



# Modul 2

R Fundamental for Data Science

# Pendahuluan

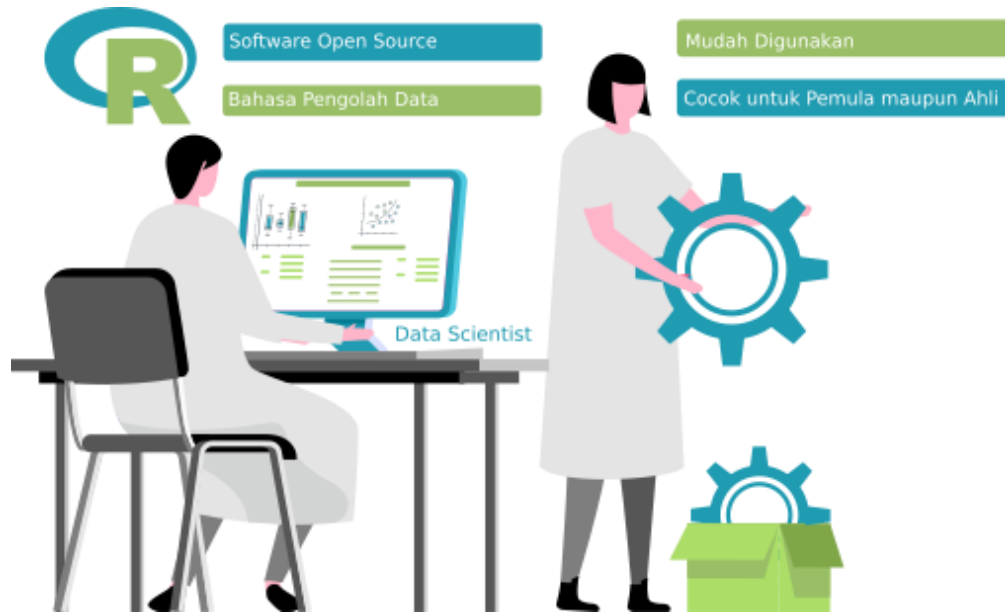


Course "R Fundamental for Data Science" ini adalah course pembuka DQLab untuk topik pengolahan data menggunakan R.

Dengan durasi singkat, 15 menit. Anda akan mempelajari dan mendapatkan hal-hal berikut pada course ini

- ☐ Mengerti **apa dan kenapa R** digunakan oleh para data scientist dan dicari oleh perusahaan besar dunia.
- ☐ Menguasai dasar bahasa R seperti penggunaan **teks, angka, dan rumus**.
- ☐ Menguasai penggunaan **tipe data** fundamental R.
- ☐ Mengerti bagaimana **membaca file Excel**.
- ☐ Mengenali bagaimana **grafik** dihasilkan di R.
- ☐ **Sertifikat pencapaian** dari PHI-Integration dan Universitas Multimedia Nusantara (Kompas Group).

## Jadi, apa itu R?



R adalah nama sebuah bahasa pemrograman sekaligus software untuk pengolahan data dan grafik. R sangat populer saat ini karena memiliki jumlah fitur yang sangat besar - mencapai puluhan ribu fitur.

Mulai dari membaca file teks, membaca database, menghasilkan berbagai grafik, menghasilkan dashboard yang menarik, sampai ke penggunaan **machine learning** - semuanya tersedia di R.

Selain itu, R bersifat gratis dan **open source**. Artinya, dengan R kita tidak perlu biaya lisensi macam-macam untuk menggunakannya secara bebas dan luas.

### Data Scientist dan R

Dunia saat ini penuh dengan data, akibat meluasnya penggunaan aplikasi smartphone dan juga pemakaian sistem pada seluruh perusahaan. Namun data-data ini belum tentu menjadi informasi yang dibutuhkan oleh organisasi dan bisnis bila tidak bisa diolah dengan baik.

Sekarang muncul sebuah disiplin ilmu untuk menjawab permasalahan tersebut, ilmu yang mempelajari bagaimana mengolah data menjadi informasi yang berguna, yaitu **data science**.

Seseorang yang bekerja dengan dibekali ilmu **data science** dinamakan **data scientist** - saat ini merupakan salah satu profesi paling **hot** menurut LinkedIn.

Untuk membantu pekerjaannya, data scientist perlu dibekali dengan aplikasi yang bagus - R dengan segala kekayaan adalah jawabannya.

# Kenapa Data Scientist menggunakan R?



Apa sebenarnya yang membuat para data scientist tertarik belajar dan menggunakan R?

Ada empat alasan utama, yaitu:

- ❑ **Lebih Mudah Dipelajari (Easier):** R relatif lebih mudah dipelajari dibandingkan dengan bahasa lain, seperti Java, C#, Javascript, dan lain-lain.
- ❑ **Lebih Cepat (Faster):** Banyak fungsi R memberikan hasil jauh lebih cepat dibandingkan dengan aplikasi lain.  
Contoh: R dapat menghasilkan berbagai visualisasi yang menarik dalam waktu singkat, sehingga *data scientist* dapat jauh lebih produktif dalam memahami data dan menghasilkan informasi.
- ❑ **Lebih kaya fitur (Richer):** Dengan puluhan ribu fitur yang terus berkembang, hampir semua permasalahan data dapat dijawab oleh R. Sebagai contoh, untuk mengatasi permasalahan optimasi stok di e-commerce, R memiliki fungsi menghasilkan rekomendasi *product packaging*.
- ❑ **Telah terbukti (Proven):** R sudah digunakan oleh banyak data scientist perusahaan besar seperti Anz, Uber, dan Facebook dan memberikan solusi riil. Tidak heran jika akhirnya dari kisah sukses ini, banyak lowongan *data scientist* mencantumkan R sebagai syarat keterampilan yang harus dimiliki.

# Code Pertama, Hello World!

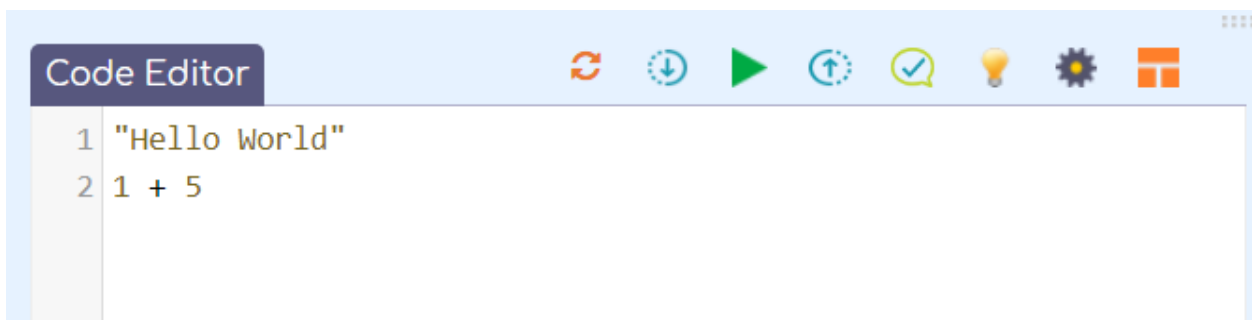
Mari kita langsung praktek dengan halaman **Live Code Editor** ini:

1. Cobalah ketik teks atau code **"Hello World"** pada bagian panel Code Editor persis di bawah bagian lesson ini.
2. Jalankan dengan menekan tombol .
3. Jika berjalan dengan lancar, Anda akan mendapatkan hasil seperti pada gambar berikut.

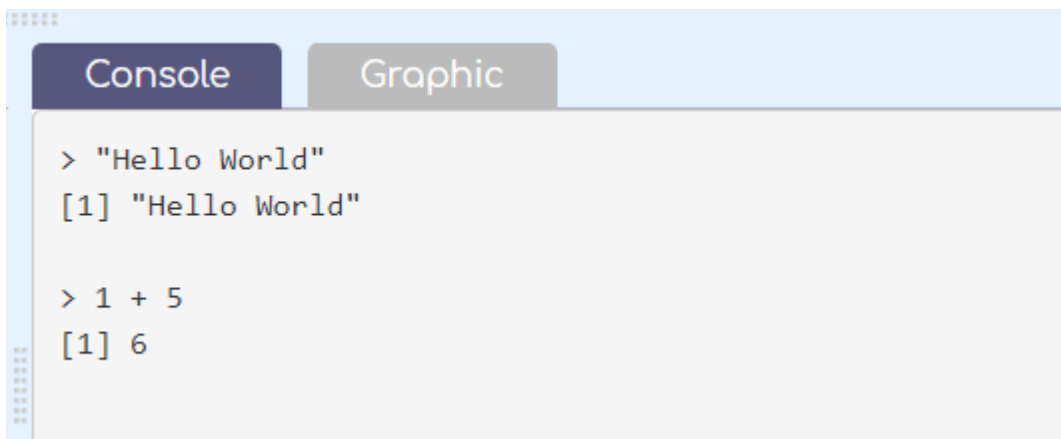


Nomor 3 adalah output yang Anda akan dapatkan dimana teks "Hello World" dikirimkan ke sistem DQLab dan dijalankan.


Sekarang mari kita tambahkan perintah berupa perhitungan matematika, ketik **1+5** di bawah "Hello World".



Jalankan code tersebut, dan jika berhasil maka akan muncul hasil berikut di Console.



Terlihat perhitungan  $1 + 5$  menghasilkan angka 6. Nah, begitulah kira-kira R menjalankan programnya.

Tugas: Kirimkan hasil ini dan lanjutkan pelajaran dengan mengklik tombol  .

# Teks, Angka dan Rumus Perhitungan

R merupakan bahasa yang berisi berbagai perintah, dari perintah melakukan perhitungan matematika sederhana sampai menghasilkan grafik.

Perintah ini bisa sangat singkat, hanya berisi angka atau teks saja. Sebagai contoh, cobalah ketik angka 9 dan teks "Budi" berikut pada Code Editor.

```
9  
"Budi"
```

Klik tombol  dan pada panel **Console** akan muncul hasil berikut.

```
> 9  
[1] 9  
> "Budi"  
[1] "Budi"
```

Catatan: Tanda > tidak perlu diketik

| Elemen ekspresi | Keterangan  |
|-----------------|---|
| >               | Prompt dari R, adalah tanda dari R untuk menerima input perintah  |
| 9               | Menampilkan angka 9   |
| [1]             | Menunjukkan posisi urutan dari hasil output pertama. Kebetulan disini output hanya ada satu item, yaitu angka 9 – jadi posisinya otomatis adalah 1. |
| "Budi"          | Menampilkan teks Budi   |

|     |   |
|-----|---|
| [1] | Menunjukkan posisi urutan dari hasil output kedua. Kebetulan disini output hanya ada satu item, yaitu teks "Budi" – jadi posisinya otomatis adalah 1. |
|-----|---|

Cobalah lanjutkan coding di atas dengan perintah.

```
9 * 3
```

Klik tombol  dan pada panel **Console** akan muncul hasil seperti berikut.

```
> 9 * 3
```

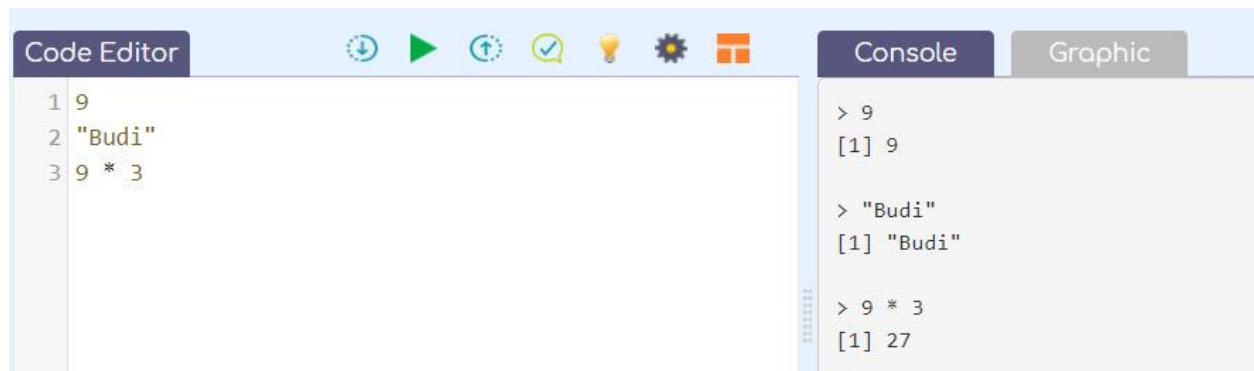
```
[1] 27
```

Berikut adalah penjelasan elemen dari output tersebut

| Elemen ekspresi | Keterangan   |
|-----------------|--|
| >               | Prompt dari R  |
| 9 * 3           | Perintah untuk melakukan perkalian angka 9 dan 3   |
| [1]             | Menunjukkan posisi urutan dari hasil output pertama. Kebetulan disini output hanya ada satu item, yaitu angka 27 – jadi posisinya otomatis adalah 1. |
| 27              | Hasil output.  |



Tampilan hasil terakhir harusnya terlihat sebagai berikut. Klik tombol **Submit Code** dan lanjutkan ke bagian berikutnya.



The screenshot shows a web-based code editor interface. On the left, the 'Code Editor' tab is active, displaying three lines of R code: `1 9`, `2 "Budi"`, and `3 9 * 3`. Above the code editor is a toolbar with icons for running, undo, redo, and other functions. On the right, the 'Console' tab is active, showing the output of the code execution. The output consists of three lines: `> 9` followed by `[1] 9`, `> "Budi"` followed by `[1] "Budi"`, and `> 9 * 3` followed by `[1] 27`. The 'Graphic' tab is also visible but inactive.

```
1 9
2 "Budi"
3 9 * 3
```

```
> 9
[1] 9

> "Budi"
[1] "Budi"

> 9 * 3
[1] 27
```

# Menampilkan dengan Fungsi Print

Pada praktek sebelumnya kita bisa menampilkan teks dan angka dengan menuliskannya secara langsung, tapi akan lebih baik apabila kita menggunakan fungsi bernama **print**.

Mari kita ketik dua perintah print berikut pada code editor.

```
print("Hello World")  
print(3 + 4)
```

Jika dijalankan akan muncul hasil berikut.

```
> print("Hello World")  
[1] "Hello World"  
  
> print(3 + 4)  
[1] 7
```

Dengan demikian, hasilnya akan sama apabila kita langsung mengetikkan teks ataupun formula tersebut. Klik tombol Submit Code untuk melanjutkan ke bagian berikutnya.

# Huruf Besar, Huruf Kecil dan Format Angka

Huruf besar dan huruf kecil sangat perlu diperhatikan pada bahasa pemrograman R, atau dengan kata lain R sangat *case sensitive*. Sebagai contoh: "Budi" dan "BUDI" adalah dua teks yang berbeda.

Selain itu, pada saat mengirimkan jawaban via live code editor di DQLab, kadang format angka - termasuk di penulisan tanggal - sangat berpengaruh terhadap penilaian apakah jawaban benar atau tidak. Sebagai contoh:

- ❑ 01 dan 1 adalah dua angka yang dianggap berbeda pada saat pengiriman jawaban.
- ❑ "12-01-1987" dan "12-1-1987" adalah hal berbeda pada saat mengirimkan jawaban.

Mari kita praktekan dengan mengetikkan enam perintah berikut di code editor dan klik "Submit Code" untuk mengirimkan jawaban.

```
01
1
"01-01-1980"
"1-1-1980"
"Budi"
"BUDI"
```


# Function

Kali ini kita akan belajar apa yang dinamakan *function*. Function adalah perintah R yang memiliki dan menerima beberapa nilai teks maupun angka sebagai parameternya.

Mari kita lihat satu fuction yang akan kita sering kita gunakan, yaitu **c**. Fungsi c ini digunakan untuk membuat urutan angka maupun teks.

Ketikkan perintah `c(10:40)` berikut pada bagian Code Editor. Perintah ini berguna untuk membuat rangkaian angka dari 10 s/d 40.

```
c(10:40)
```

Klik tombol  dan pada panel **Console** akan muncul tambahan output sebagai berikut.

```
> c(10:40)
[1] 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 [26] 3
5 36 37 38 39 40
```

Catatan: Tanda > tidak perlu diketik

Berikut adalah penjelasan hasil dari contoh di atas.

| Elemen ekspresi | Keterangan   |
|-----------------|--|
| >               | Prompt dari R  |
| c(10:40)        | Ekspresi untuk membuat rangkaian angka 10 sampai dengan 40   |
| [1]             | Menunjukkan posisi urutan (indeks) dari angka pertama dari rangkaian output di baris ke 1. Dalam hal ini angka pertama, yaitu 10 - menempati posisi urutan ke-1 dari keseluruhan output. |

|  |  |
|--|--|
| 10 11 12<br>13 14 15<br>16 17 18<br>19 20 21<br>22 23 24<br>25 26 27<br>28 29 30<br>31 32 33<br>34 | Hasil output angka yang ditampilkan pada baris pertama.  |
| [26]   | Menunjukkan posisi indeks dari angka pertama dari rangkaian output di baris ke 2. Dalam hal ini angka pertama, yaitu 35 – menempati posisi urutan ke-26 dari keseluruhan output. |
| 35 36 37<br>38 39 40   | Hasil output angka yang ditampilkan pada baris kedua.  |

Sekarang ubahlah perintah di atas merubah huruf c menjadi C (dari huruf kecil menjadi huruf besar).

### C(10:40)

Klik tombol Run dan pada panel **R Console** akan muncul tambahan output sebagai berikut.

```
> C(10:40)
object not interpretable as a factor
```

Ini artinya perintah tersebut tidak dapat dijalankan dan mengakibatkan error atau kesalahan. Huruf c dan C disini memiliki arti yang sangat berbeda.

### Tugas Praktek

Hapus seluruh perintah pada Code Editor, kemudian buat suatu rangkaian angka 5 s/d 10 pada Code Editor dengan menggunakan *function c*.



Klik tombol (submit) untuk mengirimkan code ke sistem DQLab dan melakukan scoring apakah code Anda sudah benar atau tidak.

# Variable


Angka-angka yang kita gunakan dapat juga disimpan dengan sesuatu yang dinamakan **variable**. Variable memiliki **nama** yang dapat kita definisikan dan gunakan untuk mengambil nilainya kembali. Untuk memahami hal ini, cobalah ketik perintah berikut pada Code Editor.

```
budi_berat_kg <- 68

santi_berat_kg <- 54.5

budi_berat_kg

santi_berat_kg
```

Jalankan keempat perintah di atas dengan mengklik  dan hasilnya adalah sebagai berikut.

```
> budi_berat_kg <- 68

> santi_berat_kg <- 54.5

> budi_berat_kg
[1] 68

> santi_berat_kg
[1] 54.5
```

## Penjelasan Hasil


| Elemen ekspresi     | Keterangan  |
|---------------------|---|
| budi_berat_kg <- 68 | Memasukkan data berupa angka 68 ke dalam variable bernama <b>budi_berat_kg</b> . Maksud variable tersebut dinamakan demikian untuk menyatakan bahwa angka yang dimasukkan mewakili berat badan seorang bernama Budi dalam satuan kilogram – atau singkatnya berat Budi 68 kg. |

|   |  |
|---|--|
|   | <p>Nilai dimasukkan dengan bantuan operator <code>&lt;-</code>. Operator ini disebut <b>assignment</b> operator.</p> <p>Perhatikan bahwa perintah ini juga tidak mengeluarkan output apapun di bawahnya.</p> |
| santi_berat_kg<br><code>&lt;- 55</code> | <p>Ini sama tujuannya dengan perintah di atas, memasukkan data berupa angka 55 ke dalam variable bernama <b>santi_berat_kg</b>.</p> <p>Informasi variable tersebut adalah berat Santi 55 kg.</p>             |
| budi_berat_kg                           | Menampilkan isi data dari nama variable <b>budi_berat_kg</b> .   |
| [1] 68                                  | Menampilkan angka 68 – yang merupakan isi dari variable <b>budi_berat_kg</b> .   |
| santi_berat_kg                          | Menampilkan isi data dari nama variable <b>santi_berat_kg</b> .  |
| [1] 55                                  | Menampilkan angka 55 – yang merupakan isi dari variable <b>santi_berat_kg</b> .  |

## Tugas

Lanjutkan penambahan code pada code editor dengan dua perintah berikut.

- ☐ Buat satu variable dengan nama **pi** dan isilah dengan nilai 3.14
- ☐ Tampilkan isi variable **pi**.

Jalankan dan jika muncul hasil sebagai berikut, klik tombol  (submit) untuk mengirimkan code ke sistem DQLab dan melakukan scoring apakah code Anda sudah benar atau tidak.

```
> budi_berat_kg <- 68
> santi_berat_kg <- 54.5
> budi_berat_kg
[1] 68
> santi_berat_kg
[1] 54.5
> pi <- 3.14
> pi
[1] 3.14
```





# Comment pada R

**Comment** atau komentar adalah teks yang bisa dimasukkan di R, tapi tidak dianggap sebagai code yang bisa dieksekusi. Comment ini biasanya digunakan sebagai catatan untuk menjelaskan potongan code yang ada.

Membuat **comment** sangat mudah. Caranya adalah menuliskan tanda pagar (#) yang kemudian diikuti dengan tulisan apapun.

Berikut adalah contoh perhitungan matematika yang diikuti oleh sebuah **comment**.

```
2 + 2 #Ini adalah baris komentar
```

Ketik contoh tersebut dan jika berjalan dengan lancar maka Anda akan mendapatkan hasil berikut.

```
> 2 + 2 #Ini adalah baris komentar  
[1] 4
```

Catatan: Tanda > tidak perlu diketik.

Perhatikan bahwa **comment** "Ini adalah baris komentar" tidak dimengerti oleh R dan tidak menghasilkan apapun. Hal yang berbeda dengan perhitungan "2 + 2" yang tetap menghasilkan output angka 4.

Sebagai praktek, tambahkan komentar "Ini adalah komentar penutup" pada code editor.



Jalankan dan klik tombol (submit) untuk mengirimkan hasil dan melangkah ke bagian berikutnya.

Catatan: huruf besar dan huruf kecil berpengaruh pada saat submit code.

# Kesimpulan

Selamat, Anda telah menyelesaikan bab pertama mengenai apa itu R, hubungan dengan *data science* dan kenapa *data scientist* menggunakannya sebagai alat bantu utama.

Selain itu, Anda juga telah menguasai keterampilan dasar R dengan beberapa praktek berikut:

- ☐ Mengolah dan menampilkan data dengan perintah *print* ataupun mengetikkan langsung data tersebut.
- ☐ Melakukan perhitungan matematika.
- ☐ Memberikan komentar atau *comment* untuk memberi penjelasan pada code di R.
- ☐ Menyimpan angka, teks dan hasil perhitungan ke dalam *variable* - sehingga dapat digunakan pada bagian code lain.
- ☐ Menjelaskan sifat R yang *case sensitive* - huruf besar dan huruf kecil adalah berbeda.

## Pendahuluan



List



Vector



Data Frame

Dengan berbekal teori dan praktek sebelumnya, sekarang kita telah siap untuk mengenal **tipe data** dan **operasi data** yang lebih berguna pada kasus nyata, misalkan untuk dapat ditampilkan dengan bermacam komposisi grafik yang menarik.

Untuk mencapai hal ini kita perlu mengenal tiga tipe data yang paling sering digunakan di R, yaitu **vector**, **list** dan **data frame**.

Kita akan belajar konsep dan praktek detil untuk mengenali semuanya dengan lebih baik.

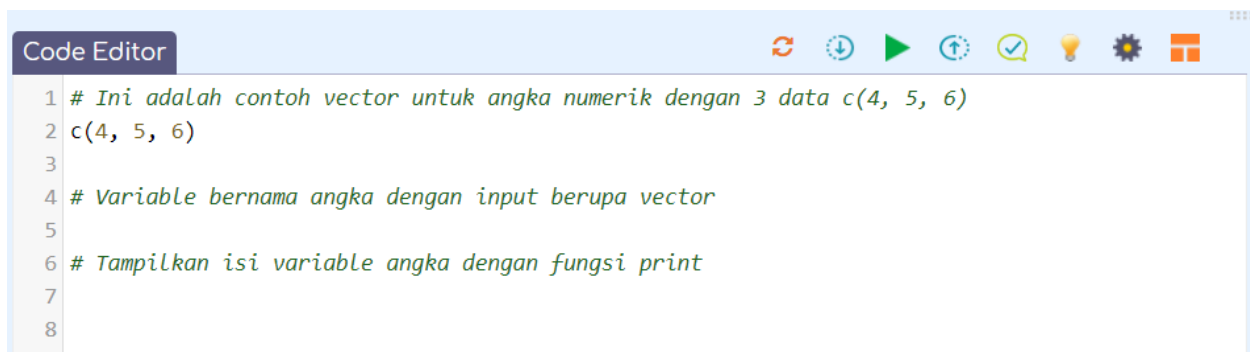
Khusus untuk **data frame** akan dicontohkan dengan fungsi pembacaan file teks, sehingga Anda akan mendapatkan pengalaman yang lebih nyata kenapa harus menggunakan tipe data ini pada praktek sehari-hari.

# Vector

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

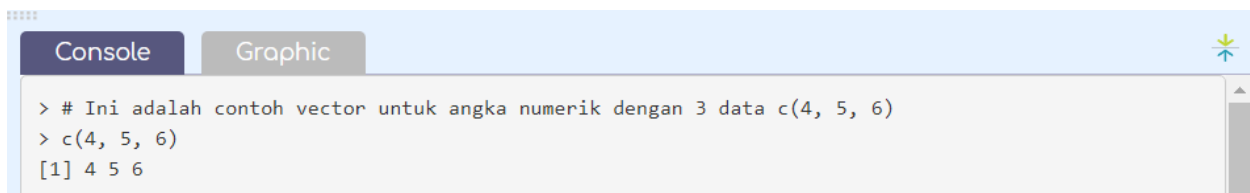
Vector didefinisikan dengan nama function yang memiliki satu huruf saja: **c**. Sebagai contoh untuk membuat vector yang isinya angka dengan nilai 2, 5, dan 7, maka perintahnya adalah **c(2, 5, 7)**.

Cobalah ketik perintah tersebut di bagian bawah komentar "Ini adalah contoh vector untuk angka numerik dengan 3 data c(4, 5, 6)" sehingga tampilan code editor tampak sebagai berikut.



```
Code Editor
1 # Ini adalah contoh vector untuk angka numerik dengan 3 data c(4, 5, 6)
2 c(4, 5, 6)
3
4 # Variable bernama angka dengan input berupa vector
5
6 # Tampilkan isi variable angka dengan fungsi print
7
8
```

Jalankan perintah tersebut sehingga Anda akan mendapatkan hasil pada Console seperti gambar berikut.



```
Console
> # Ini adalah contoh vector untuk angka numerik dengan 3 data c(4, 5, 6)
> c(4, 5, 6)
[1] 4 5 6
```

Ini artinya Anda telah membuat dan menampilkan sebuah **vector** yang berisi tiga 3 angka: 4, 5 dan 6. Terlihat tiga angka tersebut semua ditampilkan dalam satu baris.

Sekarang kita akan simpan vector ini ke dalam variable **angka** dan ditampilkan bukan dengan menggunakan fungsi **print** pada bagian bawah comment "# Variable bernama angka dengan input berupa vector".

```
Code Editor
1 # Ini adalah contoh vector untuk angka numerik dengan 3 data c(4, 5, 6)
2 c(4, 5, 6)
3
4 # Variable bernama angka dengan input berupa vector
5 angka <- c(4, 5, 6)
6
7 # Tampilkan isi variable angka dengan fungsi print
8 print(angka)
```

Jalankan code tersebut, dan jika berjalan dengan lancar maka akan tampil hasil berikut.

```
Console
> # Ini adalah contoh vector untuk angka numerik dengan 3 data c(4, 5, 6)
> c(4, 5, 6)
[1] 4 5 6

> # Variable bernama angka dengan input berupa vector
> angka <- c(4, 5, 6)

> # Tampilkan isi variable angka dengan fungsi print
> print(angka)
[1] 4 5 6
```

Terliha pada baris terakhir, variable **angka** yang kita tampilkan sama seperti tampilan awal yang menunjukkan angka-angka ini disimpan dalam bentuk **vector**. Jika semua hasil telah sesuai, klik tombol Submit Code untuk melanjutkan.

# Deretan Nilai dengan Operator :

Operator : atau titik dua adalah operator yang digunakan untuk mempersingkat penulisan dari nilai-nilai vector yang berurutan. Sebagai contoh, perintah `c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)` yang membentuk vector dengan angka 1 sampai dengan 10 dapat dipersingkat dengan `c(1:10)`.

Ketiklah empat perintah berikut pada code editor, dimana variable **angka1** diisi dengan **vector** dengan penulisan angka berurutan yang panjang. Dan variable **angka2** diisi dengan menggunakan operator `:`.

```
angka1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
print(angka1)
angka2 <- c(1:10)
print(angka2)
```

Jalankan perintah tersebut, jika berhasil akan muncul tampilan sebagai berikut.

```
angka1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

> print(angka1)
[1] 1 2 3 4 5 6 7 8 9 10

> angka2 <- c(1:10)

> print(angka2)
[1] 1 2 3 4 5 6 7 8 9 10
```

# Vector dengan Isi Teks

Selain angka, vector juga bisa diisi dengan teks. Cobalah ketik satu perintah pada code editor untuk membuat *vector* yang isinya tiga nama orang dan disimpan sebagai *variable* `nama_mahasiswa` sebagai berikut.

```
nama_mahasiswa <- c("Amira","Budi","Charlie")
```

Kemudian lanjutkan dengan perintah untuk menampilkan *variable* tersebut.

```
print(nama_mahasiswa)
```

Jika berhasil dijalankan dengan baik, kita akan memperoleh hasil seperti berikut.

```
> nama_mahasiswa <- c("Amira","Budi","Charlie")  
  
> print(nama_mahasiswa)  
[1] "Amira" "Budi"  "Charlie"
```

Terlihat vector yang ditampilkan adalah urutan teks "Amira", "Budi", dan "Charlie". Ini menunjukkan bahwa *vector* bisa berisi teks selain angka.

# Index dan Accessor pada Vector

Pada latihan sebelumnya kita sudah mengerti cara membuat **vector** untuk angka dan teks, namun belum ditunjukkan cara untuk mengambil tiap nilai dari **vector** tersebut. Perhatikan pada live code editor telah terisi **code-code** yang dapat Anda praktekkan sambil melihat penjelasan di bawah ini.

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

Mari kita langsung contohkan agar lebih jelas. Pada code editor kita telah membuat suatu variable vector angka dengan isi berupa angka numerik 20 sampai dengan 30.

```
angka <- c(20:30)
```

yang jika ditampilkan akan terlihat sebagai berikut - cobalah jalankan code untuk mendapatkan hasil ini.

```
> print(angka)
[1] 20 21 22 23 24 25 26 27 28 29 30
```

Terlihat nilai dari urutan ketiga pada vector angka ini adalah angka 22, bagaimana kita mengambil nilai ini? Seperti petunjuk di atas, kita bisa gunakan accessor dan index dengan perintah berikut.

```
print(angka[3])
```

Cobalah ketik perintah tersebut di bawah comment "# Tampilkan isi variable angka pada posisi ke 3", dan jalankan code dengan tombol Run.

Jika berjalan dengan lancar maka akan tampil hasil sebagai berikut.

```
> print(angka[3])
[1] 22
```

Kita juga bisa mengambil urutan indeks dengan menggunakan **accessor** dengan format kurung siku ganda sebagai berikut.



```
print(angka[[5]])
```

Jika berjalan dengan lancar, maka hasilnya tampak sebagai berikut.

```
> print(angka[[5]])
[1] 24
```

Bagaimana dengan indeks angka lebih dari satu dan berurutan, misalkan saya ingin mengambil posisi keempat, kelima dan keenam dan menampilkannya?

Berikut adalah perintah yang perlu Anda ketikkan - dengan indeks menggunakan operator titik dua (:).

```
print(angka[4:6])
```

Tambahkan perintah tersebut di bawah comment "# Tampilkan isi variable angka pada posisi ke 4 s/d 6" dan jalankan. Jika berhasil maka akan keluar output berikut ini.

```
> print(angka[4:6])
[1] 23 24 25
```

Sekarang kita coba beralih ke vector teks. Buatlah variable dengan nama **kode\_prodi** dengan isi "DKV", "ILKOM", dan "ICT" pada code editor - di bawah comment "# Buat vector teks dengan nama kode\_prodi yang diisi sesuai petunjuk soal".

```
kode_prodi <- c("DKV", "ILKOM", "ICT")
```

Jangan jalankan dulu, cobalah Anda ketik sendiri perintah untuk menampilkan nilai urutan ketiga pada variable kode\_prodi ini di bawah comment "# Tampilkan isi indeks ketiga dari kode\_prodi" .



Klik tombol (submit) untuk coba menguji code ke sistem DQLab dan melangkah ke subbab berikutnya jika semua berjalan dengan lancar.

# Named Vector

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

Mari kita langsung praktekkan dengan contoh, ketik *named vector* berikut - dimana kita membuat vector angka dengan tiap urutan diberi nama.

```
nilai <- c(statistik = 89, fisika = 95, ilmukomunikasi = 100)
```

Dengan mudah Anda mungkin bisa melihat bahwa vector ini mewakili nilai-nilai untuk mata kuliah "statistik", "fisika" dan "ilmukomunikasi".

Code untuk membuat dan menampilkan *named vector* ini telah ada pada code editor. Jalankan dan jika berhasil, maka akan muncul output berikut.

```
> print(nilai)
      statistik      fisika ilmukomunikasi
           89           95           100
```

Perhatikan terdapat perbedaan tampilan output *named vector* ini dengan tampilan output vector sebelumnya - yaitu sudah tidak ada awalan [1].

Nah, sekarang kita akan mengambil dan menampilkan nilai pada vector untuk nama "fisika".

```
print(nilai["fisika"])
```

Ketikkan perintah tersebut di bawah *comment* "#Menampilkan isi dengan nama fisika" dan jalankan. Hasil yang Anda dapatkan harusnya sebagai berikut.

```
> print(nilai["fisika"])
fisika
     95
```

Bagaimana, cukup jelas?

Sebagai latihan sehingga Anda benar-benar menguasai bahan *named vector* ini, cobalah buat variable *named vector* **profil** dengan input berikut:

- ☐ nama = "Budi"
- ☐ tempat\_tinggal = "Jakarta"
- ☐ tingkat\_pendidikan = "S1"

Kemudian tampilkan variable **profil** ini dengan menggunakan fungsi *print*.


Jalankan dan lakukan *review* sebelum anda klik tombol Submit Code untuk menguji *code* dan untuk melanjutkan ke subbab berikutnya

# List

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

Sebagai contoh, untuk membuat list yang isinya campuran jenis data angka 2, "Budi", dan angka 4 – maka perintahnya adalah sebagai berikut.

**list(2, "Budi", 4).**

Panel code editor telah dilengkapi rangkaian contoh *code* untuk menghasilkan list dan komentarnya sekaligus menjelaskan apa yang dilakukan. Jalankan dengan tombol  untuk melihat hasilnya.

Catatan: Perhatikan hasil eksekusi seperti di bawah. Terlihat outputnya agak berbeda dengan vector - dimana tiap output terdapat dua tampilan accessor (kurung siku tunggal dan ganda).

```
> list(2, "Budi", 4)
[[1]]
[1] 2
[[2]]
[1] "Budi"
[[3]]
[1] 4
```

## Tugas

Buatlah variable dengan nama **kota** dengan isi berikut

nama\_kota = "Makassar"

propinsi = "Sulawesi Selatan"

luas\_km2 = 199.3



Klik tombol (submit) untuk mengirimkan code ke sistem DQLab dan melakukan scoring apakah code Anda sudah benar atau tidak.

# List Index

Untuk mengambil isi list, kita bisa mengambil dari posisi index-nya, ketentuan dan caranya sama persis dengan vector.

Contoh: Untuk mengambil posisi kedua dari variable list **list\_saya** kita bisa gunakan

```
list_saya[2]
```

atau

```
list_saya[[2]].
```

Panel code editor telah dilengkapi rangkaian contoh *code* untuk list dan cara akses indeksnya beserta penjelasan dalam bentuk comment. Jalankan dengan tombol ► untuk melihat hasilnya.

## Tugas

Buatlah variable dengan nama **list\_satu** dengan isi angka 1, character “Online”, dan nilai logical TRUE. Tampilkan index pertama dari list tersebut dengan accessor kurung siku ganda.



Klik tombol (submit) untuk mengirimkan code ke sistem DQLab dan melakukan scoring apakah code Anda sudah benar atau tidak.

# Data Frame

Data frame adalah jenis struktur data yang dirancang untuk representasi table, yang terdiri dari banyak kolom dengan tiap kolom berisi list ataupun vector dengan jumlah data yang sama.

Untuk membuat data frame kita bisa gunakan function **data.frame**.

Panel code editor telah dilengkapi contoh *code* untuk membuat data frame untuk data mahasiswa. Jalankan dengan tombol ► untuk melihat hasilnya.

## Tugas

Tambahkan pada code editor, tugas-tugas berikut. Code-code yang sebelumnya ada di code editor tidak boleh dihapus

- ☐ Buatlah vector terbaru bernama **akreditasi** dengan isi ("A","A","B","A","A")
- ☐ Buat satu data frame dengan nama **info\_mahasiswa** yang terdiri dari dua vector dari contoh ditambah dengan vector **akreditasi**.
- ☐ Tampilkan data frame **info\_mahasiswa**.



Klik tombol (submit) untuk mengirimkan code ke sistem DQLab dan melakukan scoring apakah code Anda sudah benar atau tidak.

# Cara Akses Data Frame

Data frame memiliki banyak kolom dan bisa diakses dengan nama kolom yang digunakan. Caranya adalah menggunakan accessor dengan tanda \$ yang diikuti dengan nama kolom.

Contoh: `info_mahasiswa$fakultas`.

Panel code editor telah dilengkapi contoh *code* untuk membuat data frame untuk data mahasiswa dan akses kolom `jumlah_mahasiswa`. Jalankan dengan tombol ► untuk melihat hasilnya.

## Tugas

Tambahkan code untuk menampilkan kolom **fakultas** dari data frame **info\_mahasiswa**. Code yang sudah terdapat di code editor tidak boleh dihapus



Klik tombol (submit) untuk mengirimkan code ke sistem DQLab dan melakukan scoring apakah code Anda sudah benar atau tidak.

## Kesimpulan

Anda telah belajar mengenai tiga tipe data fundamental yang sangat penting di R pada bab ini, yaitu vector, list dan data frame. Tiga tipe data ini dapat diisi dengan lebih dari satu nilai.

Ringkasan dari tipe-tipe data tersebut adalah sebagai berikut.

- **Vector** hanya bisa diisi dengan salah satu tipe data saja di seluruh elemennya, misalkan angka ataupun teks. Pembuatan vector menggunakan fungsi `c`, dan bisa diakses dengan *accessor* dengan angka indeks yang diapit kurung siku. Namun jika berupa named vector, maka indeksnya adalah berupa teks.
- **List** adalah tipe data yang bisa diisi dengan lebih dari satu tipe data di seluruh elemennya, dengan campuran teks dan angka misalnya. Pembuatan list adalah menggunakan fungsi `list`, dan seperti vector elemen pada list diakses dengan accessor yang diapit kurung siku.
- **Data Frame** adalah tipe data yang terdiri dari satu atau beberapa vector ataupun list. Untuk membuat data frame kita gunakan fungsi `data.frame`, dan kita bisa mengakses menggunakan accessor `$` diikuti nama kolom, dan juga angka indeks.

Dengan memahami penggunaan vector, list dan data frame kita siap mempelajari penggunaan banyak fungsi advanced di R, seperti menghasilkan grafik dan penggunaan algoritma *machine learning*.

# Pendahuluan

Jika kita ingin eksplorasi dan menganalisa data lebih jauh, grafik adalah tipe tampilan yang tidak bisa ditawar. Dan R memiliki fungsi yang kaya untuk menghasilkan grafik.

Bab ini akan berisi pengenalan singkat dan padat untuk menunjukkan kemampuan ini dengan menggunakan ***package ggplot2***. Dengan pendekatan yang bersifat demo, Anda cukup mengikuti petunjuk sehingga bisa membayangkan apa saja yang bisa Anda lakukan dengan ggplot2.



# Package ggplot2

Pada dua bab sebelumnya, kita telah mampu menggunakan banyak perintah yang masih dalam paket *standard* di R. Untuk kebanyakan kasus, fungsi-fungsi standar ini tidak cukup.

Nah, untuk fungsi yang lebih *powerful* seperti menghasilkan grafik yang *advanced* - kita perlu menggunakan apa yang dinamakan *package*.

Untuk memahami konsep package ini, pada editor terdapat potongan code untuk menghasilkan grafik. Cobalah jalankan, dan Anda akan mendapatkan *error* seperti ini.

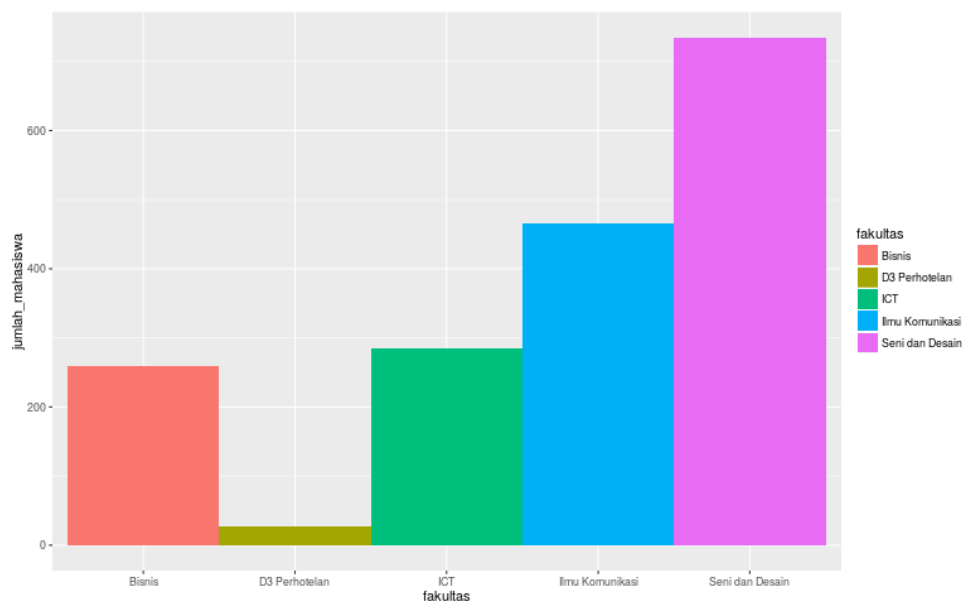
```
> gambar <- ggplot(info_mahasiswa, aes(x=fakultas, y=jumlah_mahasiswa, fill=fakultas))  
could not find function "ggplot"
```

Terlihat ada tampilan error berwarna merah yang menyatakan bahwa fungsi *ggplot* tidak ditemukan. Ini karena fungsi tersebut bukan fungsi standar, tapi harus menggunakan *package* ggplot2 dengan fungsi library.

Tambahkan code tersebut di bawah comment "#Menggunakan package ggplot2" dan jalankan kembali.

```
library("ggplot2")
```

Jika berhasil, maka akan muncul grafik berikut yang menunjukkan jumlah mahasiswa berdasarkan fakultas di salah satu universitas di Tangerang.



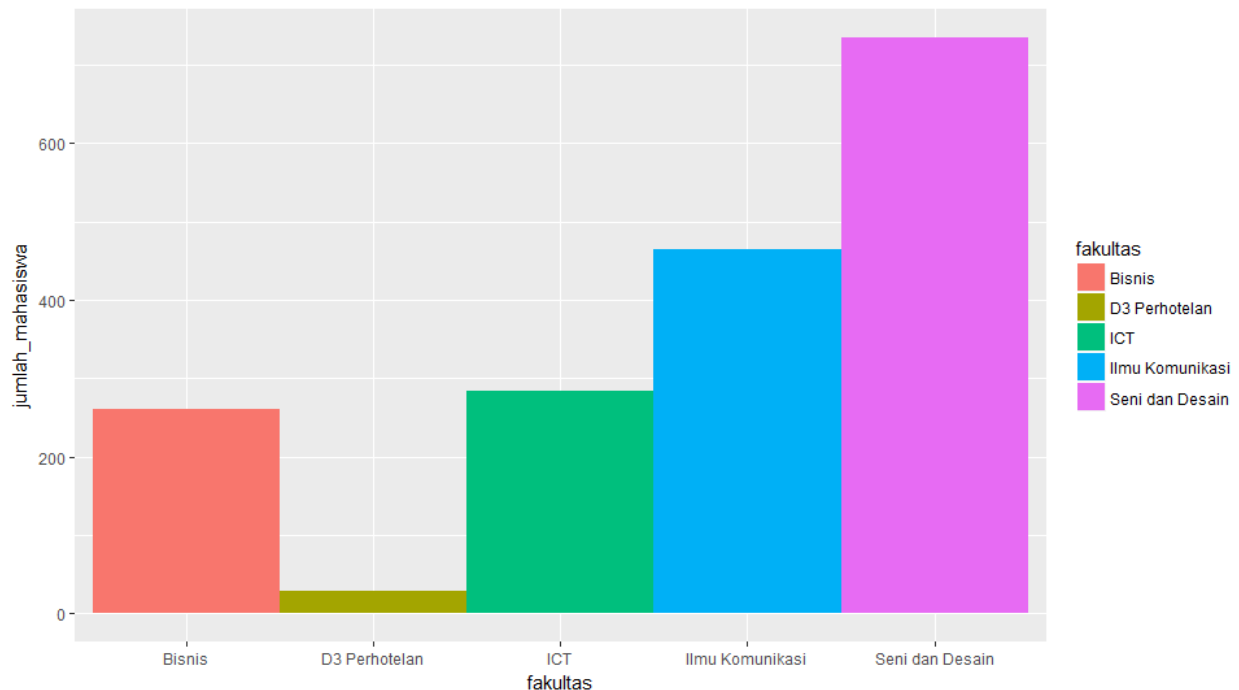
Dari grafik tersebut, terlihat Seni dan Desain adalah fakultas paling favorit dengan jumlah mahasiswa terbanyak :)

# Membuat Grafik Sebaran Mahasiswa (1)

Pada praktek kali ini, kita akan menghasilkan *bar chart* sederhana dengan menggunakan variable data frame bernama **info\_mahasiswa** yang kita hasilkan pada subbab "Vector, List dan Data Frame". Variable ini masih bersifat statis atau *hard code*, artinya data frame ini bukan berdasarkan pembacaan dari suatu file atau database.

Cara membuat grafik di R bisa menggunakan banyak cara, salah satunya dengan library **ggplot2** - dimana kita menggambar *chart* secara bertahap, yaitu dengan konsep *layering* (lapisan demi lapisan).

Untuk lebih jelasnya, pada code editor telah terdapat code-code untuk menghasilkan distribusi mahasiswa. Cobalah jalankan, dan jika lancar akan menghasilkan grafik berikut.



Terlihat kalau mahasiswa di fakultas "Seni dan Desain" merupakan fakultas dengan jumlah mahasiswa terbanyak, diikuti kemudian oleh fakultas "Ilmu Komunikasi", "ICT", "Bisnis" dan "D3 Perhotelan". Dengan histogram ini, informasi lebih mudah dilihat dan dicerna dibandingkan dengan angka bukan?

Bagaimana grafik ini dihasilkan di ggplot? Seperti informasi sebelumnya, grafik ini digambar lapis demi lapis. Dan lapisan pertama kita ibaratkan sebagai "kanvas" lukisan, dan untuk membuat "kanvas" pada contoh di code editor, kita gunakan fungsi yang namanya **ggplot**.

```
gambar <- ggplot(info_mahasiswa, aes(x=fakultas, y=jumlah_mahasiswa, fill=fakultas))
```

Disini terlihat hasil ggplot2 disimpan pada variable gambar. Variable ini yang akan menyimpan seluruh grafik dan digunakan untuk menggambar ketika digunakan sendiri pada code editor dengan perintah berikut.

```
gambar
```

Penambahan bentuk, warna dan ukuran dilakukan dengan menggunakan tanda operator plus (+) diikuti fungsi terkait. Sebagai contoh, untuk menggambar bentuk *bar chart* di atas "kanvas" kita gunakan fungsi **geom\_bar**. Sebagai contoh, untuk menggambar bar chart di atas "kanvas" kita gunakan fungsi

```
gambar <- gambar + geom_bar(width=1, stat="identity")
```

Berikut adalah penjelasan lengkap dari code yang terdapat pada code editor.

| Code  | Penjelasan  |
|---|---|
| <code>ggplot(info_mahasiswa, aes(x=fakultas, y=jumlah_mahasiswa, fill=fakultas))</code> | <p>Perintah awal untuk membentuk objek grafik ggplot2, terdiri dari:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>info_mahasiswa</b>: data frame yang kita gunakan</li> <li><input type="checkbox"/> <b>aes</b>: fungsi untuk mengambil data apa saja</li> <li><input type="checkbox"/> <b>x</b>: kolom yang kita gunakan untuk grafik sumbu x, dalam hal ini <b>fakultas</b></li> <li><input type="checkbox"/> <b>y</b>: kolom nilai yang kita gunakan untuk plot sumbu y, dalam hal ini <b>jumlah_mahasiswa</b></li> <li><input type="checkbox"/> <b>fill</b>: kolom mana yang akan kita gunakan sebagai pembeda warna. Jika fill dihilangkan maka grafik akan diisi warna abu-abu saja.</li> </ul> |
| <code>+</code>  | Tanda plus, digunakan untuk menggabungkan objek-objek ggplot2   |
| <code>geom_bar(width = 1, stat = "identity")</code>                                     | <p>Perintah untuk menambahkan bentuk geometri bar chart, dengan parametre:</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> <b>width</b>: ukuran lebar dari tiap bar chart, disini</li> <li><input type="checkbox"/> <b>stat</b>: transformasi data yang perlu dilakukan. Banyak sekali jenisnya, untuk saat ini karena kita hanya ingin plot apa adanya, kita gunakan <b>identity</b>. Untuk lebih lengkapnya mengenai stat ini akan dibahas di course "Data Visualization with ggplot2 in R".</li> </ul>   |

gambar

Variable untuk menampung objek grafik.

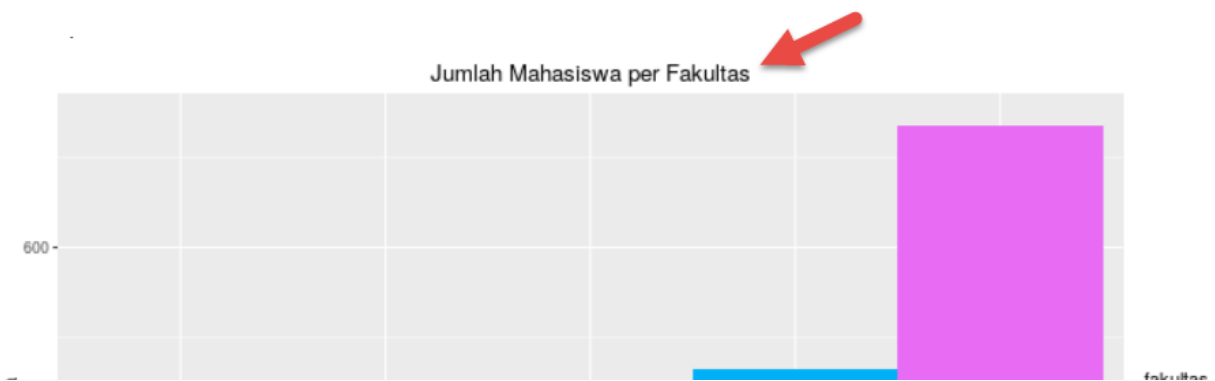
### Latihan Praktek

Mari kita sedikit improvisasi dari gambar plot tersebut dengan penambahan-penambahan berikut ini.

- Mari kita tambahkan judul "Jumlah Mahasiswa per Fakultas" pada chart untuk memperjelas konteksnya. Ini bisa dilakukan dengan menambahkan lapisan judul pada variable **gambar** dengan function **ggtitle("Jumlah Mahasiswa per Fakultas")** sebagai berikut

```
gambar <- gambar + ggtitle("Jumlah Mahasiswa per Fakultas")
```

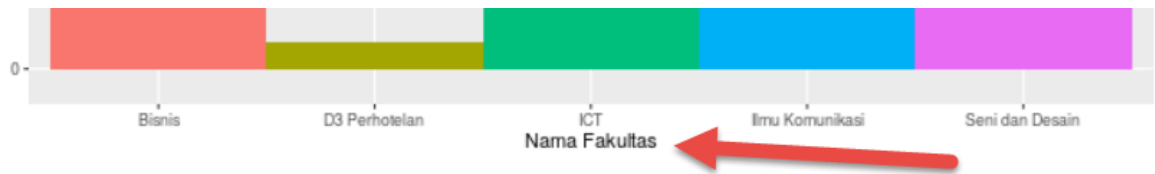
Catatan: tambahkan code tersebut di bawah comment #Menambahkan judul grafik. Jalankan code untuk memastikan terdapat tambahan judul pada chart seperti potongan gambar berikut.



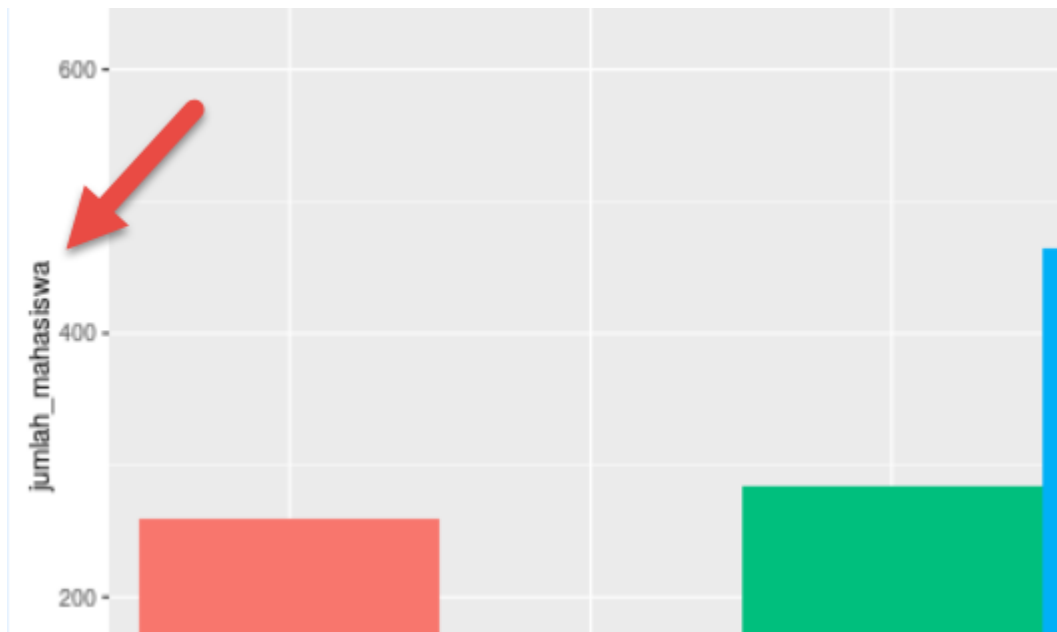
- Dengan cara yang sama di atas, cobalah tambahkan lagi di atas **gambar** dengan perintah **xlab("Nama Fakultas")**. Ini agar caption pada sumbu X berubah dari hanya "fakultas" menjadi "Nama Fakultas".

```
gambar <- gambar + xlab("Nama Fakultas")
```

Catatan: tambahkan code tersebut di bawah comment #Menambahkan caption pada sumbu x. Jalankan code untuk memastikan caption sumbu X sudah berubah seperti pada potongan gambar berikut.



- Dengan cara yang sama, cobalah tambahkan sendiri lapisan untuk merubah caption "jumlah\_mahasiswa" menjadi "Jumlah Mashasiswa" dengan perintah `ylab("Jumlah Mahasiswa")`. Jika berhasil maka caption sumbu Y akan berubah seperti berikut.



Jika semua sudah berjalan dengan baik, klik tombol Submit Code untuk mengirimkan code dan melanjutkan ke bagian berikutnya.

# Could not find function "ggplot"?

Pada saat menjalankan code dan keluar error seperti berikut.

```
> gambar <- ggplot(info_mahasiswa, aes(x=fakultas, y=jumlah_mahasiswa, fill=fakultas))  
could not find function "ggplot"
```

Apa yang perlu ditambahkan pada code kita?

- ☐ Semua benar
- ☐ `package("ggplot2")`
- ☒ `library("ggplot2")`
- ☐ `info_mahasiswa <- c("fakultas", "jumlah_mahasiswa", "fakultas")`

# Membaca File Excel

Banyak pengolahan data di R harus membaca dari file Excel. Pada contoh kali ini, kita akan menggunakan file yang berisi data jumlah mahasiswa per angkatan per tahun seperti pada tampilan berikut.

|    | A        | B               | C                        | D     | E      | F |
|----|----------|-----------------|--------------------------|-------|--------|---|
| 1  | ANGKATAN | Fakultas        | Prodi                    | Kode  | JUMLAH |   |
| 2  | 2015     | Bisnis          | Akuntansi                | AKUN  | 88     |   |
| 3  | 2016     | Bisnis          | Akuntansi                | AKUN  | 85     |   |
| 4  | 2017     | Bisnis          | Akuntansi                | AKUN  | 103    |   |
| 5  | 2016     | Seni dan Desain | Arsitek                  | AR    | 28     |   |
| 6  | 2017     | Seni dan Desain | Arsitek                  | AR    | 45     |   |
| 7  | 2015     | Seni dan Desain | Desain Komunikasi Visual | DKV   | 279    |   |
| 8  | 2016     | Seni dan Desain | Desain Komunikasi Visual | DKV   | 314    |   |
| 9  | 2017     | Seni dan Desain | Desain Komunikasi Visual | DKV   | 337    |   |
| 10 | 2015     | Seni dan Desain | Film dan Televisi        | FTV   | 284    |   |
| 11 | 2016     | Seni dan Desain | Film dan Televisi        | FTV   | 274    |   |
| 12 | 2017     | Seni dan Desain | Film dan Televisi        | FTV   | 300    |   |
| 13 | 2015     | Ilmu Komunikasi | Ilmu Komunikasi          | ILKOM | 499    |   |
| 14 | 2016     | Ilmu Komunikasi | Ilmu Komunikasi          | ILKOM | 275    |   |
| 15 | 2017     | Ilmu Komunikasi | Ilmu Komunikasi          | ILKOM | 313    |   |
| 16 | 2015     | Ilmu Komunikasi | Jurnalistik              | JR    | 1      |   |
| 17 | 2016     | Ilmu Komunikasi | Jurnalistik              | JR    | 251    |   |
| 18 | 2017     | Ilmu Komunikasi | Jurnalistik              | JR    | 186    |   |

File tersebut bernama **mahasiswa.xlsx**, dan akan kita baca dengan fungsi **read.xlsx** dari package **openxlsx** dengan cara berikut.

```
library(openxlsx)
mahasiswa <- read.xlsx("https://dqlab.id/mahasiswa.xlsx", sheet = "Sheet 1")
```

Dari code di atas, file yang dibaca dengan fungsi **read.xlsx** dan kemudian disimpan sebagai variable **mahasiswa** (yang merupakan tipe data frame). File yang dibaca adalah "mahasiswa.xlsx" pada sheet "Sheet 1".

Pada Code Editor sudah disertakan code-code untuk untuk membaca file ini, cobalah jalankan dan hasilnya akan terlihat sebagai berikut.

```
> print(mahasiswa)
```

|    | ANGKATAN | Fakultas                           | Prodi                    | Kode  | JUMLAH |
|----|----------|------------------------------------|--------------------------|-------|--------|
| 1  | 2015     | Bisnis                             | Akuntansi                | AKUN  | 88     |
| 2  | 2016     | Bisnis                             | Akuntansi                | AKUN  | 85     |
| 3  | 2017     | Bisnis                             | Akuntansi                | AKUN  | 103    |
| 4  | 2016     | Seni dan Desain                    | Arsitek                  | AR    | 28     |
| 5  | 2017     | Seni dan Desain                    | Arsitek                  | AR    | 45     |
| 6  | 2015     | Seni dan Desain                    | Desain Komunikasi Visual | DKV   | 279    |
| 7  | 2016     | Seni dan Desain                    | Desain Komunikasi Visual | DKV   | 314    |
| 8  | 2017     | Seni dan Desain                    | Desain Komunikasi Visual | DKV   | 337    |
| 9  | 2015     | Seni dan Desain                    | Film dan Televisi        | FTV   | 284    |
| 10 | 2016     | Seni dan Desain                    | Film dan Televisi        | FTV   | 274    |
| 11 | 2017     | Seni dan Desain                    | Film dan Televisi        | FTV   | 300    |
| 12 | 2015     | Ilmu Komunikasi                    | Ilmu Komunikasi          | ILKOM | 499    |
| 13 | 2016     | Ilmu Komunikasi                    | Ilmu Komunikasi          | ILKOM | 275    |
| 14 | 2017     | Ilmu Komunikasi                    | Ilmu Komunikasi          | ILKOM | 313    |
| 15 | 2015     | Ilmu Komunikasi                    | Jurnalistik              | JR    | 1      |
| 16 | 2016     | Ilmu Komunikasi                    | Jurnalistik              | JR    | 251    |
| 17 | 2017     | Ilmu Komunikasi                    | Jurnalistik              | JR    | 186    |
| 18 | 2015     | Bisnis                             | Manajemen                | MAN   | 169    |
| 19 | 2016     | Bisnis                             | Manajemen                | MAN   | 163    |
| 20 | 2017     | Bisnis                             | Manajemen                | MAN   | 160    |
| 21 | 2017     | D3 Perhotelan                      | Perhotelan               | HTL   | 28     |
| 22 | 2015     | ICT                                | Sistem Informasi         | SI    | 104    |
| 23 | 2016     | ICT                                | Sistem Informasi         | SI    | 95     |
| 24 | 2017     | ICT                                | Sistem Informasi         | SI    | 113    |
| 25 | 2015     | ICT                                | Sistem Komputer          | SK    | 20     |
| 26 | 2016     | ICT                                | Sistem Komputer          | SK    | 20     |
| 27 | 2017     | ICT                                | Sistem Komputer          | SK    | 18     |
| 28 | 2016     | ICT                                | Teknik Elektro           | TE    | 4      |
| 29 | 2017     | ICT                                | Teknik Elektro           | TE    | 7      |
| 30 | 2016     | ICT                                | Teknik Fisika            | TF    | 7      |
| 31 | 2017     | ICT                                | Teknik Fisika            | TF    | 16     |
| 32 | 2015     | ICT                                | Teknik Informatika       | TI    | 125    |
| 33 | 2016     | ICT                                | Teknik Informatika       | TI    | 168    |
| 34 | 2017     | ICT                                | Teknik Informatika       | TI    | 164    |
| 35 | 2017     | ICT Teknik Informatika Dual Degree |                          | TID   | 6      |

Terlihat hasil pembacaan Excel ini merupakan data frame yang terdiri dari lima kolom - dengan nama ANGGATAN, Fakultas, Prodi, Kode, dan JUMLAH - dan 35 baris data.

Kita akan menggunakan data frame ini sepanjang sisa bab ini, mari kita menutup subbab ini dengan menampilkan kolom Prodi dari data frame ini, caranya dengan menuliskan perintah berikut.

```
print(mahasiswa$Prodi)
```

Tambahkan pada Code Editor dan jalankan untuk menampilkan hasil berikut.



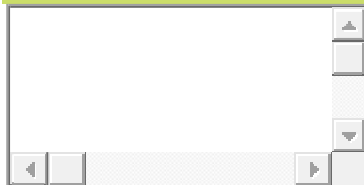
```
> print(mahasiswa$Prodi)
[1] "Akuntansi"           "Akuntansi"
[3] "Akuntansi"           "Arsitek"
[5] "Arsitek"             "Desain Komunikasi Visual"
[7] "Desain Komunikasi Visual" "Desain Komunikasi Visual"
[9] "Film dan Televisi"    "Film dan Televisi"
[11] "Film dan Televisi"    "Ilmu Komunikasi"
[13] "Ilmu Komunikasi"      "Ilmu Komunikasi"
[15] "Jurnalistik"          "Jurnalistik"
[17] "Jurnalistik"          "Manajemen"
[19] "Manajemen"            "Manajemen"
[21] "Perhotelan"           "Sistem Informasi"
[23] "Sistem Informasi"     "Sistem Informasi"
[25] "Sistem Komputer"      "Sistem Komputer"
[27] "Sistem Komputer"      "Teknik Elektro"
[29] "Teknik Elektro"       "Teknik Fisika"
[31] "Teknik Fisika"        "Teknik Informatika"
[33] "Teknik Informatika"   "Teknik Informatika"
[35] "Teknik Informatika Dual Degree"
```

Jika semua sudah sesuai, Submit Code untuk melanjutkan ke bagian berikutnya.

Lesson

5

120



1

```
#Menggunakan package ggplot2
```

2

```
library(ggplot2)
```

3

```
#Menggunakan package openxlsx
```

4

```
library(openxlsx)
```

5

6

```
#Membaca file mahasiswa.xlsx
```

7

```
mahasiswa <- read.xlsx("https://academy.dqlab.id/dataset/mahasiswa.xlsx",sheet = "Sheet 1")
```

8

```
#Menampilkan data
```

10

```
print(mahasiswa)
```

11

```
#Menampilkan kolom Prodi
```

13

```
print(mahasiswa$Prodi)
```

Code Editor

0 Hint Submit Run

Console

Graphic

&gt;

```
> #Menggunakan package ggplot2
> library(ggplot2)
```

```
> #Menggunakan package openxlsx
> library(openxlsx)
```

```
> #Membaca file mahasiswa.xlsx
> mahasiswa <- read.xlsx("https://academy.dqlab.id/dataset/mahasiswa.xlsx",sheet = "Sheet 1")
```

```
> #Menampilkan data
> print(mahasiswa)
```

|    | ANGKATAN | Fakultas        | Prodi                    | Kode | JUMLAH |
|----|----------|-----------------|--------------------------|------|--------|
| 1  | 2015     | Bisnis          | Akuntansi                | AKUN | 88     |
| 2  | 2016     | Bisnis          | Akuntansi                | AKUN | 85     |
| 3  | 2017     | Bisnis          | Akuntansi                | AKUN | 103    |
| 4  | 2016     | Seni dan Desain | Arsitek                  | AR   | 28     |
| 5  | 2017     | Seni dan Desain | Arsitek                  | AR   | 45     |
| 6  | 2015     | Seni dan Desain | Desain Komunikasi Visual | DKV  | 279    |
| 7  | 2016     | Seni dan Desain | Desain Komunikasi Visual | DKV  | 314    |
| 8  | 2017     | Seni dan Desain | Desain Komunikasi Visual | DKV  | 337    |
| 9  | 2015     | Seni dan Desain | Film dan Televisi        | FTV  | 284    |
| 10 | 2016     | Seni dan Desain | Film dan Televisi        | FTV  | 274    |

|    |      |                                    |                    |       |     |
|----|------|------------------------------------|--------------------|-------|-----|
| 11 | 2017 | Seni dan Desain                    | Film dan Televisi  | FTV   | 300 |
| 12 | 2015 | Ilmu Komunikasi                    | Ilmu Komunikasi    | ILKOM | 499 |
| 13 | 2016 | Ilmu Komunikasi                    | Ilmu Komunikasi    | ILKOM | 275 |
| 14 | 2017 | Ilmu Komunikasi                    | Ilmu Komunikasi    | ILKOM | 313 |
| 15 | 2015 | Ilmu Komunikasi                    | Jurnalistik        | JR    | 1   |
| 16 | 2016 | Ilmu Komunikasi                    | Jurnalistik        | JR    | 251 |
| 17 | 2017 | Ilmu Komunikasi                    | Jurnalistik        | JR    | 186 |
| 18 | 2015 | Bisnis                             | Manajemen          | MAN   | 169 |
| 19 | 2016 | Bisnis                             | Manajemen          | MAN   | 163 |
| 20 | 2017 | Bisnis                             | Manajemen          | MAN   | 160 |
| 21 | 2017 | D3 Perhotelan                      | Perhotelan         | HTL   | 28  |
| 22 | 2015 | ICT                                | Sistem Informasi   | SI    | 104 |
| 23 | 2016 | ICT                                | Sistem Informasi   | SI    | 95  |
| 24 | 2017 | ICT                                | Sistem Informasi   | SI    | 113 |
| 25 | 2015 | ICT                                | Sistem Komputer    | SK    | 20  |
| 26 | 2016 | ICT                                | Sistem Komputer    | SK    | 20  |
| 27 | 2017 | ICT                                | Sistem Komputer    | SK    | 18  |
| 28 | 2016 | ICT                                | Teknik Elektro     | TE    | 4   |
| 29 | 2017 | ICT                                | Teknik Elektro     | TE    | 7   |
| 30 | 2016 | ICT                                | Teknik Fisika      | TF    | 7   |
| 31 | 2017 | ICT                                | Teknik Fisika      | TF    | 16  |
| 32 | 2015 | ICT                                | Teknik Informatika | TI    | 125 |
| 33 | 2016 | ICT                                | Teknik Informatika | TI    | 168 |
| 34 | 2017 | ICT                                | Teknik Informatika | TI    | 164 |
| 35 | 2017 | ICT Teknik Informatika Dual Degree |                    | TID   | 6   |

```

> #Menampilkan kolom Prodi
> print(mahasiswa$Prodi)
[1] "Akuntansi"
[3] "Akuntansi"
[5] "Arsitek"
[7] "Desain Komunikasi Visual"
[9] "Film dan Televisi"
[11] "Film dan Televisi"
[13] "Ilmu Komunikasi"
[15] "Jurnalistik"
[17] "Jurnalistik"
[19] "Manajemen"
[21] "Perhotelan"
[23] "Sistem Informasi"
[25] "Sistem Komputer"
[27] "Sistem Komputer"
[29] "Teknik Elektro"
[31] "Teknik Fisika"
[33] "Teknik Informatika"

"Akuntansi"
"Arsitek"
"Desain Komunikasi Visual"
"Desain Komunikasi Visual"
"Film dan Televisi"
"Film dan Televisi"
"Ilmu Komunikasi"
"Ilmu Komunikasi"
"Jurnalistik"
"Jurnalistik"
"Manajemen"
"Manajemen"
"Sistem Informasi"
"Sistem Informasi"
"Sistem Komputer"
"Sistem Komputer"
"Teknik Elektro"
"Teknik Fisika"
"Teknik Informatika"
"Teknik Informatika"

```

# Package yang dibutuhkan untuk membaca file Excel?

Pada contoh praktek kita sebelumnya, package apa yang dibutuhkan untuk membaca file Excel?

- ☐ ggplot2
- ☐ arules
- ☐ xls
- ☒ openxlsx

## Membuat Grafik Sebaran Mahasiswa (2)

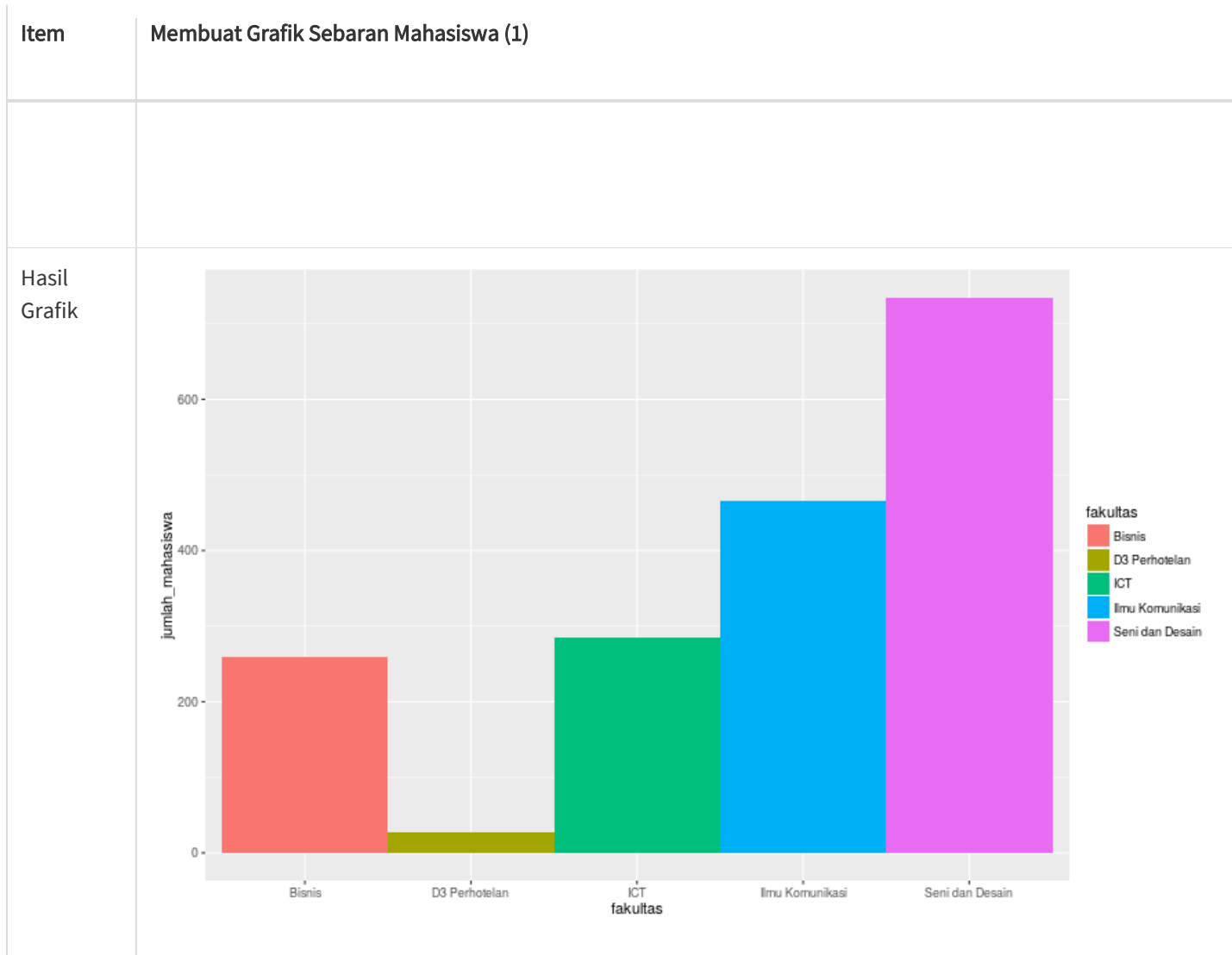
Setelah memiliki kemampuan membaca sumber data dari luar yaitu file Excel yang berisi data jumlah mahasiswa, kita akan kembali menghasilkan grafik sebaran yang sudah kita lakukan sebelumnya - tapi kali ini dengan hasil pembacaan tersebut.

Pada code editor telah diisi code-code untuk membaca file Excel dan menghasilkan grafik histogram. Cobalah jalankan, dan jika berhasil maka akan mendapatkan grafik histogram yang menunjukkan porsi jumlah mahasiswa per fakultas.

Lalu apa perbedaan praktek kita kali ini dengan subbab "Membuat Grafik Sebaran Mahasiswa (1)"?

Berikut adalah daftar perbedaannya, dari potongan code dan hasil.

| Item                  | Membuat Grafik Sebaran Mahasiswa (1)   |
|-----------------------|--|
| Sumber Data           | <pre>fakultas &lt;- c("Bisnis", "D3 Perhotelan", "ICT", "Ilmu Komunikasi", "Seni dan Desain") jumlah_mahasiswa &lt;- c(260, 28, 284, 465, 735) akreditasi &lt;- c("A", "A", "B", "A", "A")  info_mahasiswa &lt;- data.frame(fakultas, jumlah_mahasiswa, akreditasi) info_mahasiswa</pre> |
| Parameter pada ggplot | <pre>gambar &lt;- ggplot(info_mahasiswa, aes(x=fakultas, y=jumlah_mahasiswa, fill=fakultas))</pre>   |



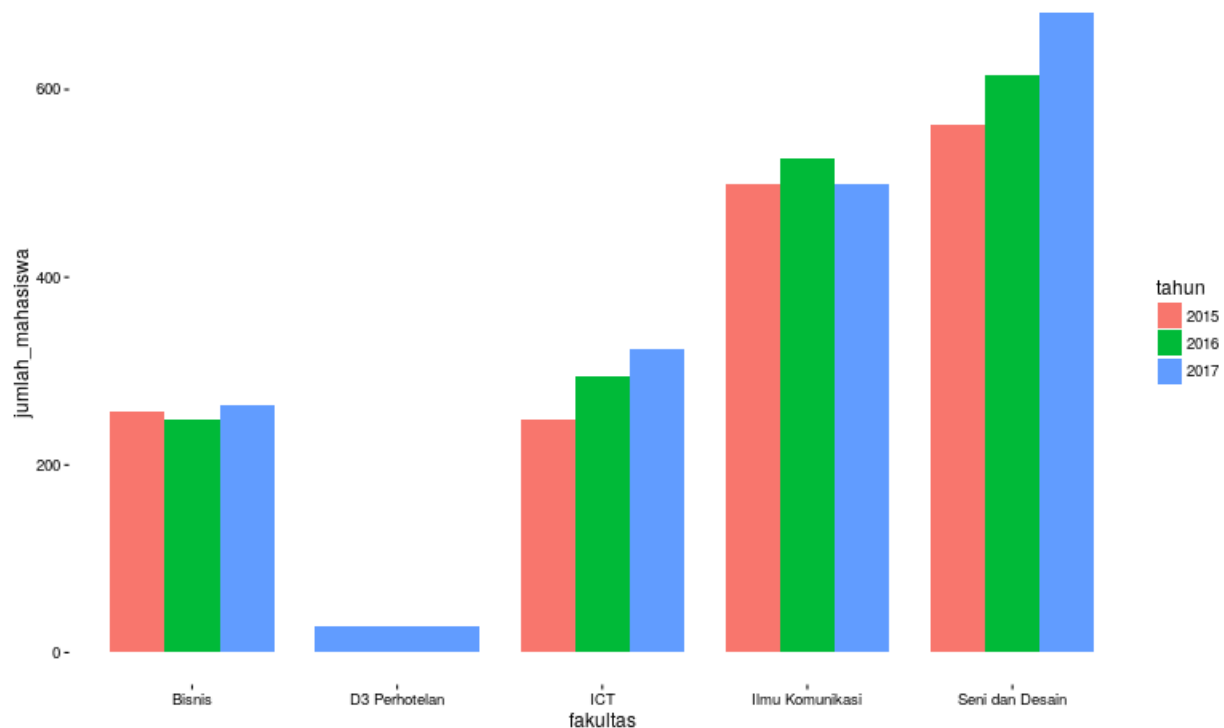
Klik tombol Submit Code untuk melanjutkan ke subbab berikutnya.

# Trend Jumlah Mahasiswa dari Tahun ke Tahun

Grafik histogram sebelumnya cukup baik, tapi informasi yang terkandung masih belum kita eksplorasi lebih detil. Sebagai contoh, bagaimana melihat trend jumlah mahasiswa per fakultas dari tahun ke tahun?

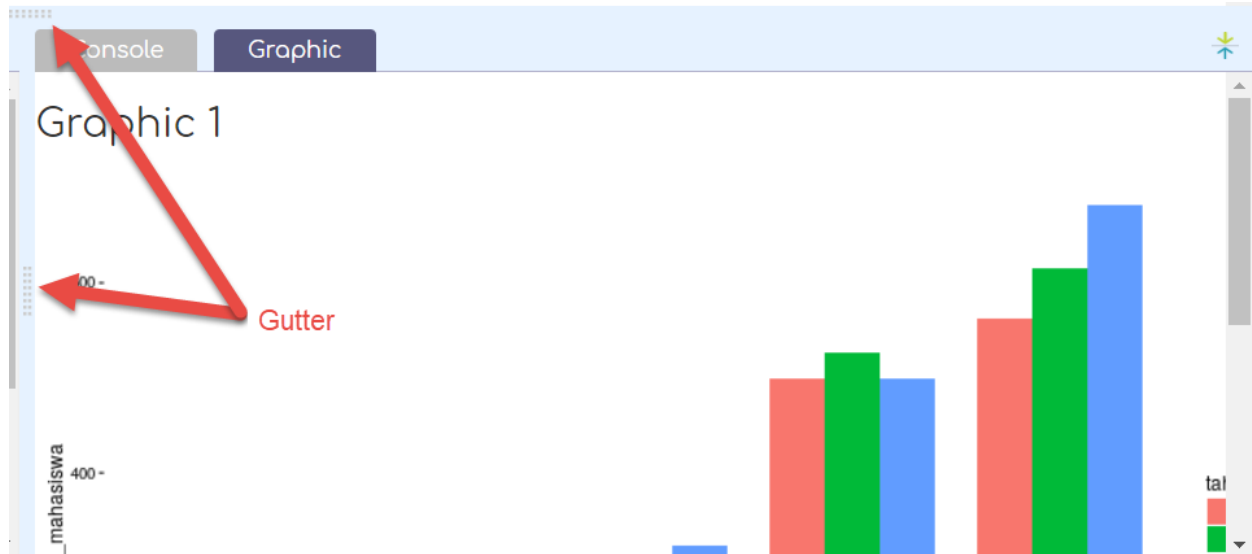
Subbab ini akan menunjukkan hal tersebut. Cobalah jalankan semua code yang telah dilengkapi di code editor, jika berjalan dengan lancar maka akan mendapatkan hasil berikut.

Catatan: Code tidak akan diterangkan pada praktek kali ini, karena melibatkan teknik yang akan kita bahas di course R lain yaitu "Data Preparation and Wrangling with R" dan "Data Visualization with ggplot2 in R".

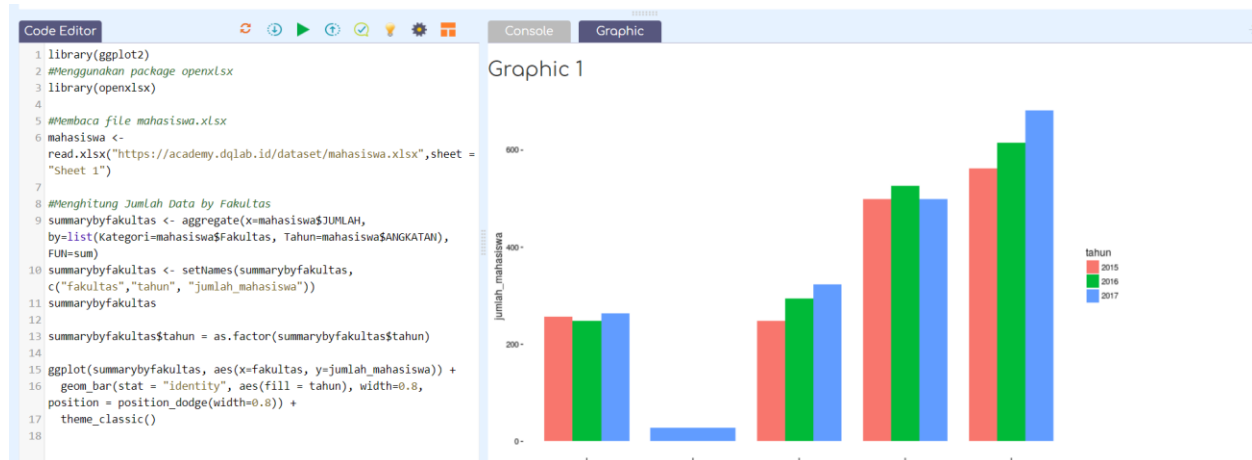


Terlihat dari tahun yang mengalami peningkatan secara konsiten adalah fakultas "ICT" dan "Seni dan Desain". Terlihat juga, fakultas "D3 perhotelan" baru ada di tahun 2017. Untuk fakultas "Bisnis" dan "Ilmu Komunikasi" fluktuatif selama tiga tahun.

Catatan: Jika Anda tidak dapat melihat keseluruhan chart karena keterbatasan panel seperti berikut, maka Anda bisa menggeser *gutter* sehingga dapat gambar penuh (lihat gambar berikutnya).



Berikut adalah gambar penuh dari chart setelah kita mengatur *layout* panel dengan menggeser *gutter*.

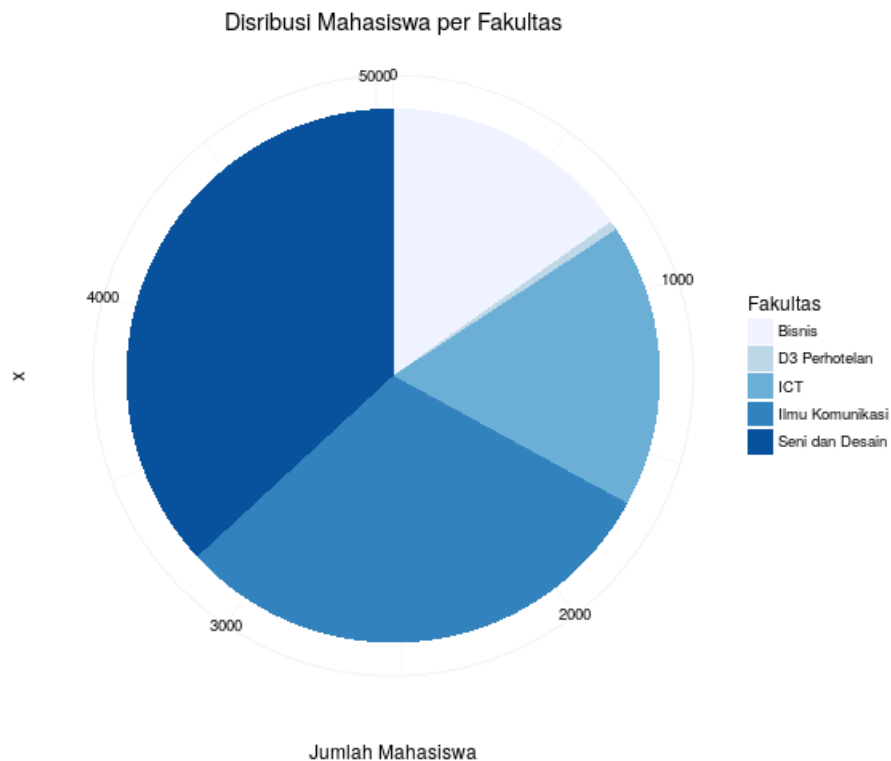




# Pie Chart

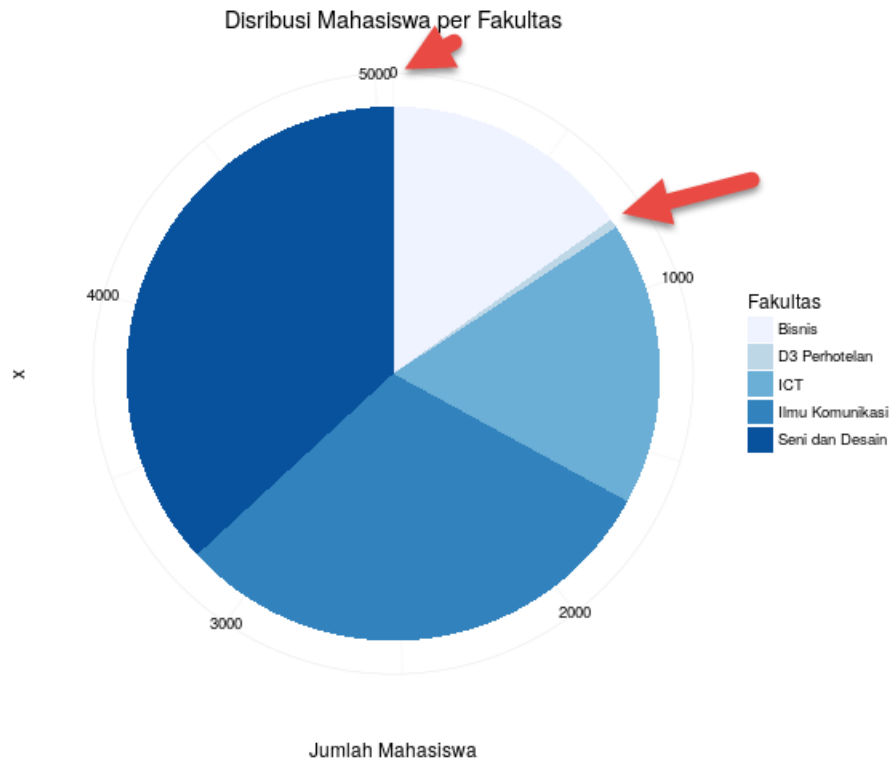
Pie chart masih merupakan chart favorit bagi banyak analis untuk menunjukkan proporsi data. Kebetulan, jumlah mahasiswa per fakultas adalah kasus proporsi yang bisa ditampilkan dengan pie chart.

Pada code editor kita telah diisi dengan code yang menghasilkan pie chart dengan menggunakan fungsi `ggplot` dan `coord_polar`. Cobalah jalankan code tersebut, dimana kita akan mendapatkan hasil berikut.

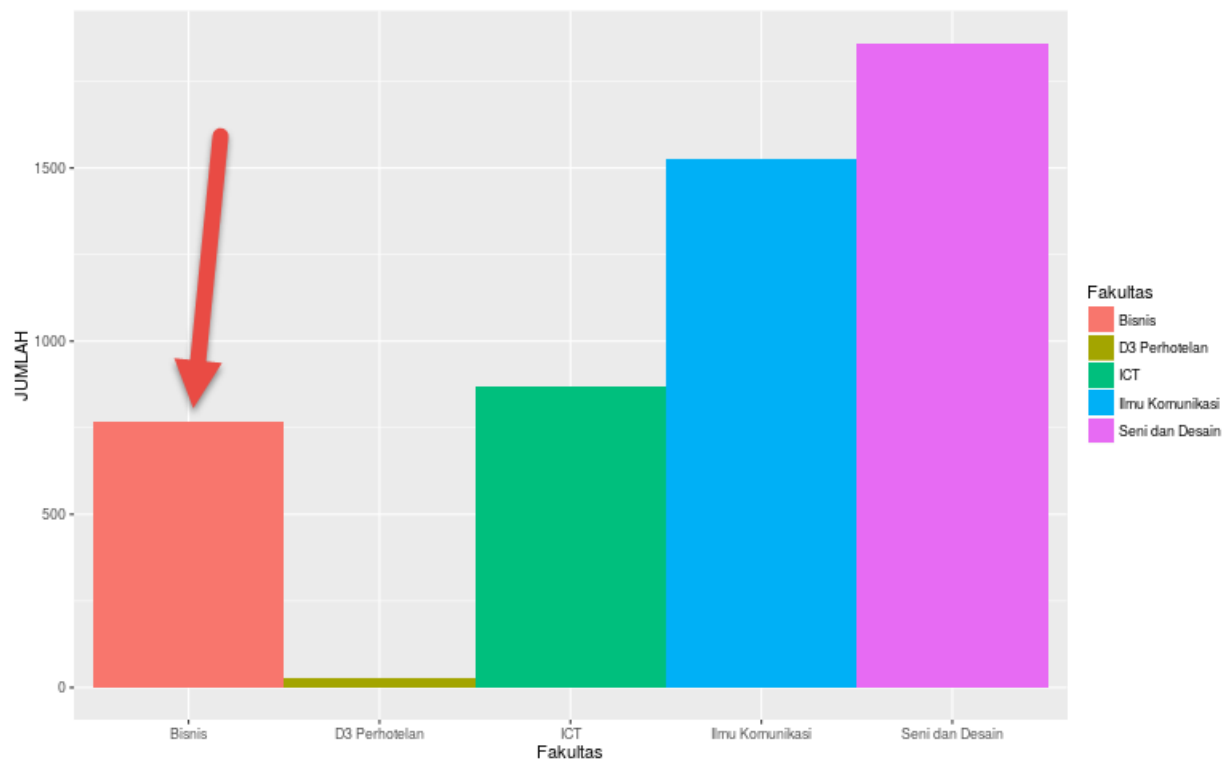


Terlihat porsi fakultas "Seni dan Desain" dan "Ilmu Komunikasi" menempati porsi terbesar. Angka 0 s/d 5000 di luar dari pie chart menunjukkan rentang jumlah mahasiswa secara akumulatif.

Mari kita lihat contoh, mari kita lihat untuk fakultas Bisnis yang menempati porsi pie chart pada angka 0 dan di bawah 1000 (ditunjukkan dengan anak panah merah).



Ini konsisten dengan informasi dari grafik histogram sebelumnya, bahwa jumlah mahasiswa memang di bawah angka 1000.



Dan kalau secara insting dari grafik tersebut, jika kita tambahkan jumlah mahasiswa fakultas "Bisnis", "D3 Perhotelan", dan "ICT" maka totalnya tidak akan mencapai 2000 mahasiswa. Ini juga sudah tergambarkan di pie chart ini.

