



Modul 6

Data Visualization in Data Science using R

Pendahuluan

Selamat datang di course Data Visualization with ggplot2, sebuah library tambahan di R yang sangat populer digunakan untuk visualisasi – populer karena grafik dapat dihasilkan dengan struktur perintah yang sangat generik tapi jelas dan rapi.

Sepanjang course ini Anda akan mempelajari



- Apa dan kenapa diperlukan ggplot2
- **Plot:** Dasar "Kanvas" untuk Grafik
- Melengkapi Plot dengan Data dan Aesthetic Mapping
- Menggambar dengan Layer
- Scatter Plot dan Pengaturan Warna
- Visualisasi Trend dengan Grafik Line Chart
- Membuat Bar Chart
- Melihat Komposisi dengan Pie Chart
- Histogram untuk Menampilkan Distribusi Kepadatan Penduduk Jakarta
- Membagi Grafik dengan Faceting

Dan untuk mempelajari ggplot ini disarankan mengikuti pengantar dan prakteknya secara terurut, bab demi bab. Ini dikarenakan function-function penting akan dicilil sepanjang *course* sebagai strategi penyampaian materi – dengan tujuan agar Anda tidak terlalu terbebani dengan materi yang disampaikan selanjutnya.

Dengan dasar yang diperkenalkan di *course* ini, diharapkan Anda dapat memiliki kemampuan untuk memahami dan berkreaitivitas lebih lanjut dengan ggplot untuk menghasilkan berbagai grafik yang menarik.

Klik tombol Next untuk melanjutkan.

Apa itu ggplot2?

Mana pernyataan berikut yang tepat untuk menjelaskan apa itu ggplot2?

- ☒ Sebuah pustaka (library) yang terdapat di R
- ☐ Sebuah pustaka (library) di Microsoft Excel
- ☐ Sebuah set komponen untuk machine learning
- ☒ Sebuah set komponen untuk menggambar grafik
- ☐ Sebuah modul untuk machine learning

Menggunakan Package ggplot2

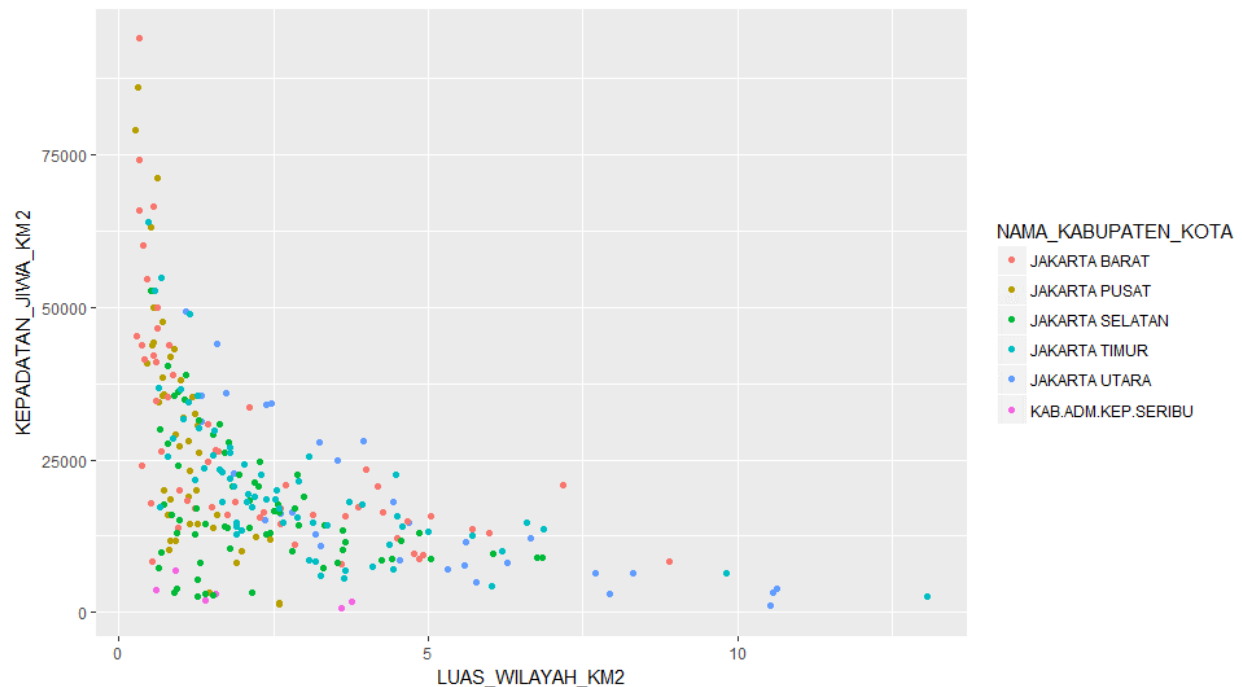
Berikut adalah perintah untuk menggunakan package ggplot2:

- ☒ `library(ggplot2)`
- ☐ `packages.install("ggplot2")`
- ☐ `install.packages("ggplot2")`
- ☐ `library(ggplot)`
- ☒ `library("ggplot2")`

Konsep Komponen di ggplot2

Kekuatan ggplot adalah memiliki definisi yang jelas untuk komponen-komponen visual sehingga dapat digunakan untuk membangun visualisasi yang lengkap.

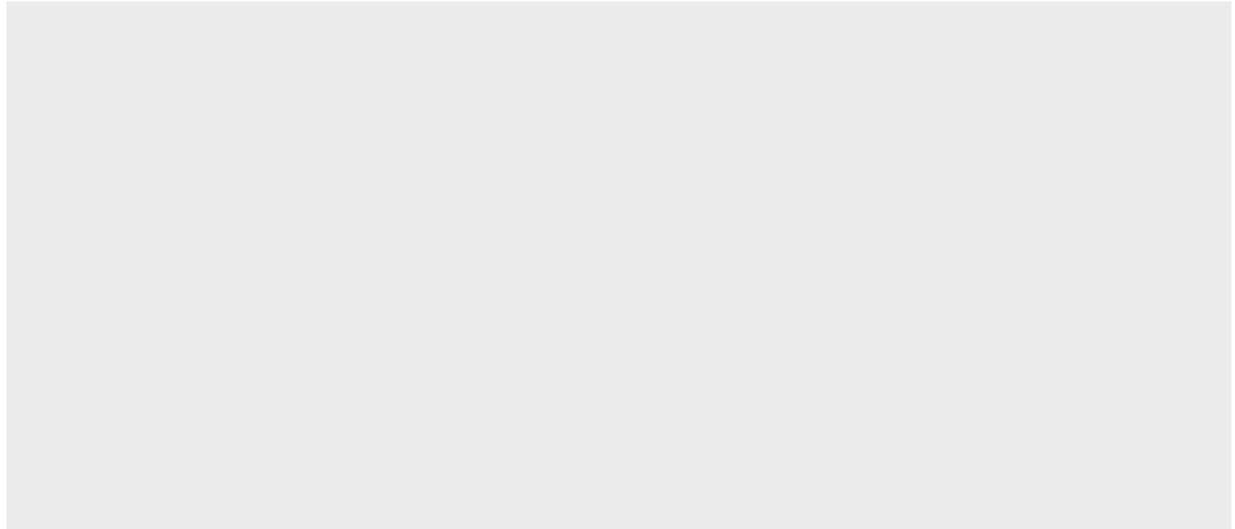
Sebagai contoh, cobalah kita lihat grafik di bawah ini – yang berisi informasi tingkat kepadatan penduduk dibandingkan dengan luas wilayah di propinsi DKI Jakarta.



Ada empat komponen ggplot2 pada grafik tersebut, yaitu **plot**, **data**, **aesthetic mapping** dan **layer**.

Plot

Paling awal adalah adanya plot atau "kanvas" kosong dimana berbagai grafik dapat "digambar" di atasnya.



Plot dapat dihasilkan dengan *function* bernama **ggplot**.

Catatan Penting: Bedakan antara ggplot2 sebagai package dan ggplot sebagai function disini.

Data dan Aesthetic Mapping

Sebelum digambar, kanvas ini tentunya perlu memiliki data yang akan ditampilkan. Berikut adalah data yang kita gunakan untuk contoh grafik di atas. Data ini adalah data kependudukan dari Pemerintah Provinsi DKI Jakarta tahun 2013 yang diambil dari situs data.go.id.

A	B	C	D	E	F	G
TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.76	1705
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	3.59	628
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. UNTUNG JAWA	0.59	3625
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. TIDUNG	1.57	3084
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. PARI	1.39	1968
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO UTARA	1.12	18987
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO SELATAN	1.14	14465
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	KEBON KELAPA	0.78	15890
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	DURI PULO	0.72	35628
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	PASAR BARU	1.89	8041
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARANG ANYAR	0.51	63122
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARTINI	0.55	49862
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	GUNUNG SAHARI UTARA	1.98	9933
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	MANGGA DUA SELATAN	1.29	26203
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	KEMAYORAN	0.55	44202
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	KEBON KOSONG	1.13	28014
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	HARAPAN MULIA	0.91	29205
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	SERDANG	0.82	41837
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	GUNUNG SAHARI SELATAN	0.53	43858
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	CEMPAKA BARU	0.99	38088
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	SUMUR BATU	1.15	23271
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	UTAN PANJANG	1.05	31889
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SENEN	SENEN	0.81	10158
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SENEN	KENARI	0.91	11757

Dan dari contoh grafik awal, terlihat tidak semua kolom itu ditampilkan. Yang diperlukan hanya dua kolom, yaitu **Luas Wilayah (KM2)** dan **Kepadatan (Jiwa / KM2)**.

	A	B	C	D	E	F	G	H
1	TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)	
2	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779	
3	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.76	1705	
4	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	3.59	628	
5	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. UNTUNG JAWA	0.59	3625	
6	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. TIDUNG	1.57	3084	
7	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. PARI	1.39	1968	
8	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350	
9	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584	
10	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO UTARA	1.12	18987	
11	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO SELATAN	1.14	14465	
12	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	KEBON KELAPA	0.78	15890	
13	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	DURI PULO	0.72	35628	
14	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	PASAR BARU	1.89	8041	
15	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARANG ANYAR	0.51	63122	
16	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARTINI	0.55	49862	
17	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	GUNUNG SAHARI UTARA	1.98	9933	
18	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	MANGGA DUA SELATAN	1.29	26203	
19	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	KEMAYORAN	0.55	44202	
20	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	KEBON KOSONG	1.13	28014	
21	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	HARAPAN MULIA	0.91	29205	
22	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	SERDANG	0.82	41837	
23	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	GUNUNG SAHARI SELATAN	0.53	43858	
24	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	CEMPAKA BARU	0.99	38088	

Kolom **Luas Wilayah (KM2)** akan mengisi absis (sumbu-x) dan **Kepadatan (Jiwa / KM2)** akan mengisi ordinat (sumbu-y). Dan kita gunakan kolom ketiga, yaitu Nama Kabupaten/Kota untuk pembeda warna (color).

Komponen-komponen **x**, **y** dan **color** disebut komponen grafik atau **aesthetic** di ggplot2. Dan penggunaan ketiga kolom data ini ke dalam **aesthetic** disebut **aesthetic mapping**.

	A	B	C	D	E	F	G
1	TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)
2	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779
3	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.7	1705
4	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	0.59	628
5	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. UNTUNG JAWA	0.59	3625
6	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. TIDUNG	1.57	3084
7	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PARI	1.39	1968
8	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350
9	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584
10	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR		1.12	987
11	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR		1.14	465
12	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR		0.78	890
13	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	DURI PULO	0.72	35628
14	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	PASAR BARU	1.89	8041
15	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARANG ANYAR	0.51	63122

Dengan adanya **plot**, **data**, dan **aesthetic mapping** ini menjadi bahan baku untuk melakukan step berikutnya: menggambar grafik itu sendiri dengan layer!

Layer

Layer adalah objek grafik itu sendiri. Sesuai namanya, objek-objek tersebut disusun dalam bentuk lapisan (layer).

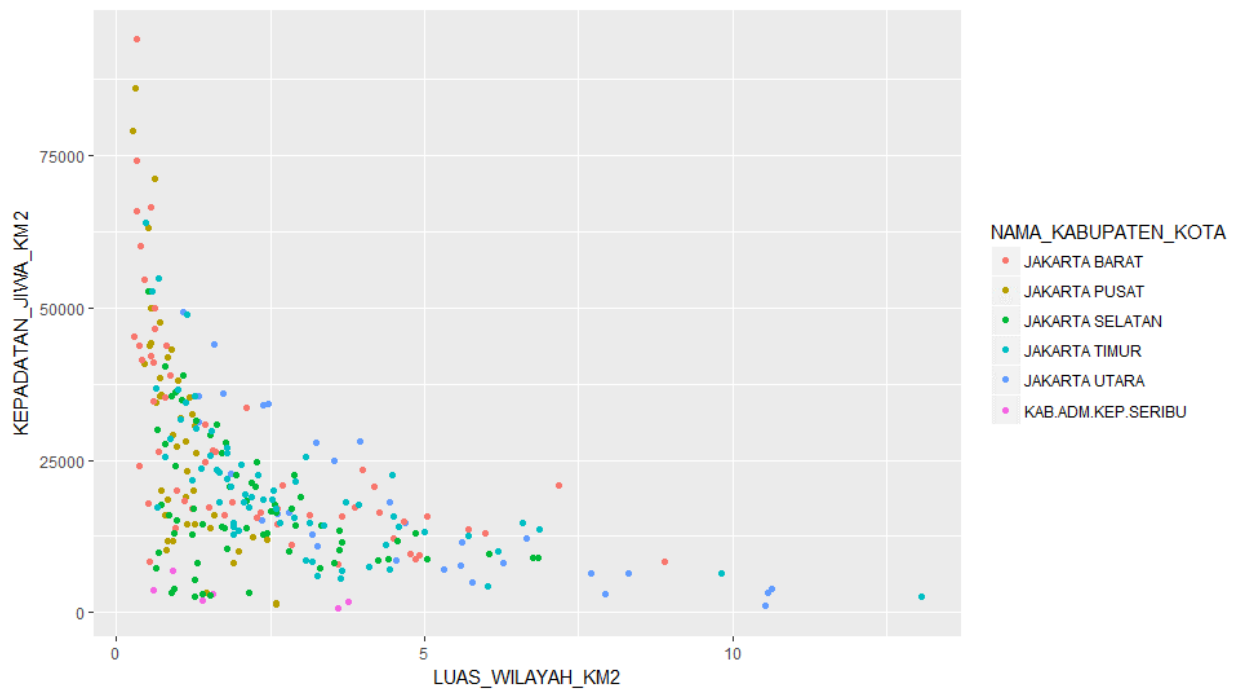
Setiap lapisan memiliki objek-objek berikut:

- **Geom**: Bentuk geometri seperti garis (line), batang (bar), titik (point), dan lain-lain.
- **Stat**: Atau suatu fungsi untuk melakukan transformasi statistik terhadap data input.

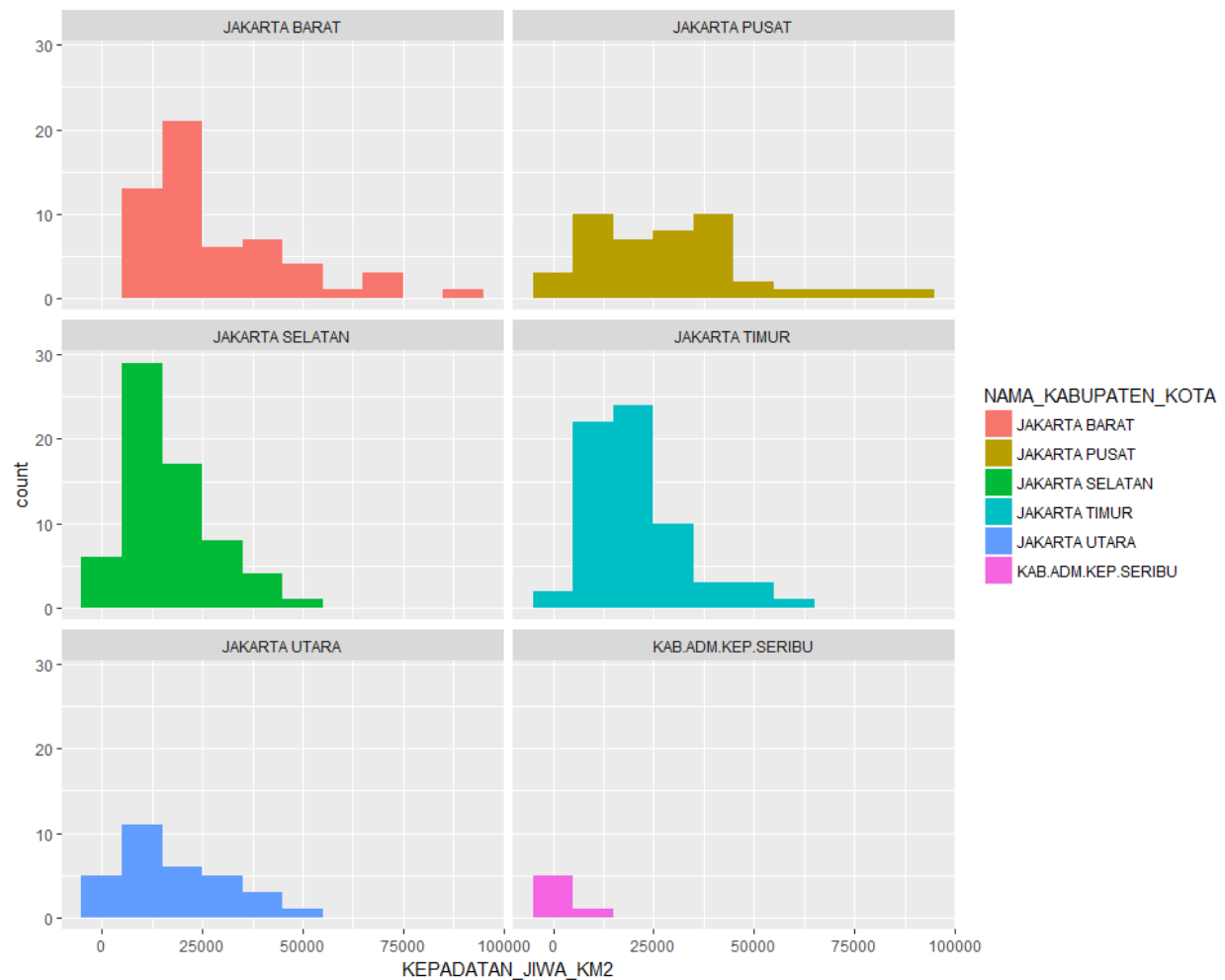
Contoh paling sederhana adalah transformasi data untuk kepadatan jiwa dari angka menjadi range atau interval per lima ribuan. Jadi data input dengan angka 8041 diubah menjadi interval angka 8001-8500. Transformasi ini disebut dengan bin.

- **Position**: Posisi dari beberapa data yang memiliki nilai yang sama. Jika diplot sebagai scatter plot misalnya, tentunya data-data tersebut akan menumpuk di satu titik. Apakah perlu ditambahkan nilai acak tertentu sehingga pas digambarkan, terlihat datanya lebih tersebar? Jika iya, maka ini namanya **jitter**. Jika kita tidak ingin mengubah apa-apa, ini dinamakan **identity**.

Nah, kembali ke contoh di atas. Jika layer ditambahkan ke plot objek menggunakan point maka bentuknya menjadi berikut.



Dengan objek plot yang sama, jika tambahkan dengan variasi layer lain – yaitu dengan `geom_histogram` – maka hasilnya adalah grafik berikut.



Untuk memperjelas, berikut adalah diagram summary untuk proses tampilan grafik dan komponen yang terlibat.

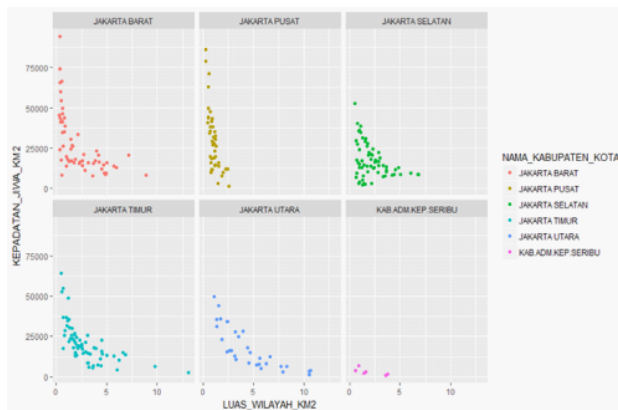
plot



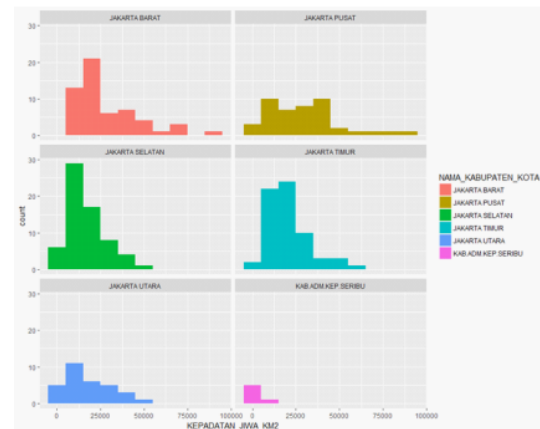
- data
- aesthetic mapping

+ layer

+ layer



- geom: point
- stat: identity
- position: identity



- geom: histogram
- stat: bin
- position: identity

Dengan mengerti akan fitur fundamental ini, kita dapat membuat banyak grafik yang lebih kompleks dengan ggplot dengan cepat.

Klik tombol Next untuk melanjutkan.

Komponen mana yang merupakan "kanvas" ggplot2?

Komponen mana yang berfungsi sebagai "kanvas" di ggplot2?

- ☐ Data
- ☒ Plot
- ☐ Aesthetic Mapping
- ☐ Color
- ☐ Layer dan tanda +

Bagaimana menentukan x dan y pada plot?

Dari data yang dijadikan input oleh ggplot2, komponen apa yang digunakan untuk memberikan nilai x dan y pada plot?

- ☐ Data
- ☐ Plot
- ☒ Aesthetic Mapping
- ☐ Color
- ☐ Layer dan tanda +

Bagaimana menghasilkan grafik?

Empat komponen minimal yang diperlukan untuk menghasilkan suatu grafik

- ☒ Aesthetic Mapping
- ☒ Data
- ☒ Layer dan tanda +
- ☒ Plot
- ☒ Color

Kesimpulan

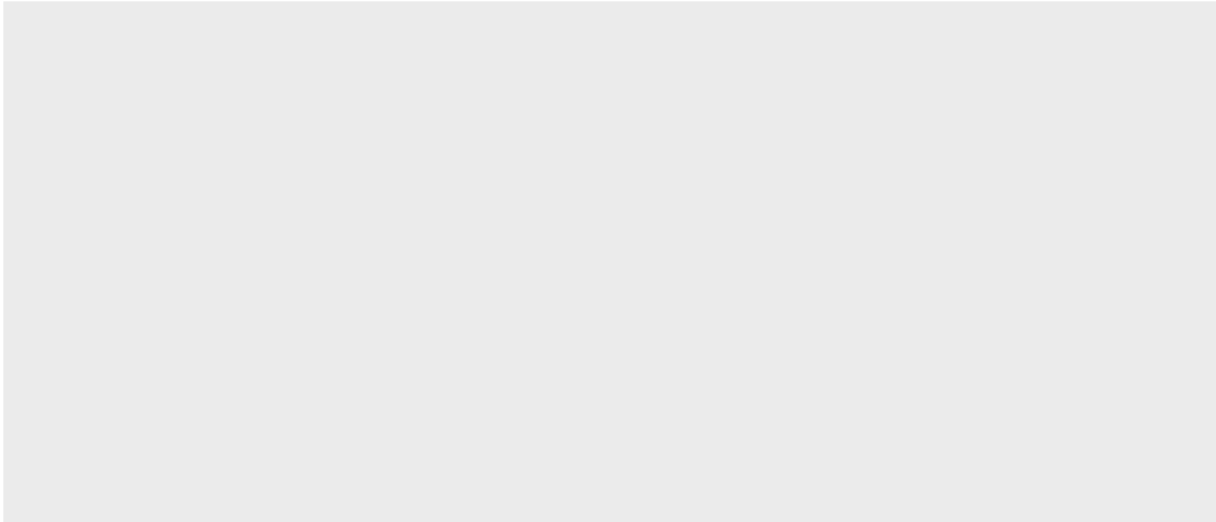
Selamat, Anda telah menyelesaikan bab penjelasan tentang ggplot2 dengan materi berikut:

- ggplot2 sebagai package untuk visualisasi data di R.
- pendekatan yang digunakan untuk ggplot2 adalah dengan suatu struktur tata bahasa (grammar) dimana komponen-komponen grafik dinamakan dengan detil dan jelas.
- komponen-komponen penting di ggplot2 adalah plot, data, aesthetic mapping dan layer – yang merupakan komponen minimal untuk menghasilkan suatu grafik.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Plot adalah komponen paling dasar di ggplot2, tanpa plot seluruh grafik tidak akan bisa ditampilkan. Dengan demikian, plot dapat dianalogikan sebagai "kanvas" gambar kita. Plot dibuat dengan function **ggplot**. Dan jika tidak digabungkan dengan komponen lain, maka tampilannya akan terlihat sebagai kotak persegi kosong dengan latar belakang abu-abu seperti gambar berikut.



Bab ini akan membahas bagaimana kita menghasilkan kanvas, menambahkan judul, label, merubah warna latar, menggunakan variable untuk objek plot, dan melihat detail dari objek plot dengan function *summary*.

Klik tombol Next untuk melanjutkan ke materi.

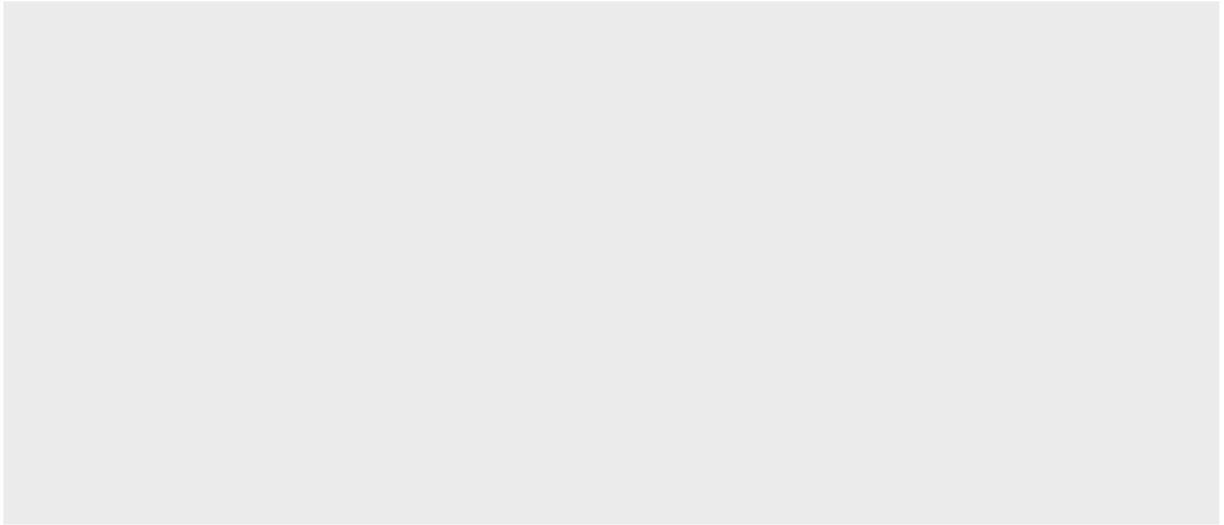
Function apa yang digunakan untuk menghasilkan Plot?

Function apa yang digunakan untuk menghasilkan plot?

- ☐ aes
- ☐ summary
- ☐ data
- ☐ ggplot2
- ☒ ggplot

Membuat Kanvas Kosong

Untuk membuat plot, kita akan gunakan function **ggplot**. Function ini akan menghasilkan gambar sebagai berikut.



Tugas Praktek

Gantilah bagian [...] pada code editor dengan function ggplot() dan jalankan.

Code Editor

```
library(ggplot2)
```

#Ketik function ggplot() di bawah ini

```
ggplot()
```

Console

Menambahkan Judul

Setiap orang akan sangat terbantu melihat suatu grafik apabila memiliki judul yang jelas. Ini dapat ditambahkan di ggplot dengan fungsi **labs(title = "....")**.

Contoh perintah (syntax) dari labs adalah sebagai berikut:

```
[objek plot] + labs(title="Judul")
```

Berikut adalah penjelasan lengkap elemen-elemen perintah di atas.

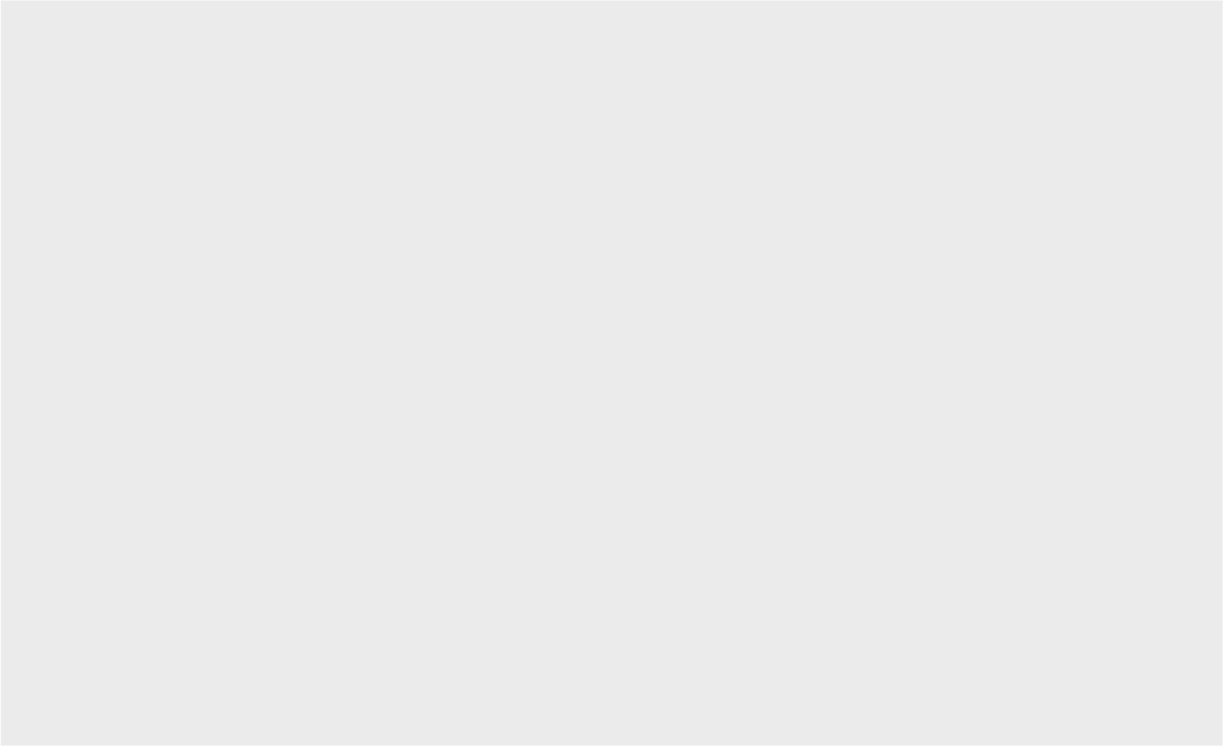
Komponen	Deskripsi
[objek plot]	Objek plot, untuk saat ini adalah objek yang dihasilkan oleh function ggplot()
+	Tanda plus, operator untuk menambahkan komponen lain ke dalam plot
labs	Function untuk menghasilkan komponen label text untuk ditambahkan ke dalam plot
title	Parameter judul, bagian dari function labs
" Judul"	Nilai teks untuk judul, ini bisa diganti sesuai keinginan kita

Tugas Praktek

Tambahkan plot – function ggplot() – yang ada di code editor dengan judul "Luas Wilayah vs Kepadatan Penduduk DKI Jakarta - Periode 2013".

Jika berhasil dijalankan, maka hasilnya akan terlihat sebagai berikut.

Luas Wilayah vs Kepadatan Penduduk DKI Jakarta



Petunjuk: Ganti bagian [...] dengan code yang sesuai, dan ingat huruf besar dan kecil sangat berpengaruh.

Code Editor

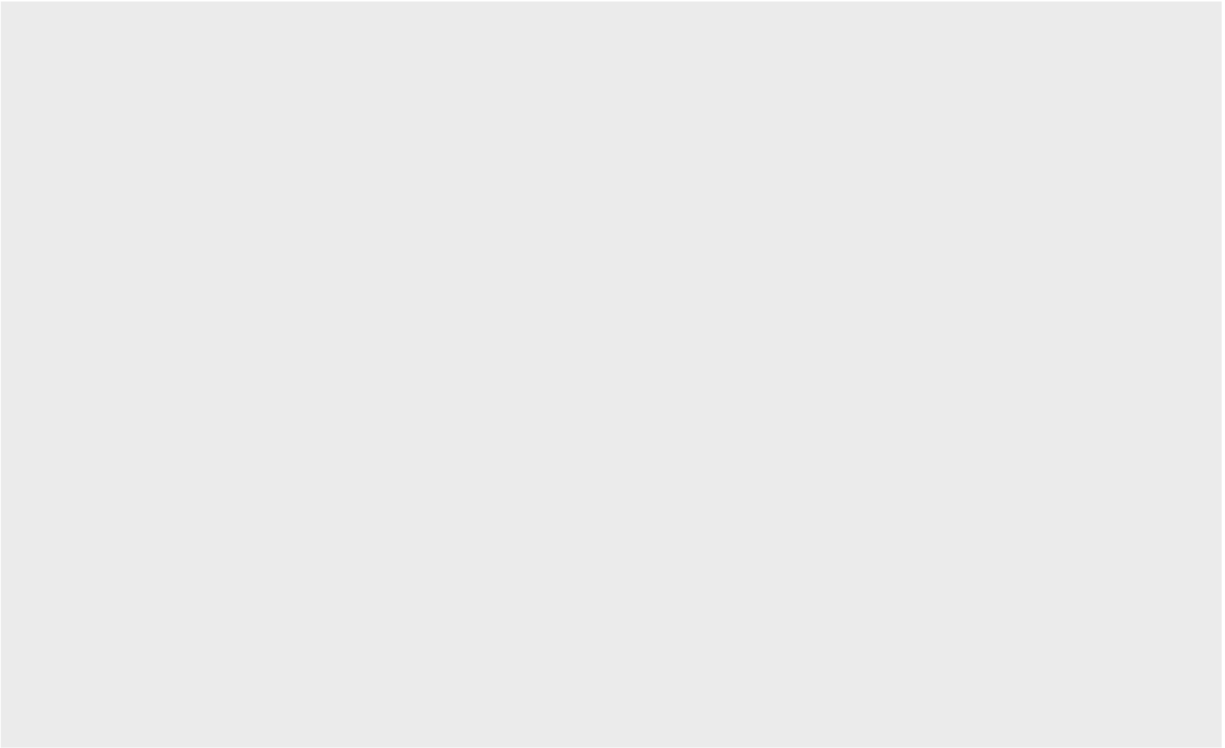
```
library(ggplot2)
```

```
#Penambahan judul dengan menggunakan fungsi labs
```

```
ggplot() + labs(title="Luas Wilayah vs Kepadatan Penduduk DKI Jakarta-Periode 2013")
```

Console

Luas Wilayah vs Kepadatan Penduduk DKI Jakarta



Plot disimpan sebagai Variable

Komponen plot dan layer bisa disimpan saja dalam satu variable. Berikut adalah contoh perintah dimana kita memiliki satu variable bernama **plot.jakarta** yang menyimpan plot dan judul. Kemudian menampilkan grafik dengan hanya menuliskan nama variable tersebut.

```
plot.jakarta <- ggplot()
plot.jakarta <- plot.jakarta + labs(title="Luas Wilayah vs Kepadatan Penduduk DKI Jakarta - Periode 2013")
plot.jakarta
```

Tugas Praktek

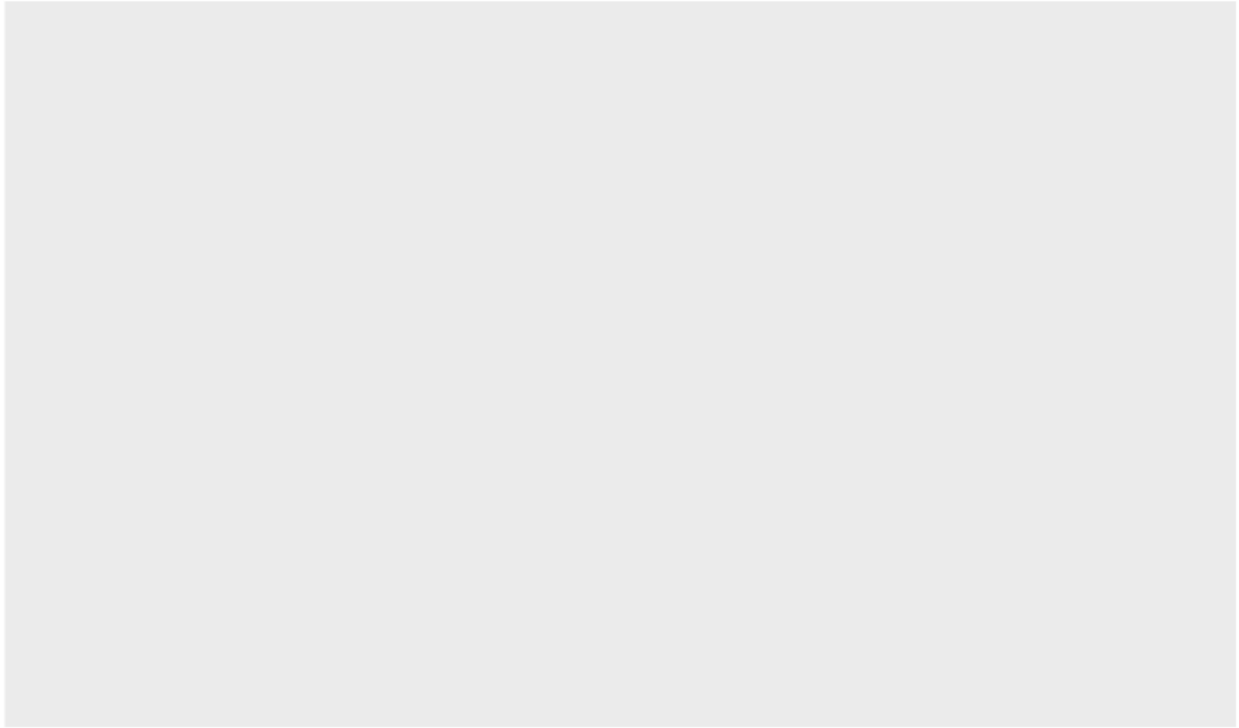
Pada code editor telah terdapat code lengkap untuk menghasilkan suatu plot. Rubahlah code tersebut sehingga plot ini disimpan ke variable bernama **plot.jakarta** sesuai contoh di atas, termasuk untuk menampilkan grafik.

Code Editor

```
library(ggplot2)
plot.jakarta <- ggplot()
plot.jakarta <- plot.jakarta + labs(title="Luas Wilayah vs Kepadatan Penduduk DKI Jakarta - Periode 2013")
plot.jakarta
```

Console

Luas Wilayah vs Kepadatan Penduduk DKI Jakarta - Periode 2013



Menambahkan Label pada Sumbu X dan Y

Function labs yang sudah diperkenalkan pada tiga praktek sebelumnya bukan hanya bisa diisi dengan judul dan subjudul, tapi bisa juga untuk mengisi teks pada sumbu x dan y.

Berikut adalah konstruksi perintahnya:

`[objek plot] + labs(x="Label X", y = "Label Y")`

Berikut adalah penjelasan lengkap elemen-elemen perintah di atas.

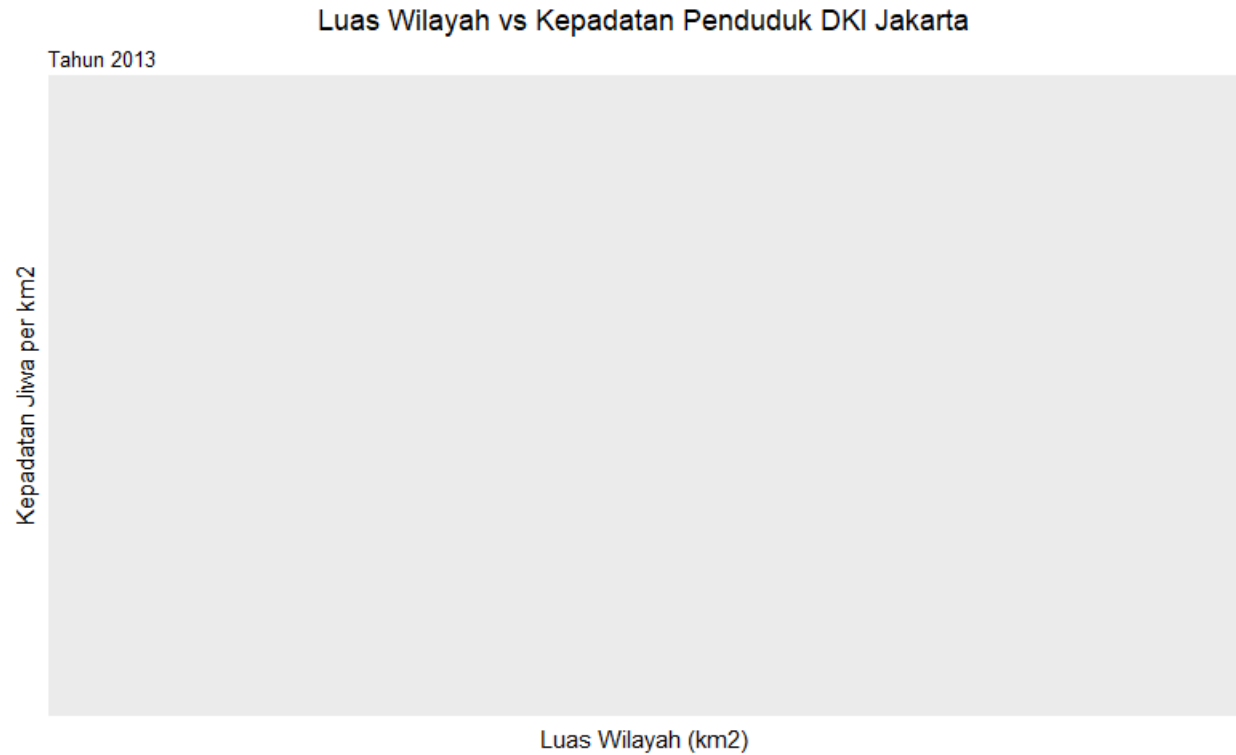
Komponen	Deskripsi
[objek plot]	Objek plot, untuk saat ini adalah objek yang dihasilkan oleh function ggplot()
+	Tanda plus, operator untuk menambahkan komponen lain ke dalam plot
labs	Function untuk menghasilkan komponen label text untuk ditambahkan ke dalam plot
x	Parameter untuk sumbu x, bagian dari function labs
"Label X"	Nilai teks untuk x, ini bisa diganti sesuai keinginan kita
subtitle	Parameter untuk sub judul, bagian dari function labs
"Label Y"	Nilai teks untuk y, ini bisa diganti sesuai keinginan kita

Catatan: Function labs ini bisa digunakan berkali-kali pada plot. Ini artinya kita bisa menambahkan x dan y pada function labs sebelumnya, atau menambahkan function labs baru untuk mengisi label x dan y.

Tugas Praktek

Rubah bagian [...] pada code editor untuk menambahkan pada variable plot.jakarta laebl untuk sumbu x dengan teks " Luas Wilayah (km2)" dan sumbu y dengan teks " Kepadatan Jiwa per km2".

Jika berhasil dijalankan, maka hasilnya akan terlihat sebagai berikut.



Code Editor

```
library(ggplot2)
plot.jakarta <- ggplot()
plot.jakarta <- plot.jakarta + labs(title="Luas Wilayah vs Kepadatan Penduduk DKI
Jakarta - Periode 2013", subtitle="Tahun 2013")
plot.jakarta <- plot.jakarta + labs(x="Luas Wilayah (km2)", y="Kepadatan Jiwa per km2")
plot.jakarta
```

Console

Luas Wilayah vs Kepadatan Penduduk DKI Jakarta - Periode 2013



Fungsi summary untuk objek ggplot

Tidak setiap saat kita harus menganalisa plot dan komponennya dengan cara ditampilkan. Ketika sudah disimpan di variable, kita bisa melihat detilnya dalam bentuk tekstual dengan menggunakan fungsi **summary**.

Mari kita langsung praktekkan saja melalui tugas berikut.

Tugas Praktek

Perhatikan seluruh perintah yang ada pada code editor, ini merupakan kumpulan code yang telah kita pelajari pada bab ini.

Pada baris terakhir terdapat penggunaan fungsi summary dengan input variable **plot.jakarta**. Ini untuk menampilkan detil apa saja yang terdapat di dalam sebuah plot.

Jika berhasil dijalankan, maka hasilnya akan terlihat sebagai berikut. Untuk saat ini, tidak ada yang perlu dijelaskan kecuali mengenal fungsi ini sebagai alat untuk menganalisa objek-objek plot kita. Kita akan menggunakan function ini lebih detil di bab berikutnya.

```
data: [x]
faceting: facet_null()
```

Code Editor

```
library(ggplot2)

plot.jakarta <- ggplot()

plot.jakarta <- plot.jakarta + labs(title="Luas Wilayah vs Kepadatan Penduduk DKI Jakarta")

plot.jakarta <- plot.jakarta + labs(x = "Luas Wilayah (km2)", y="Kepadatan Jiwa per km2")

summary(plot.jakarta)
```

Console

```
> library(ggplot2)

> plot.jakarta <- ggplot()

> plot.jakarta <- plot.jakarta + labs(title="Luas Wilayah vs Kepadatan Penduduk DKI J
akarta")

> plot.jakarta <- plot.jakarta + labs(x = "Luas Wilayah (km2)", y="Kepadatan Jiwa per
km2")

> summary(plot.jakarta)
data: [x]
faceting: facet_null()
```

Kesimpulan

Plot adalah komponen paling dasar di ggplot2, tanpa plot seluruh grafik tidak akan bisa ditampilkan. Dengan demikian, plot adalah "kanvas" grafik. Untuk membuat plot kita gunakan fungsi bernama **ggplot()**

Kita dapat menambahkan komponen grafik lain di atas plot. Dan pada bab ini kita telah menambahkan:

- Teks judul dengan fungsi **labs(title="...")**.
- Teks pada sumbu x dan y dengan fungsi **labs(x="...", y="...")**.

Selain itu juga ditunjukkan bagaimana sebaiknya objek plot disimpan ke variable. Dengan cara ini, kita bisa mengolahnya dengan lebih rapi. Terakhir, kita menampilkan informasi plot ini dengan menggunakan fungsi **summary**.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Plot bisa menerima input data, dan dari data tersebut bisa dipilih beberapa kolom yang akan "dipetakan" sebagai input untuk komponen visual.

Untuk jelasnya, mari kita lihat data yang dibuka dengan aplikasi Excel pada gambar berikut. Data ini adalah data luas wilayah dan kependudukan pemerintah provinsi DKI Jakarta.

	A	B	C	D	E	F	G
1	TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)
2	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779
3	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.73	1705
4	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	0.59	628
5	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. UNTUNG JAWA	0.59	3625
6	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. TIDUNG	1.57	3084
7	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PARI	1.39	1968
8	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350
9	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584
10	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR		1.12	987
11	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR		1.14	465
12	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR		0.78	890
13	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	DURI PULO	0.72	35628
14	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	PASAR BARU	1.89	8041
15	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARANG ANYAR	0.51	63122

Pada gambar terlihat banyak kolom informasi seperti TAHUN, NAMA PROVINSI, NAMA KABUPATEN/KOTA, dll.

Ada tiga kolom yang di-highlight dengan warna kuning, hijau dan biru muda. Ketiga kolom ini masing-masing akan dipetakan di dalam grafik untuk komponen visual x, y dan color (warna). Pemetaan data ini disebut dengan **aesthetic mapping**.

Pada praktek selanjutnya, kita akan membaca data dan melakukan pemetaan. Input data pada ggplot2 harus bertipe **data.frame**, tidak boleh yang lain.

Klik tombol Next untuk melanjutkan materi.

Data pada ggplot2

Data tipe apa yang bisa diberikan sebagai input untuk plot?

- ☐ Matrix
- ☐ List
- ☐ Vector
- ☐ Semuanya Benar
- ☒ Data Frame

Aesthetic Mapping

Dari text penjelasan pada Pendahuluan, manakah yang merupakan variable aesthetic mapping?

- ☐ plot
- ☒ x
- ☐ data.frame
- ☒ y
- ☒ color

Dataset Kepadatan Penduduk Jakarta

Dataset yang akan kita gunakan sepanjang praktek course ini adalah data wilayah dan kepadatan kependudukan pemerintah provinsi DKI Jakarta tahun 2013. Sumber dataset ini dapat didownload dari data.go.id, dengan url lengkap berikut:

<https://data.go.id/dataset/kepadatan-dan-luas-wilayah-per-kelurahan-dki-jakarta>

Data kependudukan dipilih karena hampir seluruh organisasi, termasuk bisnis dan institusi pemerintah memerlukan ini untuk mengambil keputusan strategik. Sebagai contoh, dimana saya perlu membuka toko atau kantor cabang? Produk apa yang bakal laku di area tersebut?

Nah, atas dasar itulah data kependudukan dipilih untuk *course* visualisasi data kali ini.

Berikut adalah tampilan data kependudukan DKI Jakarta tersebut jika dibuka di aplikasi Notepad. Perhatikan tanda pemisah antar kolom menggunakan koma.

```

dkikepadatankelurahan2013.csv - Notepad
File Edit Format View Help
TAHUN,NAMA PROVINSI,NAMA KABUPATEN/KOTA,NAMA KECAMATAN,NAMA KELURAHAN,LUAS WILAYAH (KM2),KEPADATAN
2013,PROVINSI DKI JAKARTA,KAB.ADM.KEP.SERIBU,KEP. SERIBU UTR,P. PANGGANG, 0.91 ,6779,,,,,,,,,23
2013,PROVINSI DKI JAKARTA,KAB.ADM.KEP.SERIBU,KEP. SERIBU UTR,P. KELAPA, 3.76 ,1705,,,,,,,,,84,8
2013,PROVINSI DKI JAKARTA,KAB.ADM.KEP.SERIBU,KEP. SERIBU UTR,P. HARAPAN, 3.59 ,628,,,,,,,,,255,
2013,PROVINSI DKI JAKARTA,KAB.ADM.KEP.SERIBU,KEP. SERIBU SLT,P. UNTUNG JAWA, 0.59 ,3625,,,,,,,,,
2013,PROVINSI DKI JAKARTA,KAB.ADM.KEP.SERIBU,KEP. SERIBU SLT,P. TIDUNG, 1.57 ,3084,,,,,,,,,98,7
2013,PROVINSI DKI JAKARTA,KAB.ADM.KEP.SERIBU,KEP. SERIBU SLT,P. PARI, 1.39 ,1968,,,,,,,,,113,11
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,GAMBIR,GAMBIR, 2.58 ,1350,,,,,,,,,166,174,130,165,176,1
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,GAMBIR,CIDENG, 1.26 ,14584,,,,,,,,,850,748,749,798,779,
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,GAMBIR,PETOJO UTARA, 1.12 ,18987,,,,,,,,,954,920,914,94
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,GAMBIR,PETOJO SELATAN, 1.14 ,14465,,,,,,,,,752,675,691,
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,GAMBIR,KEBON KELAPA, 0.78 ,15890,,,,,,,,,592,491,447,52
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,GAMBIR,DURI PULO, 0.72 ,35628,,,,,,,,,1213,1106,1105,10
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,SAWAH BESAR,PASAR BARU, 1.89 ,8041,,,,,,,,,714,611,595,
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,SAWAH BESAR,KARANG ANYAR, 0.51 ,63122,,,,,,,,,1575,1485
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,SAWAH BESAR,KARTINI, 0.55 ,49862,,,,,,,,,1307,1177,1172
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,SAWAH BESAR,GUNUNG SAHARI UTARA, 1.98 ,9933,,,,,,,,,817
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,SAWAH BESAR,MANGGA DUA SELATAN, 1.29 ,26203,,,,,,,,,168
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,KEMAYORAN,KEMAYORAN, 0.55 ,44202,,,,,,,,,1164,1063,1013
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,KEMAYORAN,KEBON KOSONG, 1.13 ,28014,,,,,,,,,1644,1542,1
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,KEMAYORAN,HARAPAN MULIA, 0.91 ,29205,,,,,,,,,1256,1213,
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,KEMAYORAN,SERDANG, 0.82 ,41837,,,,,,,,,1603,1559,1605,1
2013,PROVINSI DKI JAKARTA,JAKARTA PUSAT,KEMAYORAN,GUNUNG SAHARI SELATAN, 0.53 ,43858,,,,,,,,,10

```

Dan ini adalah tampilan jika data tersebut dibuka di aplikasi Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1	TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)													35-39 Laki-Laki	35-
2	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779														231
3	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.76	1705														84
4	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	3.59	628														255
5	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. UNTUNG JAWA	0.59	3625														199
6	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. TIDUNG	1.57	3084														98
7	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. PARI	1.39	1968														113
8	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350														166
9	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584														850
10	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO UTARA	1.12	18987														954
11	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO SELATAN	1.14	14465														752
12	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	KEBON KELAPA	0.78	15890														592
13	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	DURI PULO	0.72	35628														1213
14	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	PASAR BARU	1.89	8041														714
15	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARANG ANYAR	0.51	63122														1575
16	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARTINI	0.55	49862														1307
17	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	GUNUNG SAHARI UTARA	1.98	9933														817
18	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	MANGGA DUA SELATAN	1.29	26203														1683
19	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	KEMAYORAN	0.55	44202														1164
20	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	KEBON KOSONG	1.13	28014														1644
21	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	HARAPAN MULIA	0.91	29205														1256
22	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	SERDANG	0.82	41837														1603
23	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	GUNUNG SAHARI SELATAN	0.53	43858														1071
24	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	CEMPAKA BARU	0.99	38088														1750
25	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	SUMUR BATU	1.15	23271														1452
26	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	KEMAYORAN	UTAN PANJANG	1.05	31889														1610
27	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SENIEN	SENIEN	0.81	10158														208

Data tersebut memiliki 25 variable kolom dengan penjelasan sebagai berikut:

1. **TAHUN:** Tahun
2. **NAMA PROVINSI:** Nama provinsi di DKI Jakarta, dan nilainya hanya ada satu
3. **NAMA KABUPATEN/KOTA:** Nama kabupaten/kota di DKI Jakarta
4. **NAMA KECAMATAN:** Nama kecamatan di DKI Jakarta
5. **NAMA KELURAHAN:** Nama kelurahan di DKI Jakarta
6. **LUAS WILAYAH (KM2):** Luas wilayah (km persegi)
7. **KEPADATAN (JIWA/KM2):** Kepadatan penduduk (jiwa/km2)
8. **35-39 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 35-39 tahun
9. **35-39 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 35-39 tahun
10. **40-44 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 40-44 tahun
11. **40-44 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 40-44 tahun
12. **45-49 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 45-49 tahun
13. **45-49 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 45-49 tahun
14. **50-54 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 50-54 tahun
15. **50-54 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 50-54 tahun
16. **55-59 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 55-59 tahun
17. **55-59 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 55-59 tahun
18. **60-64 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 60-64 tahun
19. **60-64 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 60-64 tahun
20. **65-69 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 65-69 tahun
21. **65-69 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 65-69 tahun
22. **70-74 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur 70-74 tahun
23. **70-74 Perempuan:** Jumlah penduduk perempuan dengan rentang umur 70-74 tahun
24. **>75 Laki-Laki:** Jumlah penduduk laki-laki dengan rentang umur di atas 75 tahun
25. **>75 Perempuan:** Jumlah penduduk perempuan dengan rentang umur di atas 75 tahun

Perhatikan seluruh kolom berisi informasi berupa angka, kecuali empat kolom berikut: **NAMA PROVINSI**, **NAMA KABUPATEN/KOTA**, **NAMA KECAMATAN**, dan **NAMA KELURAHAN**. Dataset ini memiliki detail sampai ke tingkat kelurahan. Contoh yang cukup bagus untuk kita olah.

Klik tombol Next untuk melanjutkan ke praktek berikutnya.

Membaca Dataset Kependudukan dengan read.csv

Langkah pertama yang perlu kita lakukan adalah membaca dataset kependudukan tersebut dari file teks menjadi data.frame di R dengan perintah **read.csv**.

Mari kita praktekkan penggunaan read.csv ini dengan mengetikkan perintah berikut pada code editor.

```
penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

Berikut adalah penjelasan dari perintah di atas

Komponen	Deskripsi
penduduk.dki	nama variable yang digunakan untuk menampung data dari contoh dataset
read.csv	function yang digunakan untuk membaca contoh dataset yang berupa file
https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv	lokasi dataset yang terdapat di web DQLab. Jika lokasi file dan aplikasi R terdapat di komputer lokal Anda, maka gantilah dengan lokasi file di lokal. Misalkan c:\data\customer_segment s.txt
sep=","	Parameter pemisah (separator) antar kolom data. Kita gunakan tanda koma untuk dataset penduduk DKI Jakarta.

Jika terjadi error berikut, cobalah periksa kembali penulisan code – huruf besar, huruf kecil dan juga penulisan lokasi file – dengan teliti.

Error in file(file, "rt") : cannot open the connection

Jika tidak terjadi error maka langkah selanjutnya adalah menampilkan isi data dengan mengetikkan nama variable.

```
penduduk.dki
```

Hasil eksekusi perintah ini sebagian akan tampak sebagai berikut.

	TAHUN	NAMA.PROVINSI	NAMA.KABUPATEN.KOTA	NAMA.KECAMATAN	NAMA.KELURAHAN	LUAS.WILAYAH..KM2.	KEPADATAN..JIWA.KM2.
1	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779
2	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.76	1705
3	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	3.59	628
4	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. UNTUNG JAWA	0.59	3625
5	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. TIDUNG	1.57	3084
6	2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU SLT	P. PARI	1.39	1968
7	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350
8	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584
9	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO UTARA	1.12	18987
10	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	PETOJO SELATAN	1.14	14465
11	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	KEBON KELAPA	0.78	15890
12	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	DURI PULO	0.72	35628
13	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	PASAR BARU	1.89	8041
14	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARANG ANYAR	0.51	63122
15	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	KARTINI	0.55	49862
16	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	GUNUNG SAHARI UTARA	1.98	9933
17	2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	SAWAH BESAR	MANGGA DIA SELATAN	1.29	26203

Terlihat isi data dari tujuh kolom pertama dan terdapat nomor baris pada tiap data yang ditampilkan.

Perhatikan jika nama kolom asal terdapat karakter yang bukan angka atau huruf, akan diubah menjadi tanda titik setelah dibaca dengan read.csv.

Sebagai contoh, "NAMA PROVINSI" diubah menjadi "NAMA.PROVINSI", "LUAS WILAYAH (KM2)" diubah menjadi "LUAS.WILAYAH..KM2".

Catatan: Walaupun nama kolom ini terlihat tidak standar dan tidak menyenangkan, kita akan biarkan nama kolom tetap seperti ini tanpa melakukan proses *cleansing*. Tetapi pada saat plotting, seperti telah ditunjukkan pada bab sebelumnya, kita bisa memberi label text sesuai keinginan kita. Disitulah kesempatan kita untuk merapikan tampilan teks dari kolom.

Jika kita hanya ingin menampilkan beberapa kolom tertentu, misalkan untuk kolom "NAMA.KECAMATAN" dan "NAMA.KELURAHAN" maka perintahnya adalah sebagai berikut.

```
penduduk.dki[c("NAMA.KECAMATAN", "NAMA.KELURAHAN")]
```

maka akan muncul hasil sebanyak 267 baris data, dimana sebagian datanya terlihat sebagai berikut:

	NAMA.KECAMATAN	NAMA.KELURAHAN
1	KEP. SERIBU UTR	P. PANGGANG

2	KEP. SERIBU UTR	P. KELAPA
3	KEP. SERIBU UTR	P. HARAPAN
4	KEP. SERIBU SLT	P. UNTUNG JAWA
5	KEP. SERIBU SLT	P. TIDUNG
6	KEP. SERIBU SLT	P. PARI
7	GAMBIR	GAMBIR
8	GAMBIR	CIDENG
9	GAMBIR	PETOJO UTARA
10	GAMBIR	PETOJO SELATAN
11	GAMBIR	KEBON KELAPA
12	GAMBIR	DURI PULO
13	SAWAH BESAR	PASAR BARU
14	SAWAH BESAR	KARANG ANYAR
15	SAWAH BESAR	KARTINI
16	SAWAH BESAR	GUNUNG SAHARI UTARA
17	SAWAH BESAR	MANGGA DUA SELATAN
18	KEMAYORAN	KEMAYORAN
19	KEMAYORAN	KEBON KOSONG
...		

Tugas Praktek

Lengkapi bagian [...] pada code editor untuk menampilkan data frame dengan kolom "NAMA.KELURAHAN" dan "LUAS.WILAYAH..KM2.".

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <-
```

```
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

```
# Tampilkan data frame dari kolom " NAMA.KELURAHAN " dan  
"LUAS.WILAYAH..KM2."
```

```
penduduk.dki[c("NAMA.KELURAHAN", "LUAS.WILAYAH..KM2.")]
```

Console

```

> library(ggplot2)

> #Membaca data csv dan dimasukkan ke variable penduduk.dki
> penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

> # Tampilkan data frame dari kolom " NAMA.KELURAHAN " dan "LUAS.WILAYAH..KM2."
> penduduk.dki[c("NAMA.KELURAHAN","LUAS.WILAYAH..KM2.")]
  NAMA.KELURAHAN LUAS.WILAYAH..KM2.
1      P. PANGGANG          0.91
2      P. KELAPA           3.76
3      P. HARAPAN          3.59
4    P. UNTUNG JAWA          0.59
5      P. TIDUNG           1.57
6      P. PARI             1.39
7      GAMBIR              2.58
8      CIDENG              1.26
9    PETOJO UTARA           1.12
10   PETOJO SELATAN         1.14
11   KEBON KELAPA           0.78
12   DURI PULO              0.72
13   PASAR BARU             1.89
14   KARANG ANYAR           0.51
15   KARTINI                0.55
16  GUNUNG SAHARI UTARA      1.98
17  MANGGA DUA SELATAN       1.29
18   KEMAYORAN              0.55
19   KEBON KOSONG           1.13
20   HARAPAN MULIA          0.91
21   SERDANG                0.82
22  GUNUNG SAHARI SELATAN     0.53
23   CEMPAKA BARU           0.99
24   SUMUR BATU             1.15
25   UTAN PANJANG           1.05
26   SENEN                  0.81
27   KENARI                 0.91
28   PASEBAN                0.71
29   KRAMAT                 0.71
30   KWITANG                0.45
31   BUNGUR                 0.64
32  CEMPAKA PUTIH TIMUR       2.22
33  CEMPAKA PUTIH BARAT       1.22
34   RAWASARI               1.25
35   MENTENG                2.44
36   PEGANGSAAN             0.98
37   CIKINI                 0.82
38   GONDANGDIA             1.46
39   KEBON SIRIH            0.83
40   GELORA                 2.59
41  BENDUNGAN HILIR          1.58

```

42	KARET TENGSIN	1.53
43	PETAMBURAN	0.90
44	KEBON MELATI	1.26
45	KEBON KACANG	0.71
46	KAMPUNG BALI	0.73
47	JOHAR BARU	1.19
48	KAMPUNG RAWA	0.30
49	GALUR	0.27
50	TANAH TINGGI	0.62
51	PENJARINGAN	3.95
52	KAMAL MUARA	10.53
53	KAPUK MUARA	10.56
54	PEJAGALAN	3.23
55	PLUIT	7.71
56	TANJUNG PRIOK	5.59
57	SUNTER JAYA	4.68
58	PAPANGGO	2.80
59	SUNGAI BAMBU	2.36
60	KEBON BAWANG	1.73
61	SUNTER AGUNG	6.65
62	WARAKAS	1.09
63	KOJA	3.27
64	TUGU UTARA	2.37
65	LAGOA	1.58
66	RAWA BADAK UTARA	1.33
67	TUGU SELATAN	1.86
68	RAWA BADAK SELATAN	1.33
69	CILINCING	8.31
70	SUKAPURA	5.61
71	MARUNDA	7.92
72	KALI BARU	2.47
73	SEMPER TIMUR	3.17
74	ROROTAN	10.64
75	SEMPER BARAT	4.44
76	PADEMANGAN TIMUR	2.61
77	PADEMANGAN BARAT	3.53
78	ANCOL	5.77
79	KELAPA GADING TIMUR	5.31
80	PEGANGSAAN DUA	6.28
81	KELAPA GADING BARAT	4.53
82	CENGKARENG BARAT	4.26
83	DURI KOSAMBI	5.03
84	RAWA BUAYA	4.67
85	KEDAUNG KALI ANGKE	2.61
86	KAPUK	7.18
87	CENGKARENG TIMUR	4.18
88	GROGOL	1.22
89	TANJUNG DUREN UTARA	1.11
90	TOMANG	1.88
91	JELAMBAR	1.44
92	TANJUNG DUREN SELATAN	1.76
93	JELAMBAR BARU	1.44
94	WIJAYA KUSUMA	2.61
95	TAMAN SARI	0.68
96	KRUKUT	0.55

97	MAPHAR	0.59
98	TANGKI	0.37
99	MANGGA BESAR	0.51
100	KEAGUNGAN	0.32
101	GLODOK	0.38
102	PINANGSIA	0.96
103	TAMBORA	0.28
104	KALI ANYAR	0.32
105	DURI UTARA	0.40
106	TANAH SEREAL	0.62
107	KERENDANG	0.32
108	JEMBATAN BESI	0.55
109	ANGKE	0.80
110	JEMBATAN LIMA	0.46
111	PEKOJAN	0.78
112	ROA MALAKA	0.53
113	DURI SELATAN	0.42
114	KEBON JERUK	2.69
115	SUKABUMI UTARA	1.60
116	SUKABUMI SELATAN	1.57
117	KELAPA DUA	1.50
118	DURI KEPAS	3.86
119	KEDOYA UTARA	3.14
120	KEDOYA SELATAN	2.28
121	KALIDERES	5.72
122	SEMANAN	5.98
123	TEGAL ALUR	4.00
124	KAMAL	4.49
125	PEGADUNGAN	8.89
126	PALMERAH	2.11
127	SLIPI	0.97
128	KOTA BAMBU UTARA	0.63
129	JATI PULO	0.87
130	KEMANGGISAN	2.33
131	KOTA BAMBU SELATAN	0.61
132	KEMBANGAN UTARA	3.65
133	MERUYA UTARA	4.76
134	MERUYA SELATAN	2.85
135	SRENGSENG	4.92
136	JOGLO	4.86
137	KEMBANGAN SELATAN	3.60
138	TEBET TIMUR	1.39
139	TEBET BARAT	1.72
140	MENTENG DALAM	2.58
141	KEBON BARU	1.30
142	BUKIT DURI	1.08
143	MANGGARAI SELATAN	0.51
144	MANGGARAI	0.95
145	SETIA BUDI	0.94
146	KARET SEMANGGI	0.90
147	KARET KUNINGAN	1.79
148	KARET	0.94
149	MENTENG ATAS	0.90
150	PASAR MANGGIS	0.78
151	GUNTUR	0.65

152	KUNINGAN TIMUR	2.15
153	MAMPANG PRAPATAN	0.78
154	BANGKA	3.30
155	PELA MAMPANG	1.62
156	TEGAL PARANG	1.06
157	KUNINGAN BARAT	0.98
158	PASAR MINGGU	2.79
159	JATI PADANG	2.50
160	CILANDAK TIMUR	3.53
161	RAGUNAN	5.05
162	PEJATEN TIMUR	2.88
163	PEJATEN BARAT	2.90
164	KEBAGUSAN	2.26
165	KEBAYORAN LAMA UTARA	1.78
166	PONDOK PINANG	6.84
167	CIPULIR	1.94
168	GROGOL UTARA	3.33
169	GROGOL SELATAN	2.85
170	KEBAYORAN LAMA SELATAN	2.57
171	CILANDAK BARAT	6.05
172	LEBAK BULUS	4.41
173	PONDOK LABU	3.61
174	GANDARIA SELATAN	1.76
175	CIPETE SELATAN	2.37
176	MELAWAI	1.26
177	GUNUNG	1.32
178	KRAMAT PELTA	1.23
179	SELONG	1.40
180	RAWA BARAT	0.69
181	SENAYAN	1.53
182	PULO	1.27
183	PETOGOGAN	0.86
184	GANDARIA UTARA	1.52
185	CIPETE UTARA	1.83
186	PANCORAN	1.24
187	KALIBATA	2.20
188	RAWA JATI	0.67
189	DUREN TIGA	2.45
190	PENGADEGAN	0.95
191	CIKOKO	0.72
192	JAGAKARSA	4.85
193	SRENGSENG SAWAH	6.75
194	CIGANJUR	3.61
195	LENTENG AGUNG	2.28
196	TANJUNG BARAT	3.65
197	CIPEDAK	4.24
198	PESANGGRAHAN	2.10
199	BINTARO	4.56
200	PETUKANGAN UTARA	2.99
201	PETUKANGAN SELATAN	2.11
202	ULUJAMI	1.71
203	PISANGAN BARU	0.68
204	UTAN KAYU UTARA	1.05
205	KAYU MANIS	0.57
206	PAL MERIAM	0.65

207	KEBON MANGGIS	0.78
208	UTAN KAYU SELATAN	1.12
209	PULO GADUNG	1.29
210	PISANGAN TIMUR	1.80
211	CIPINANG	1.54
212	JATINEGARA KAUM	1.23
213	RAWAMANGUN	2.60
214	KAYU PUTIH	4.37
215	JATI	2.15
216	KAMPUNG MELAYU	0.48
217	BIDARA CINA	1.26
218	BALI MESTER	0.67
219	RAWA BUNGA	0.88
220	CIPINANG CEMPEDAK	1.67
221	CIPINANG MUARA	2.90
222	CIPINANG BESAR SELATAN	1.63
223	CIPINANG BESAR UTARA	1.15
224	KRAMAT JATI	1.52
225	KAMPUNG TENGAH	2.03
226	DUKUH	1.98
227	BATU AMPAR	2.55
228	BALE KAMBANG	1.67
229	CILILITAN	1.80
230	CAWANG	1.79
231	GEDONG	2.65
232	BARU	1.89
233	CIJANTUNG	2.37
234	KALISARI	2.89
235	PEKAYON	3.14
236	JATINEGARA	6.60
237	RAWA TERATE	4.10
238	PENGGILINGAN	4.48
239	CAKUNG TIMUR	9.81
240	PULO GEBANG	6.86
241	UJUNG MENTENG	4.43
242	CAKUNG BARAT	6.19
243	DUREN SAWIT	4.58
244	PONDOK BAMBU	5.00
245	KLENDER	3.08
246	PONDOK KELAPA	5.72
247	MALAKA SARI	1.38
248	MALAKA JAYA	0.99
249	PONDOK KOPI	2.06
250	MAKASAR	1.85
251	PINANG RANTI	1.89
252	KEBON PALA	2.30
253	HALIM PERDANA KUSUMAH	13.07
254	CIPINANG MELAYU	2.53
255	CIRACAS	3.93
256	CIBUBUR	4.50
257	KELAPA DUA WETAN	3.37
258	SUSUKAN	2.19
259	RAMBUTAN	2.09
260	CIPAYUNG	3.08
261	CILANGKAP	6.03

262	PONDOK RANGGON	3.66
263	MUNJUL	1.90
264	SETU	3.25
265	BAMBU APUS	3.17
266	LUBANG BUAYA	3.72
267	CEGER	3.63

Memasukkan Data ke Plot

Data dapat dimasukkan ke dalam plot melalui argumen di function **ggplot** dengan syntax berikut.

```
ggplot(data = ...)
```

Untuk contoh data kependudukan kita, maka perintah lengkapnya adalah sebagai berikut.

```
ggplot(data = penduduk.dki)
```

Tugas Praktek

Pada code editor, ganti bagian [...] dengan:

- variable **plot.dki** yang menyimpan plot dengan input data variable **penduduk.dki**.
- tampilkan summary dari **plot.dki**.

Jika berhasil dijalankan, maka Anda harusnya mendapatkan output summary sebagai berikut. Terlihat saat ini plot memiliki komponen data, yang tidak tampil ketika kita membuat plot pada bab sebelumnya.

```
data: TAHUN, NAMA.PROVINSI, NAMA.KABUPATEN.KOTA, NAMA.KECAMATAN,
      NAMA.KELURAHAN, LUAS.WILAYAH.KM2., KEPADATAN.JIWA.KM2., X, X.1,
      X.2, X.3, X.4, X.5, X.6, X.7, X.8, X.9, X.10, X.11, X35.39.Laki.Laki,
      X35.39.Perempuan, X40.44.Laki.Laki, X40.44.Perempuan,
      X45.49.Laki.Laki, X45.49.Perempuan, X50.54.Laki.Laki,
      X50.54.Perempuan, X55.59.Laki.Laki, X55.59.Perempuan,
      X60.64.Laki.Laki, X60.64.Perempuan, X65.69.Laki.Laki,
      X65.69.Perempuan, X70.74.Laki.Laki, X70.74.Perempuan, X.75.Laki.Laki,
      X.75..Perempuan [267x37]
faceting: facet_null()
```

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <-
```

```
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

Masukkan data ke dalam plot dan simpan sebagai variable plot.dki, dan tampilkan summary dari plot tersebut

```
plot.dki <- ggplot(data = penduduk.dki)
```

```
summary(plot.dki)
```

Console

```
> library(ggplot2)

> #Membaca data csv dan dimasukkan ke variable penduduk.dki
> penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

> # Masukkan data ke dalam plot dan simpan sebagai variable plot.dki, dan tampilkan summary dari plot tersebut
> plot.dki <- ggplot(data = penduduk.dki)

> summary(plot.dki)
data: TAHUN, NAMA.PROVINSI, NAMA.KABUPATEN.KOTA, NAMA.KECAMATAN,
      NAMA.KELURAHAN, LUAS.WILAYAH..KM2., KEPADATAN..JIWA.KM2., X, X.1,
      X.2, X.3, X.4, X.5, X.6, X.7, X.8, X.9, X.10, X.11, X35.39.Laki.Laki,
      X35.39.Perempuan, X40.44.Laki.Laki, X40.44.Perempuan,
      X45.49.Laki.Laki, X45.49.Perempuan, X50.54.Laki.Laki,
      X50.54.Perempuan, X55.59.Laki.Laki, X55.59.Perempuan,
      X60.64.Laki.Laki, X60.64.Perempuan, X65.69.Laki.Laki,
      X65.69.Perempuan, X70.74.Laki.Laki, X70.74.Perempuan, X.75.Laki.Laki,
      X.75..Perempuan [267x37]
faceting: facet_null()
```

Memetakan x, y dan color dengan function aes

Setelah data dimasukkan pada plot, saatnya kita memetakan kolom yang diperlukan pada data tersebut ke elemen visual. Pemetaan ini disebut dengan aesthetic mapping.

Untuk melakukan hal ini, kita gunakan function aes yang memiliki syntax berikut.

```
aes([nama_elemen_1] = [nama_kolom_atau_nilai_1], [nama_elemen_2] = [nama_kolom_atau_nilai_1] ...)
```

Berikut adalah penjelasan syntaxnya.

Komponen	Deskripsi
Aes	Function untuk memetakan elemen-elemen visual dengan nilai atau variable
[nama_elemen_1]	Nama elemen visual sebagai parameter pertama. Nama elemen ini bisa x, y dan color yang akan kita gunakan di tugas praktek.
[nama_kolom_atau_nilai_1]	Nama kolom yang terdapat pada data atau nilai konstanta. Sebagai contoh, untuk nama kolom bisa NAMA.KECAMATAN sedangkan untuk nilai konstanta bisa dimasukkan nilai teks maupun angka.
[nama_elemen_2]	Nama elemen visual sebagai parameter kedua. Nama elemen ini bisa x, y dan color yang akan kita gunakan di tugas praktek.
[nama_kolom_atau_nilai_2]	Nama kolom yang terdapat pada data atau nilai konstanta. Sebagai contoh, untuk nama kolom bisa NAMA.KELURAHAN sedangkan untuk nilai konstanta bisa dimasukkan nilai teks maupun angka.
...	Pasangan nama elemen dan variable/nilai lainnya.

Kembali ke awal kebutuhan kita, mari kita lihat gambar berikut. Terlihat kita ingin memetakan x, y dan color ke kolom NAMA KELURAHAN, LUAS WILAYAH (KM2) dan KEPADATAN (JIWA/KM2).

A	B	C	D	E	F	G
TAHUN	NAMA PROVINSI	NAMA KABUPATEN/KOTA	NAMA KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PANGGANG	0.91	6779
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. KELAPA	3.76	1705
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. HARAPAN	3.59	628
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. UNTUNG JAWA	0.59	3625
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. TIDUNG	1.57	3084
2013	PROVINSI DKI JAKARTA	KAB.ADM.KEP.SERIBU	KEP. SERIBU UTR	P. PARI	1.39	1968
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	GAMBIR	2.58	1350
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584
2013	PROVINSI DKI JAKARTA	JAKARTA PUSAT	GAMBIR	CIDENG	1.26	14584

Diagram showing mapping: 'Color' points to column C (NAMA KABUPATEN/KOTA), 'x' points to column F (LUAS WILAYAH (KM2)), and 'y' points to column G (KEPADATAN (JIWA/KM2)).

Tetapi kita lihat di bagian bawah gambar, nama-nama kolom yang akan dimapping mengalami sedikit perubahan ketika dibaca di R, dimana spasi dan non huruf maupun angka diubah menjadi titik. Sebagai contoh, NAMA KABUPATEN/KOTA diubah menjadi NAMA.KABUPATEN.KOTA.

Berikut adalah contoh penggunaan aes berdasarkan kebutuhan tersebut.

```
aes(x = LUAS.WILAYAH..KM2., y=KEPADATAN..JIWA.KM2., color=NAMA.KABUPATEN.KOTA)
```

Dan function aes ini harus dipasangkan dengan function ggplot yang telah memiliki data, sehingga konstruksi lengkap codenya adalah sebagai berikut.

```
ggplot(data=penduduk.dki, aes(x =
LUAS.WILAYAH..KM2., y=KEPADATAN..JIWA.KM2.,
color=NAMA.KABUPATEN.KOTA))
```

Untuk lebih jelasnya mari kita lakukan tugas praktek berikut.

Tugas Praktek

Pada code editor, ganti bagian [...] dengan aesthetic mapping dengan pemetaan berikut:

- x = LUAS.WILAYAH..KM2.,
- y=KEPADATAN..JIWA.KM2.
- color=NAMA.KABUPATEN.KOTA

Jika berhasil dijalankan, maka Anda harusnya mendapatkan output summary sebagai berikut. Terlihat bagian data dan mapping sudah terisi sesuai keinginan kita.

```
data: TAHUN, NAMA.PROVINSI, NAMA.KABUPATEN.KOTA, NAMA.KECAMATAN, NAMA.KELURAHAN,
      LUAS.WILAYAH..KM2., KEPADATAN..JIWA.KM2., X, X.1, X.2, X.3, X.4, X.5, X.6, X.7, X.8,
      ,
```

```
X.9, X.10, X.11, X35.39.Laki.Laki, X35.39.Perempuan, X40.44.Laki.Laki,
X40.44.Perempuan, X45.49.Laki.Laki, X45.49.Perempuan, X50.54.Laki.Laki,
X50.54.Perempuan, X55.59.Laki.Laki, X55.59.Perempuan, X60.64.Laki.Laki,
X60.64.Perempuan, X65.69.Laki.Laki, X65.69.Perempuan, X70.74.Laki.Laki,
X70.74.Perempuan, X.75.Laki.Laki, X.75..Perempuan [267x37]
mapping: x = LUAS.WILAYAH..KM2., y = KEPADATAN..JIWA.KM2., colour = NAMA.KABUPATEN.K
OTA
faceting: facet_null()
```

Code Editor

```
library(ggplot2)

#Membaca data csv dan dimasukkan ke variable penduduk.dki
penduduk.dki <-
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

aes(x = LUAS.WILAYAH..KM2.,
y=KEPADATAN..JIWA.KM2.,color=NAMA.KABUPATEN.KOTA)

summary(plot.dki)
```

Console

```
> library(ggplot2)

> #Membaca data csv dan dimasukkan ke variable penduduk.dki
> penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

> aes(x = LUAS.WILAYAH..KM2., y=KEPADATAN..JIWA.KM2.,color=NAMA.KABUPATEN.KOTA)
* x      -> LUAS.WILAYAH..KM2.
* y      -> KEPADATAN..JIWA.KM2.
* colour -> NAMA.KABUPATEN.KOTA

> summary(plot.dki)
data: TAHUN, NAMA.PROVINSI, NAMA.KABUPATEN.KOTA, NAMA.KECAMATAN,
      NAMA.KELURAHAN, LUAS.WILAYAH..KM2., KEPADATAN..JIWA.KM2., X, X.1,
      X.2, X.3, X.4, X.5, X.6, X.7, X.8, X.9, X.10, X.11, X35.39.Laki.Laki,
      X35.39.Perempuan, X40.44.Laki.Laki, X40.44.Perempuan,
      X45.49.Laki.Laki, X45.49.Perempuan, X50.54.Laki.Laki,
      X50.54.Perempuan, X55.59.Laki.Laki, X55.59.Perempuan,
      X60.64.Laki.Laki, X60.64.Perempuan, X65.69.Laki.Laki,
      X65.69.Perempuan, X70.74.Laki.Laki, X70.74.Perempuan, X.75.Laki.Laki,
      X.75..Perempuan [267x37]
```

```
faceting: facet_null()
```

Menampilkan Plot hasil Mapping

Masih terkait dengan praktek sebelumnya, ada pertanyaan yang muncul di benak kita. Seperti apa tampilan dari plot yang telah dilengkapi dengan data dan aesthetic mapping?

Untuk menjawab hal ini, cobalah jalankan tugas praktek berikut.

Tugas Praktek

Perhatikan seluruh perintah yang ada pada code editor, ini merupakan kumpulan code yang telah kita pelajari pada bab ini.

Jika berhasil dijalankan, maka hasilnya akan terlihat sebagai berikut. Terlihat plot tidak kosong, tapi sudah terisi dengan label LUAS.WILAYAH..KM2. pada sumbu x, label KEPADATAN..JIWA.KM2 pada sumbu y, dan sudah ada garis-garis penjelas grid pada plot.

Code Editor

```
library(ggplot2)

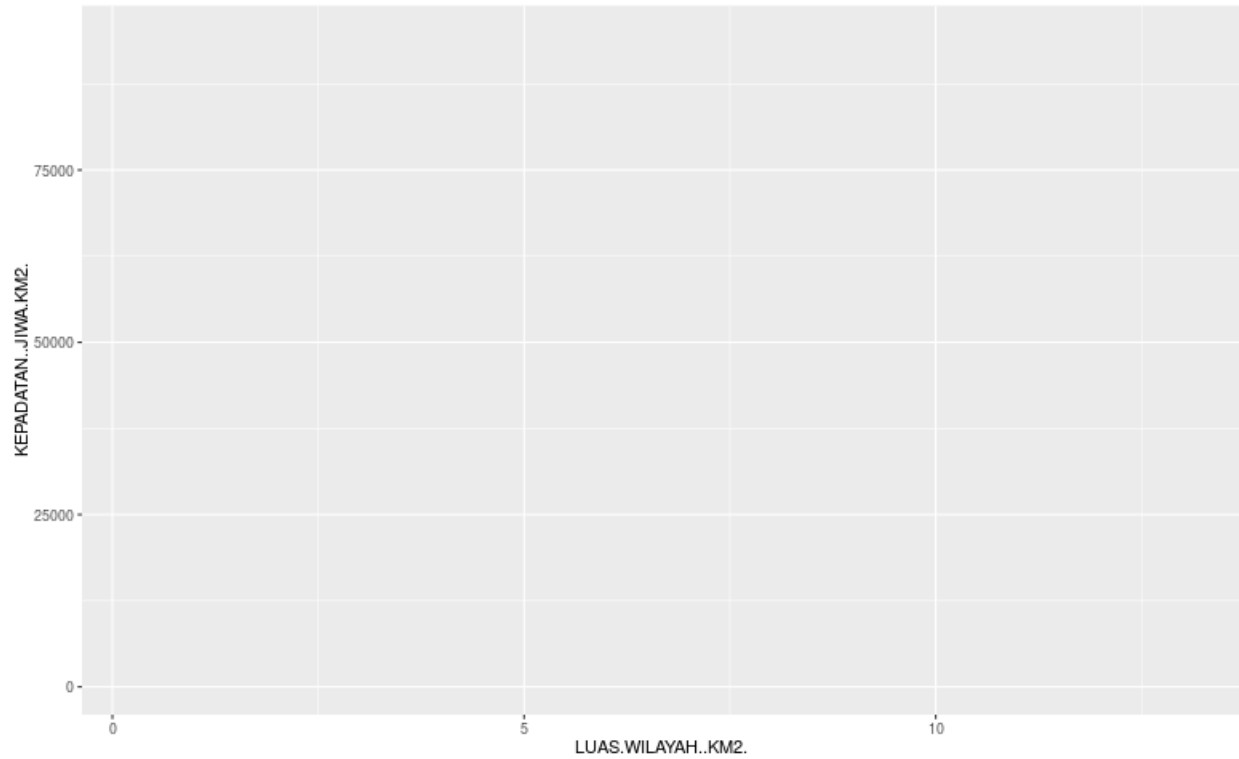
#Membaca data csv dan dimasukkan ke variable penduduk.dki

penduduk.dki <-
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

plot.dki <- ggplot(data=penduduk.dki, aes(x = LUAS.WILAYAH..KM2.,
y=KEPADATAN..JIWA.KM2., color=NAMA.KABUPATEN.KOTA))

plot.dki
```

Console



Kesimpulan

Plot memerlukan dua input sebagai bahan bakunya agar bisa digunakan lebih jauh, yaitu data dan aesthetic mapping (pemetaan beberapa kolom data ke elemen visual).

Pendahuluan

Mengulang kembali sedikit konsep yang telah dipaparkan di bab awal, grafik bisa dihasilkan dengan menambahkan **layer** secara berlapis di atas plot.

Setiap layer terdiri dari objek-objek berikut:

- **Geom**: Bentuk geometri seperti garis (line), batang (bar), titik (point), dan lain-lain.
- **Stat**: Atau suatu fungsi untuk melakukan transformasi statistik terhadap data input.
- Contoh paling sederhana adalah transformasi data untuk kepadatan jiwa dari angka menjadi range atau interval per lima ribuan. Jadi data input dengan angka 8041 diubah menjadi interval angka 8001-8500. Transformasi ini disebut dengan **bin**. Jika kita tidak ingin mengubah apa-apa, stat yang kita gunakan adalah **identity**.
- **Position**: Posisi dari beberapa data yang memiliki nilai yang sama. Jika diplot sebagai scatter plot misalnya, tentunya data-data tersebut akan menumpuk di satu titik. Apakah perlu ditambahkan nilai acak tertentu sehingga pas digambarkan, terlihat datanya lebih tersebar? Jika iya, maka ini namanya **jitter**. Jika kita tidak ingin merubah posisi, maka kita gunakan **identity**.

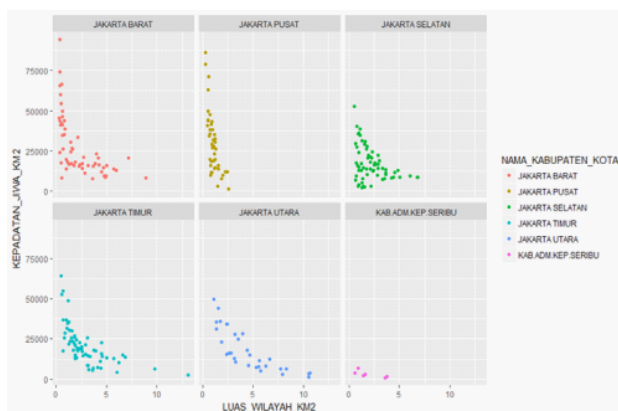
Berikut adalah diagram summary untuk proses menampilkan grafik dengan objek plot dan layer.

plot



- data
- aesthetic mapping

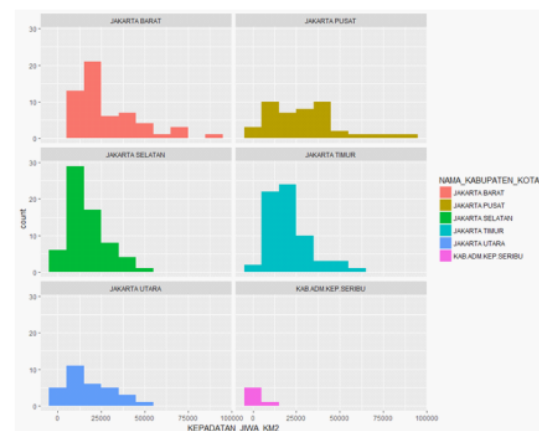
+ layer



- geom: point
- stat: identity
- position: identity

Klik tombol Next untuk melanjutkan

+ layer



- geom: histogram
- stat: bin
- position: identity

Proses menampilkan grafik

Mana proses menampilkan grafik yang benar di bawah ini?

- ☒ Inisialisasi plot, penambahan layer, dan menampilkan grafik
- ☐ Inisialisasi layer, penambahan plot, dan menampilkan grafik
- ☐ Inisialisasi layer, coordinate, plot dan tampilkan grafik
- ☐ Inisialisasi layer, inisialisasi koordinat, penambahan plot, dan menampilkan grafik
- ☐ Semua benar

Mana yang bukan komponen layer?

Dari daftar di bawah ini, mana yang bukan termasuk sebagai layer?

- ☐ Data
- ☐ Aesthetic Mapping
- ☐ Geom Object
- ☐ Statistic Transformation (stat)
- ☒ Plot
- ☐ Position Adjustment

Default pada aesthetic (masuk ke mana?)

Berikut adalah pernyataan yang benar mengenai default

- ☒ Data dan aesthetic mapping pada plot adalah default untuk seluruh layer
- ☒ Data dan aesthetic mapping pada tiap layer akan override default pada plot
- ☐ Data dan aesthetic mapping pada tiap layer tidak bisa diberikan jika terdapat data dan aesthetic pada plot
- ☐ Data dan aesthetic bisa tidak dimasukkan pada plot object => need to test
- ☐ Semua benar

Melihat Detail Plot dan Layers

Jika p adalah variable object plot dan layers, function apakah yang digunakan untuk melihat detilnya?

- ☐ inspect
- ☐ print
- ☐ p
- ☒ summary
- ☐ semuanya bisa

Scatter Plot Kepadatan Penduduk Jakarta dengan function layer

Untuk menggambar grafik sebenarnya pada ggplot, kita menambahkan apa yang dinamakan layer. Layer bisa dihasilkan dengan menggunakan function **layer** dengan syntax berikut:

```
layer(geom = "...", stat = "...", position = "...")
```

Keterangan

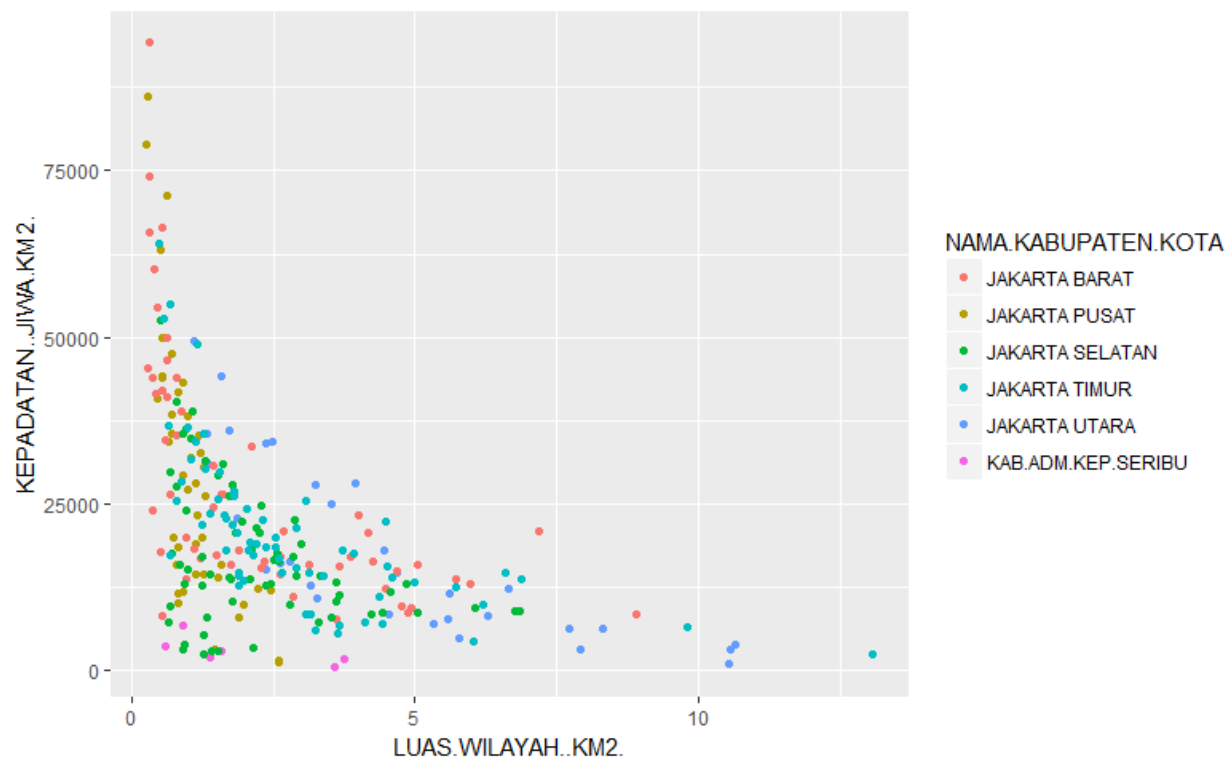
Komponen	Deskripsi
Layer	Function yang digunakan untuk membentuk layer untuk ditambahkan pada plot
geom	Bentuk geometri yang digunakan untuk layer. Beberapa daftar geometri yang bisa diberikan adalah sebagai berikut: <ul style="list-style-type: none">• point: untuk menggambar grafik berupa titik yang menunjukkan hubungan antar variable atau <i>scatter plot</i>• line: diagram garis• hline: diagram garis horizontal• bar: diagram batang• histogram: histogram• dll
stat	Jenis transformasi untuk merubah data dari bentuk asli ke bentuk lain. Beberapa daftar stat yang bisa digunakan antara lain adalah sebagai berikut: <ul style="list-style-type: none">• identity: tidak melakukan transformasi• bin: membagi data menjadi interval• sum: menjumlahkan nilai-nilai yang unik• summary: melakukan summary dari semua nilai y untuk setiap titik data x• dll
position	Perubahan posisi terhadap tiap titik yang terdapat pada layer. Beberapa daftar position yang bisa digunakan antara lain adalah sebagai berikut:

Komponen	Deskripsi
	<ul style="list-style-type: none"> • identity: tidak ada perubahan posisi • jitter: merubah posisi dengan jarak acak tertentu dari posisi awal, ini digunakan untuk memberi kejelasan jika banyak titik yang <i>overlapping</i> (saling tindih) • stack: menyusun posisi dalam bentuk tumpukan • dll

Jika kita ingin menggambar scatter plot, maka konstruksi layer yang digunakan akan berbentuk sebagai berikut:

```
layer(geom = "point", stat = "identity", position = "identity")
```

Ganti bagian [...] pada editor dengan code tersebut. Jika berhasil dijalankan, maka Anda harusnya mendapatkan output grafik sebagai berikut.



Terlihat grafik titik (scatter plot) dihasilkan data-data kependudukan diambil Luas Wilayah untuk sumbu x, Kepadatan untuk sumbu y, dan pewarnaan dilakukan sesuai Nama Kabupaten.

Code Editor

```
library(ggplot2)

#Membaca data csv dan dimasukkan ke variable penduduk.dki

penduduk.dki <-
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

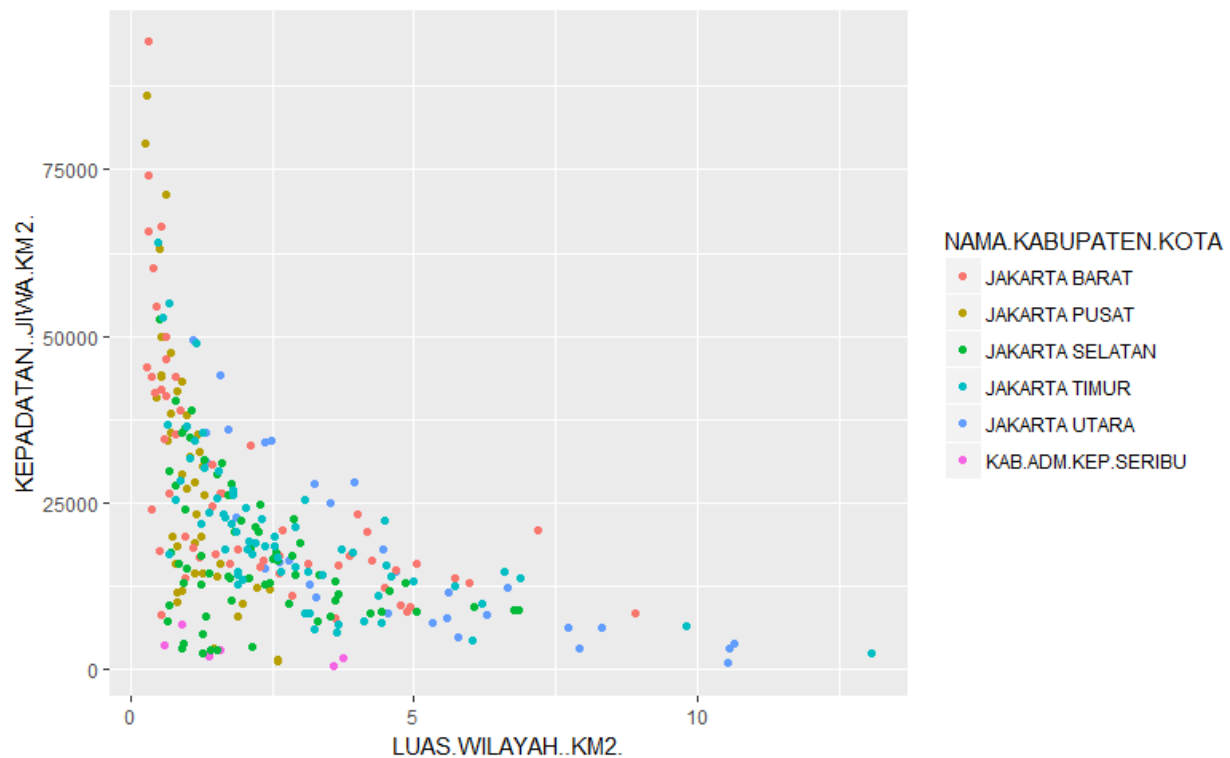
#Menambahkan data dan aesthetic mapping

plot.dki <- ggplot(data=penduduk.dki, aes(x = LUAS.WILAYAH..KM2.,
y=KEPADATAN..JIWA.KM2., color=NAMA.KABUPATEN.KOTA))

#Menambahkan layer untuk menghasilkan grafik scatter plot

plot.dki + layer(geom = "point", stat = "identity", position = "identity")
```

Console



Scatter Plot Kepadatan Penduduk Jakarta dengan geom_point

Perintah layer untuk menggambar scatter plot pada subbab sebelumnya, memiliki *shortcut function* yang lebih pendek. Function tersebut bernama **geom_point**.

Dengan demikian, fungsi layer yang sebelumnya harus diketik sebagai berikut.

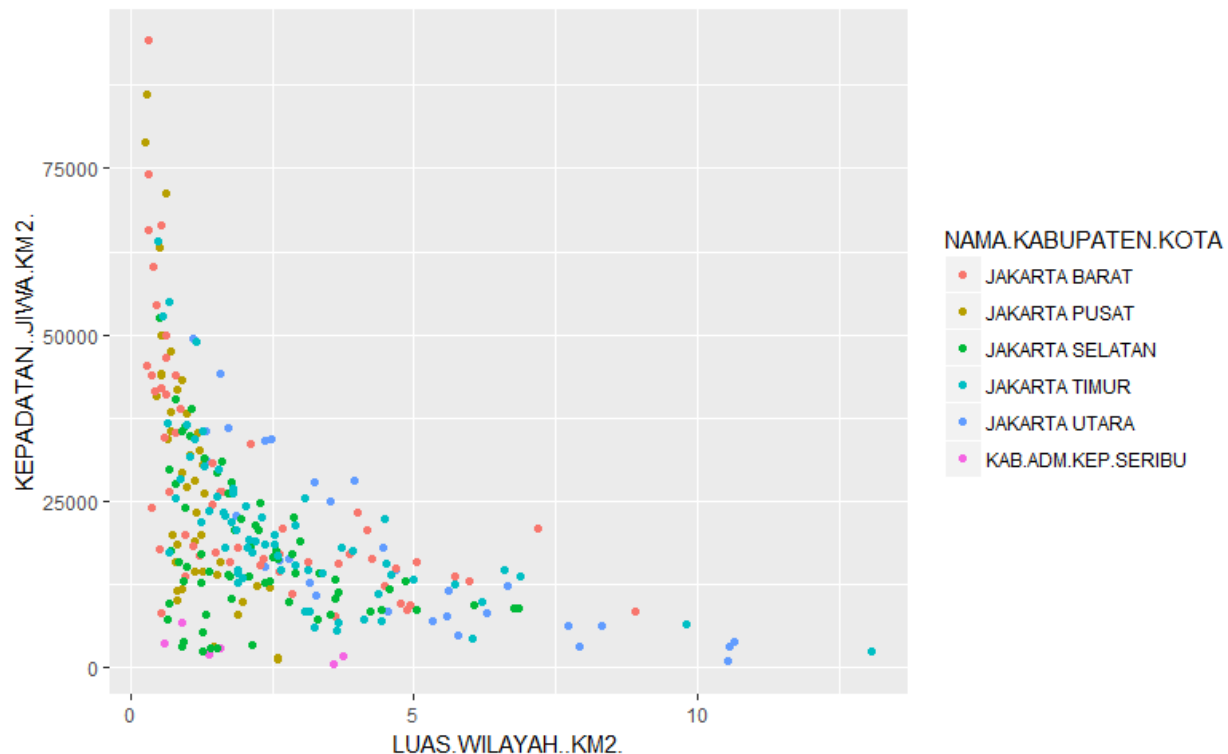
```
layer(geom = "point", stat = "identity", position = "identity")
```

dapat dirubah menjadi

```
geom_point()
```

Catatan: Stat dan position tidak perlu diisi, karena fungsi geom_point ini secara otomatis akan memiliki nilai "identity" untuk stat dan position. Berbeda ketika menggunakan layer, dimana parameter stat dan position harus diisi lengkap.

Ketik geom_point() untuk menggantikan bagian [...] Jika berjalan dengan baik, grafik yang sama dengan praktek sebelumnya akan dihasilkan sebagai berikut.



Code Editor

```
library(ggplot2)

#Membaca data csv dan dimasukkan ke variable penduduk.dki

penduduk.dki <-
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")

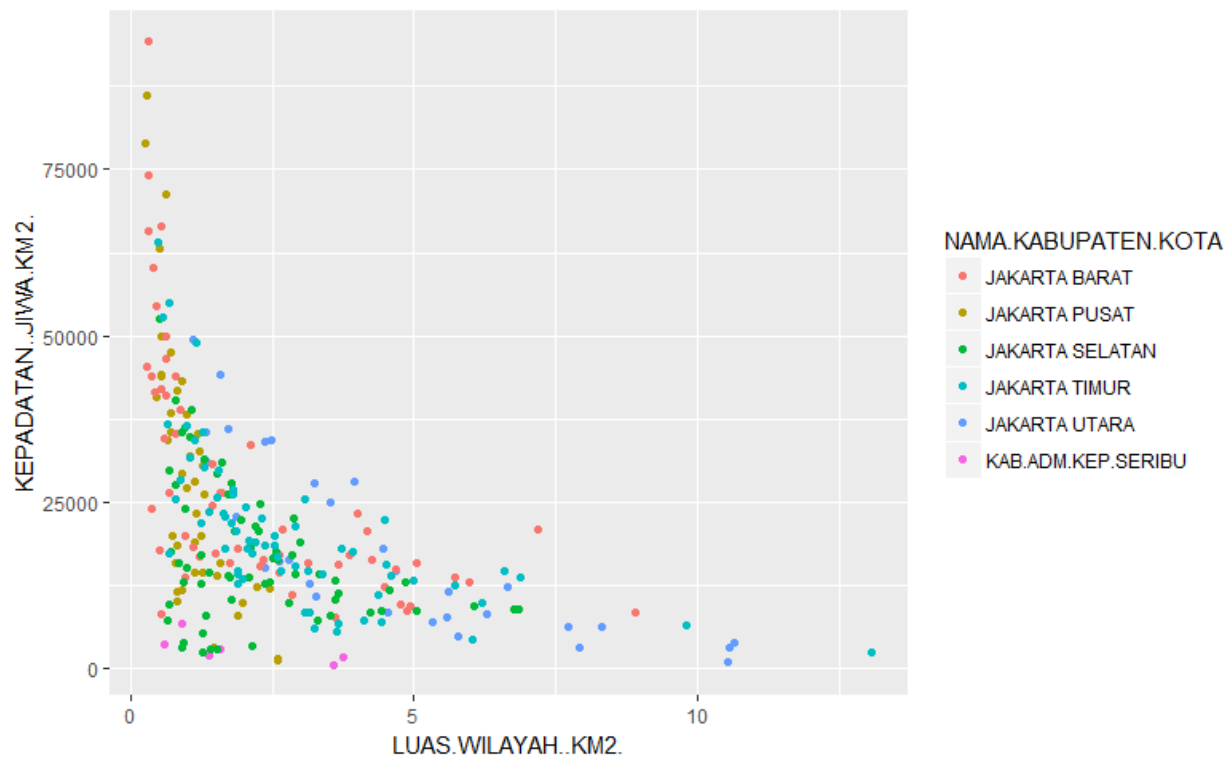
#Menambahkan data dan aesthetic mapping

plot.dki <- ggplot(data=penduduk.dki, aes(x = LUAS.WILAYAH..KM2.,
y=KEPADATAN..JIWA.KM2., color=NAMA.KABUPATEN.KOTA))

#Menambahkan layer scatter plot dengan geom_point

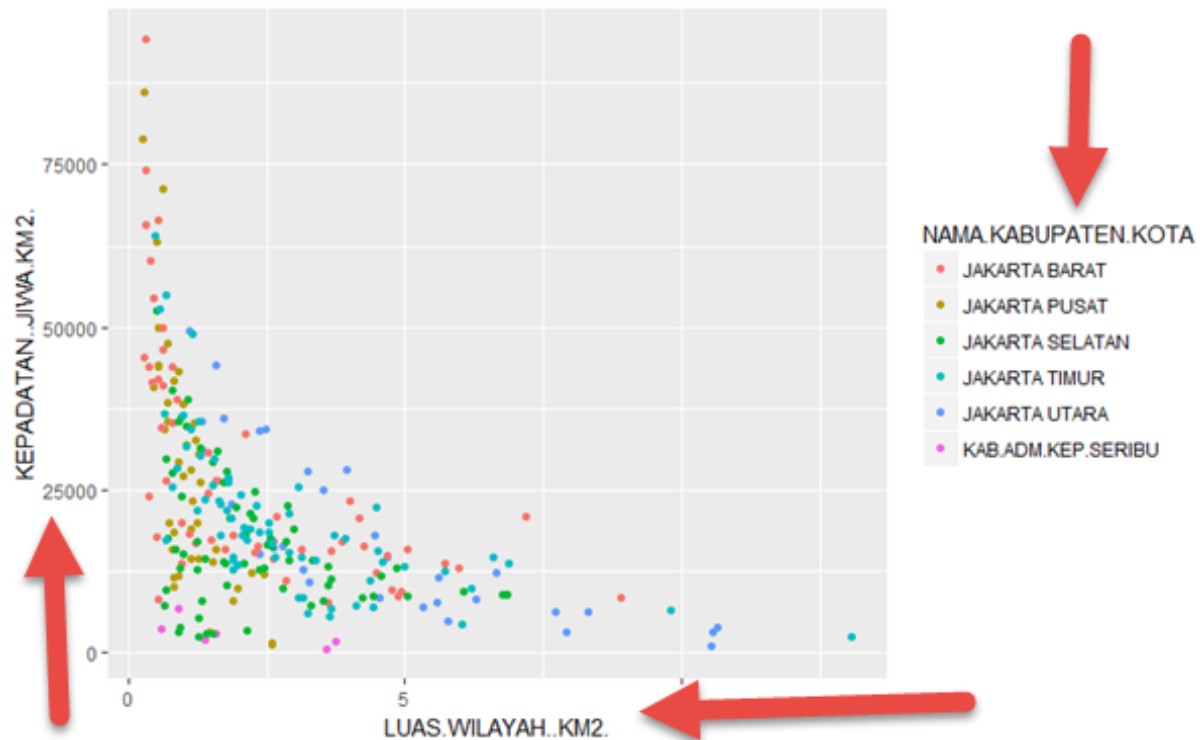
plot.dki + geom_point()
```

Console



Menambahkan Judul dan Label

Scatter plot yang dihasilkan sudah menarik untuk kita, namun label teks pada x, y dan color masih terasa kurang bagus. Bukankah begitu? Mari kita perhatikan.

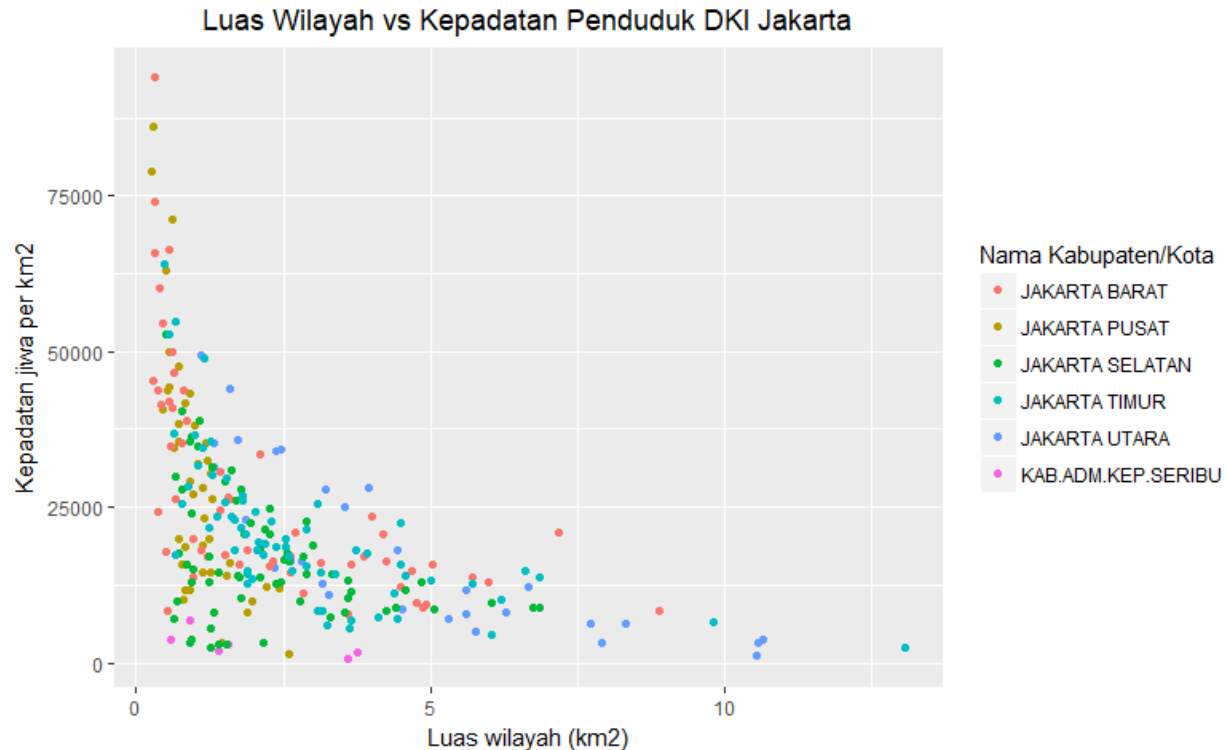


Kita bisa merubah label ini sesuai dengan keinginan kita menggunakan function **labs** yang sudah kita pelajari sebelumnya.

Gantilah bagian ... pada code editor dengan function **labs** dengan parameter-parameter berikut.

- title = "Luas Wilayah vs Kepadatan Penduduk DKI Jakarta".
- x = "Luas wilayah (km2)"
- y = "Kepadatan Jiwa per km2"
- color = "Nama Kabupaten/Kota"

Jika berjalan lancar harusnya didapatkan hasil sebagai berikut.



Catatan: Pada code terdapat perintah baru, yaitu **theme(plot.title = element_text(hjust=0.5))**. Ini adalah code untuk menempatkan judul di tengah plot.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <-  
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

```
#Menambahkan data dan aesthetic mapping
```

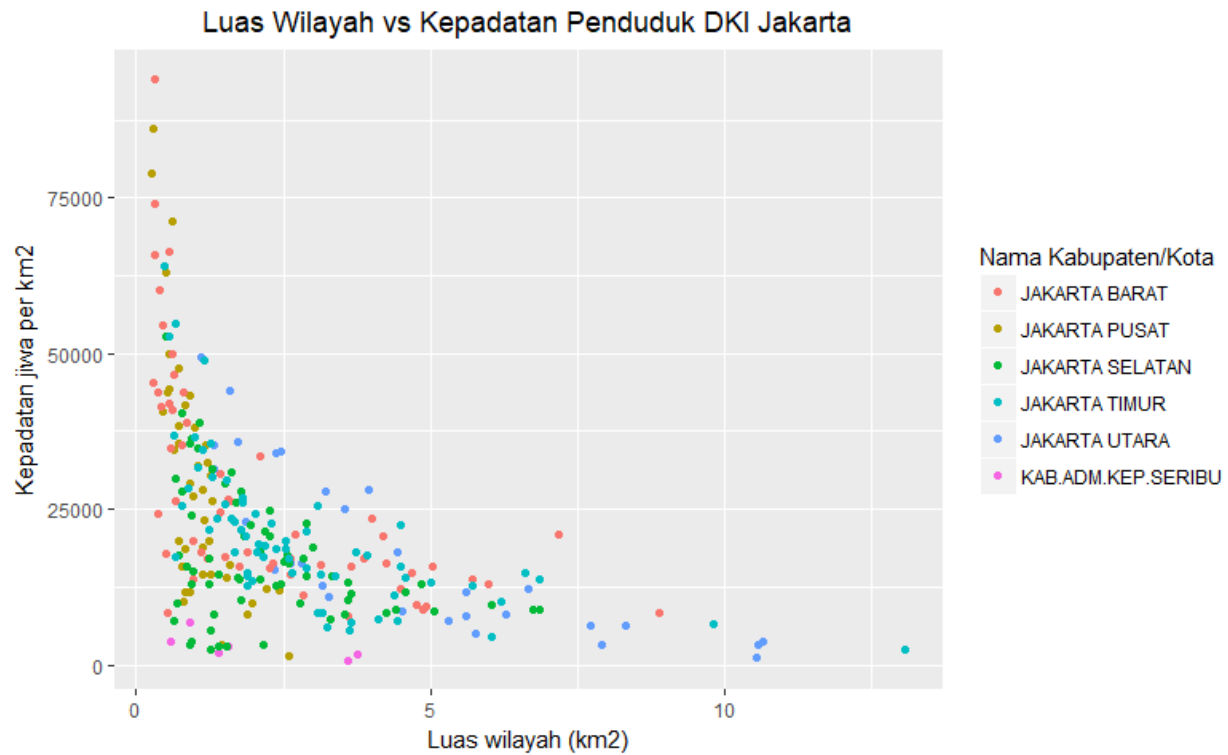
```
plot.dki <- ggplot(data=penduduk.dki, aes(x = LUAS.WILAYAH..KM2.,  
y=KEPADATAN..JIWA.KM2., color=NAMA.KABUPATEN.KOTA))
```

```
#Menambahkan Layer dan labels
```

```
plot.dki + geom_point() +  
theme(plot.title = element_text(hjust=0.5)) +
```

```
labs(title="Luas Wilayah vs Kepadatan Penduduk DKI Jakarta", x="Luas wilayah (km2)", y="Kepadatan Jiwa per km2", color="Nama Kabupaten/Kota")
```

Console



Kesimpulan

Layer diperlukan agar grafik sebenarnya dapat digambar di atas objek plot. Layer dibuat dengan function **layer**. Parameter function layer yang harus diisi terdiri dari **geom**, **stat** dan **position**.

Dengan pengisian nilai yang berbeda-beda terhadap tiga parameter ini, kita bisa menghasilkan berbagai macam grafik. Pada bab ini, grafik yang kita hasilkan adalah "scatter plot" dimana parameter geom bernilai "point", stat dan position bernilai "identity".

Selain menggunakan function layer, tiap objek juga memiliki function yang memiliki prefix geom_, stat_ , dan position_. Misalkan, untuk geom bertipe point memiliki function geom_point. Ini dibuat dengan tujuan agar beberapa nilai default untuk tiap-tiap objek sudah dispesifikasikan di function-function berprefix.

Dan bab ditutup dengan praktek dimana kita bisa menyelesaikan satu grafik lengkap dengan judul, dan label-label yang lebih mudah dibaca.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Histogram adalah tipe visualisasi yang sangat cocok untuk menggambarkan data distribusi dari jumlah populasi data. Dan dataset kependudukan adalah contoh yang baik dimana kita bisa menggambarkan distribusi kepadatan penduduk dengan jumlah kelurahan.

Untuk membuat histogram, kita gunakan geom bertipe histogram dan stat bin, yang bisa diwakili oleh function `geom_histogram`.

Klik tombol Next untuk melanjutkan.

Layer `geom_histogram` dan Lebar Interval

Untuk menghasilkan histogram maka kita gunakan function `geom_histogram` untuk ditambahkan ke plot.

Layer histogram ini memerlukan stat bertipe bin, yaitu membagi data menjadi interval. Namun dengan penggunaan **`geom_histogram`**, **`stat`** ini sudah menjadi **`bin`** secara *default* jadi tidak perlu lagi dispesifikasikan.

Secara default geom histogram tidak memerlukan aesthetic mapping untuk sumbu y, cukup menggunakan sumbu x.

Namun ada parameter yang perlu diinput, yaitu besarnya interval **`bin`**. Ini menggunakan parameter **`binwidth`**.

Contoh untuk membuat layer `geom_histogram` adalah sebagai berikut.

`[objek plot] + geom_histogram(binwidth=5000)`

Komponen	Deskripsi
<code>[objek plot]</code>	Objek plot dari hasil function <code>ggplot()</code> ditambah dengan komponen visual atau layer
<code>+</code>	Tanda plus, operator untuk menambahkan komponen lain ke dalam plot
<code>geom_histogram</code>	Function untuk menghasilkan layer dengan geom histogram
<code>binwidth</code>	Lebar interval data, dalam hal ini 5000

Tugas Praktek

Pada code editor telah tersedia code untuk membaca data kependudukan DKI. Ganti bagian [...] dengan layer histogram dan dengan interval 10000.

Jika berhasil maka grafik yang tampil adalah sebagai berikut. Terlihat bahwa jumlah kelurahan dengan penduduk antara 10 s/d 20 ribu adalah yang terbanyak.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

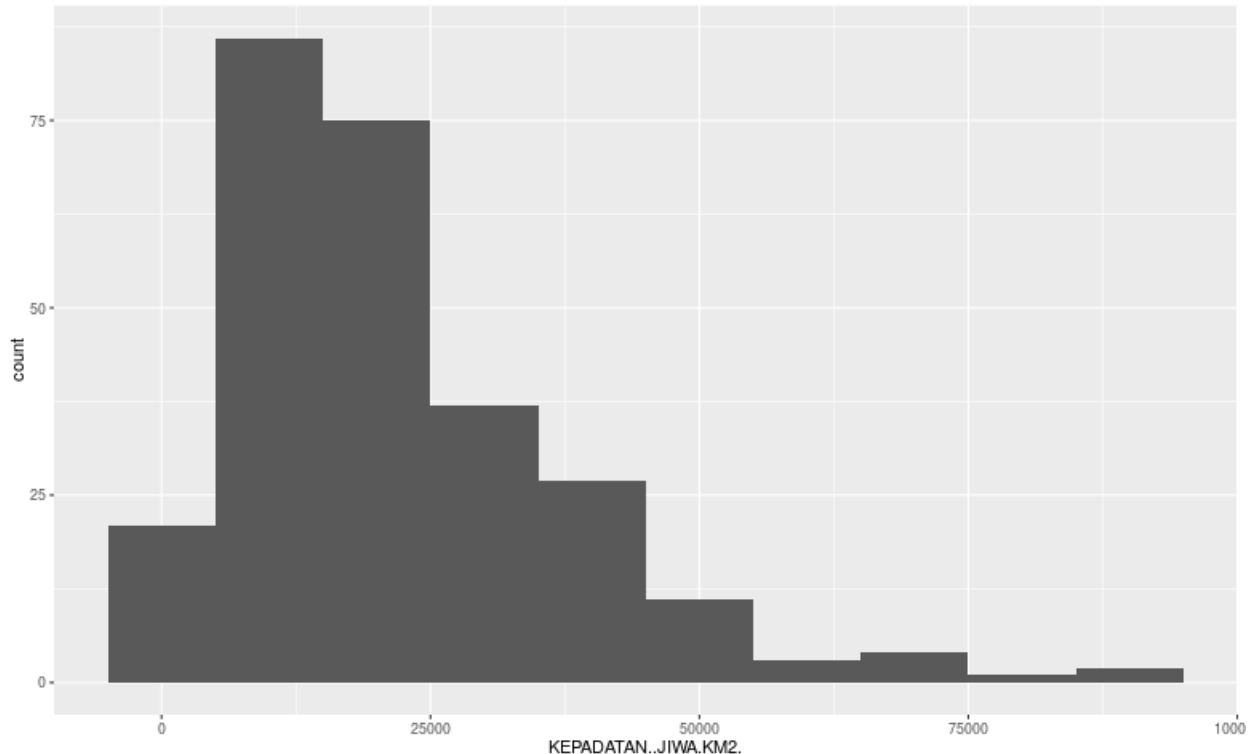
```
penduduk.dki <-  
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

```
#Menambahkan data dan aesthetic mapping
```

```
plot.dki <- ggplot(data=penduduk.dki, aes(x = KEPADATAN..JIWA.KM2.))
```

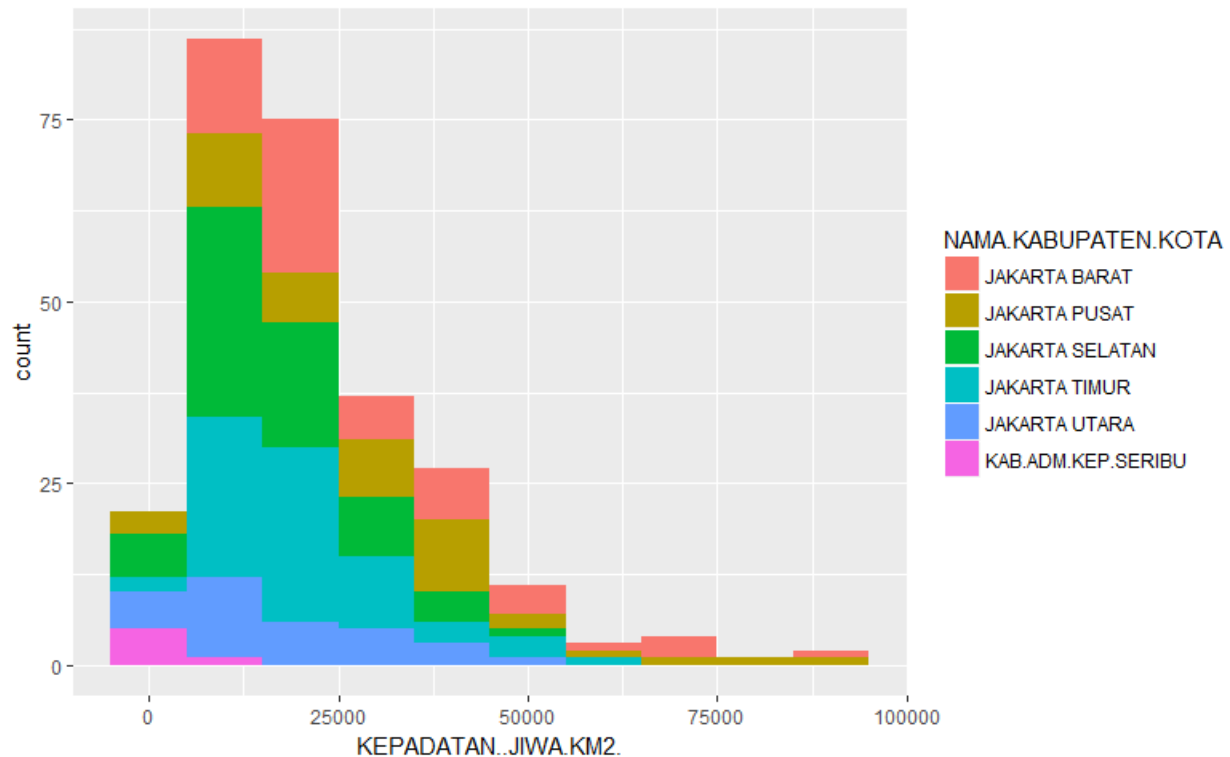
```
plot.dki + geom_histogram(binwidth=10000)
```

Console



Penggunaan aesthetic fill

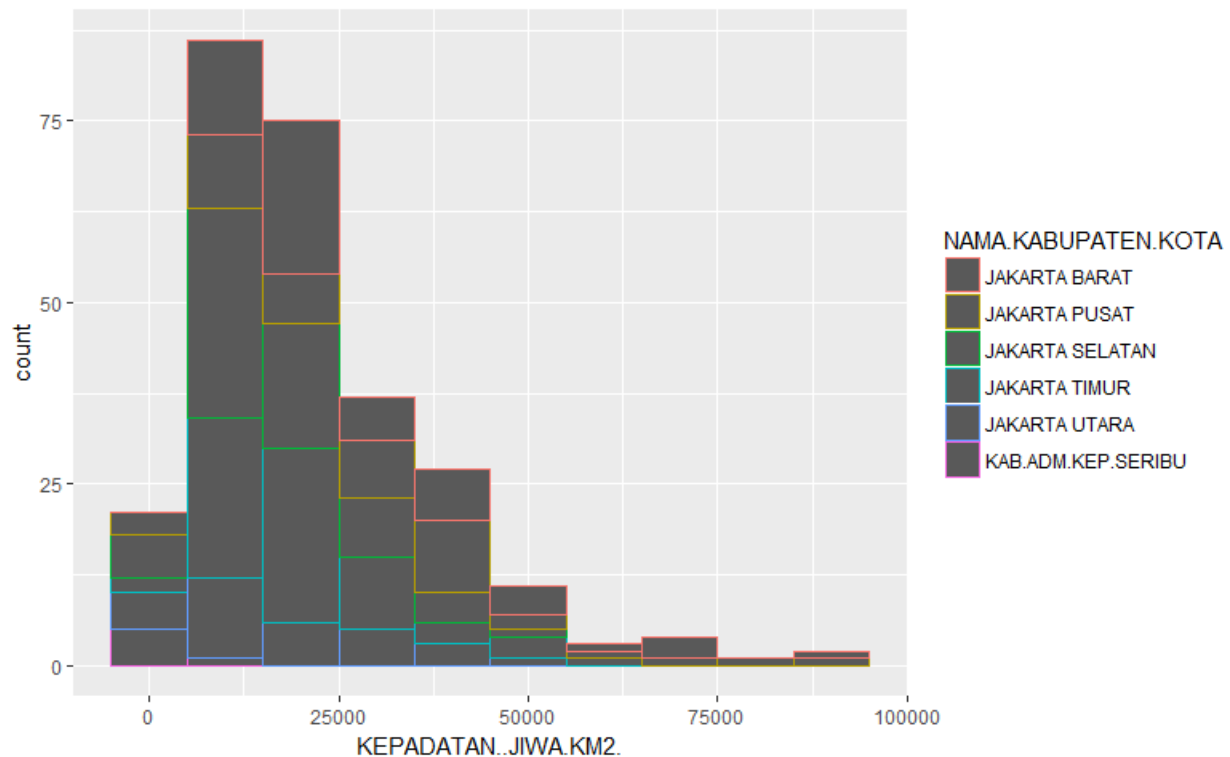
Kita dapat melakukan banyak penambahan informasi pada histogram sebelumnya. Salah satunya adalah melihat porsi jumlah kelurahan berdasarkan nama kabupaten / kota pada tiap rentang histogram seperti berikut.



Untuk ini kita menggunakan aesthetic fill dengan syntax berikut.

```
aes(fill = NAMA.KABUPATEN.KOTA)
```

Catatan: aesthetic color sebenarnya juga bisa digunakan namun hasilnya akan tampak sebagai berikut.



Tugas Praktek

Lengkapi code editor dimana histogram yang dihasilkan bisa ditambahkan dengan aesthetic fill – input berupa kolom NAMA.KABUPATEN.KOTA.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <-
```

```
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

```
plot.dki <- ggplot(data=penduduk.dki, aes(x = KEPADATAN..JIWA.KM2., aes(fill =  
NAMA.KABUPATEN.KOTA)))
```

```
plot.dki + geom_histogram(binwidth = 10000)
```

Console

(LANGSUNG PRAKTEK)

Kesimpulan

Histogram adalah tipe visualisasi yang sangat cocok untuk menggambarkan data distribusi dari jumlah populasi data. Dan dataset kependudukan adalah contoh yang baik dimana kita bisa menggambarkan distribusi kepadatan penduduk dengan jumlah kelurahan.

Praktek pada bab ini cukup simpel dan straightforward, dimana kita telah menggunakan function `geom_histogram` dengan parameter `binwidth` untuk menghasilkan grafik distribusi.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Line chart atau grafik garis adalah tipe visualisasi yang sangat baik untuk menggambarkan apa impact (pengaruh) dari perubahan suatu variable dari satu titik ke titik lain atau trend— dan variable yang paling umum digunakan adalah waktu.

Sebagai contoh, di bidang ekonomi untuk menggambarkan inflasi dari bulan ke bulan. Namun tentunya tidak harus selalu waktu. Perubahan lain, misalkan pengaruh kecepatan lari dengan peningkatan detak jantung.

Untuk membuat line chart standar, kita gunakan geom bertipe line dan stat identity, yang bisa diwakili oleh function `geom_line`.

Pada bab berikut kita akan gunakan dataset tambahan, yaitu tingkat inflasi bulanan tahun 2017 untuk negara Indonesia dan Singapura. Selain plotting, diperkenalkan juga konsep factoring untuk menangani data dan grouping untuk grafik.

Klik tombol Next untuk melanjutkan.

Dataset Inflasi

Untuk dataset untuk inflasi kita akan "loncat" sedikit ke dataset lain, yaitu dataset inflasi sepanjang tahun 2017 untuk negara Indonesia dan Singapore. Dataset ini dikumpulkan oleh DQLab untuk keperluan pelatihan dan dapat didownload di:

<https://academy.dqlab.id/dataset/inflasi.csv>

Dataset ini hanya memiliki tiga kolom, yaitu:

- **Bulan:** bulan dan tahun data inflasi.
- **Negara:** negara yang mengalami inflasi.
- **Inflasi:** tingkat inflasi dalam persentase.

	A	B	C	D
1	Bulan	Negara	Inflasi	
2	Jan-17	Indonesia	0.0349	
3	Feb-17	Indonesia	0.0383	
4	Mar-17	Indonesia	0.0361	
5	Apr-17	Indonesia	0.0417	
6	May-17	Indonesia	0.0433	
7	Jun-17	Indonesia	0.0437	
8	Jul-17	Indonesia	0.0388	
9	Aug-17	Indonesia	0.0382	
10	Sep-17	Indonesia	0.0372	
11	Oct-17	Indonesia	0.0358	
12	Jan-17	Singapura	0.025	
13	Feb-17	Singapura	0.027	
14	Mar-17	Singapura	0.024	
15	Apr-17	Singapura	0.022	
16	May-17	Singapura	0.019	
17	Jun-17	Singapura	0.016	
18	Jul-17	Singapura	0.017	
19	Aug-17	Singapura	0.019	
20	Sep-17	Singapura	0.022	
21	Oct-17	Singapura	0.02	
22				

File ini berbentuk csv dan akan dibaca dengan menggunakan function **read.csv**.

Membaca data inflasi

Langkah pertama yang perlu kita lakukan adalah membaca dataset inflasi tersebut dari file teks menjadi data.frame di R dengan perintah **read.csv**.

Berdasarkan keterangan mengenai dataset ini dan praktek di dua bab sebelumnya, kita langsung lakukan praktek sebagai berikut.

Tugas Praktek

Ganti bagian [...] dengan kedua perintah berikut pada code editor yang berfungsi untuk membaca dan menampilkan data inflasi .

```
inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv",
sep=",")
```

inflasi.indo.sing

Code Editor

#Membaca data csv dan dimasukkan ke variable inflasi.indo.sing

```
inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep=",")
```

inflasi.indo.sing

Console

```
> #Membaca data csv dan dimasukkan ke variable inflasi.indo.sing
> inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep="
,")

> inflasi.indo.sing
      Bulan   Negara Inflasi
1 Jan-2017 Indonesia 0.0349
2 Feb-2017 Indonesia 0.0383
3 Mar-2017 Indonesia 0.0361
4 Apr-2017 Indonesia 0.0417
5 May-2017 Indonesia 0.0433
6 Jun-2017 Indonesia 0.0437
7 Jul-2017 Indonesia 0.0388
8 Aug-2017 Indonesia 0.0382
9 Sep-2017 Indonesia 0.0372
10 Oct-2017 Indonesia 0.0358
11 Jan-2017 Singapura 0.0250
12 Feb-2017 Singapura 0.0270
13 Mar-2017 Singapura 0.0240
14 Apr-2017 Singapura 0.0220
15 May-2017 Singapura 0.0190
16 Jun-2017 Singapura 0.0160
17 Jul-2017 Singapura 0.0170
18 Aug-2017 Singapura 0.0190
```

Error pada saat Plotting Line Chart

Untuk membuat line chart dari data inflasi, kita lakukan langkah-langkah dari ggplot2:

- Membuat plot, dengan function ggplot()
- Mengisi data dari pembacaan file dengan function read.csv()
- Membuat aesthetic mapping, dengan function aes
- Menambahkan layer, dengan function geom_line()

Berikut adalah tugas praktek yang coba kita lakukan. Untuk praktek kali ini, sengaja kita buat hasilnya error.

Tugas Praktek

Tambahkan layer untuk line chart pada bagian [...] di code editor yang telah dilengkapi dengan plot, data, dan aesthetic mapping.

Jika berjalan dengan baik, maka akan muncul dua hasil, satu grafik berupa plot kosong dan satu tampilan error sebagai berikut. Ini artinya sistem kebingungan karena tiap Bulan pada sumbu x memiliki dua data (observation). Kondisi ini akan kita perbaiki pada praktek selanjutnya.

geom_path: Each group consists of only one observation. Do you need to adjust the group aesthetic?

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable inflasi.indo.sing
```

```
inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep=";",)
```

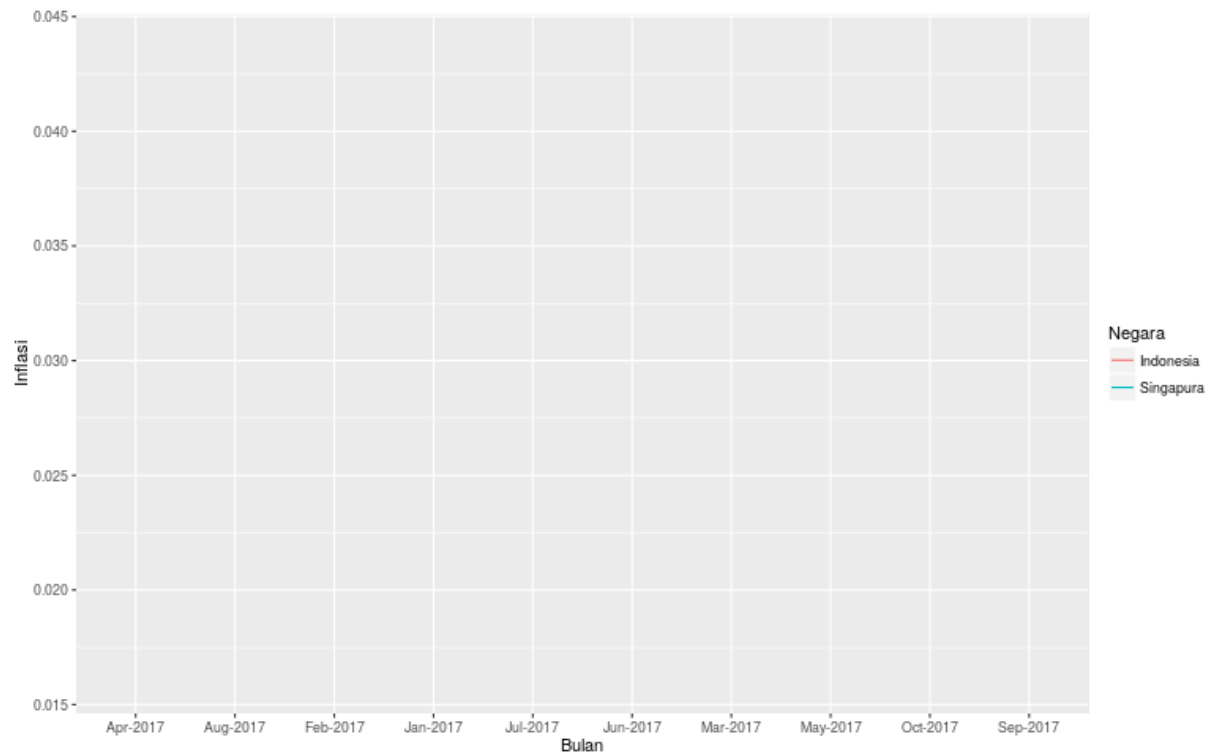
```
#Menambahkan data dan aesthetic mapping
```

```
plot.inflasi <- ggplot(data=inflasi.indo.sing, aes(x = Bulan, y=Inflasi, color=Negara))
```

#Menambahkan layer

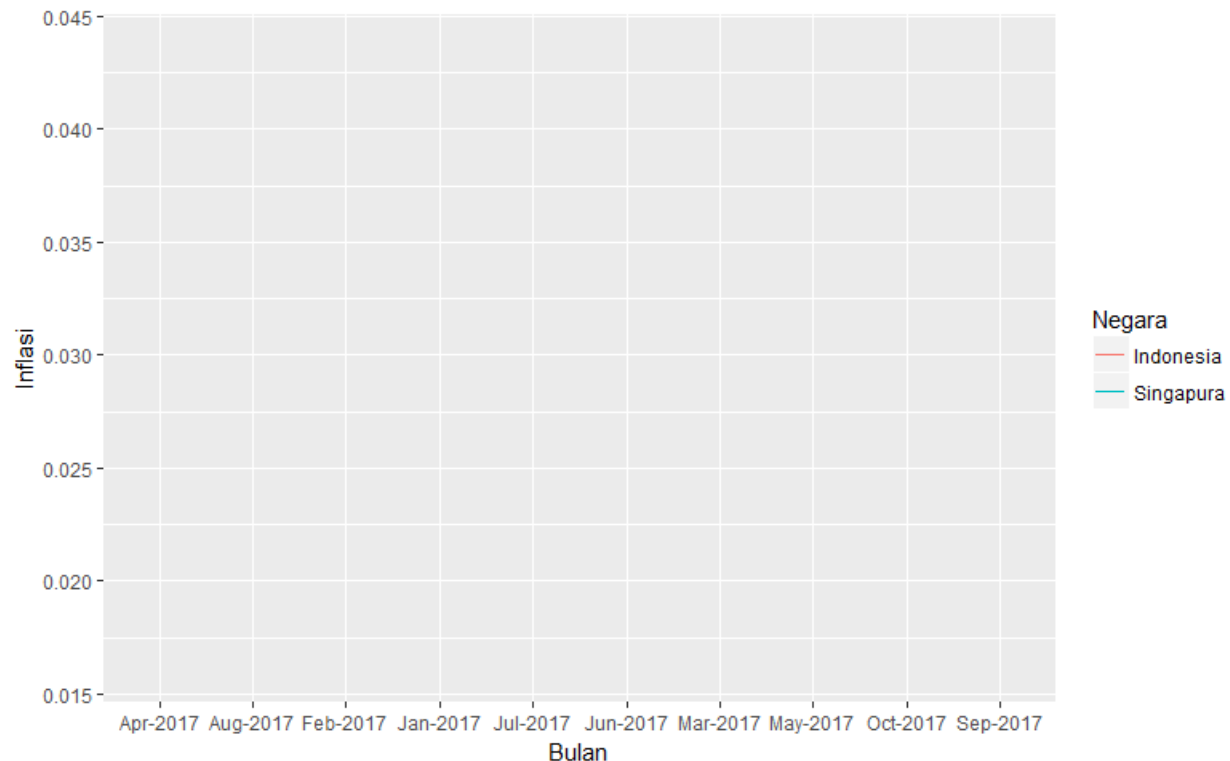
plot.inflasi + geom_line()

Console



Menggunakan Pengelompokan Data (grouping)

Pada praktek sebelumnya, kita tidak mendapatkan grafik trend inflasi dari negara Indonesia dan Singapura. Hasil akhirnya terlihat sebagai berikut.



Penyebabnya adalah sistem kebingungan karena data untuk tiap bulan memiliki dua titik data, yang akan menjadi kendala karena tiap line perlu satu titik data untuk sumbu x dan y.

Ini karena kita belum membagi data dengan aesthetic grouping dengan konstruksi berikut. Walaupun pada gambar kita sudah memiliki aesthetic color yang membagi Indonesia dan Singapura, tapi kita masih perlu memiliki aesthetic group sebagai berikut:

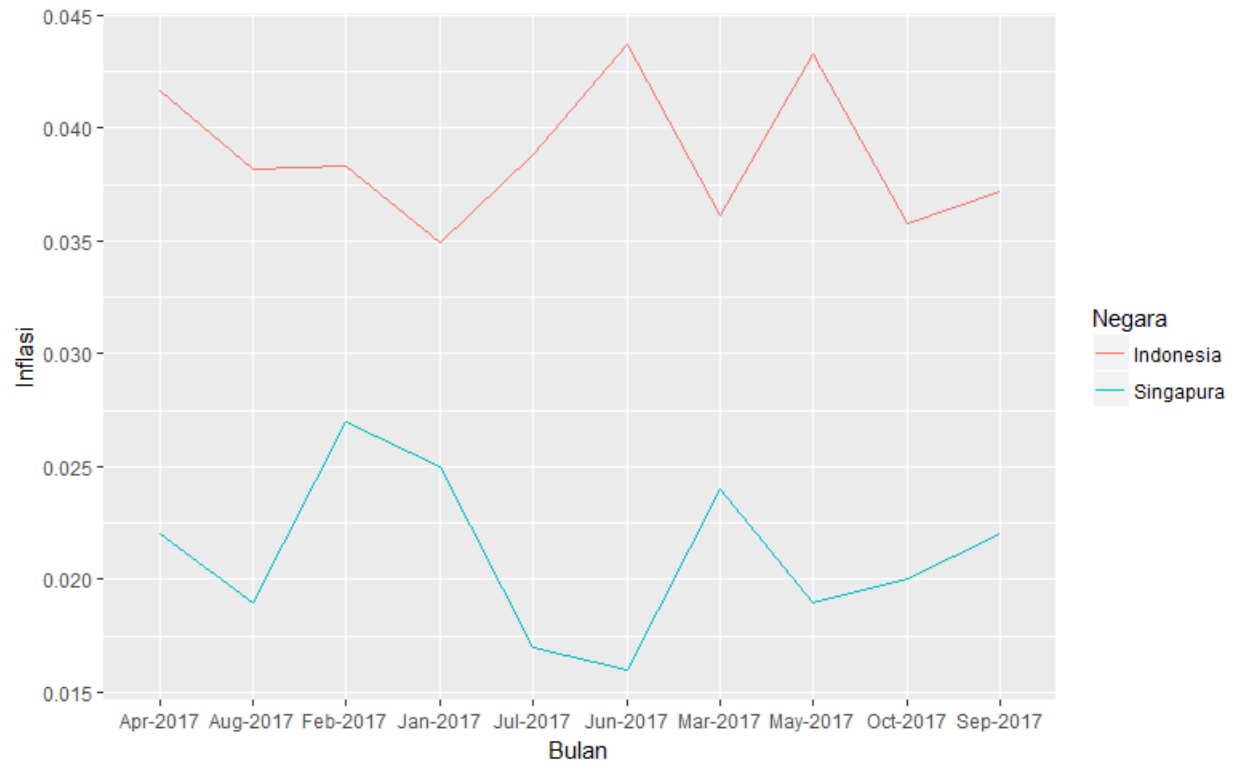
```
aes(..., group=Negara)
```

Dimana ... mewakili pasangan aesthetic mapping lainnya.

Tugas Praktek

Tambahkan aesthetic group dengan Negara pada bagian [...] di code editor, sehingga jika dijalankan akan menghasilkan grafik berikut.

Catatan: Perhatikan jika teks Bulan pada sumbu X masih pada urutan yang salah. Jangan khawatir, ini akan kita perbaiki di praktek berikutnya.



Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable inflasi.indo.sing
```

```
inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep=",")
```

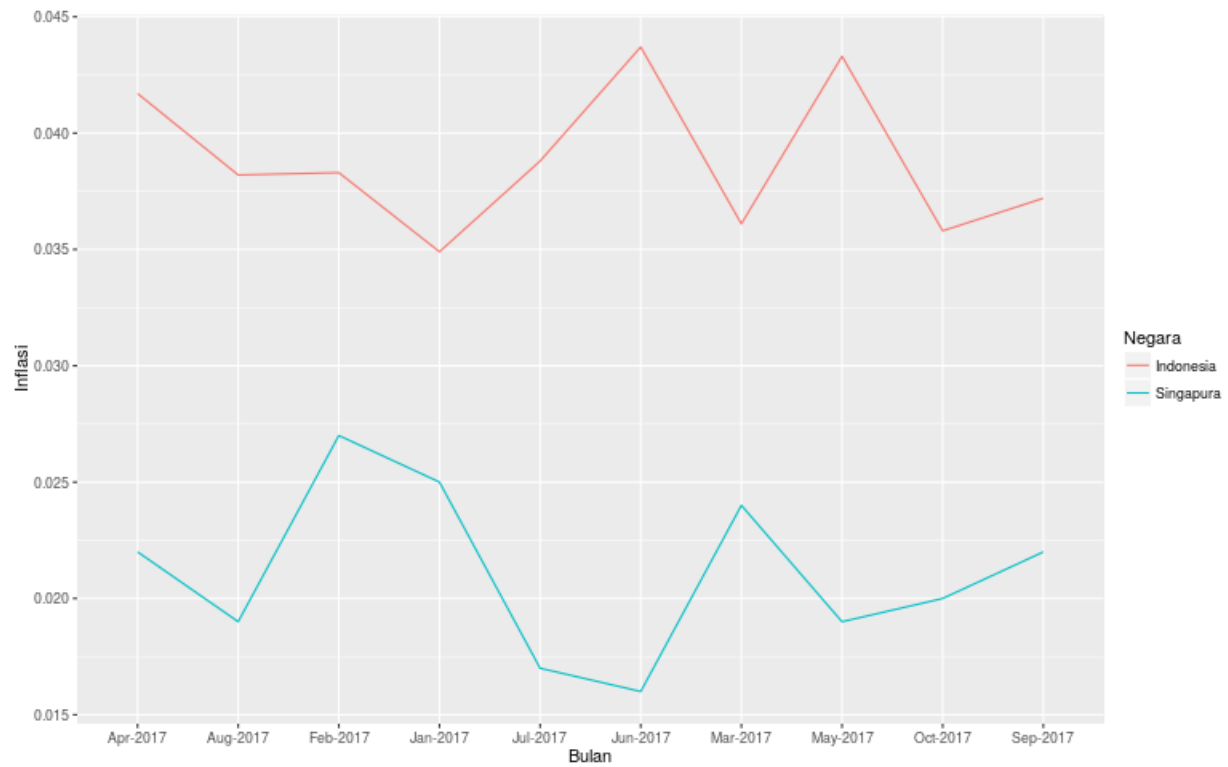
```
#Menambahkan data dan aesthetic mapping
```

```
plot.inflasi <- ggplot(data=inflasi.indo.sing, aes(x = Bulan, y=Inflasi, color=Negara, group=Negara))
```

```
#Menambahkan Layer
```

```
plot.inflasi + geom_line()
```

Console



Memperbaiki Urutan Bulan dengan Factoring

Urutan bulan yang salah pada praktek sebelumnya karena masalah pengurutan internal di sistem R, yang mengurutkan berdasarkan alfabet.

Jika kita lihat internal struktur dengan function str, maka untuk kolom bulan terlihat sebagai berikut.

```
> str(inflasi.indo.sing)
'data.frame': 20 obs. of 3 variables:
 $ Bulan : Factor w/ 10 levels "Apr-2017","Aug-2017",...: 4 3 7 1 8 6 5 2 10 9 ...
 $ Negara : Factor w/ 2 levels "Indonesia","Singapura": 1 1 1 1 1 1 1 1 1 1 ...
 $ Inflasi: num 0.0349 0.0383 0.0361 0.0417 0.0433 0.0437 0.0388 0.0382 0.0372 0.0358 ...
```

Kita dapat secara manual merubah urutan ini dengan **factor** dan parameter levels dimasukkan manual.

Dengan data bulan dari Jan s/d Oct 2017 maka konstruksi factor adalah sebagai berikut.

```
factor(inflasi.indo.sing$Bulan, levels = c("Jan-2017", "Feb-2017", "Mar-2017", "Apr-2017", "May-2017", "Jun-2017", "Jul-2017", "Aug-2017", "Sep-2017", "Oct-2017"))
```

Ini perlu dimasukkan kembali ke kolom bulan untuk merubah urutannya dengan perintah berikut.

```
inflasi.indo.sing$Bulan <- factor(inflasi.indo.sing$Bulan, levels = c("Jan-2017", "Feb-2017", "Mar-2017", "Apr-2017", "May-2017", "Jun-2017", "Jul-2017", "Aug-2017", "Sep-2017", "Oct-2017"))
```

Tugas Praktek

Gantilah bagian [...] dengan factoring sehingga hasil output terakhir yang tampil adalah sebagai berikut, sehingga data akan siap untuk ditampilkan.

```
'data.frame': 20 obs. of 3 variables:
 $ Bulan : Factor w/ 10 levels "Jan-2017","Feb-2017",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ Negara : Factor w/ 2 levels "Indonesia","Singapura": 1 1 1 1 1 1 1 1 1 1 ...
 $ Inflasi: num 0.0349 0.0383 0.0361 0.0417 0.0433 0.0437 0.0388 0.0382 0.0372 0.0358 ...
```

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable inflasi.indo.sing
```

```
inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep=",")
```

```
inflasi.indo.sing$Bulan = factor(inflasi.indo.sing$Bulan, levels = c("Jan-2017", "Feb-2017", "Mar-2017", "Apr-2017", "May-2017", "Jun-2017", "Jul-2017", "Aug-2017", "Sep-2017", "Oct-2017"))
```

```
str(inflasi.indo.sing)
```

Console

```
> library(ggplot2)

> #Membaca data csv dan dimasukkan ke variable inflasi.indo.sing
> inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep="
,")

> inflasi.indo.sing$Bulan = factor(inflasi.indo.sing$Bulan, levels = c("Jan-2017", "F
eb-2017", "Mar-2017", "Apr-2017", "May-2017", "Jun-2017", "Jul-2017", "Aug-2017", "Se
p-2017", "Oct-2017"))

> str(inflasi.indo.sing)
'data.frame':   20 obs. of  3 variables:
 $ Bulan   : Factor w/ 10 levels "Jan-2017","Feb-2017",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ Negara  : Factor w/ 2 levels "Indonesia","Singapura": 1 1 1 1 1 1 1 1 1 1 ...
 $ Inflasi: num  0.0349 0.0383 0.0361 0.0417 0.0433 0.0437 0.0388 0.0382 0.0372 0.035
8 ...
```


Plotting Ulang dengan hasil Factoring

Dengan proses perbaikan pengurutan yang telah kita lakukan sebelumnya, saatnya kita plotting ulang data inflasi kita.

Tugas Praktek

yang ada pada code editor, ini merupakan kumpulan code yang telah kita lakukan dan perbaiki sepanjang bab ini. Tapi pada code ini juga ditambahkan layer geom text yang digunakan untuk menandai label-label nilai pada titik di line chart.

Jalankan langsung code tersebut tanpa perubahan atau tambahan apapun.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable inflasi.indo.sing
```

```
inflasi.indo.sing <- read.csv("https://academy.dqlab.id/dataset/inflasi.csv", sep=",")
```

```
inflasi.indo.sing$Bulan = factor(inflasi.indo.sing$Bulan,
```

```
                                levels = c("Jan-2017", "Feb-2017", "Mar-2017", "Apr-2017", "May-  
2017", "Jun-2017", "Jul-2017", "Aug-2017", "Sep-2017", "Oct-2017"))
```

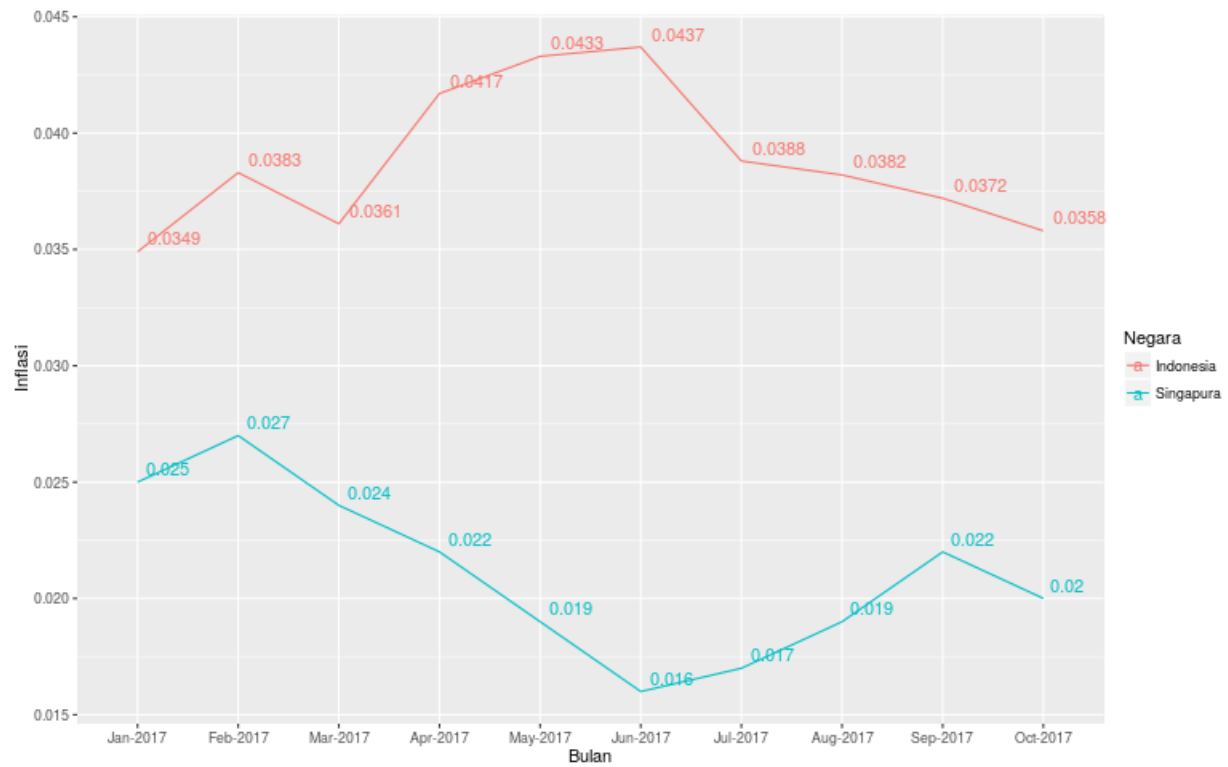
```
#Menambahkan data dan aesthetic mapping
```

```
plot.inflasi <- ggplot(data=inflasi.indo.sing, aes(x = Bulan, y=Inflasi, color=Negara,  
group=Negara))
```

```
#Menambahkan Layer dan labels
```

```
plot.inflasi + geom_line() + geom_text(aes(label=Inflasi),hjust=-0.2, vjust=-0.5)
```

Console



Kesimpulan

Line chart menggunakan geom line telah kita praktekan pada bab untuk menampilkan tren inflasi negara Indonesia dan Singapura untuk periode Januari s/d Oktober 2017.

Terlihat juga bahwa dataset yang kita baca dari file teks ini tidak bisa langsung digunakan begitu saja karena ada urutan data pada kolom Bulan yang perlu diubah.

Dengan treatment data yang baik dan dengan penambahan aesthetic baru – yaitu grouping – kita telah menyelesaikan pembuatan line chart sederhana dengan ggplot.

Klik tombol Next untuk melanjutkan ke bab berikutnya.

Pendahuluan

Dengan praktek yang telah kita lakukan untuk menghasilkan scatter plot dan line chart, kita melangkah ke praktek untuk membuat bar chart dan pie chart.

Bar chart atau grafik batang adalah salah satu tipe visualisasi yang sudah umum digunakan yang sangat baik untuk melakukan perbandingan data multi kategori. Sedangkan pie chart sangat berguna untuk melihat rasio.

Kita kembali akan menggunakan dataset kependudukan Jakarta, namun telah dirubah untuk kepentingan kedua chart ini.

Klik tombol Next untuk melanjutkan ke materi.

Dataset Kependudukan Jakarta versi DQLab

Pada dataset asli kependudukan Jakarta, terdapat kolom-kolom jumlah penduduk per kabupaten yang tersebar ke banyak kolom dengan pembagian umur dan jenis kelamin di nama kolom, seperti terlihat pada gambar berikut.

KEPADATAN (JIWA/KM2)	35-39 Laki-Laki	35-39 Perempuan	40-44 Laki-Laki	40-44 Perempuan	45-49 Laki-Laki	45-49 P
6779	231	235	233	210	171	
1705	84	88	99	88	72	
628	255	238	232	234	212	
3625	199	185	178	176	162	
3084	98	75	73	94	67	
1968	113	112	108	80	66	
1350	166	174	130	165	176	
14584	850	748	749	798	779	
18987	954	920	914	943	871	
14465	752	675	691	691	659	

Kolom-kolom ini akan butuh banyak data treatment atau data preparation sebelum dapat digunakan untuk kepentingan penggunaan grafik kita untuk bab ini.

Oleh sebab itu, DQLab telah melakukan proses tersebut sehingga datanya menjadi seperti bentuk berikut.

D	E	F	G	H	I	J
KECAMATAN	NAMA KELURAHAN	LUAS WILAYAH (KM2)	KEPADATAN (JIWA/KM2)	JENIS KELAMIN	RENTANG UMUR	JUMLAH
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	35-39	5316
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	35-39	3221
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	40-44	3137
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	40-44	6358
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	45-49	3675
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	45-49	3768
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	50-54	7443
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	50-54	3431
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	55-59	3430
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	55-59	6861
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	60-64	3040
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	60-64	2816
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	65-69	5856
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	65-69	2320
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	70-74	2195
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	70-74	4515
ARENG	CENGKARENG BARAT	4.26	16409	Laki-laki	>75	1663
ARENG	CENGKARENG BARAT	4.26	16409	Perempuan	>75	1761
ARENG	CENGKARENG TIMUR	4.18	20725	Laki-laki	25-29	7146

Catatan: Detil transformasi data yang dilakukan akan dibahas di course "Data Preparation / Wrangling with R"

Berikut adalah nama kolom dataset tersebut.

1. **TAHUN:** Tahun
2. **NAMA PROVINSI:** Nama provinsi di DKI Jakarta, dan nilainya hanya ada satu
3. **NAMA KABUPATEN/KOTA:** Nama kabupaten/kota di DKI Jakarta
4. **NAMA KECAMATAN:** Nama kecamatan di DKI Jakarta
5. **NAMA KELURAHAN:** Nama kelurahan di DKI Jakarta
6. **LUAS WILAYAH (KM2):** Luas wilayah (km persegi)
7. **KEPADATAN (JIWA/KM2):** Kepadatan penduduk (jiwa/km2)
8. **JENIS KELAMIN:** Jenis kelamin dari populasi
9. **RENTANG UMUR:** Rentang umur dari populasi
10. **JUMLAH:** Jumlah penduduk

Dengan demikian, dataset ini telah kita ubah strukturnya tanpa menghilangkan informasi aslinya. File csv dari dataset ini dapat didownload di url berikut:

<https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv>

Klik tombol Next untuk melanjutkan ke praktek berikutnya.

Menghasilkan Bar Chart Pertama

Untuk menghasilkan bar chart kita bisa gunakan function **geom_bar**. Function ini akan menghasilkan sebuah layer untuk kita tambahkan ke plot.

Secara default, stat yang digunakan oleh **geom_bar** adalah **count** sehingga bar chart hanya akan menghitung jumlah baris data point per x, dan tidak membutuhkan aesthetic mapping **y**.

Berikut adalah contoh perintah dimana kita menghasilkan bar chart dengan dataset kependudukan DKI Jakarta.

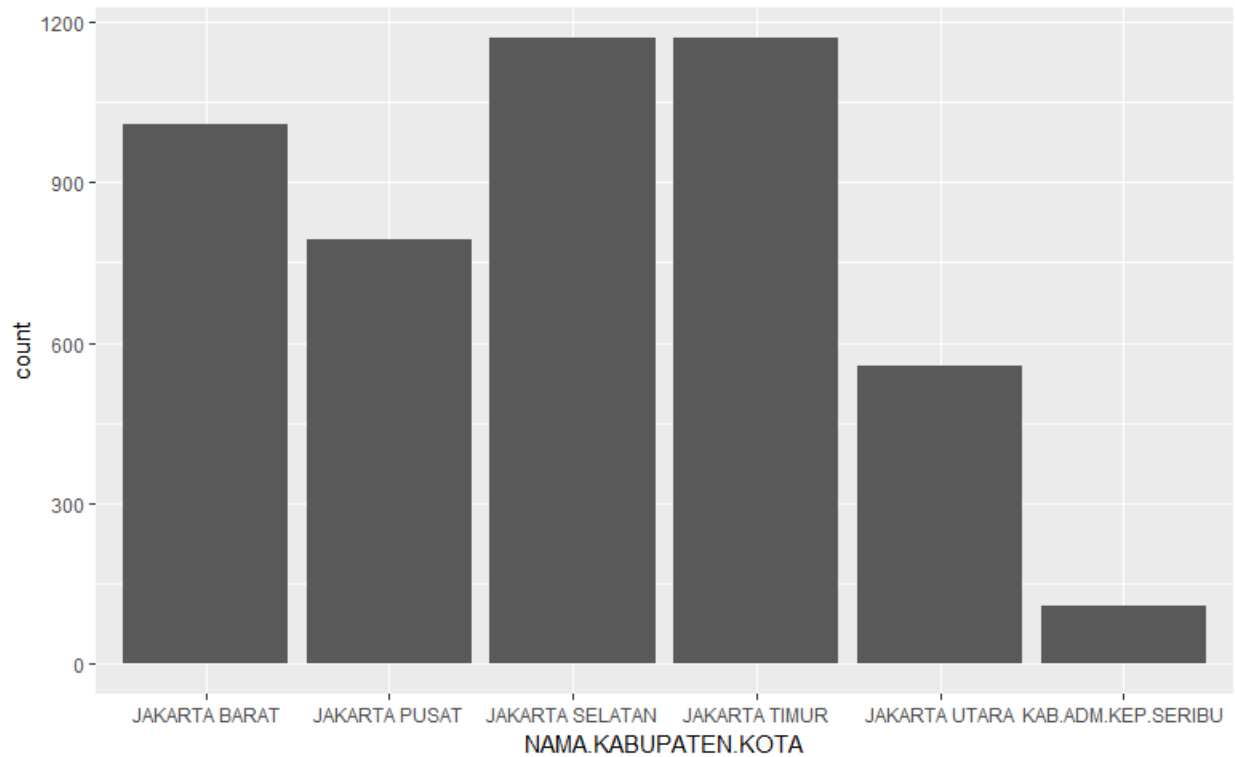
```
plot.dki <- ggplot(data=penduduk.dki, aes(x = NAMA.KABUPATEN.KOTA))  
plot.dki + geom_bar()
```

Ini akan menghasilkan gambar chart berikut. Kelihatan aneh karena jumlah baris data untuk Laki-laki dan Perempuan di dataset kita seimbang. Dan karena stat untuk **geom_bar** adalah **count**, maka yang ditampilkan adalah jumlah baris data. Ini akan kita rubah pada praktek selanjutnya.



Tugas Praktek

Di dalam code editor telah tersedia code untuk membaca dataset baru kita. Lengkapi [...] dengan code yang sesuai sehingga dapat menghasilkan bar chart berikut.



Code Editor

```
library(ggplot2)

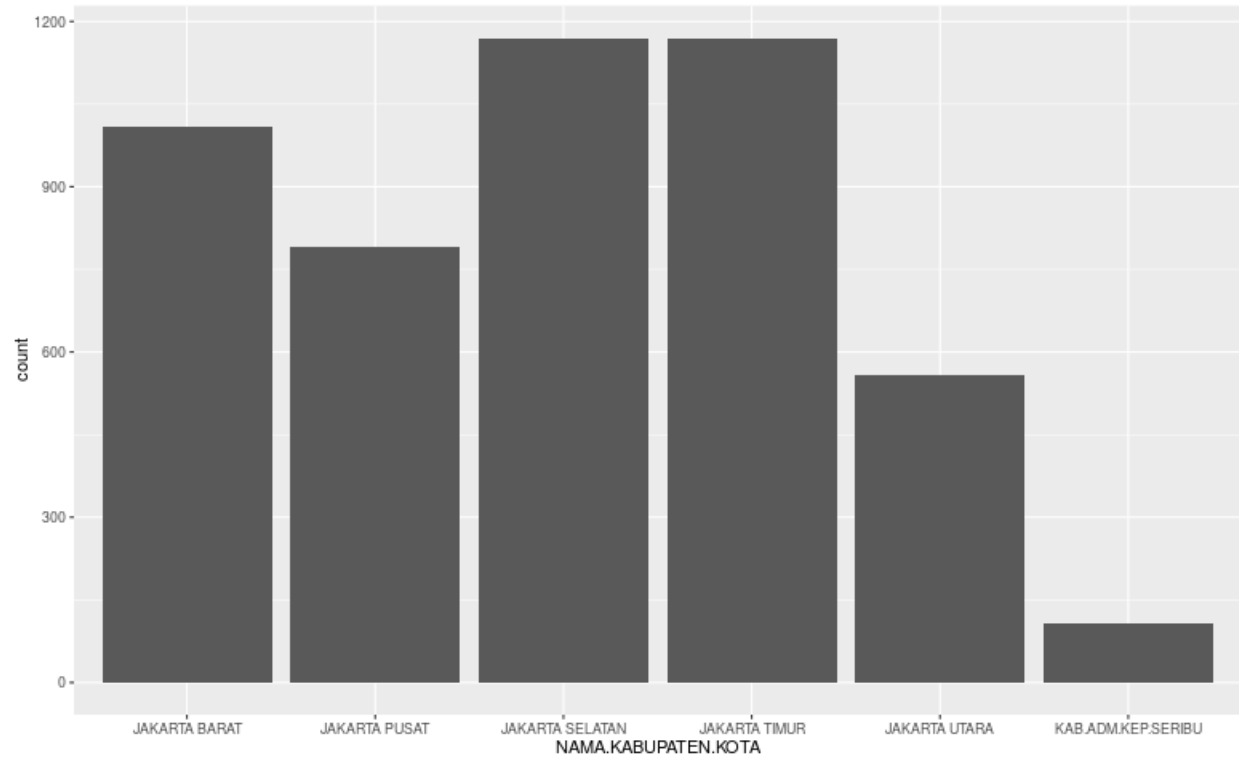
#Membaca data csv dan dimasukkan ke variable penduduk.dki

penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-
dqlab.csv", sep=",")

plot.dki <- ggplot(data=penduduk.dki, aes(x = NAMA.KABUPATEN.KOTA))

plot.dki + geom_bar()
```

Console



Aesthetic Y dan Stat Identity

Pada praktek sebelumnya, bar chart kita masih mirip dengan histogram. Dan itu bukan yang kita inginkan.

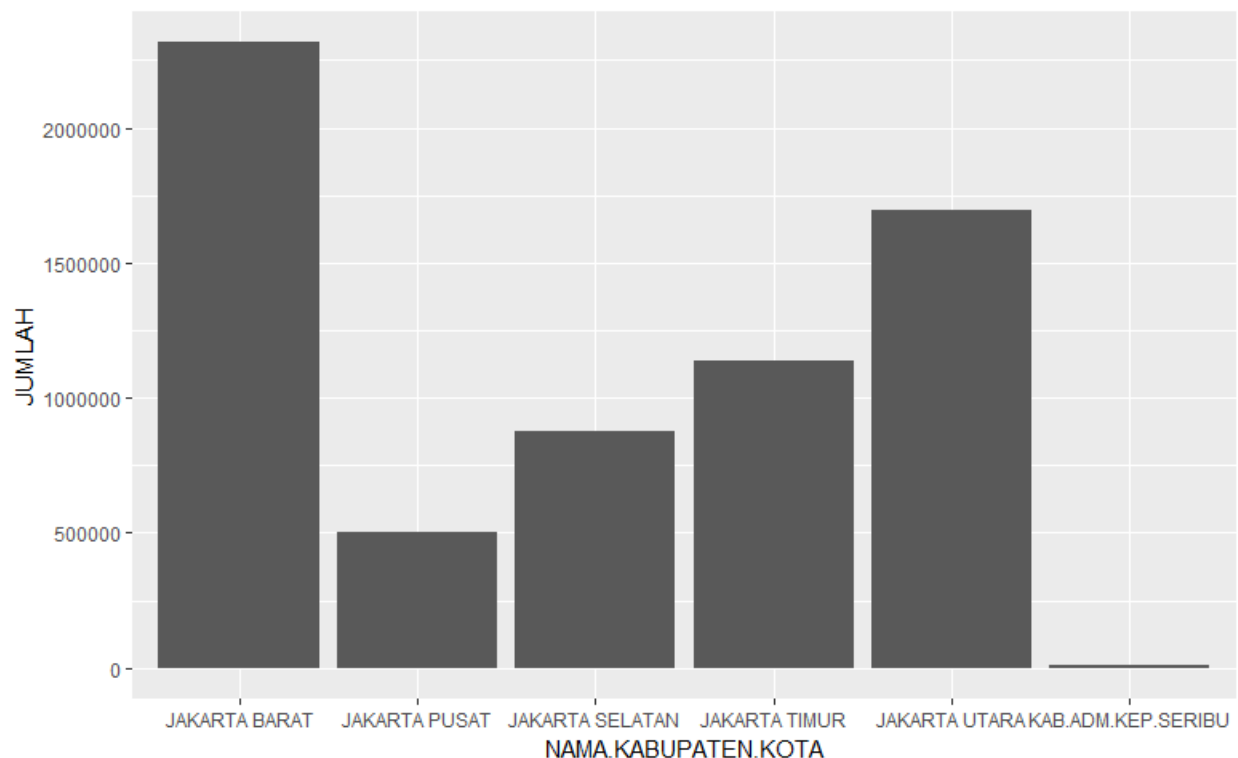
Kita akan memplot bar chart, dimana aesthetic y menggunakan stat penjumlahan (sum) - bukan count - dari data kepadatan penduduk.

Ini bisa dilakukan dengan mengisi dua hal berikut:

- Mengisi aesthetic y dengan kolom grouping yang kita inginkan - yaitu kolom bernama **JUMLAH** - pada plot (ggplot).
`y=JUMLAH`
- Mengisi parameter stat dengan nilai **identity** pada bagian layer (geom_bar)..
`stat="identity"`

Tugas Praktek

Di dalam code editor telah tersedia code yang perlu kita lengkapi. Ganti bagian [...] dengan contoh code yang sesuai dengan penjelasan di atas sehingga dapat menghasilkan bar chart berikut.



Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv", sep=",")
```

```
#Menghasilkan bar chart
```

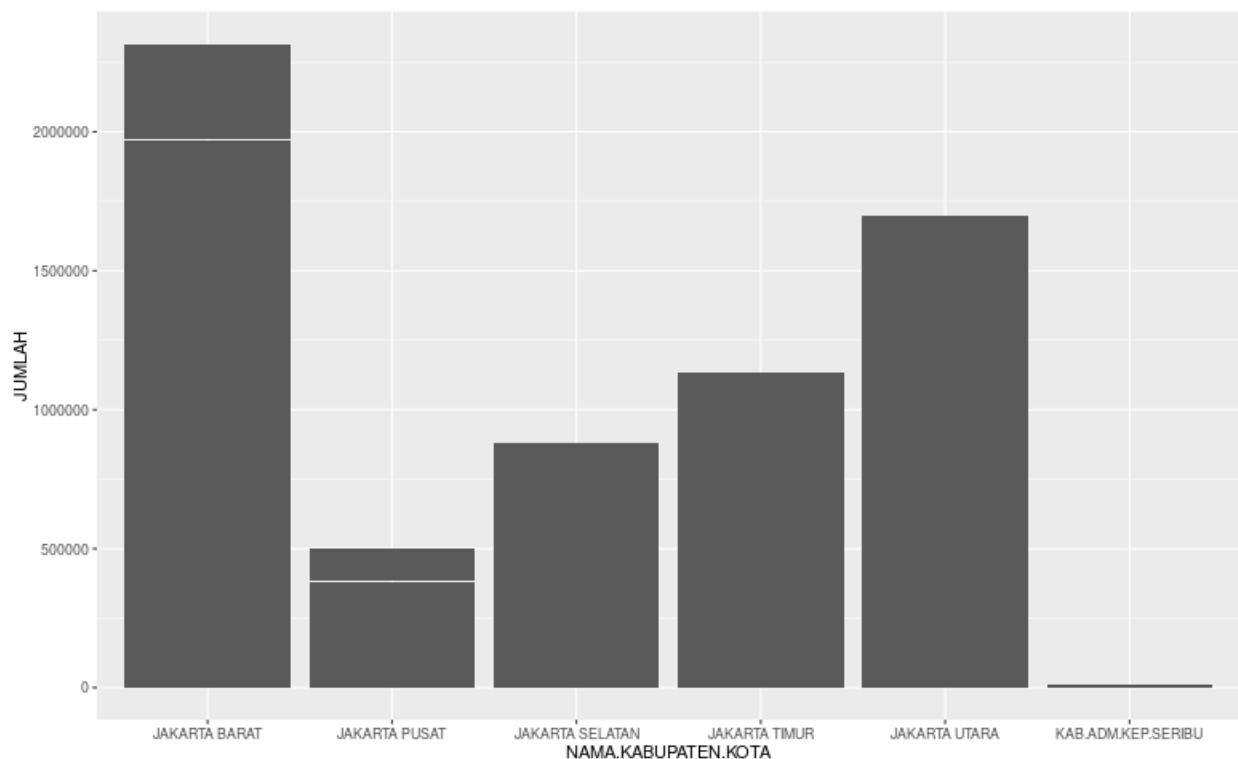
```
#Membuat plot
```

```
plot.dki <- ggplot(data=penduduk.dki, aes(x = NAMA.KABUPATEN.KOTA, y = JUMLAH))
```

```
#Menambahkan layer pada plot
```

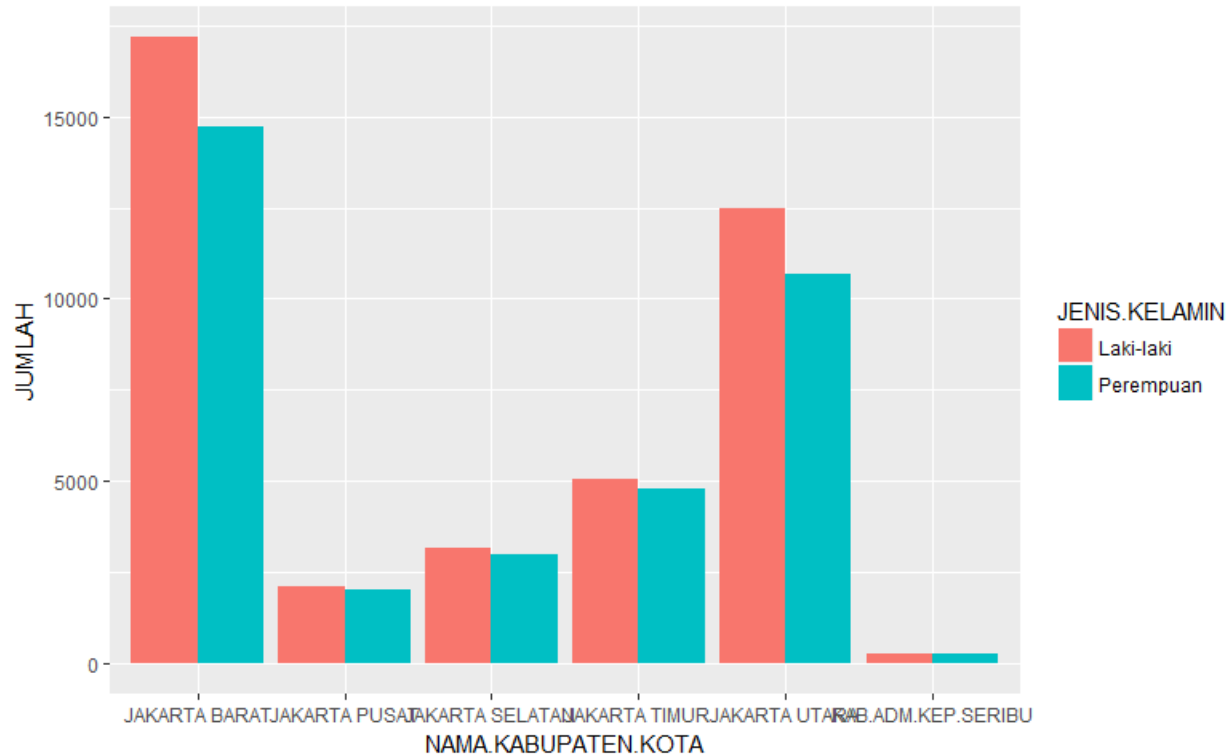
```
plot.dki + geom_bar(stat="identity")
```

Console



Aesthetic Fill dan Position Dodge

Pada subbab sebelumnya, kita melihat efek pergantian nilai **stat** pada `geom_bar` dari `count` (nilai default) menjadi **identity** untuk mendapatkan total penjumlahan dari angka jumlah penduduk. Saatnya kita bisa lebih memperbaiki bar chart kita dengan informasi grouping seperti berikut.



Kita abaikan label yang menumpuk terlebih dahulu, terlihat bar chart ini telah memiliki informasi jumlah penduduk, dengan series berdasarkan jenis kelamin dan kemudian dikategorikan berdasarkan nama kabupaten / kota.

Untuk melakukan hal ini, ada dua hal yang mesti kita lakukan:

- Aesthetic **fill** pada objek plot perlu diberikan input kolom **JENIS.KELAMIN**.
`fill=JENIS.KELAMIN`
- Agar data antara Laki-laki dan Perempuan tidak menjadi satu bar, **position** pada layer kita isi dengan nilai **dodge** – dimana objek satu dengan objek yang lain diposisikan bersisian.
`position="dodge"`

Tugas Praktek

Di dalam code editor telah tersedia code yang perlu kita lengkapi. Ganti bagian [...] dengan code yang sesuai sehingga dapat menghasilkan bar chart seperti terlihat pada bagian Lesson.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv", sep=",")
```

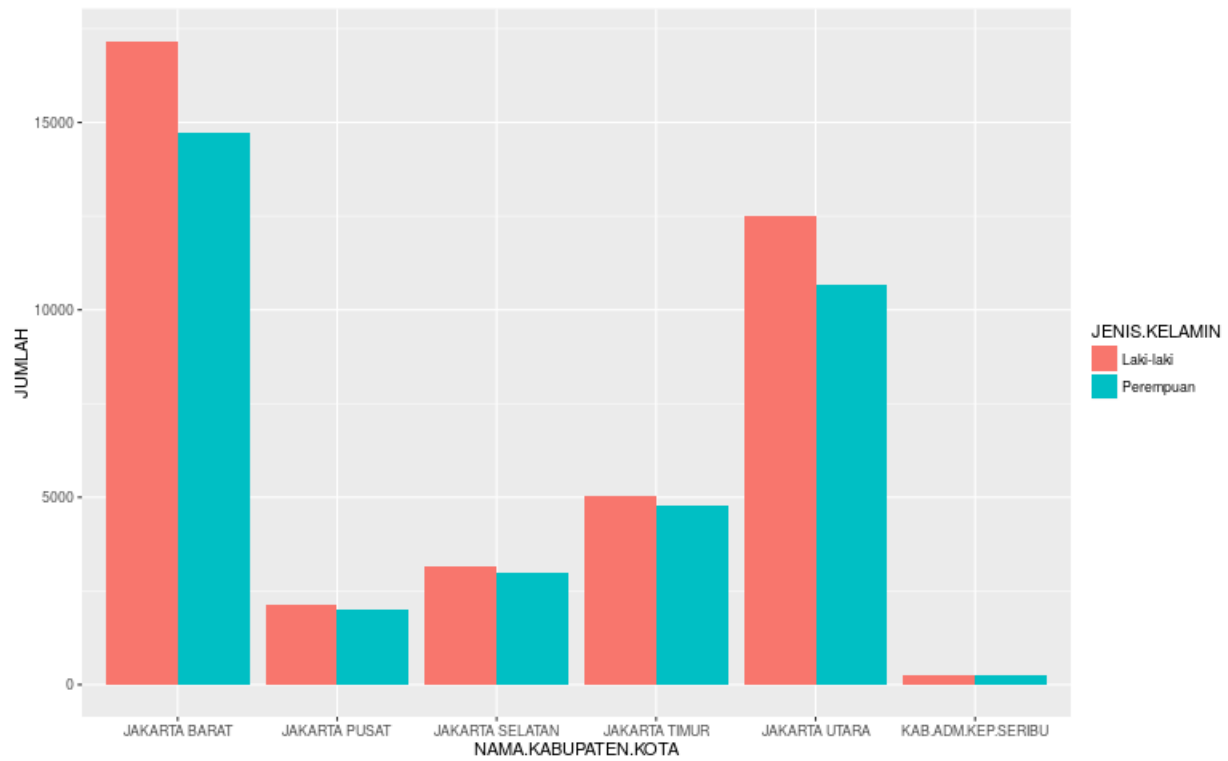
```
#Bagian plot
```

```
plot.dki <- ggplot(data=penduduk.dki, aes(x = NAMA.KABUPATEN.KOTA, y=JUMLAH, fill = JENIS.KELAMIN))
```

```
#Bagian penambahan layer
```

```
plot.dki + geom_bar(stat="identity", position="dodge")
```

Console



Fungsi Aggregate

Data frame yang kita gunakan pada plot, sering sekali harus dilakukan summary terlebih sehingga dapat diproses lebih mudah oleh layer di ggplot.

Fungsi summary yang bisa kita gunakan adalah function **aggregate**, yang memiliki minimal tiga input sebagai berikut:

- x: merupakan list dari kolom data yang ingin kita jadikan summary
- FUN: function summary yang bisa kita gunakan, misalkan sum, mean, dan lain-lain
- by: list berisi pasangan daftar field yang ingin kita gunakan

Untuk lebih jelasnya, kita langsung gunakan contoh sebagai berikut:

```
aggregate(x=list(JUMLAH=penduduk.dki$JUMLAH), FUN=mean, by = list(NAMA.KABUPATEN.KOTA=penduduk.dki$NAMA.KABUPATEN.KOTA, JENIS.KELAMIN=penduduk.dki$JENIS.KELAMIN))
```

Komponen	Deskripsi
aggregate	Fungsi aggregate yang akan menghasilkan data.frame baru
x=list(RATARATA=penduduk.dki\$JUMLAH)	Parameter x diisi dengan list. Kebetulan listnya isinya adalah 1 variable bernama RATARATA yang diambil dari kolom penduduk.dki\$JUMLAH
FUN=mean	Fungsi yang digunakan adalah rata-rata (mean) untuk summary
by = list(NAMA.....KELAMIN)	Dua kolom group by digunakan yaitu NAMA.KABUPATEN.KOTA dan JENIS.KELAMIN

Jika dijalankan perintahnya akan menghasilkan data frame berikut:

	NAMA.KABUPATEN.KOTA	JENIS.KELAMIN	RATARATA
1	JAKARTA BARAT	Laki-laki	2336.01389
2	JAKARTA PUSAT	Laki-laki	683.05556
3	JAKARTA SELATAN	Laki-laki	760.94359
4	JAKARTA TIMUR	Laki-laki	988.83932
5	JAKARTA UTARA	Laki-laki	3087.00358
6	KAB.ADM.KEP.SERIBU	Laki-laki	78.85185
7	JAKARTA BARAT	Perempuan	2259.58333
8	JAKARTA PUSAT	Perempuan	579.71212
9	JAKARTA SELATAN	Perempuan	740.14530
10	JAKARTA TIMUR	Perempuan	953.80513
11	JAKARTA UTARA	Perempuan	2990.49821
12	KAB.ADM.KEP.SERIBU	Perempuan	75.00000

Tugas Praktek

Di dalam code editor telah tersedia code yang perlu kita lengkapi. Ganti bagian [...] dengan function aggregate dari teks lesson di atas, namun dengan fungsi sum. Kemudian nama RATARATA diganti dengan JUMLAH.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv", sep=",")
```

```
#Melakukan agregasi
```

```
aggregate(x=list(JUMLAH=penduduk.dki$JUMLAH), FUN=sum, by = list(NAMA.KABUPATEN.KOTA=penduduk.dki$NAMA.KABUPATEN.KOTA, JENIS.KELAMIN=penduduk.dki$JENIS.KELAMIN))
```

Console

```
> library(ggplot2)

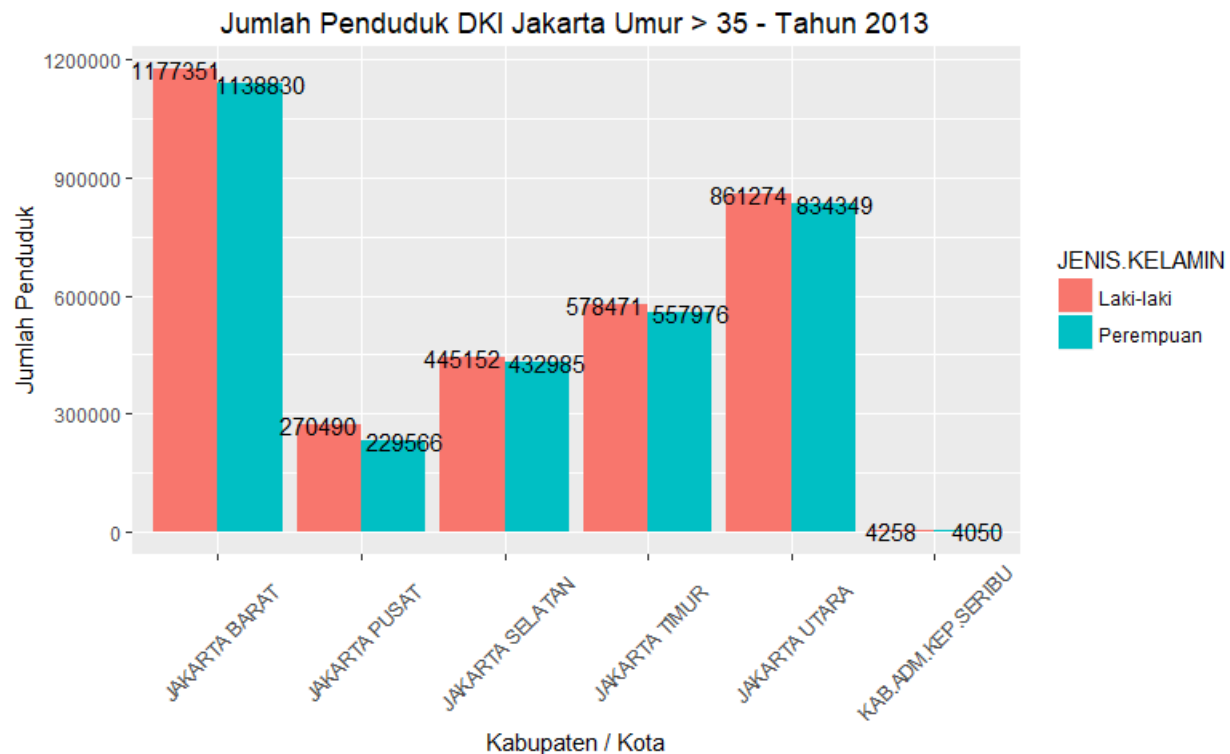
> #Membaca data csv dan dimasukkan ke variable penduduk.dki
> penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv", sep=",")

> #Melakukan agregasi
> aggregate(x=list(JUMLAH=penduduk.dki$JUMLAH), FUN=sum, by = list(NAMA.KABUPATEN.KOTA=penduduk.dki$NAMA.KABUPATEN.KOTA, JENIS.KELAMIN=penduduk.dki$JENIS.KELAMIN))
  NAMA.KABUPATEN.KOTA JENIS.KELAMIN  JUMLAH
1      JAKARTA BARAT      Laki-laki 1177351
```


2	JAKARTA PUSAT	Laki-laki	270490
3	JAKARTA SELATAN	Laki-laki	445152
4	JAKARTA TIMUR	Laki-laki	578471
5	JAKARTA UTARA	Laki-laki	861274
6	KAB.ADM.KEP.SERIBU	Laki-laki	4258
7	JAKARTA BARAT	Perempuan	1138830
8	JAKARTA PUSAT	Perempuan	229566
9	JAKARTA SELATAN	Perempuan	432985
10	JAKARTA TIMUR	Perempuan	557976
11	JAKARTA UTARA	Perempuan	834349
12	KAB.ADM.KEP.SERIBU	Perempuan	4050

"Merapikan" Tampilan Bar Chart

Kalau kita perhatikan grafik bar chart yang telah kita hasilkan. Masih ada beberapa hal yang perlu dirapikan. Selain penamaan label, contoh perapian yang perlu kita lakukan adalah memperbaiki posisi isi data pada sumbu x sehingga tidak menumpuk, sehingga tampilannya sebagai berikut.



Ini dapat menggunakan tambahan komponen tema berikut yang belum dibahas di bab-bab sebelumnya untuk menghasilkan grafik tersebut:

komponen	Keterangan
<code>aggregate(...)</code>	Menggunakan fungsi aggregate yang dibahas pada praktek sebelumnya
<code>theme(...)</code>	Function untuk mengatur teks pada grafik
<code>axis.text.x = element_text(angle=45,vjust = 0.5))</code>	Melakukan rotasi text data (axis.text.x) sebesar 45 derajat ke kiri (angle=45), dan penempatan teks secara vertical di tengah-tengah (vjust=0.5).
<code>plot.title = element_text(hjust=0.5)</code>	Mengatur posisi teks pada title ke tengah
<code>geom_bar(stat="identity", position="dodge")</code>	Grafik bar chart dimana posisi data yang dispesifikasikan di fill, ditempatkan bersisian (position="dodge")

label = JUMLAH	Penempatan kolom JUMLAH sebagai label untuk marking data pada aesthetic mapping
geom_text(position = position_dodge(1.2))	Mengatur posisi text untuk nilai data saat ini di atas bar plot

Tugas Praktek

Di dalam code editor terdapat code lengkap dari seluruh komponen di atas. Anda tinggal pelajari detilnya untuk memahami apa saja yang diperlukan, tidak ada yang perlu dirubah atau ditambahkan.

Code Editor

```
library(ggplot2)

#Membaca data csv dan dimasukkan ke variable penduduk.dki

penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv", sep=",")

#Melakukan agregasi

p <- aggregate(x=list(JUMLAH=penduduk.dki$JUMLAH), FUN=sum, by =
list(NAMA.KABUPATEN.KOTA=penduduk.dki$NAMA.KABUPATEN.KOTA,
JENIS.KELAMIN=penduduk.dki$JENIS.KELAMIN))

#Plot grafik

plot.dki <- ggplot(data=p, aes(x = NAMA.KABUPATEN.KOTA, y=JUMLAH,
fill=JENIS.KELAMIN, label = JUMLAH))

plot.dki <- plot.dki + geom_bar(stat="identity", position="dodge")

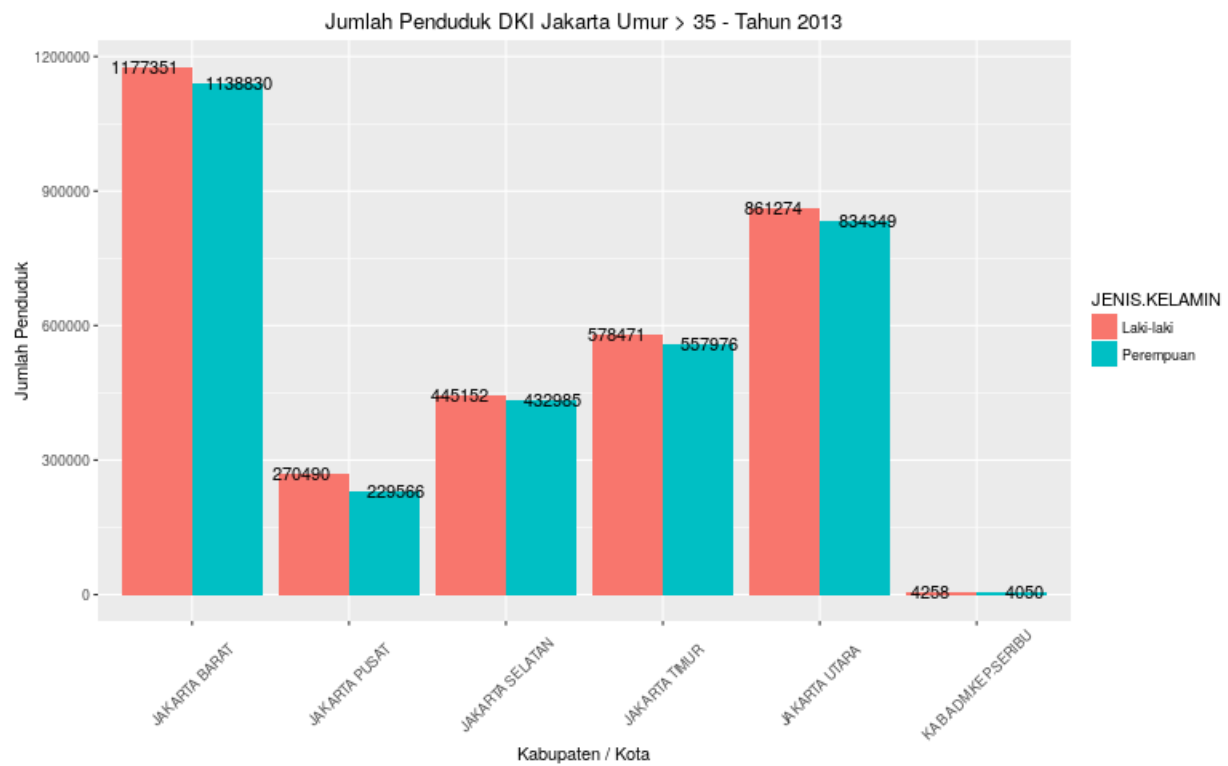
plot.dki <- plot.dki + labs(title="Jumlah Penduduk DKI Jakarta Umur > 35 - Tahun 2013",
x="Kabupaten / Kota", y="Jumlah Penduduk")

plot.dki <- plot.dki + theme(axis.text.x = element_text(angle=45,vjust = 0.5), plot.title =
element_text(hjust=0.5))

plot.dki <- plot.dki + geom_text(position = position_dodge(1.2))

plot.dki
```

Console



Pie Chart dengan Koordinat Polar

Pie chart adalah jenis grafik yang banyak digunakan secara umum untuk menampilkan rasio. Dan uniknya, di ggplot2 tidak ada fungsi khusus yang menghasilkan pie chart ini.

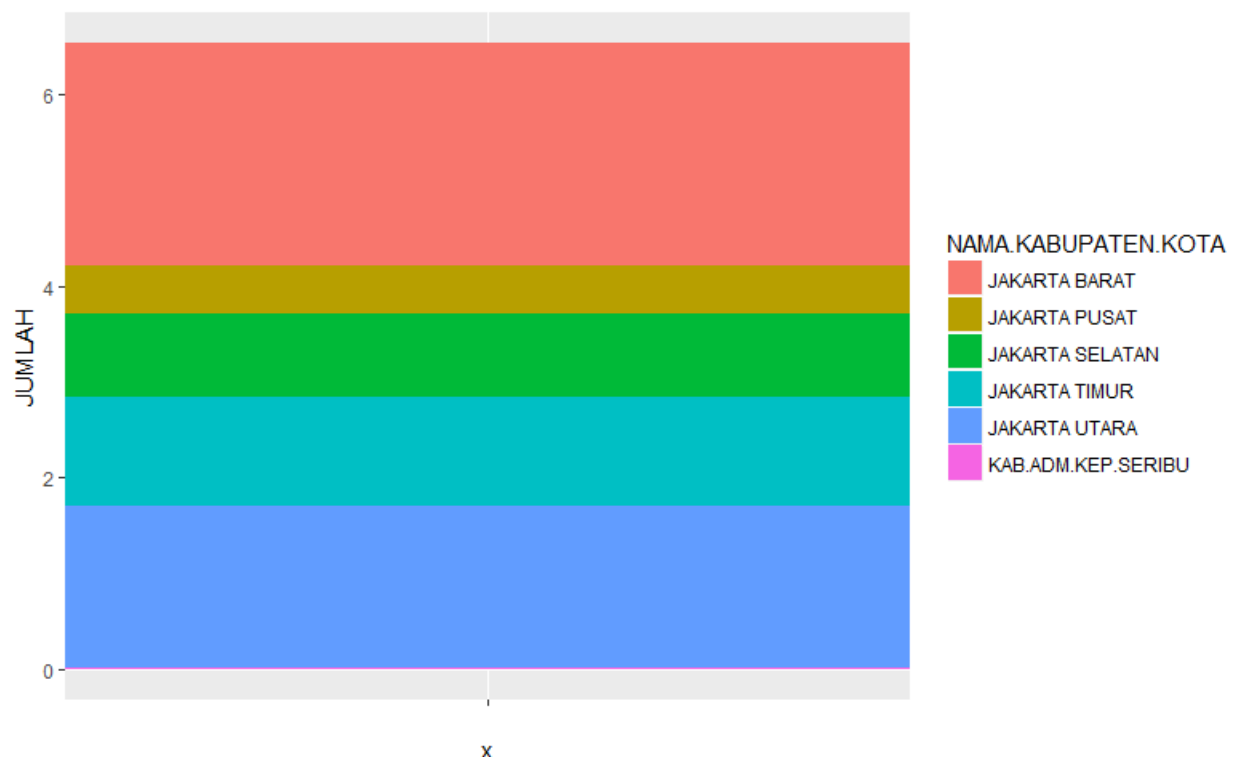
Pie chart dihasilkan dengan proses yang cukup unik, dimana koordinat dari geom bar diubah ke polar.

Untuk lebih jelasnya, kita gunakan contoh langkah demi langkah.

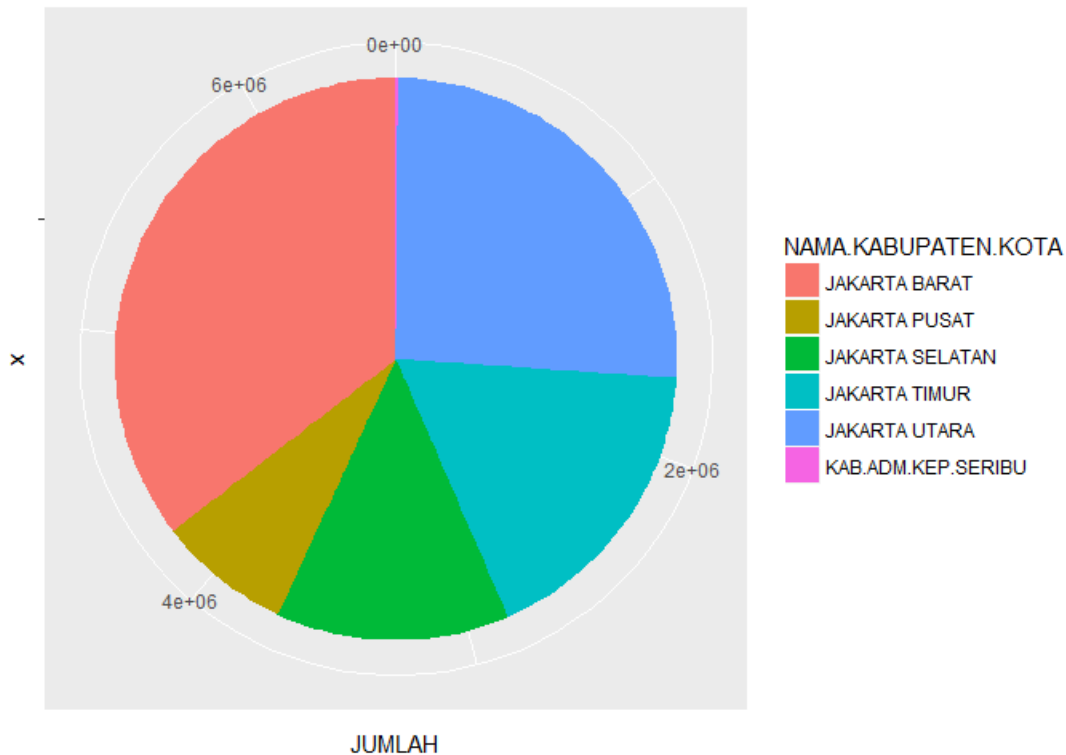
Jika kita memiliki plot data kependudukan di variable p, dan kita ingin membentuk suatu grafik geom bar yang tidak ada pemetaan x-nya, tapi sumbu y diisi dengan JUMLAH, dan pengisian warna menggunakan NAMA.KABUPATEN.KOTA maka perintahnya adalah sebagai berikut.

```
plot.dki <- ggplot(data=p, aes(x="", y=JUMLAH, fill = NAMA.KABUPATEN.KOTA))  
plot.dki + geom_bar(width = 1, stat = "identity")
```

Perintah ini akan menghasilkan grafik berikut.



Nah, bayangkan grafik ini kemudian diputar ke dalam koordinat polar sehingga menjadi seperti berikut.



Perputaran ini membuat bar chart tadi ditransformasikan menjadi pie chart.

Walaupun terlihat tidak sempurna, untuk saat ini kita tidak akan membahas terlalu jauh lagi mengenai pie chart kecuali Anda melakukan tugas praktek untuk menghasilkan grafik di atas.

Tugas Praktek

Di dalam code editor terdapat code lengkap dari seluruh komponen di atas. Anda tinggal pelajari detilnya untuk memahami apa saja yang diperlukan, tidak ada yang perlu dirubah atau ditambahkan.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <- read.csv("https://academy.dqlab.id/dataset/datakependudukandki-dqlab.csv", sep=",")
```

```
#Melakukan agregasi
```

```
p <- aggregate(x=list(JUMLAH=penduduk.dki$JUMLAH), FUN=sum, by = list(NAMA.KABUPATEN.KOTA=penduduk.dki$NAMA.KABUPATEN.KOTA))
```

```
#Plot grafik pie chart
```

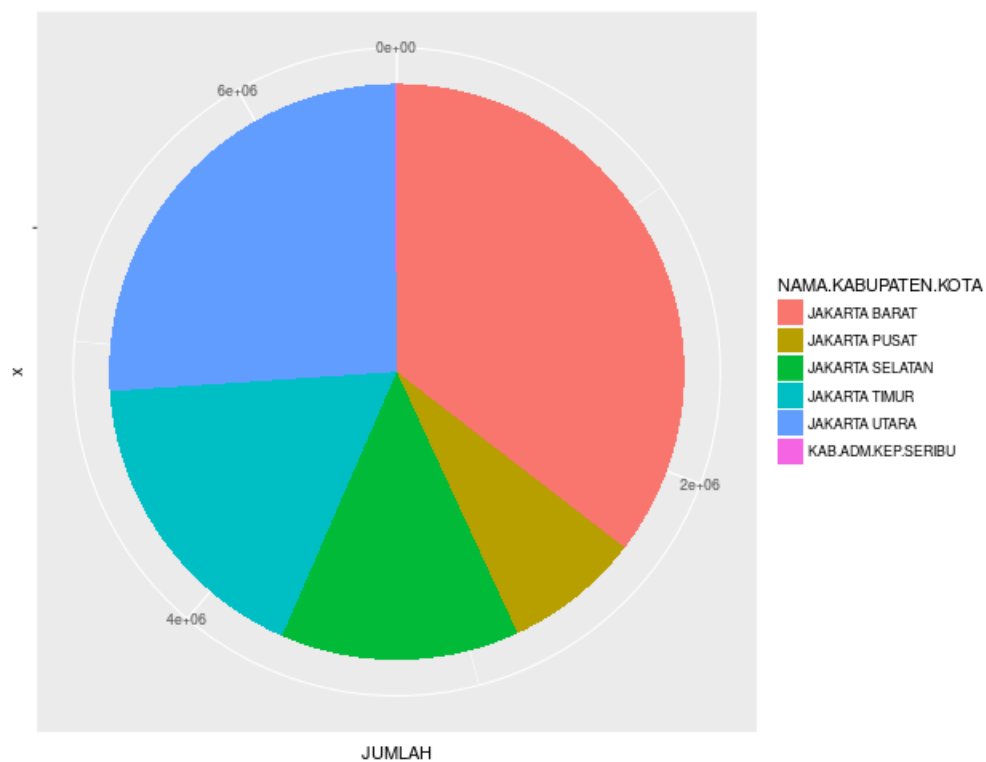
```
plot.dki <- ggplot(data=p, aes(x="", y=JUMLAH, fill = NAMA.KABUPATEN.KOTA))
```

```
plot.dki <- plot.dki + geom_bar(width = 1, stat = "identity")
```

```
plot.dki <- plot.dki + coord_polar("y", start=0)
```

```
plot.dki
```

Console



Kesimpulan

Pada bab ini, kita telah menyelesaikan teori dan praktek untuk membuat bar chart dan pie chart.

Walaupun kelihatan sederhana, untuk menghasilkan bar chart kita perlu mengerti beberapa function dan proses sebagai berikut:

- Dataset kependudukan yang telah kita gunakan pada bab-bab sebelumnya sangat sulit digunakan dan perlu ditransformasikan terlebih dahulu.
- Geom bar secara default hanya memerlukan aesthetic mapping x, karena secara internal geom bar ini menggunakan `stat = "count"`.
- Jika ingin menghitung agregasi sum, lebih baik kita gunakan variable baru yang menampung hasil agregasi dengan function *aggregate*.
- Untuk membuat bar chart dengan *grouping* yang bersisian, kita gunakan aesthetic mapping `fill` dan positioning `dodge`.
- Bar chart dan pie chart adalah grafik dari ggplot2 yang memiliki karakteristik yang hampir sama, kecuali koordinat polarnya.

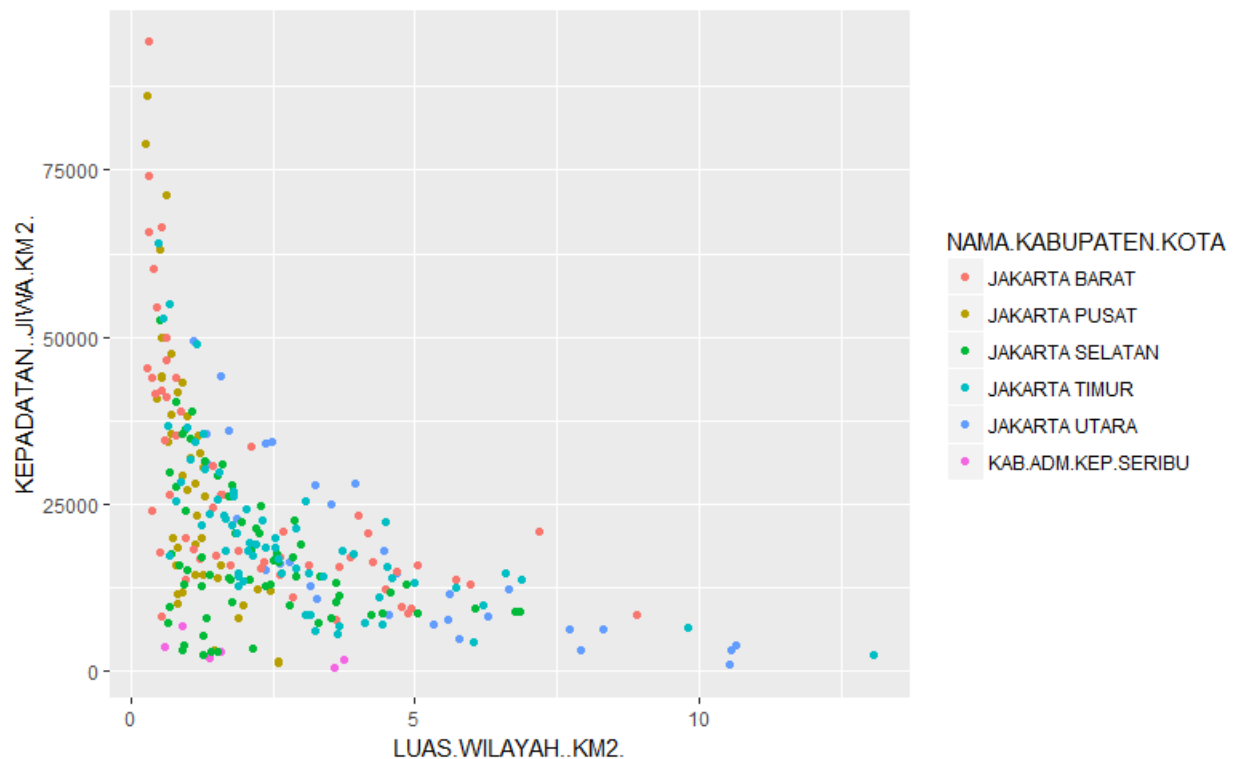
Klik tombol Next untuk melanjutkan ke bab berikutnya.

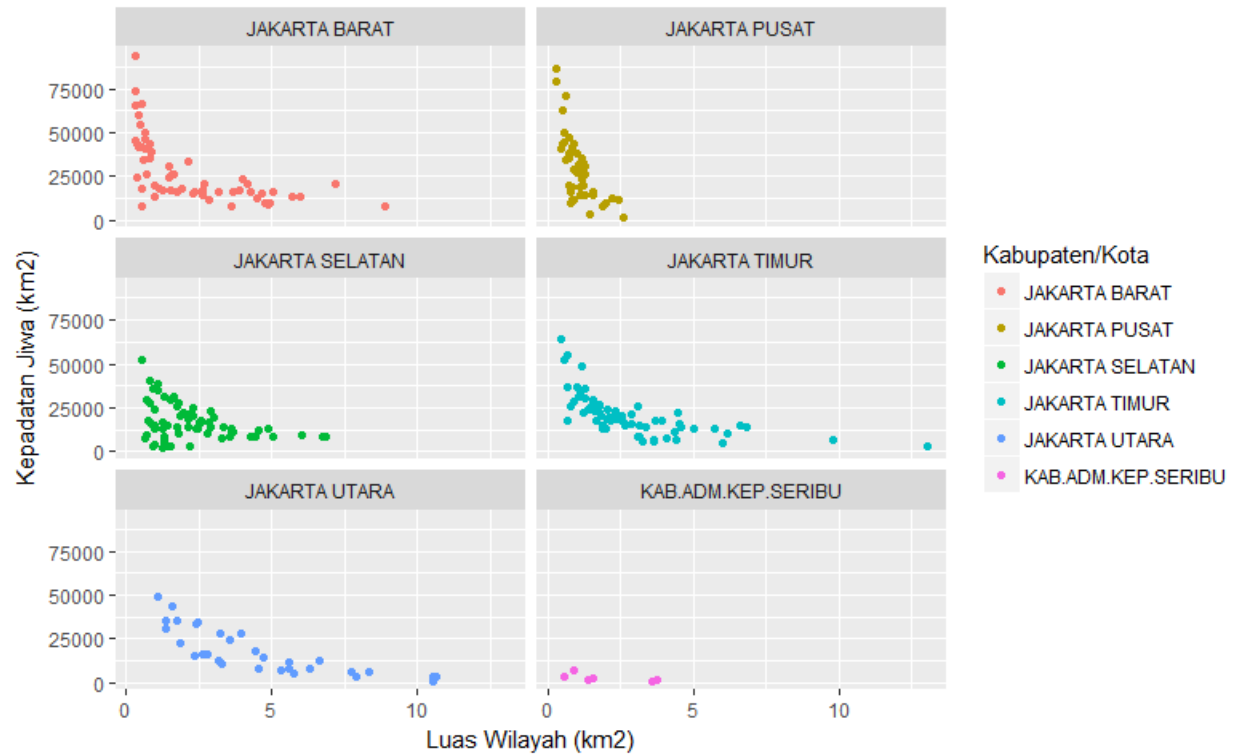
Pendahuluan: Apa dan kenapa Faceting?

Faceting pada ggplot2 adalah:

- proses pembagian dataset menjadi potongan beberapa subset data
- dan menghasilkan visualisasi dari potongan-potongan tersebut
- dengan tujuan agar data lebih mudah dimengerti

Dua gambar berikut adalah contoh faceting dari scatter plot tunggal menjadi 6 potongan scatter plot, dipotong berdasarkan kolom NAMA.KABUPATEN.KOTA.





Bab berikut akan membahas penggunaan praktis dari faceting.
Klik tombol Next untuk melanjutkan ke materi.

Faceting pada Scatter Plot

Ada dua function untuk facet, yaitu `facet_wrap` dan `facet_grid`. Kedua function tersebut fungsinya sama, yaitu membagi subset menjadi layout kotak matrix seperti pada contoh di teks pengenalan.

Functiono `facet_grid` memerlukan dua variable, misalkan provinsi dan kabupaten untuk membagi dataset. Sedangkan function `facet_wrap` hanya memerlukan satu variable, dataset dibagi berdasarkan jumlah kolom yang diinginkan.

Kita akan ambil scatter plot pada bab awal untuk dataset kependudukan dan kita bagi menjadi berdasarkan variable `NAMA.KABUPATEN.KOTA` dan 2 kolom dengan `facet_wrap`.

Konstruksi lengkap untuk contoh tersebut adalah sebagai berikut:

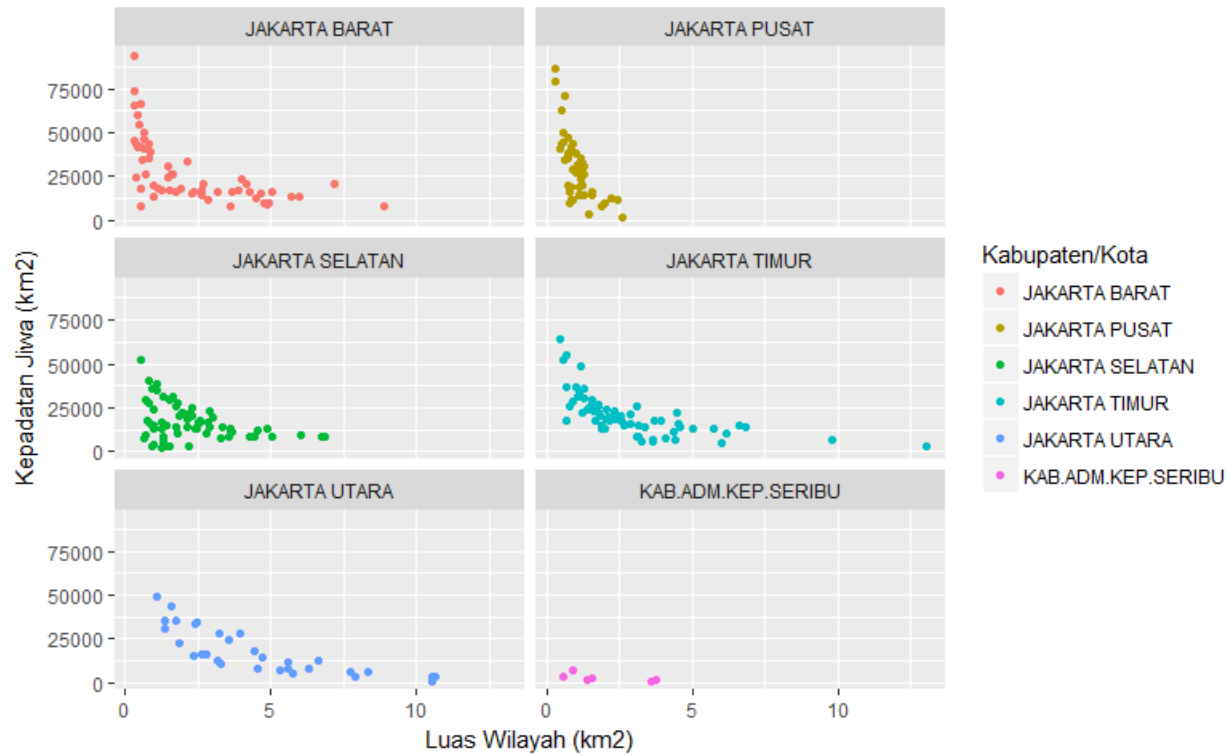
```
facet_wrap( ~ NAMA.KABUPATEN.KOTA, ncol=2)
```

Komponen	Deskripsi
<code>facet_wrap</code>	Fungsi faceting pada ggplot2
<code>~ NAMA.KABUPATEN.KOTA</code>	Facet wrap hanya menggunakan satu variable saja. Dan variable yang digunakan diletakkan di sebelah kanan tanda <code>~</code> .
<code>ncol=2</code>	Jumlah kolom pembagian adalah 2

Tugas Praktek

Di dalam code editor terdapat code lengkap perlu dilengkapi bagian [...] dengan function `facet_wrap` seperti pada teks lesson di atas.

Jika berjalan lancar maka grafik yang dihasilkan terlihat sebagai berikut. Terlihat kepadatan untuk Jakarta Pusat sangat tidak berimbang dibandingkan kabupaten lainnya di DKI Jakarta.



Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <-  
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

```
#Menambahkan data dan aesthetic mapping
```

```
plot.dki <- ggplot(data=penduduk.dki, aes(x = LUAS.WILAYAH..KM2.,  
y=KEPADATAN..JIWA.KM2., color=NAMA.KABUPATEN.KOTA))
```

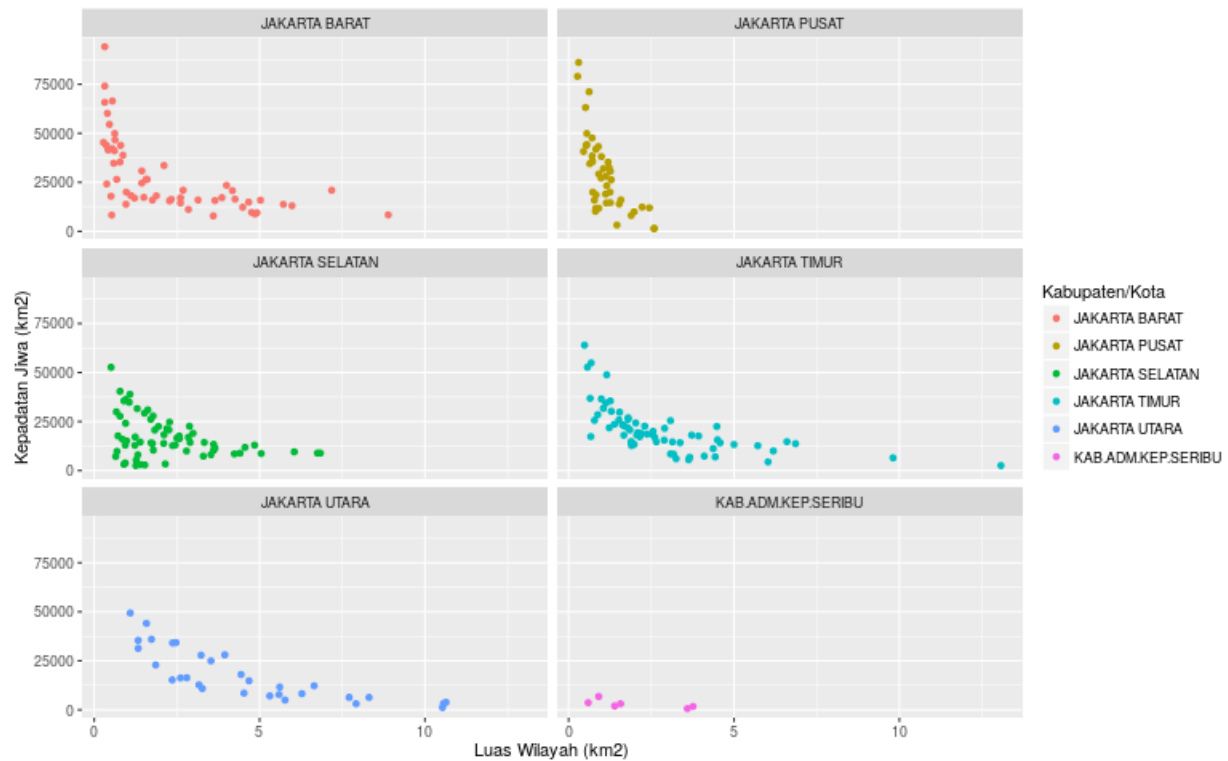
```
#Menambahkan layer
```

```
plot.dki <- plot.dki + layer(geom = "point", stat="identity", position = "identity")
```

```
plot.dki <- plot.dki + labs(x="Luas Wilayah (km2)", y="Kepadatan Jiwa (km2)",
color="Kabupaten/Kota")
```

```
plot.dki + facet_wrap( ~ NAMA.KABUPATEN.KOTA, ncol=2)
```

Console



Faceting pada Histogram

Dengan menggunakan function yang sama kita juga bisa memecah histogram menjadi facet grafik.

Tugas Praktek

Di dalam code editor terdapat code lengkap perlu dilengkapi bagian [...] dengan function `facet_wrap` seperti contoh pada praktek sebelumnya.

Code Editor

```
library(ggplot2)
```

```
#Membaca data csv dan dimasukkan ke variable penduduk.dki
```

```
penduduk.dki <-  
read.csv("https://academy.dqlab.id/dataset/dkikepadatankelurahan2013.csv", sep=",")
```

```
#Menambahkan data dan aesthetic mapping
```

```
plot.dki <- ggplot(data=penduduk.dki, aes(x=KEPADATAN..JIWA.KM2.,  
fill=NAMA.KABUPATEN.KOTA))
```

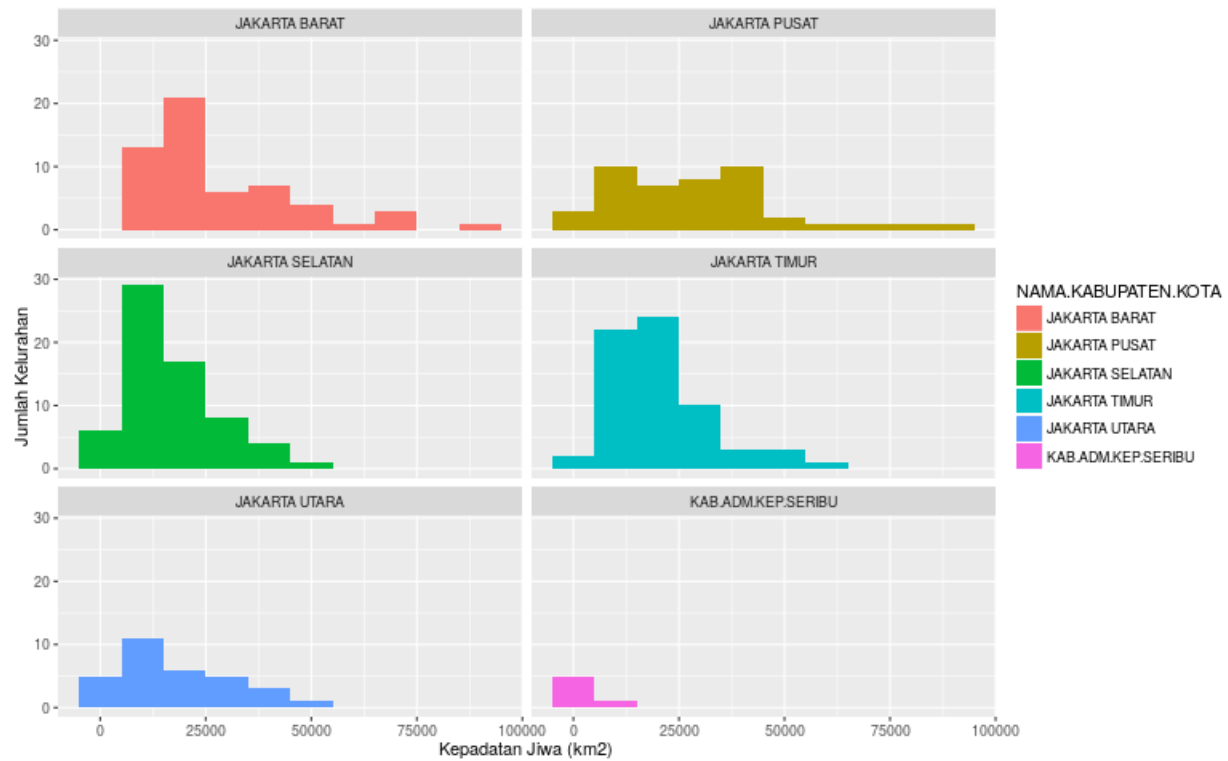
```
#Menambahkan layer
```

```
plot.dki <- plot.dki + geom_histogram(binwidth=10000)
```

```
plot.dki <- plot.dki + labs(x="Kepadatan Jiwa (km2)", y="Jumlah Kelurahan",  
color="Kabupaten/Kota")
```

```
plot.dki + facet_wrap( ~ NAMA.KABUPATEN.KOTA, ncol=2)
```

Console



Kesimpulan

Selamat, Anda telah menyelesaikan seluruh teori pengantar dan praktek mengenai visualisasi dengan ggplot2.

Sepanjang course ini Anda telah mempelajari fundamental data visualisasi ggplot2 dengan pembelajaran kunci sebagai berikut:

- Konsep rancangan ggplot2 sebagai grammar of graphics yang memecah satu grafik menjadi komponen-komponen visual.
- Komponen plot sebagai kanvas dasar sebelum bisa menghasilkan berbagai macam grafik.
- Data dan aesthetic mapping sebagai isi untuk plot dan grafik.
- Layer sebagai komponen grafik itu sendiri, yang terdiri dari geom, stat dan position.
- Penggunaan layer dan transformasi data yang diperlukan untuk menghasilkan scatter plot, histogram, line chart, bar chart, dan pie chart.
- Faceting untuk memecah grafik sehingga lebih mudah dianalisa.

Dengan dasar ini kita sudah dapat menghasilkan berbagai grafik yang dapat bercerita mengenai informasi dari data yang dikandungnya.

What Next?

Topik ggplot2 adalah topik yang sangat luas. Banyak jenis grafik dan function yang belum kita bahas. Untuk fitur-fitur lain di ggplot2 kita akan mempelajari melalui project-project yang ada di dalam track, ataupun melalui track ggplot tersendiri.

Stay tuned!