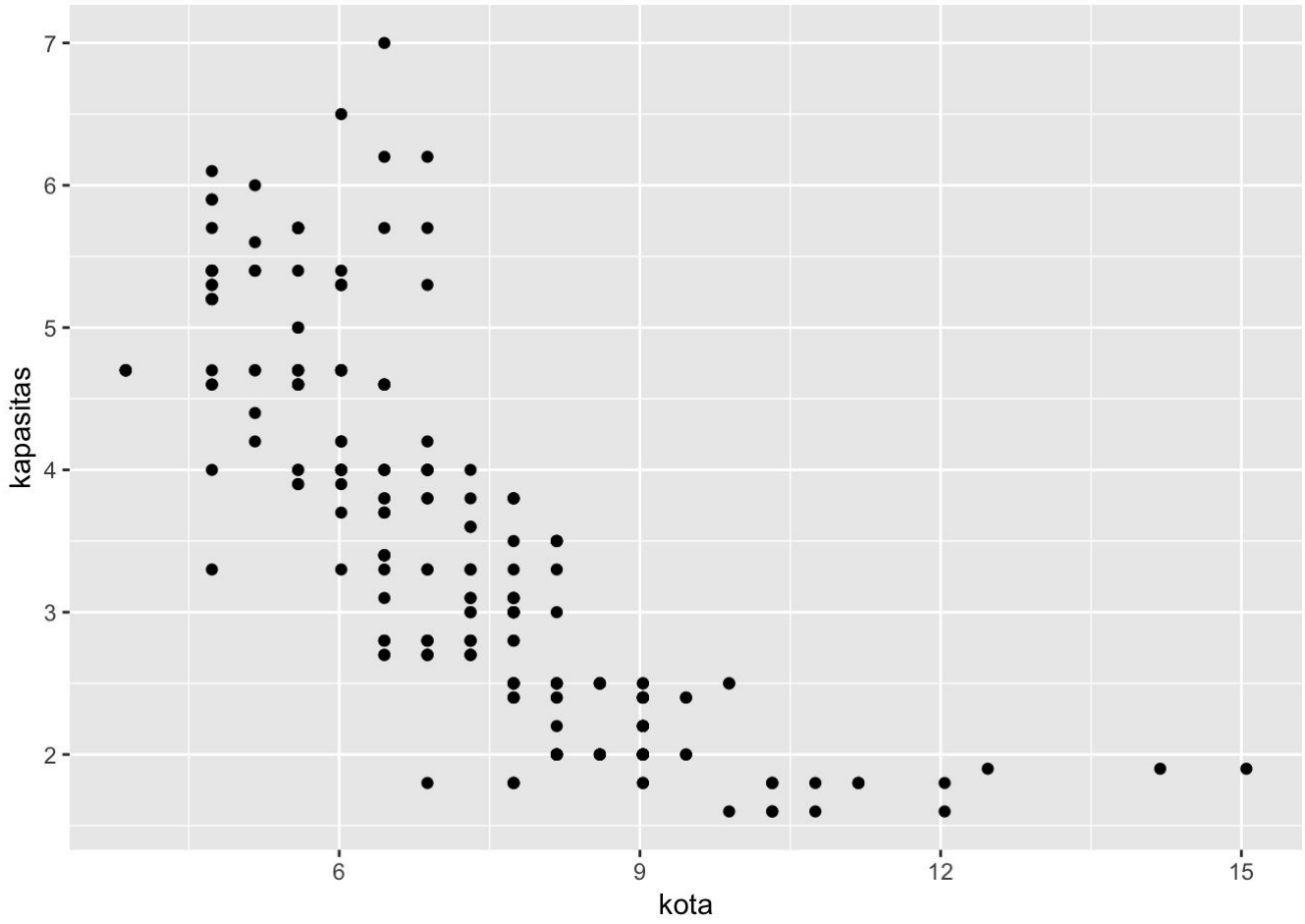
Berikut gambar penjelasannya (sumber: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php):



- Korelasi positif: peningkatan nilai pada variabel pertama berbanding lurus dengan nilai variabel kedua. Variabel A naik, variabel B naik.
- Korelasi negatif: peningkatan nilai pada variabel pertama berbanding terbalik dengan nilai variabel kedua. Variabel A naik, variabel B turun.
- Tidak ada korelasi: tidak ada kolerasi antara variabel pertama dan kedua.

Misalnya, Anda akan mengecek apakah ada korelasi antara variabel ‘kapasitas’ dan ‘kota’ pada set data `df.mobil` .

```
# Membuat grafik sebar antara variabel kapasitas dan kota
# pada set data df.mobil menggunakan fungsi geom_point()
df.mobil %>% ggplot(aes(x = kota, y = kapasitas)) + geom_point()
```



Anda bisa lihat grafik sebar diatas. Sebarannya berpola korelasi negatif. Semakin rendah nilai kapasitas mesin, maka efisiensi penggunaan bensin mobil di dalam kota akan semakin tinggi, demikian sebaliknya.

Selain melalui grafik sebar, koefisien korelasi juga bisa dihitung dengan menggunakan fungsi `cor.test()` .

```
# Mengitung korelasi antara variabel kapasitas dan kota
# pada set data df.mobil menggunakan fungsi cor.test()
  cor.test(df.mobil$kapasitas, df.mobil$kota)
```

```
##
##  Pearson's product-moment correlation
##
## data:  df.mobil$kapasitas and df.mobil$kota
## t = -20.205, df = 232, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8406782 -0.7467508
## sample estimates:
##       cor
## -0.798524
```

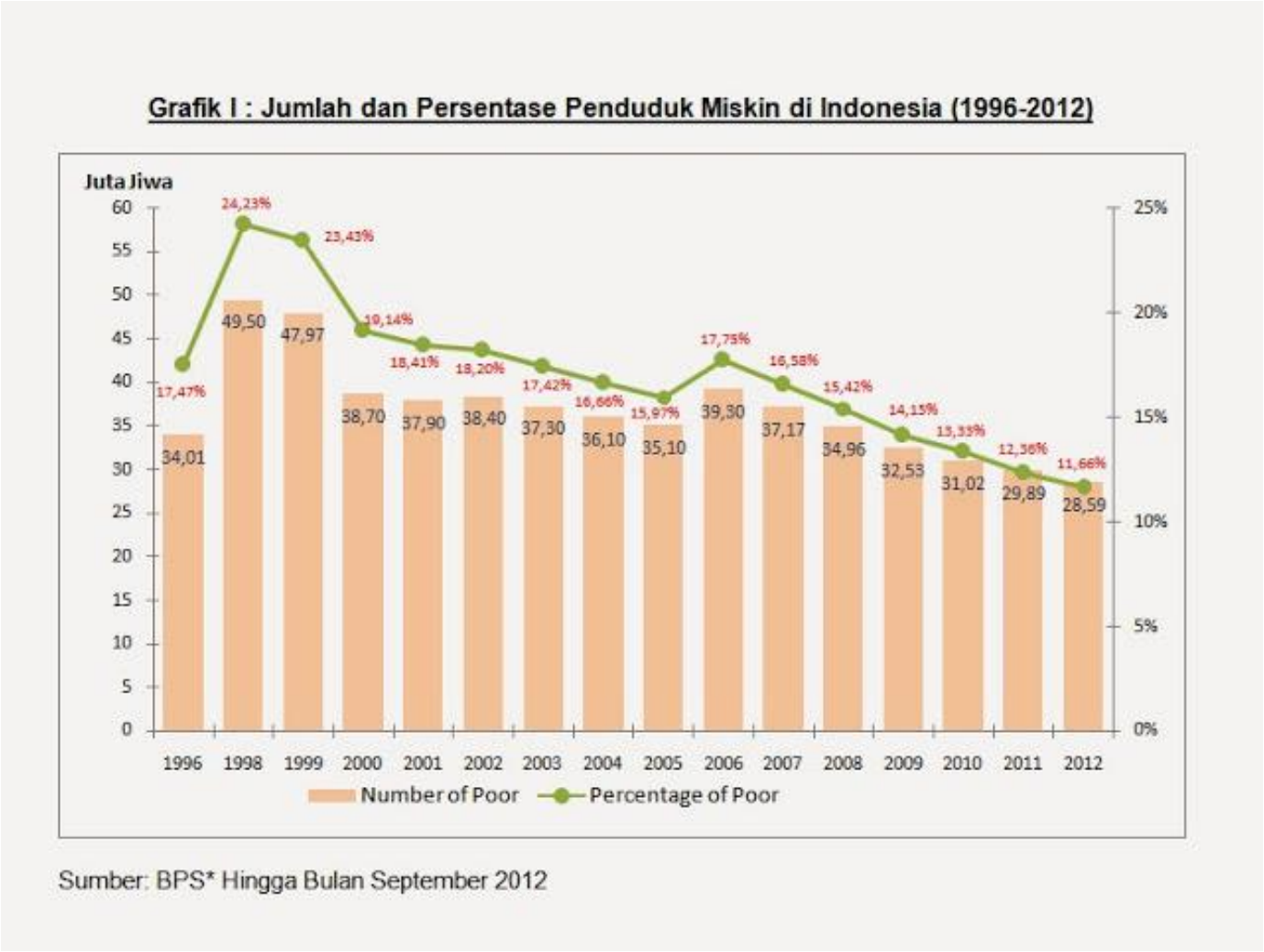
Nilai koefisien korelasinya `-0.798524` . Korelasi negatif.

Korelasinya cukup kuat. Kekuatan korelasi antar dua variabel ditentukan oleh nilai koefisiennya. Semakin mendekati angka `1` atau `-1` , maka korelasinya semakin kuat.

2.3.2 *Trend*

Trend dapat didefinisikan sebagai arah atau pergerakan. Bisa naik atau turun. Biasanya ditunjukkan melalui grafik garis atau grafik batang.

Berikut contoh grafik trend (sumber: http://ririnzuliyaningsih.blogspot.com/2015/05/kemiskinan-dan-kesenjangan-pembangunan.html)



Grafik ini menunjukkan *trend* kemiskinan di Indonesia dari tahun 1996 sampai dengan 2012. Trend naik ditunjukkan mulai tahun 1996 - 1999. Dari tahun 2006 sampai tahun 2012 terlihat penurunan jumlah penduduk miskin.

Kesimpulannya, terjadi penurunan *trend* kemiskinan di Indonesia dalam rentang waktu 16 tahun sejak 1996.

Selain *insight* dari *trend*, Anda juga bisa mencermati dari rentang 16 tahun tersebut, jumlah penduduk miskin tertinggi ada di tahun berapa?

Tahun 1998.

Lalu, Anda coba gali lagi, ada apa di tahun 1998 yang menyebabkan angka kemiskinan menjadi paling tinggi dibandingkan tahun lainnya?

Krisis moneter.

Apakah ketika terjadi krisis di Indonesia, selalu diikuti dengan peningkatan jumlah kemiskinan di tahun yang sama?

Di tahun 2008 terjadi krisis ekonomi, namun tidak diikuti dengan meningkatnya jumlah penduduk miskin.

Kenapa tidak terjadi peningkatan jumlah kemiskinan di tahun 2008? Apa bedanya krisis tahun 1998 dengan tahun 2008? Apa yang menyebabkan krisis? Apakah ada kaitan antara krisis ekonomi dengan kemiskinan?

Dan seterusnya.

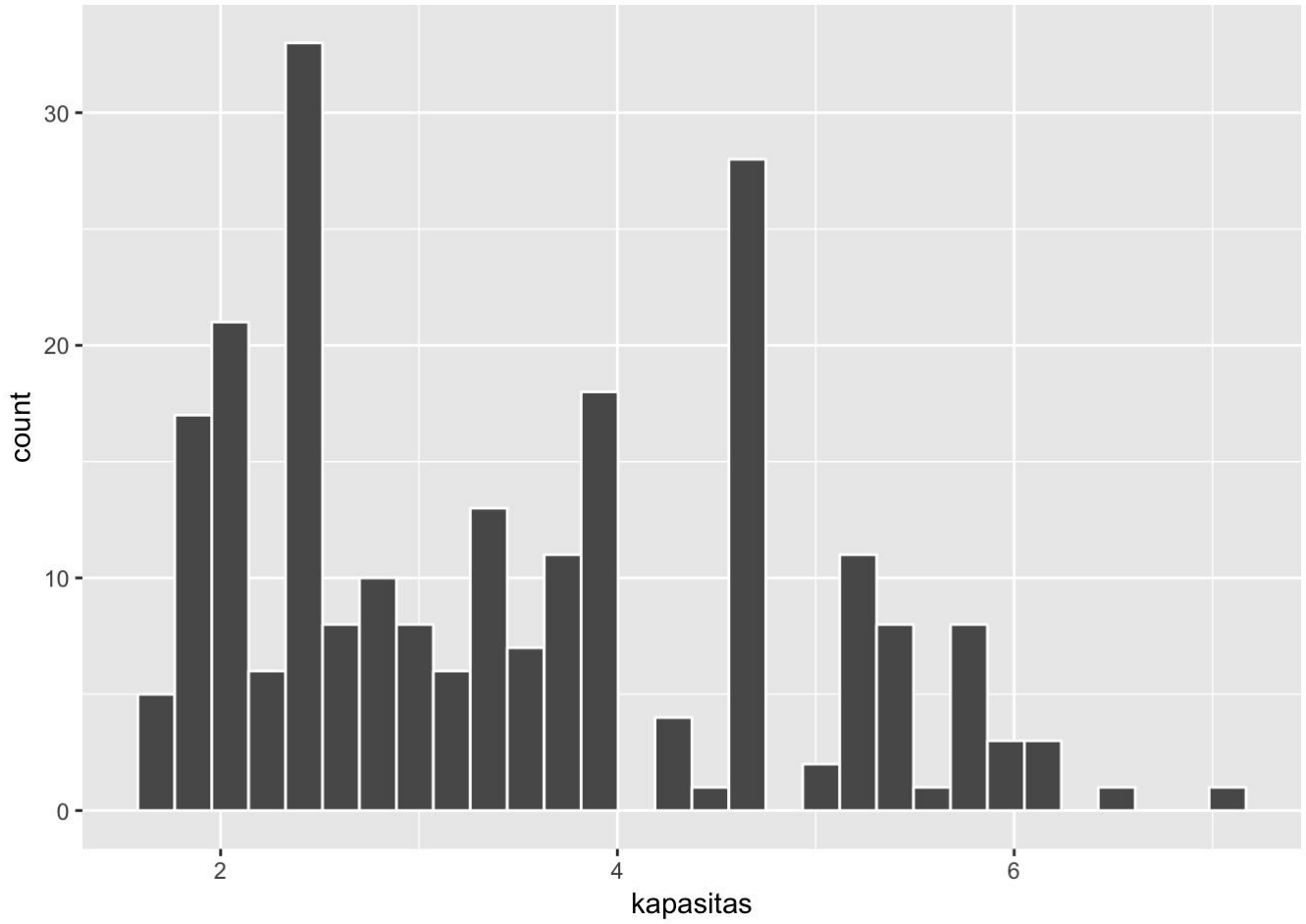
Diiringi dengan rasa penasaran serta antusias, munculkan pertanyaan-pertanyaan yang bisa membuat Anda berpikir kritis. Dengan cara ini, Anda bisa menggali *insight* dan informasi dari sebuah set data atau grafik.

2.3.3 Distribusi

Distribusi sering digunakan untuk mengetahui nilai terendah, tertinggi, rata-rata, median, serta rentang dari suatu variabel pada set data. Grafik histogram biasa dipakai untuk menunjukkan pola distribusi ini.

Sebagai contoh, Anda ingin mengetahui distribusi jumlah mobil berdasarkan kapasitas mesinnya pada set data `df.mobil` .

```
# Membuat grafik histogram kapasitas mesin
# pada set data df.mobil menggunakan fungsi geom_histogram()
df.mobil %>% ggplot(aes(x = kapasitas)) + geom_histogram(color = "white")
```

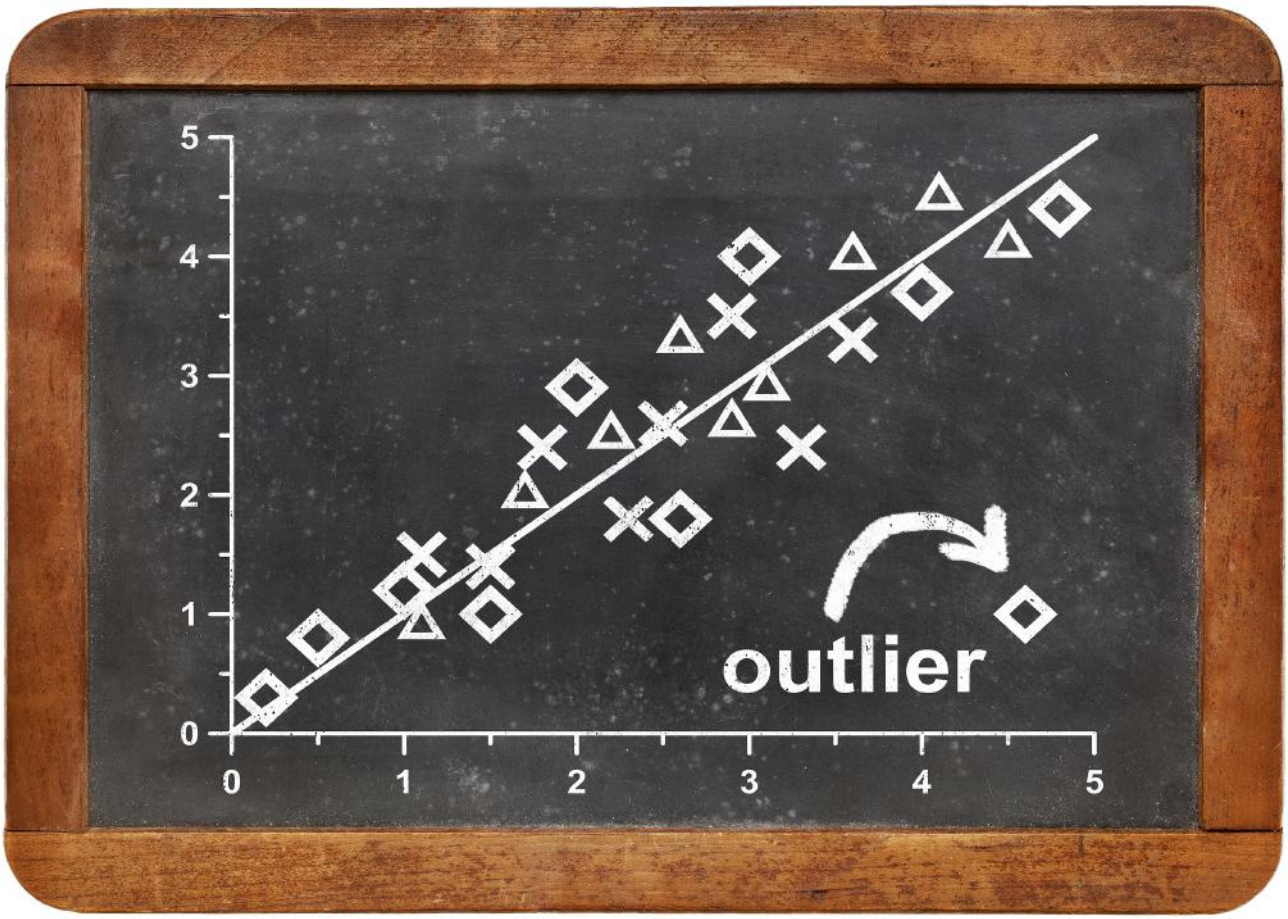


Anda bisa lihat, jumlah mobil terbanyak pada set data `df.mobil` memiliki kapasitas mesin antara 1 - 2.5 liter.

2.3.4 *Outlier*

Outlier atau pencilan adalah data obervasi yang menyimpang terlalu jauh atau bernilai ekstrim dibandingkan data lainnya. Data *outlier* ini bisa terjadi karena beberapa hal, diantaranya kesalahan dalam pemasukan data, kesalahan dalam pengambilan sampel, atau memang ada data – data ekstrim yang tidak dapat dihindarkan keberadaannya. Untuk mengidentifikasinya dapat dilakukan dengan beberapa cara antara lain melalui boxplot atau scatterplot.

Ilustrasi *outlier* (sumber: http://disabilitydunktank.com/tag/social-security-disability-fraud/):



Misalkan Anda seorang penyidik yang bekerja di KPK atau kepolisian. Anda bisa mendapatkan data pendapatan yang diterima pejabat-pejabat Pegawai Negeri Sipil (PNS) tertentu dari bank. Setelah Anda analisis data tersebut, Anda menemukan bahwa rata-rata pendapatan pejabat PNS adalah sekitar 100 juta rupiah per bulan. Anda gali lebih dalam lagi, ternyata ada 3 pejabat yang memiliki pendapatan 10 miliar rupiah per bulan. Inilah yang dinamakan *outlier*.

Apa yang harus dilakukan dengan data *outlier* ini? Tergantung tujuan Anda. Jika Anda seorang penyidik seperti contoh di atas, tentu Anda memfokuskan pada 3 pejabat tersebut. Data *outlier* ini dapat dijadikan sebuah *insight* bernilai bagi Anda. Anda bisa melakukan penyidikan lebih lanjut terhadap ketiga pejabat tersebut. Dari mana pendapatan mereka berasal?

Jika Anda, misalkan seorang penentu kebijakan di pemerintahan dan ingin mengkaji usulan kenaikan gaji PNS, maka data *outlier* tersebut sebaiknya disingkirkan. *Outlier* dapat menyebabkan penyimpangan hasil analisis data yang Anda lakukan.

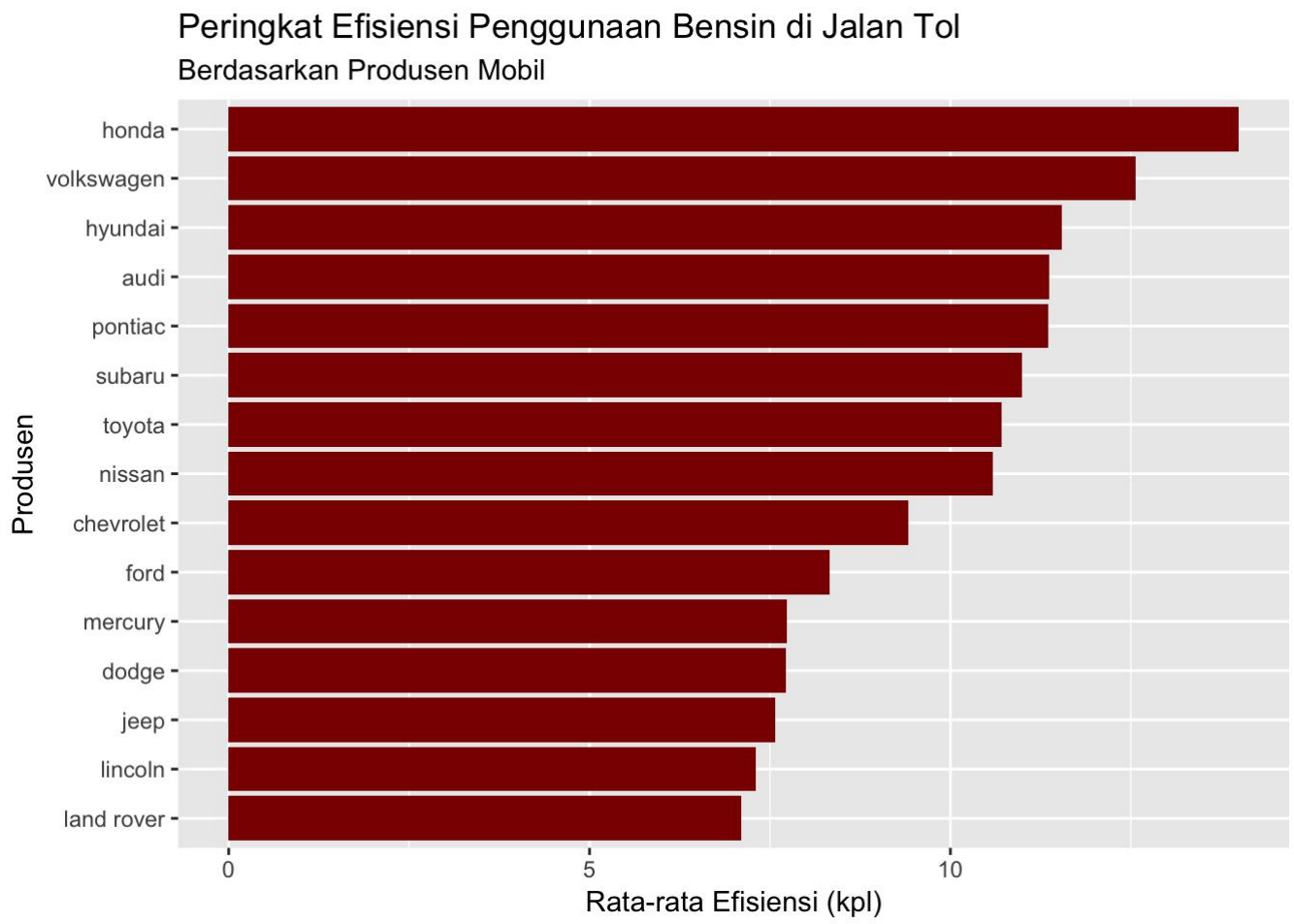
2.3.5 Perbandingan dan Peringkat

Cara ini sering dilakukan jika Anda ingin mengetahui nilai tertinggi dan terendah serta perbandingan antara satu data obervasi dengan yang lainnya. Yang sering digunakan biasanya grafik batang.

Perbandingan dan Peringkat ini sudah disinggung sebelumnya di bagian (#mengurutkan-data).

Langsung ke contoh. Misalkan Anda ingin mengetahui peringkat produsen mobil yang memiliki efisiensi penggunaan bensin (kpl) tertinggi di jalan tol dari set data `df.mobil` .

```
# Mengelompokkan berdasarkan variabel produsen
# lalu meringkas nilai rata-rata efisiensi di jalan tol
# dan menampilkannya dalam grafik batang
df.mobil %>%
  group_by(produsen) %>%
  summarise(rerata.tol = mean(jalan.tol)) %>%
  ggplot(aes(y = rerata.tol, x = reorder(produsen, rerata.tol))) +
  geom_bar(stat = "identity", fill = "#8B0000") +
  labs(x = "Produsen", y = "Rata-rata Efisiensi (kpl)") +
  ggtitle("Peringkat Efisiensi Penggunaan Bensin di Jalan Tol",
          subtitle = "Berdasarkan Produsen Mobil") +
  coord_flip()
```



Dengan melihat grafik batang di atas, Anda dapat mengetahui produsen yang memiliki efisiensi tertinggi adalah Honda, sedangkan yang terendah adalah Land Rover. Selain itu, Anda juga bisa membandingkan efisiensi dari masing-masing produsen mobil lainnya.

Langkah Kilat 3 Latihan, Latihan, Latihan

Ya, tidak ada yang spesial atau hal yang *wah* di langkah ketiga ini. Untuk bisa mahir analisis data, Anda harus banyak latihan.

Baca kembali 2 langkah sebelumnya, terutama langkah kedua. Kuasai teknik dasar analisis terlebih dahulu.

Jalankan kode programnya, *copy paste* boleh, pelajari cara kerja dan konsepnya. Setelah Anda sudah tidak merasa asing dengan sistem pemrograman R, coba Anda ketik kode programnya, tidak *copy paste*. Dengan mengetik, Anda akan lebih cepat terbiasa dengan beberapa fungsi yang sering digunakan untuk mengolah data.

Lalu, Anda coba latihan dengan set data yang lain. Jelajahi set data tersebut dengan fungsi-fungsi yang sudah dibahas di buku ini.

Berikut akan disajikan bagaimana cara melakukan analisis data dari mulai proses pengambilan sampai dengan mendapatkan *insight*. Fungsi-fungsi yang digunakan pada contoh kali ini sebagian sudah dibahas di bagian sebelumnya dan sebagian akan dibahas beriringan dengan penulisan kode.

3.1 Unduh, Inisiasi, dan Mempersiapkan Set Data

Set data yang akan digunakan dapat diunduh secara manual pada tautan berikut:

- Data Jumlah Penduduk Jakarta Berdasarkan Jenis Kelamin dan Kewarganegaraan

Anda bisa langsung inisiasi dan menyimpan set data ini ke dalam variabel tanpa harus mengunduhnya terlebih dahulu.

```
# Meyimpan alamat tautan (URL) data penduduk Jakarta berdasarkan jenis kelamin dan kewar
alamat <- "http://data.jakarta.go.id/dataset/aed0be8f-b36e-457c-9210-6f5e970b987c/resour

# Membaca file csv dari set data dan memasukkannya ke dalam variabel
df.penduduk <- read.csv(alamat, sep = ",")
```

Set data kependudukan Jakarta tersedia dalam bentuk file csv, sehingga untuk membaca data tersebut digunakan fungsi `read.csv` .

Melihat struktur dari set data `df.penduduk` .

```
str(df.penduduk)
```

```
## 'data.frame':    1068 obs. of  8 variables:
##  $ tahun          : int  2017 2017 2017 2017 2017 2017 2017 2017 2017 2017 ...
##  $ provinsi       : Factor w/ 1 level "DKI Jakarta": 1 1 1 1 1 1 1 1 1 ...
##  $ kabupaten      : Factor w/ 6 levels "JAKARTA BARAT",..: 6 6 6 6 6 6 2 2 2 2 ...
##  $ kecamatan      : Factor w/ 44 levels "CAKUNG","CEMPAKA PUTIH",..: 22 22 22 21 21 2
##  $ kelurahan      : Factor w/ 267 levels "ANCOL","ANGKE",..: 165 164 163 168 167 166
##  $ kewarganegaraan: Factor w/ 2 levels "WNA","WNI": 2 2 2 2 2 2 2 2 2 ...
##  $ jenis_kelamin  : Factor w/ 2 levels "Laki-laki","Perempuan": 1 1 1 1 1 1 1 1 1 .
##  $ jumlah         : int  3426 3546 1241 1204 2797 1688 1645 9802 11224 9333 ...
```

Seperti yang Anda lihat, beberapa variabel seperti provinsi, kabupaten, kecamatan, kelurahan, kewarganegaraan, serta jenis kelamin sudah bertipe faktor, sehingga tidak perlu diubah lagi. Variabel yang lain seperti tahun dan jumlah sudah benar bertipe integer.

Yang perlu sedikit diubah adalah penamaan variabel kabupaten. Di DKI Jakarta, penamaan wilayah administratif setelah propinsi lebih tepatnya kota, bukan kabupaten.

Masih ingat bagaimana cara mengubah nama variabel?

```
# Mengubah nama variabel kabupaten menjadi kota pada set data df.penduduk
names(df.penduduk)[3] <- "kota"
```

Bagaimana dengan informasi jumlah penduduk Jakarta?

```
# Jumlah total penduduk Jakarta
sum(df.penduduk$jumlah)
```

```
## [1] 10348570
```

```r
# Jumlah penduduk Jakarta terbanyak berdasarkan kecamatan
df.penduduk %>%
  select(kecamatan, jumlah) %>%
  group_by(kecamatan) %>%
  summarise(jumlah = sum(jumlah)) %>%
  arrange(-jumlah) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   kecamatan     jumlah
##   <fct>          <int>
## 1 CENGKARENG    513064
## 2 CAKUNG        509194
## 3 DUREN SAWIT   406998
## 4 CILINCING     403028
## 5 TANJUNG PRIOK 395022
```

```r
# Jumlah penduduk Jakarta paling sedikit berdasarkan kecamatan
df.penduduk %>%
  select(kecamatan, jumlah) %>%
  group_by(kecamatan) %>%
  summarise(jumlah = sum(jumlah)) %>%
  arrange(jumlah) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   kecamatan       jumlah
##   <fct>            <int>
## 1 KEP. SERIBU SLT  11342
## 2 KEP. SERIBU UTR  16249
## 3 MENTENG          92026
## 4 CEMPAKA PUTIH    99263
## 5 GAMBIR          101832
```

```
# Peringkat jumlah penduduk Jakarta berdasarkan kota
df.penduduk %>%
  select(kota, jumlah) %>%
  group_by(kota) %>%
  summarise(jumlah = sum(jumlah)) %>%
  arrange(-jumlah)
```

```
## # A tibble: 6 x 2
##   kota                jumlah
##   <fct>                <int>
## 1 JAKARTA TIMUR      2946926
## 2 JAKARTA BARAT      2327258
## 3 JAKARTA SELATAN    2190919
## 4 JAKARTA UTARA      1716591
## 5 JAKARTA PUSAT      1139285
## 6 KAB.ADM.KEP.SERIBU   27591
```

3.2 Visualisasi Grafik

Set data kependudukan `df.penduduk` ini dari awal sudah lumayan rapih dan tidak diperlukan proses manipulasi. Langkah berikutnya menggunakan `ggplot` untuk memvisualisasikan data agar bisa mendapatkan `insight` dari set data kependudukan ini.

```
# Inisiasi paket ggplot dan dplyr
library(ggplot2)
library(dplyr)

df.penduduk %>%
  ggplot(aes(x = reorder(kecamatan, jumlah, sum), y = jumlah)) +
  geom_bar(stat = "identity", fill = "#8B0000") +
  coord_flip() +
  theme(axis.text.y = element_text(size = 6)) +
  scale_y_continuous(labels = scales::comma_format()) +
  labs(x = "Kecamatan", y = "Jumlah Penduduk") +
  ggtitle("Perbandingan Jumlah Penduduk Jakarta Berdasarkan Kecamatan",
          subtitle = "Tahun 2017")
```

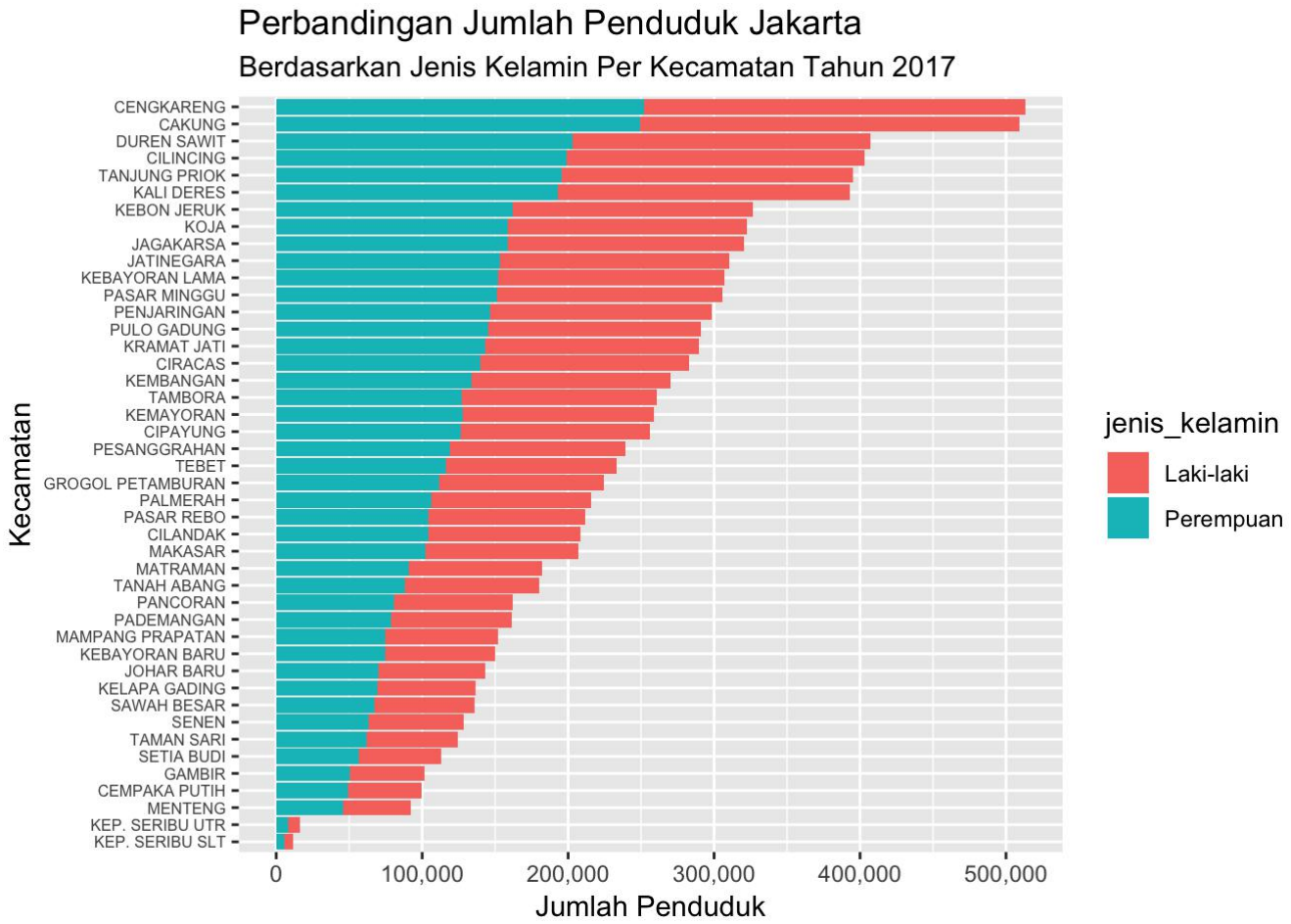
Penjelasan kode:

- `ggplot(aes(x = reorder(kecamatan, jumlah, sum), y = jumlah))` : mengurutkan faktor kecamatan berdasarkan jumlah dengan fungsi `reorder` .

- `theme(axis.text.y = element_text(size = 6))` : mengatur huruf pada axis y menjadi berukuran 6.
- `scale_y_continuous(labels = scales::comma_format()` : mengatur skala angka menjadi format yang mudah dibaca.

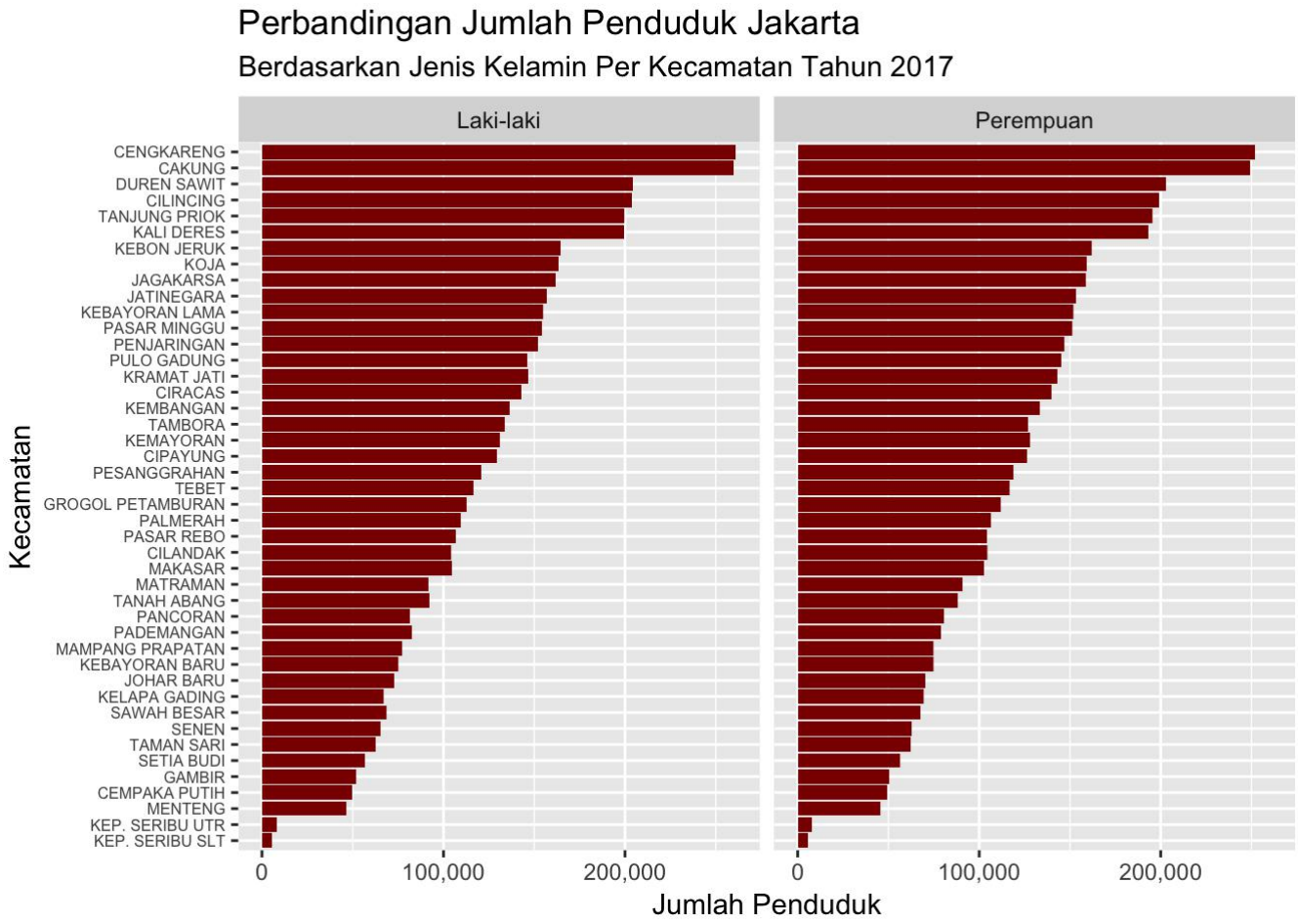
```
df.penduduk %>%
  ggplot(aes(x = reorder(kota, jumlah, sum), y = jumlah)) +
  geom_bar(stat = "identity", fill = "#8B0000") +
  coord_flip() +
  theme(axis.text.y = element_text(size = 8)) +
  theme(axis.text.x = element_text(size = 7)) +
  scale_y_continuous(labels = scales::comma_format()) +
  labs(x = "Kota", y = "Jumlah Penduduk") +
  ggtitle("Perbandingan Jumlah Penduduk Jakarta Berdasarkan Kota",
          subtitle = "Tahun 2017")
```

```
df.penduduk %>%
  ggplot(aes(x = reorder(kecamatan, jumlah, sum), y = jumlah)) +
  geom_bar(stat = "identity", aes(fill = jenis_kelamin)) +
  coord_flip() +
  theme(axis.text.y = element_text(size = 6)) +
  scale_y_continuous(labels = scales::comma_format()) +
  labs(x = "Kecamatan", y = "Jumlah Penduduk") +
  ggtitle("Perbandingan Jumlah Penduduk Jakarta",
          subtitle = "Berdasarkan Jenis Kelamin Per Kecamatan Tahun 2017")
```



Argumen `fill` pada kode di atas berfungsi untuk menampilkan atribut estetik berdasarkan variabel jenis_kelamin, sehingga pada grafik terlihat komposisi penduduk laki-laki dan perempuan yang berbeda warna.

```
df.penduduk %>%
  ggplot(aes(x = reorder(kecamatan, jumlah, sum), y = jumlah)) +
  geom_bar(stat = "identity", fill = "#8B0000") +
  coord_flip() +
  theme(axis.text.y = element_text(size = 6)) +
  scale_y_continuous(labels = scales::comma_format()) +
  facet_wrap(~jenis_kelamin) +
  labs(x = "Kecamatan", y = "Jumlah Penduduk") +
  ggtitle("Perbandingan Jumlah Penduduk Jakarta",
          subtitle = "Berdasarkan Jenis Kelamin Per Kecamatan Tahun 2017")
```



Fungsi `facet_wrap()` membuat grafik terbagi menjadi dua kolom berdasarkan jenis kelamin. Opsi ini mempermudah pengguna untuk melihat lebih detail untuk masing-masing kategori.

3.3 Visualisasi Peta

Visualisasi data juga dapat ditampilkan dalam bentuk peta. Selain cukup informatif, visualisasi peta juga secara estetis indah dipandang dan keren.

Untuk membuat visualisasi peta Jakarta, Anda perlu menginstal paket berikut terlebih dahulu.

```
install.packages("devtools")
install.packages("sf")
devtools::install_github("tidyverse/ggplot2")
devtools::install_github("rasyidstat/indonesia")
```

Inisiasi paket dan persiapan visualiasi peta Jakarta.

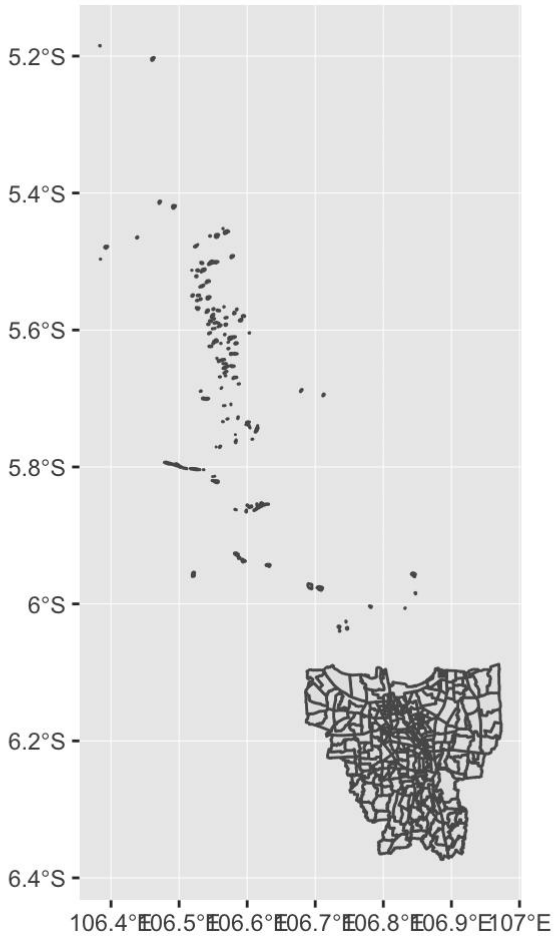
```
# Persiapan data spasial peta Jakarta dengan pembagian wilayah kelurahan
kelurahan.jkt <- id_map("jakarta", "kelurahan")

# Menampilkan struktur set data kelurahan.jkt
str(kelurahan.jkt)
```

```
## Classes 'sf' and 'data.frame':   377 obs. of  9 variables:
##  $ kode_provinsi : int  31 31 31 31 31 31 31 31 31 31 ...
##  $ nama_provinsi : Factor w/ 1 level "DAERAH KHUSUS IBUKOTA JAKARTA": 1 1 1 1 1 1 1 1
##  $ kode_kota     : int  3171 3172 3172 3172 3172 3171 3172 3172 3171 3172 ...
##  $ nama_kota     : Factor w/ 6 levels "Jakarta Barat",..: 3 4 4 4 4 3 4 4 3 4 ...
##  $ kode_kecamatan: int  3171010 3172020 3172030 3172030 3172010 3171010 3172010 31726
##  $ nama_kecamatan: Factor w/ 45 levels "Cakung","Cempaka Putih",..: 11 7 6 6 34 11 34
##  $ kode_kelurahan: num  3.17e+09 3.17e+09 3.17e+09 3.17e+09 3.17e+09 ...
##  $ nama_kelurahan: Factor w/ 267 levels "Ancol","Angke",..: 35 23 202 162 179 236 87
##  $ geometry      :sfc_POLYGON of length 377; first list element: List of 1
##   ..$ : num [1:1283, 1:2] 107 107 107 107 107 ...
##   ..- attr(*, "class")= chr  "XY" "POLYGON" "sfg"
##  - attr(*, "sf_column")= chr "geometry"
##  - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA NA NA
##   ..- attr(*, "names")= chr  "kode_provinsi" "nama_provinsi" "kode_kota" "nama_kota"
```

Melalui fungsi `str()` diketahui bahwa set data `kelurahan.jkt` bertipe ‘sf’ dan ‘data.frame’. Simple features atau sf adalah bentuk set data yang didalamnya berisi variabel geometri dua dimensi (titik, garis, poligon). Di dalam set data `kelurahan.jkt` diwakili oleh variabel ‘geometry’.

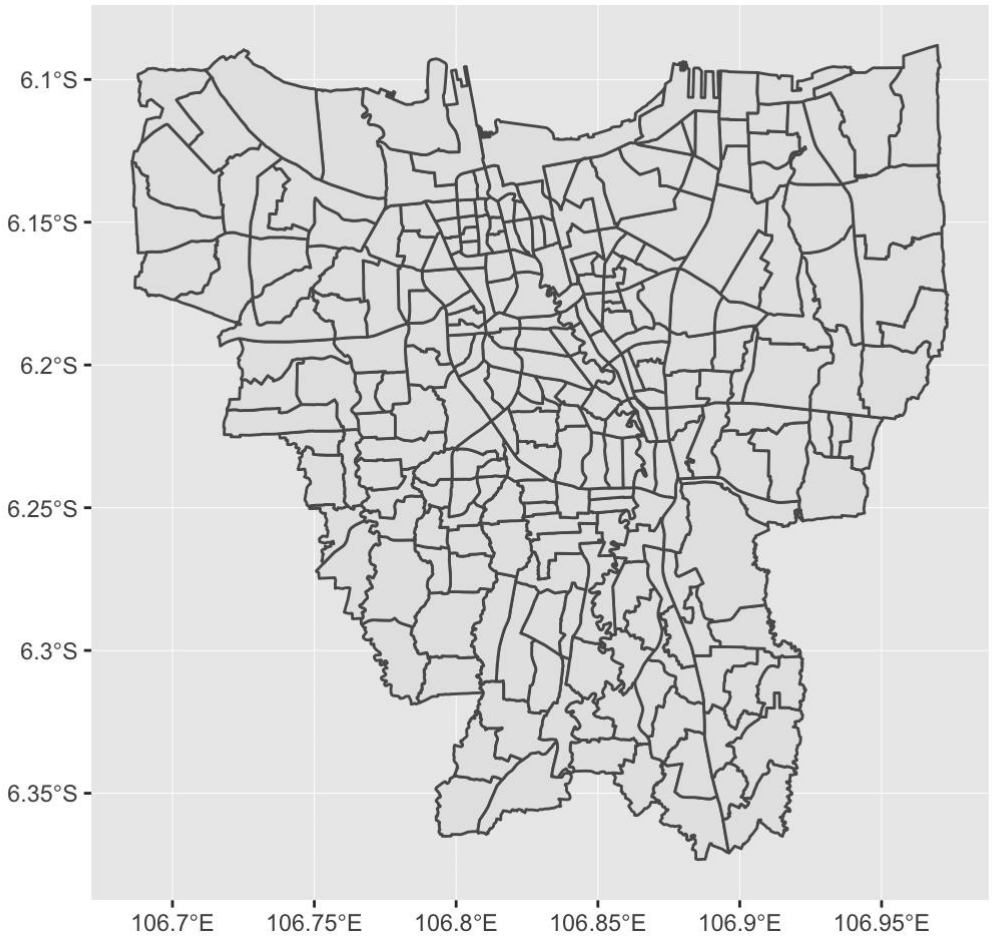
Untuk memvisualisasikan peta, gunakan paket `ggplot` diikuti dengan fungsi `geom_sf()` .

```
# Visualisasi peta dasar Jakarta
kelurahan.jkt %>% ggplot() + geom_sf()
```



Visualisasi peta dasar Jakarta yang dihasilkan dari kode di atas tidak terlalu bagus. Seperti yang Anda lihat, Kepulauan Seribu termasuk dalam wilayah Jakarta, sehingga visualisasi yang dihasilkan mencakup seluruh wilayah administratif sesuai variabel pada set data `kelurahan.jkt` . Agar visualisasi yang dihasilkan lebih estetis, maka wilayah Kepulauan Seribu akan dikeluarkan dari set data.

```
# Mengeluarkan data observasi kecamatan KEPULAUAN SERIBU
# dan memvisualisasikan peta dengan ggplot dan fungsi geom_sf
kelurahan.jkt %>%
  filter(!nama_kecamatan %in% c("KEPULAUAN SERIBU UTARA", "KEPULAUAN SERIBU SELATAN")) %
  ggplot() + geom_sf()
```



Nah, hasilnya sudah lumayan bagus. Peta difokuskan pada wilayah Jakarta yang ada di pulau Jawa, tidak termasuk wilayah Kepulauan Seribu.

Selanjutnya yang harus dilakukan adalah menggabungkan set data `kelurahan.jkt` dan `df.penduduk` sehingga visualisasi peta kepadatan penduduk Jakarta dapat ditampilkan.

Syarat agar dua set data bisa digabung: ada satu variabel memiliki isi yang sama pada kedua set data. Pada contoh ini, variabel 'jumlah' pada set data `df.penduduk` akan digabungkan ke set data `kelurahan.jkt` melalui variabel kelurahan/nama_kelurahan.

Jadi, harus dipastikan dulu penamaan variabel kelurahan/nama_kelurahan beserta isinya pada kedua set data harus sama.

```
# Meringkas variabel kelurahan dan jumlah penduduk pada set data df.penduduk
# lalu menyimpannya di variabel jml.penduduk
jml.penduduk <-
  df.penduduk %>%
    group_by(kelurahan) %>%
    summarise(jumlah = sum(jumlah))
```

```
# Mengambil isi variabel nama_kelurahan pada set data keluahan.jkt
# dan menyimpannya di variabel kelurahan
kelurahan <- sort(unique(kelurahan.jkt$nama_kelurahan))


# Menggabungkan set data jml.penduduk dan vektor kelurahan
# menggunakan fungsi data.frame
cek.kelurahan <- data.frame(jml.penduduk, nama_kelurahan = kelurahan)


# Menampilkan isi set data cek.kelurahan
head(cek.kelurahan)


##      kelurahan jumlah nama_kelurahan
## 1        ANCOL  28870          Ancol
## 2        ANGKE  34663          Angke
## 3 BALE KAMBANG  32083   Bale Kambang
## 4  BALI MESTER  11290     Balimester

## 5   BAMBU APUS  28952     Bambu Apus
## 6       BANGKA  25245         Bangka


# Mengecek adakah perbedaan penamaan kelurahan dengan membuat
# kolom baru bernama cek_nama yang berisi data logikal menggunakan
# menggunakan fungsi mutate() dari paket dplyr pada set data cek.kelurahan
cek.kelurahan <-
  cek.kelurahan %>%
  mutate(cek_nama = ifelse(tolower(kelurahan) %in% tolower(nama_kelurahan), "TRUE", "FAl
```

Operator `==` bersifat *case sensitive*, artinya perbedaan huruf besar/kecil juga menjadi faktor pengetesan logika.

Contohnya: `ifelse("ANCOL" == "Ancol", "TRUE", "FALSE")` akan menghasilkan output `FALSE` .

Fungsi `tolower()` yang diaplikasikan di variabel kelurahan dan nama_kelurahan membuat isi dari kedua variabel tersebut menjadi huruf kecil semua sehingga pengetesan logika menjadi lebih optimal.

```
# Mengecek nama kelurahan yang berbeda dengan fungsi filter
cek.kelurahan %>%
  select(kelurahan, cek_nama) %>%
  filter(cek_nama == "FALSE")
```

```
##                kelurahan cek_nama
## 1             BALI MESTER    FALSE
## 2  HALIM PERDANA KUSUMAH    FALSE
## 3           HARAPAN MULIA    FALSE
## 4              KALIDERES    FALSE
## 5              KERENDANG    FALSE
## 6        KOTA BAMBU UTARA    FALSE
## 7              P. HARAPAN    FALSE
## 8              P. KELAPA    FALSE
## 9             P. PANGGANG    FALSE
## 10               P. PARI    FALSE
## 11             P. TIDUNG    FALSE
## 12         P. UNTUNG JAWA    FALSE
## 13             PAL MERIAM    FALSE
## 14               PALMERAH    FALSE
```

Ternyata ada 14 nama kelurahan yang berbeda. Untuk memperbaiki tulisan nama kelurahan, harus dibandingkan dulu versi nama kelurahan dari kedua set data tersebut.

```
# Mengecek nama kelurahan yang berbeda dengan fungsi filter
cek.kelurahan %>%
  select(nama_kelurahan) %>%
  filter(grepl("mester|halim|harapan|deres|endang|bambu|
               |kelapa|panggang|pari|tidung|untung|meriam|merah",
               nama_kelurahan, ignore.case = TRUE))
```

```
##          nama_kelurahan
## 1            Balimester
## 2            Bambu Apus
## 3  Halim Perdana Kusuma
## 4         Harapan Mulya
## 5            Kali Deres
## 6          Kebon Kelapa
## 7            Kelapa Dua
## 8      Kelapa Dua Wetan
## 9   Kelapa Gading Barat
## 10  Kelapa Gading Timur
## 11   Kota Bambu Selatan
## 12      Kotabambu Utara
## 13             Krendang
## 14            Pal Merah
## 15            Palmeriam
## 16         Pondok Bambu
## 17        Pondok Kelapa
## 18        PULAU HARAPAN
## 19         PULAU KELAPA
## 20       PULAU PANGGANG
## 21           PULAU PARI
## 22         PULAU TIDUNG
## 23    PULAU UNTUNG JAWA
## 24         Sungai Bambu
```

Dari hasil di atas, terdapat 14 perbedaan penamaan kelurahan di antara dua set data sebagai berikut:

1. bali mester - balimester
2. halim perdana kusumah - halim perdana kusuma
3. harapan mulia - harapan mulya
4. kalideres - kali deres
5. kerendang - krendang
6. kota bambu utara - kotabambu utara
7. p. harapan - pulau harapan
8. p. kelapa - pulau kelapa
9. p. panggang - pulau panggang

10. p. pari - pulau pari
11. p. tidung - pulau tidung
12. p. untung jawa - pulau untung jawa
13. pal meriam - palmeriam
14. palmerah - pal merah

Selanjutnya, mengubah nama kelurahan di salah satu set data. Untuk contoh kali ini, nama kelurahan pada set data `jml.penduduk` akan diubah mengikuti penamaan kelurahan set data `kelurahan.jkt` .

Setelah penamaan kelurahan sudah sama persis, penggabungan kedua set data ini bisa dilakukan.

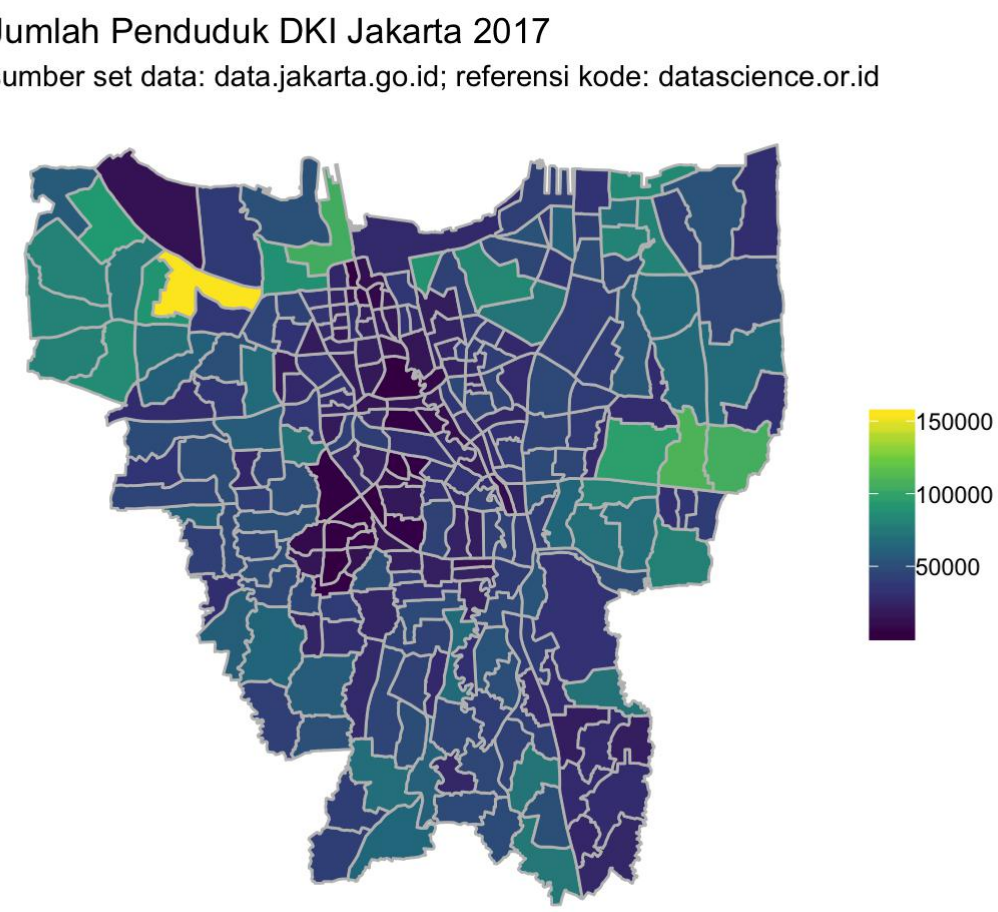
```
# Mengubah nama kelurahan menjadi huruf kecil semua pada set data
# jml.penduduk dan kelurahan.jkt untuk memudahkan manipulasi string
jml.penduduk$kelurahan <- as.factor(tolower(jml.penduduk$kelurahan))
kelurahan.jkt$nama_kelurahan <- as.factor(tolower(kelurahan.jkt$nama_kelurahan))

# Mengubah penamaan kelurahan pada set data jml.penduduk
# mengikuti peamaan kelurahan set data kelurahan.jkt
# menggunakan fungsi fct_recode
jml.penduduk <-
  mutate(jml.penduduk, kelurahan =
           fct_recode(kelurahan,
             "balimester" = "bali mester",
             "halim perdana kusuma" = "halim perdana kusumah",
             "harapan mulya" = "harapan mulia",
             "kali deres" = "kalideres",
             "krendang" = "kerendang",
             "kotabambu utara" = "kota bambu utara",
             "pulau harapan" = "p. harapan",
             "pulau kelapa" = "p. kelapa",
             "pulau panggang" = "p. panggang",
             "pulau pari" = "p. pari",
             "pulau tidung" = "p. tidung",
             "pulau untung jawa" = "p. untung jawa",
             "palmeriam" = "pal meriam",
             "pal merah" = "palmerah"
  ))
```

String di sebelah kiri `=` adalah nama baru, sedangkan di sebelah kanan `=` adalah nama lama.

Fungsi `fct_recode()` pada dasarnya sama dengan fungsi `gsub()`. Perbedaannya, mengubah string dengan menggunakan `fct_recode()` dilakukan pada variabel faktor, sedangkan `gsub()` pada tipe variabel karakter.

Jika `gsub()` digunakan untuk memanipulasi string pada tipe variabel faktor, maka ketika diaplikasikan variabel tersebut akan berubah menjadi tipe variabel karakter.

```
# Menggabungkan set data jml.penduduk ke set data kelurahan.jkt
# menggunakan fungsi left_join dari paket dplyr
kelurahan.jkt <-
  left_join(kelurahan.jkt, jml.penduduk, by = c("nama_kelurahan" = "kelurahan"))


# Memvisualisasikan peta jumlah penduduk Jakarta
kelurahan.jkt %>%
  filter(!nama_kecamatan %in% c("KEPULAUAN SERIBU UTARA", "KEPULAUAN SERIBU SELATAN")) %
  ggplot() +
  geom_sf(aes(fill = jumlah), color = "gray75") +
  scale_fill_viridis_c() +
  ggtitle("Jumlah Penduduk DKI Jakarta 2017",
          subtitle = "sumber set data: data.jakarta.go.id; referensi kode: datascience.c
  theme(panel.background = element_blank()) +
  theme(axis.title = element_blank()) +
  theme(axis.text = element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(legend.title = element_blank())
```

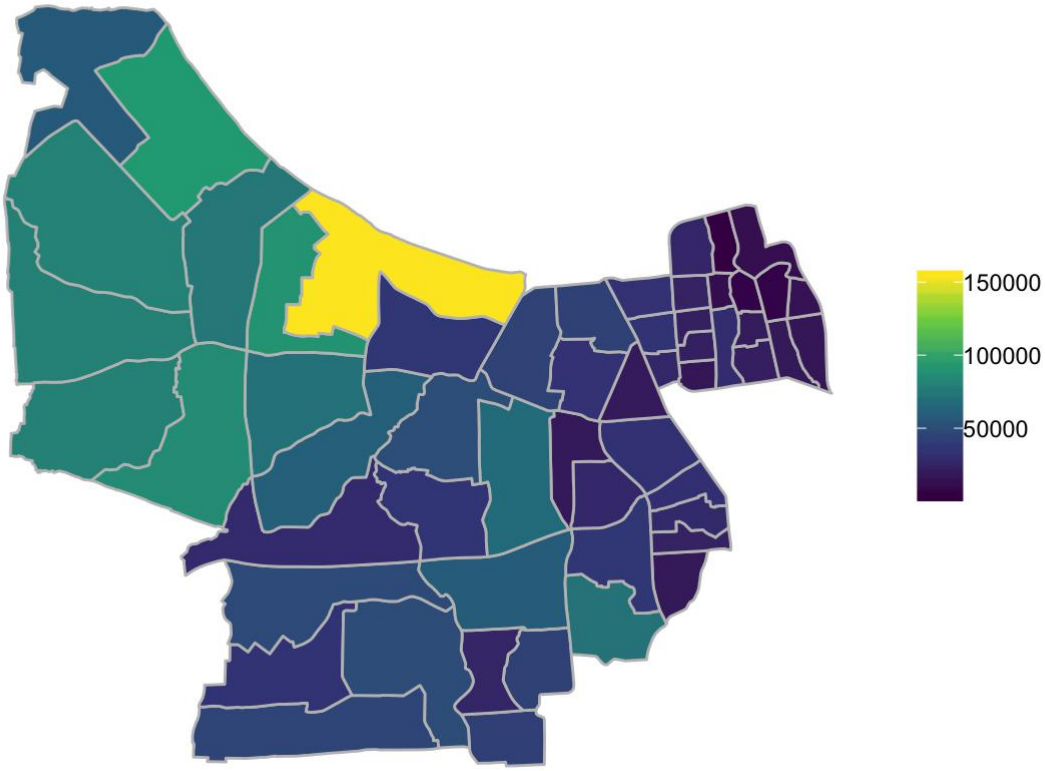


Jika Anda ingin menampilkan salah satu kota di DKI Jakarta, tinggal menambahkan saringan set data untuk kota terkait. Contoh berikut menampilkan kota Jakarta Barat.

```
# Memvisualisasikan peta jumlah penduduk Jakarta Barat
kelurahan.jkt %>%
  filter(!nama_kecamatan %in% c("KEPULAUAN SERIBU UTARA", "KEPULAUAN SERIBU SELATAN") &
           nama_kota == "Jakarta Barat") %>%
  ggplot() +
  geom_sf(aes(fill = jumlah), color = "gray75") +
  scale_fill_viridis_c() +
  ggtitle("Jumlah Penduduk Jakarta Barat 2017",
          subtitle = "sumber set data: data.jakarta.go.id; referensi kode: datascience.c
  theme(panel.background = element_blank()) +
  theme(axis.title = element_blank()) +
  theme(axis.text = element_blank()) +
  theme(axis.ticks = element_blank()) +
  theme(legend.title = element_blank())
```

Jumlah Penduduk Jakarta Barat 2017

sumber set data: data.jakarta.go.id; referensi kode: datascience.or.id



3.4 *Insight*

Beberapa *insight* atau informasi yang bisa didapat dari hasil eksplorasi di atas antara lain sebagai berikut:

1. Jumlah penduduk terbanyak ada di kecamatan Cengkareng, sedangkan untuk kategori kota ada di Jakarta Timur.
2. Jumlah penduduk paling sedikit terdapat di Kepulauan Seribu.
3. DKI Jakarta dalam angka jumlah penduduk tahun 2017:
    - Total = 10.348.570.
    - Kecamatan terbanyak, Cengkareng = 513.064.
    - Kecamatan/kota tersedikit, Kepulauan Seribu = 27.591.
    - Kota terbanyak, Jakarta Timur = 2.946.926.

3.5 Referensi

Berikut referensi yang bisa Anda jadikan panduan untuk lebih cepat menguasai R:

- *Cheat Sheet* RStudio. *Cheat Sheet* atau contekan yang dirangkum dalam 2-3 lembar ini membantu dan mempermudah Anda untuk menavigasi fungsi-fungsi yang diperlukan ketika proses eksplorasi data.
- Kursus online gratis dari DataCamp. Tidak hanya belajar teori dan perintah dasar pemrograman R, Anda juga akan disuguhi dan dilatih dengan banyak praktek penulisan kode. Setiap bagian akan ada instruksi juga kuis yang harus Anda jawab dengan menuliskan kode yang sesuai.
- Googling. Anda juga bisa mendapatkan referensi melalui Google dengan mengetikkan kata kunci secara spesifik terkait permasalahan tentang R yang Anda hadapi.

Akhir kata, semoga buku yang ditulis secara amatir ini dapat membawa banyak manfaat bagi Anda.

Terima kasih sudah membaca buku ini sampai akhir.

dan..

SELAMAT BELAJAR

> “Anyone who stops learning is old, whether at twenty or eighty. Anyone who keeps learning stays young.”
>
> —- **Henry Ford**