

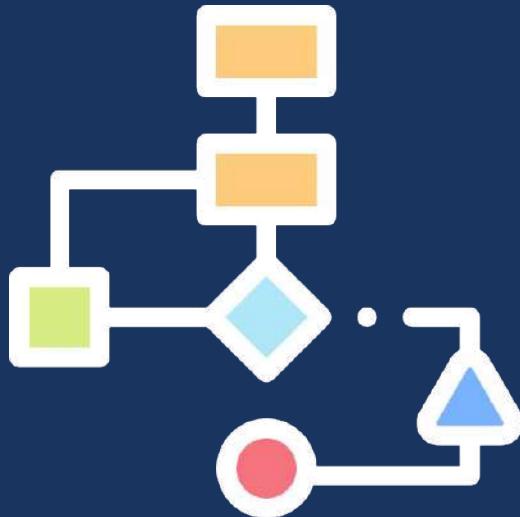
Introduction Algorithm

PART I

OUTLINE



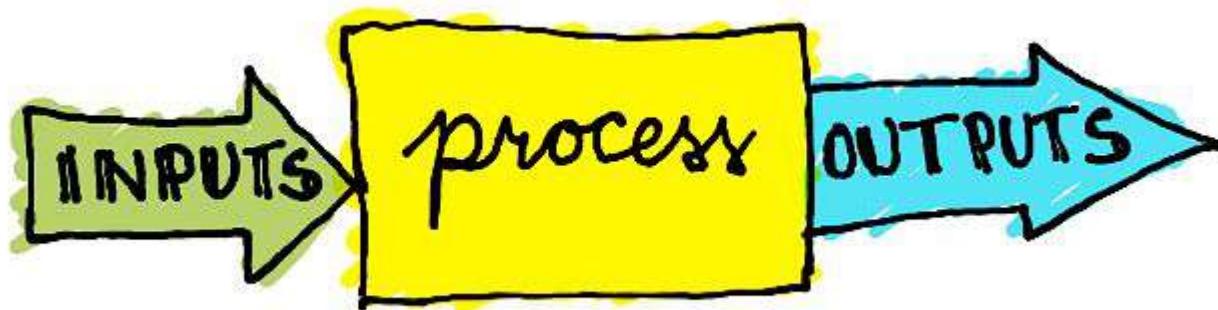
- Apa Itu Algoritma ?
- Karakteristik Algoritma
- Algoritma Dasar
- Pseudocode
- Flowchart



Pengenalan Algoritma



APA ITU ALGORITMA ?



Algoritma adalah **prosedur komputasi** yang didefinisikan dengan baik yang mengambil beberapa **nilai** sebagai **input** dan menghasilkan beberapa nilai sebagai **output**.



CONTOH ALGORITMA



*Check Prime
Number*



Sorting



Searching



KARAKTERISTIK ALGORITMA



Memiliki batas
(Awal dan akhir)



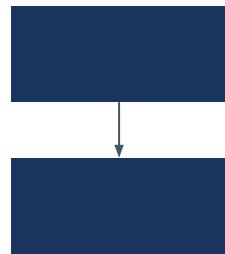
Instruksi terdefinisi
dengan baik



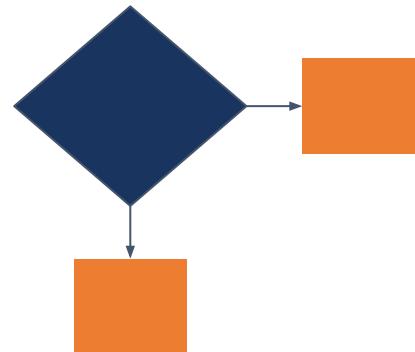
Efektif & Efisien



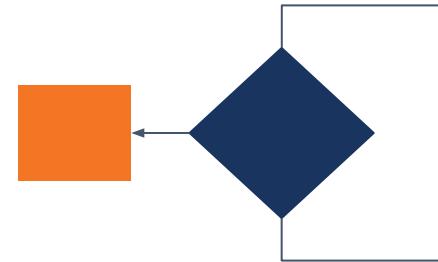
ALGORITMA DASAR



Sequential



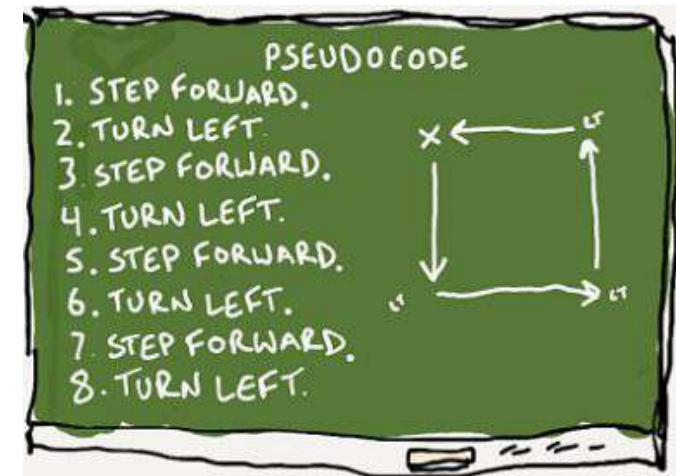
Branching



Looping



Pseudo Code?



KASUS 1

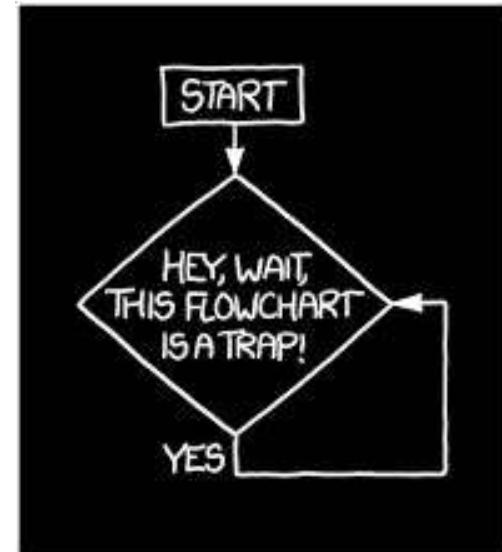
MENGHITUNG LUAS SEGITIGA

1. INPUT Alas and Tinggi
2. CALCULATE Luas = (Alas x Tinggi) / 2
3. PRINT Luas

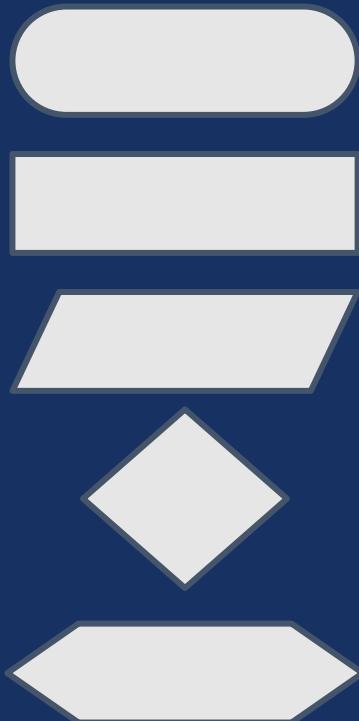


FLOWCHART

Flowchart adalah suatu bagan dengan simbol tertentu yang menggambarkan urutan dan hubungan antar proses secara mendetail.



SIMBOL FLOWCHART



Mulai / Selesai

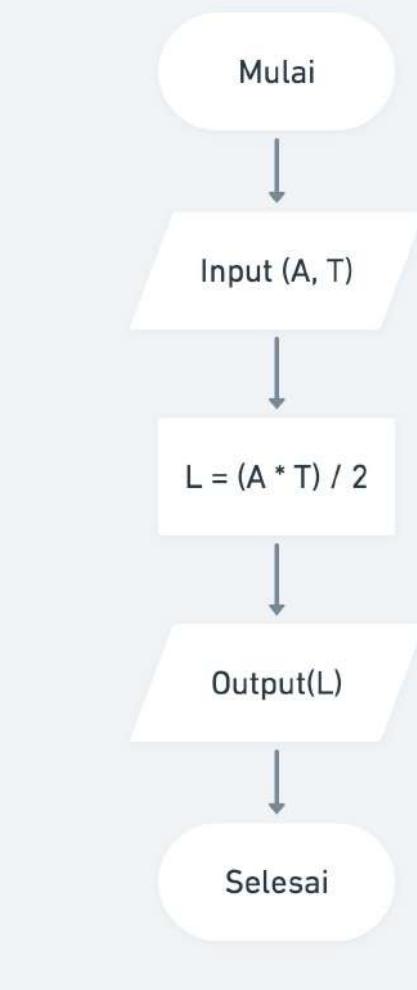
Proses

Input / Output

Perkondision

Looping

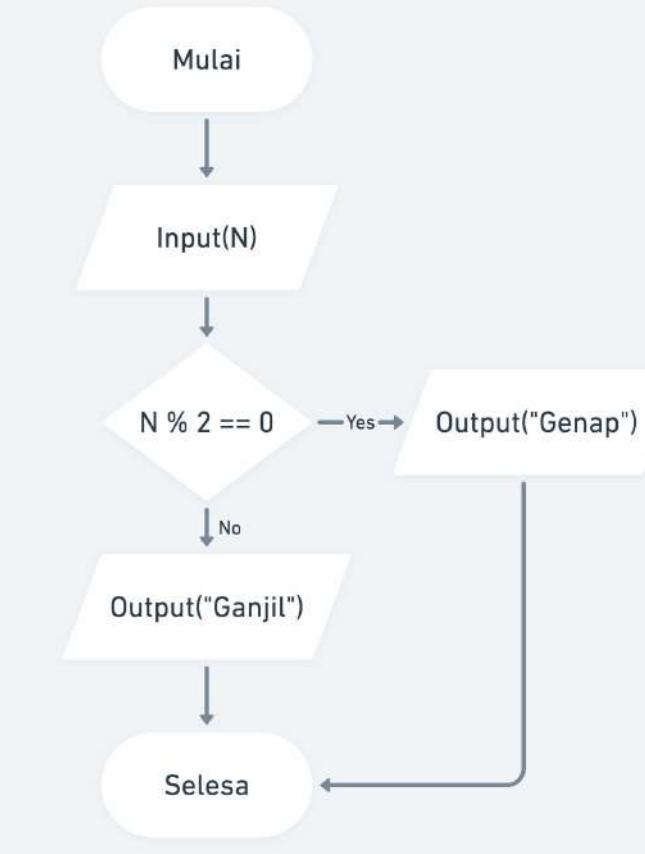
KASUS 1



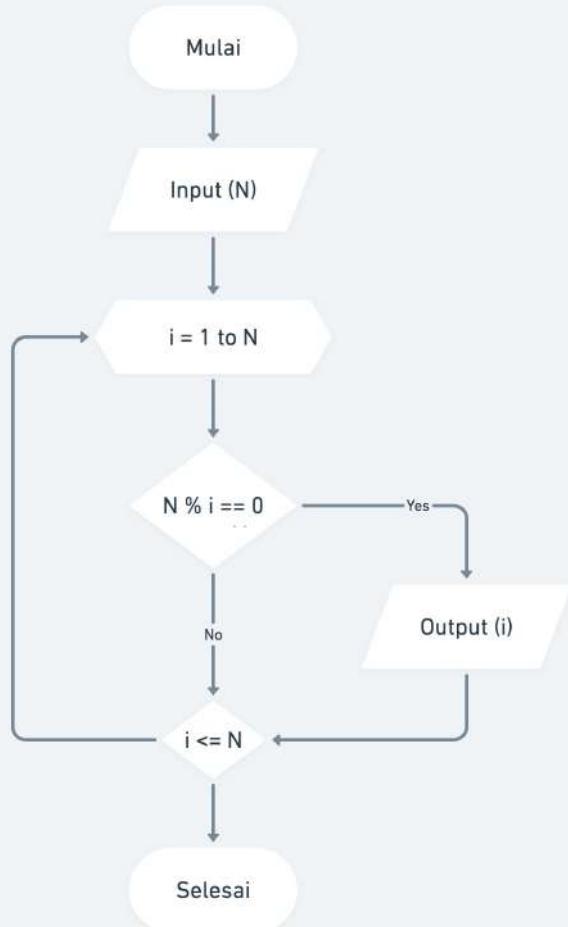
MENGHITUNG LUAS SEGITIGA

KASUS 2

MENENTUKAN BILANGAN GANJIL

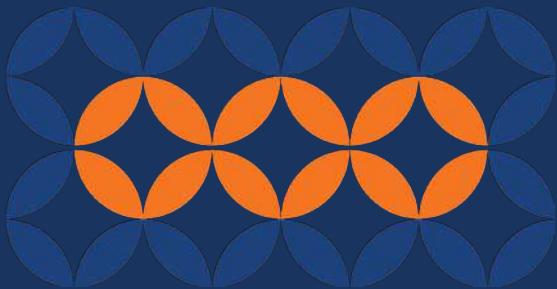


KASUS 3



MENCETAK FAKTOR BILANGAN

LIVE



Version Control and Branch Management (Git)

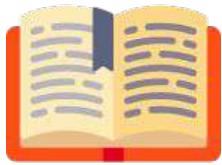


OUR RULES

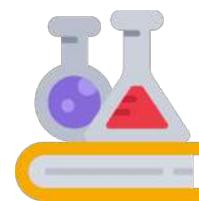




TIME ALLOCATION



Explanation



Challenge



Review

OUTLINE



- What is Versioning?
- Git Install
- Setting Up
 - Git Init, Clone, Config
- Saving Changes
 - Git Status, Add, Commit, Diff, Stash, Ignore
- Inspecting Repository and Undoing
 - Git Log, Checkout, Reset
- Syncing
 - Git Remote, Fetch, Push, Pull
- Branches
- Pull Request
- Workflow Collaboration



Let's get started!





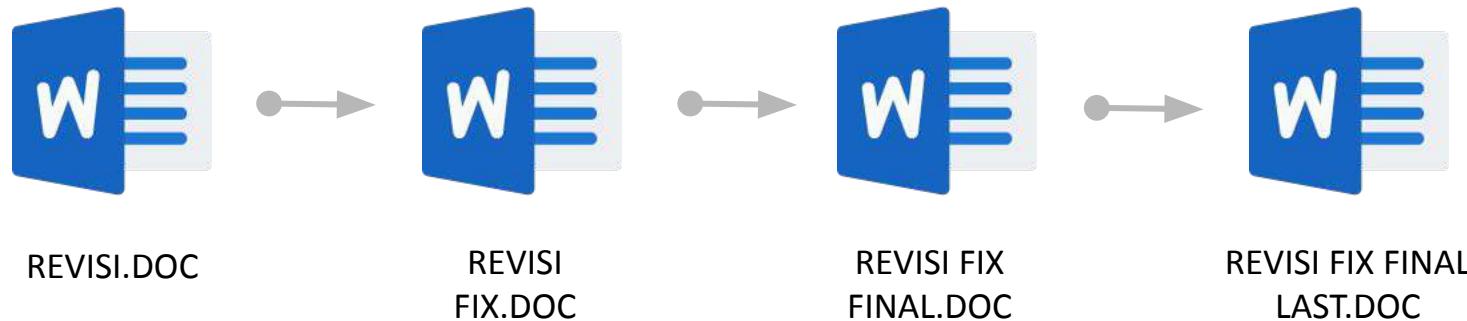
APA ITU VERSIONING?

f x

mengatur versi dari source code program



THE PROBLEM





“

“Revision Is A Must,
Don’t Expect Every Code is Perfect”

TOOLS

Version Control System (VCS)
Source Code Manager (SCM)
Revision Control System (RCS)



VERSION CONTROL SYSTEM

Single User

SCCS - 1972 Unix only
RCS - 1982 Cross platform, text only

Centralized

CVS - 1986 File focus
Perforce - 1995
Subversion - 2000 - track directory structure
Microsoft Team Foundation Server - 2005

Distributed

Git - 2005
Mercurial - 2005
Bazaar - 2005



GIT

Salah satu **version control system** populer yang digunakan para **developer** untuk mengembangkan **software** secara **bersama-bersama**

Real World **Collaboration**



TERDISTRIBUSI

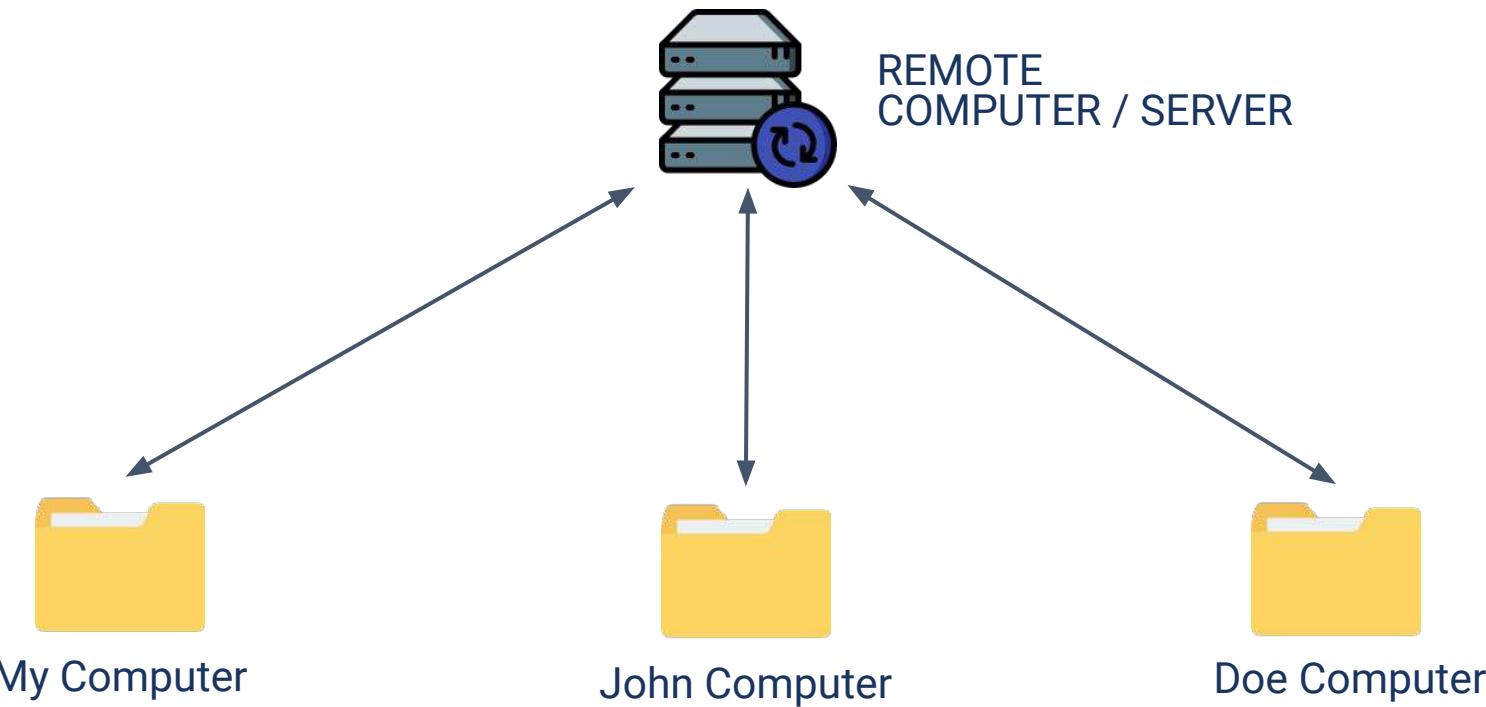
Bukan tersentralisasi

DIBUAT OLEH

Linus Torvalds (2005) - Linux Kernel
<https://github.com/torvalds/linux>

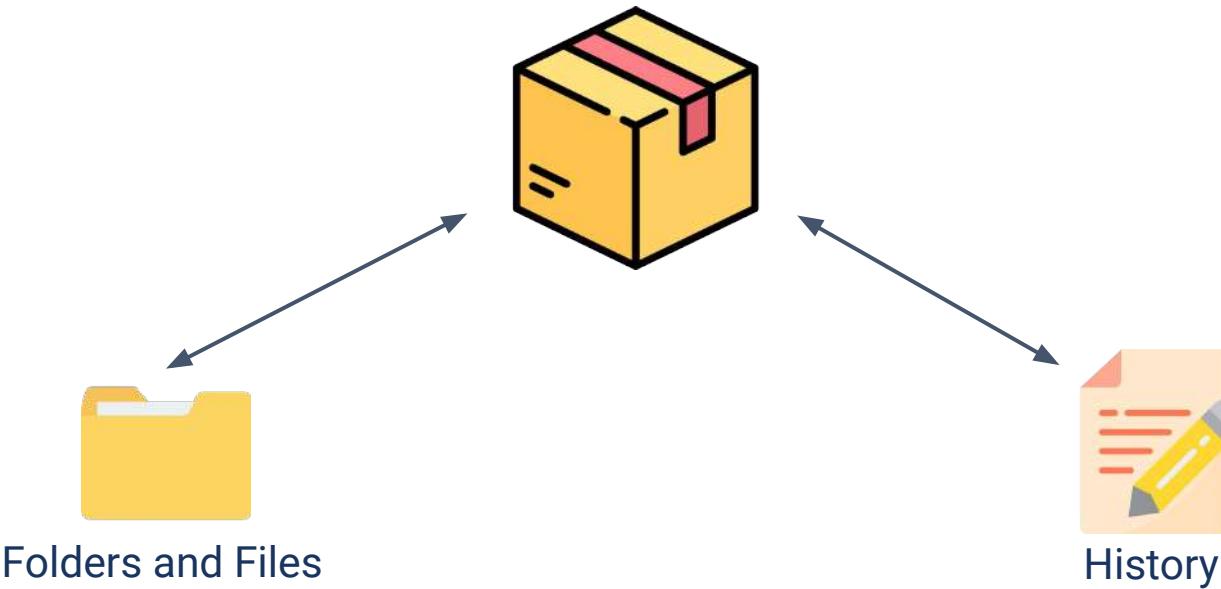


EVERYONE SHOULD SYNC TO THE REMOTE SERVER





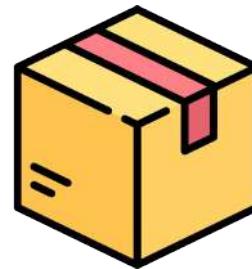
GIT REPOSITORY (FOLDER PROJECT)





Git track every file changes.

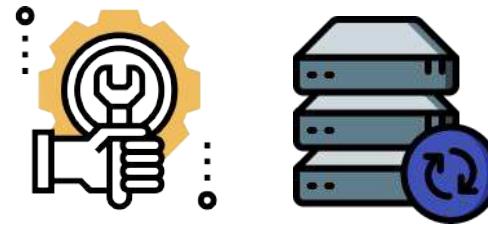
Your changes, John's changes, everyone!



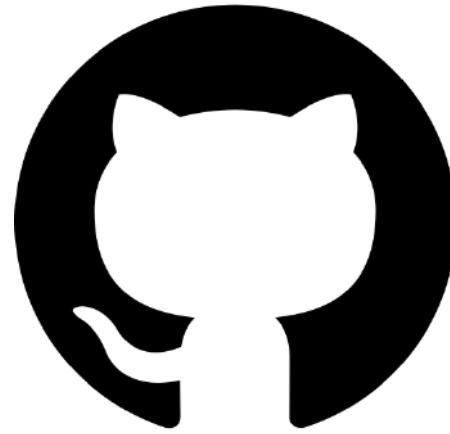
Git can undo to some 'points'
We call it as Commit
Commit = the record of changes



It is quite complicated to setup **git server**,
We need service to be the server!



REMOTE COMPUTER / SERVER



github = git hosting service
Go to github.com and **create new repository!**

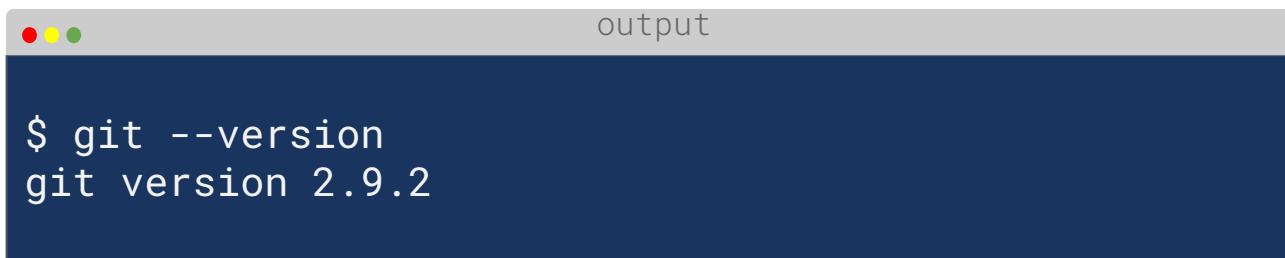


GIT INSTALL



INSTALL GIT ON

1. Download the latest **Git for Mac installer**.
2. Follow the prompts to install Git.
3. Open a terminal and verify the installation was successful by typing: `git --version`:



A screenshot of a Mac OS X terminal window titled "output". The window has the characteristic red, yellow, and green title bar buttons. The main pane contains the command `$ git --version` followed by the output "git version 2.9.2".

```
$ git --version
git version 2.9.2
```



INSTALL GIT WINDOWS



1. Download the latest [Git for Windows installer](#).
2. When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.
3. Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).



INSTALL GIT LINUX

1. From your shell, install Git using apt-get:

```
● ● ●                               output
$ sudo apt-get update
$ sudo apt-get install git
```

2. Verify the installation was successful by typing git --version:

```
● ● ●                               output
$ git --version
git version 2.9.2
```



SETTING UP



GIT INIT, CLONE, CONFIG

```
output

# git config
$ git config --global user.name "John Done"
$ git config --global user.email "johndoe@email.com"

# start with init
$ git init
$ git remote add <remote_name> <remote_repo_url>
$ git push -u <remote_name> <local_branch_name>

# start with existing project, start working on the project
$ git clone ssh://john@example.com/path/to/my-project.git
$ cd my-project
```



SAVING CHANGES



THE STAGING AREA





GIT STATUS, ADD, COMMIT

```
● ● ●          output

$ git status

$ git add <directory>
$ git add hello.py
$ git add .

$ git commit -m "add config file"
```



COMMIT MESSAGE

“If applied, this commit will *your subject line here*”

For example:

- If applied, this commit will *refactor subsystem X for readability*
- If applied, this commit will *update getting started documentation*
- If applied, this commit will *remove deprecated methods*
- If applied, this commit will *release version 1.0.0*

Notice how this doesn't work for the other non-imperative forms:

- If applied, this commit will *fixed bug with Y*
- If applied, this commit will *changing behavior of X*
- If applied, this commit will *more fixes for broken stuff*
- If applied, this commit will *sweet new API methods*



GIT DIFF AND STASH

```
● ● ●          output

# git diff
# change file
# add staging area
$ git diff --staged

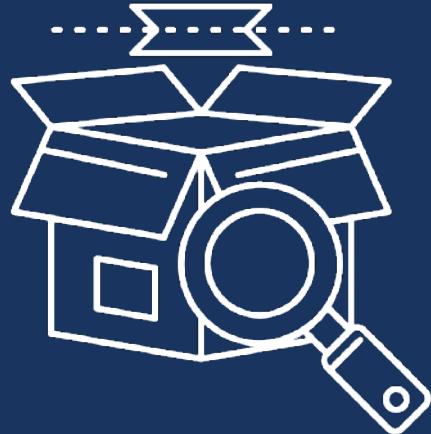
# stashing your work
$ git stash

# re-applying your stashed changes
$ git stash apply
```



FILE .gitignore

Pattern	Example matches	Explanation*
*.log	Debug.log foo.log .log logs/debug.log	An asterisk is a wildcard that matches zero or more characters.
logs	logs logs/debug.log logs/latest/foo.bar build/logs build/logs/debug.log	If you don't append a slash, the pattern will match both files and the contents of directories with that name. In the example matches on the left, both directories and files named <i>logs</i> are ignored



INSPECTING REPOSITORY



GIT LOG, CHECKOUT

```
output

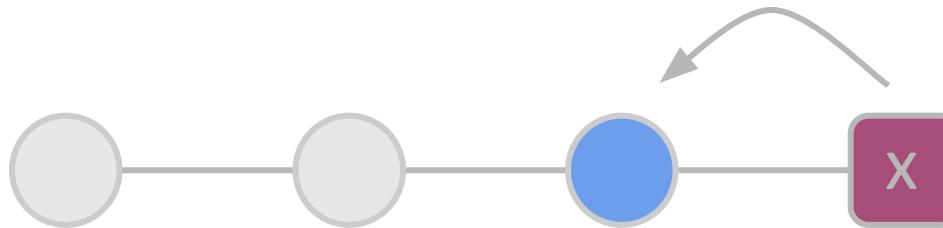
# viewing an old revision
$ git log --oneline

# b7119f2 Continue doing crazy things
# 872fa7e Try something crazy
# a1e8fb5 Make some important changes to hello.txt

$ git checkout a1e8fb5
```



GIT RESET



--soft	--hard
uncommit changes, changes are left staged (index).	uncommit + unstage + delete changes, nothing left.



GIT RESET

```
output

# viewing an old revision
$ git log --oneline

# b7119f2 Continue doing crazy things
# 872fa7e Try something crazy
# a1e8fb5 Make some important changes to hello.txt

$ git reset a1e8fb5 --soft
```



SYNCING



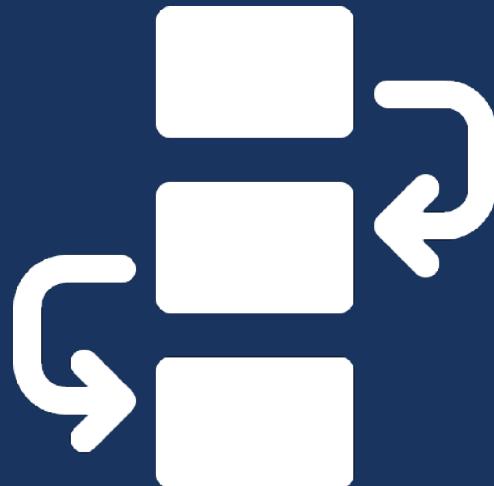
GIT PUSH, FETCH & PULL

```
output

# git remote
$ git remote -v
$ git remote add origin http://dev.example.com/john.git

# fetch and pull
$ git fetch
$ git pull origin master

# push
$ git push origin master
$ git push origin feature/login-user
```



BRACHES



GIT BRANCHING

output

```
# show all branch
$ git branch --list

# create a new branch called <branch>
$ git branch <branch>

# force delete the specified branch
$ git branch -D <branch>

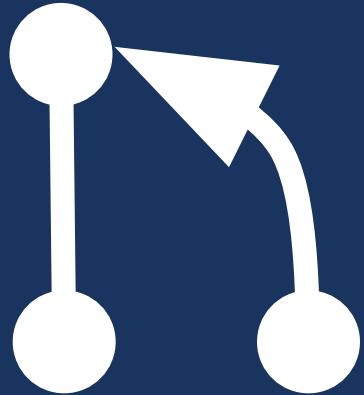
# list remote branch
$ git branch -a
```



GIT MERGE

output

```
# Start a new feature
$ git checkout -b new-feature master
# Edit some files
$ git add <file>
$ git commit -m "Start a feature"
# Edit some files
$ git add <file>
$ git commit -m "Finish a feature"
# Merge in the new-feature branch
$ git checkout master
$ git merge new-feature
$ git branch -d new-feature
```



PULL REQUEST



PULL REQUEST

iswanulumam / learn-git

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Label issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with good first issue

Dismiss

Filters ▾ is:pr is:open

Labels 9 Milestones 0

New pull request

The screenshot shows a GitHub repository named 'iswanulumam / learn-git'. The 'Pull requests' tab is selected, indicated by an orange underline. A modal window is open, titled 'Label issues and pull requests for new contributors', explaining that GitHub will help potential first-time contributors discover issues labeled with 'good first issue'. Below the modal, there are standard GitHub navigation links for Code, Issues, Projects, Wiki, Security, Insights, and Settings. At the bottom, there are filters for 'is:pr is:open', a 'Labels' section (9), a 'Milestones' section (0), and a prominent green button labeled 'New pull request' which is enclosed in a red dashed box.



In case of fire



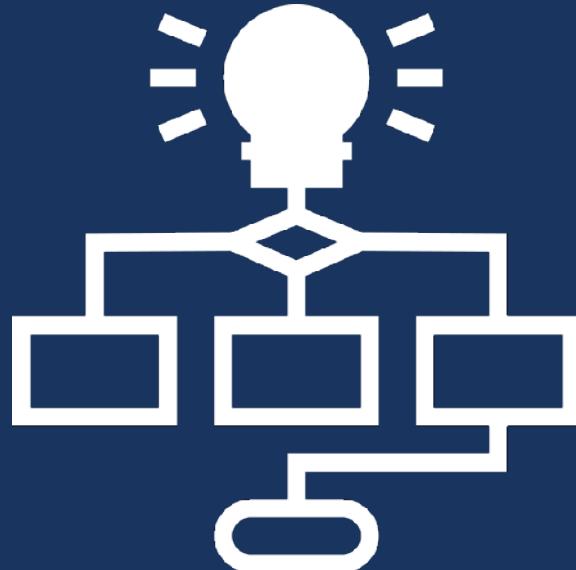
1. git commit



2. git push



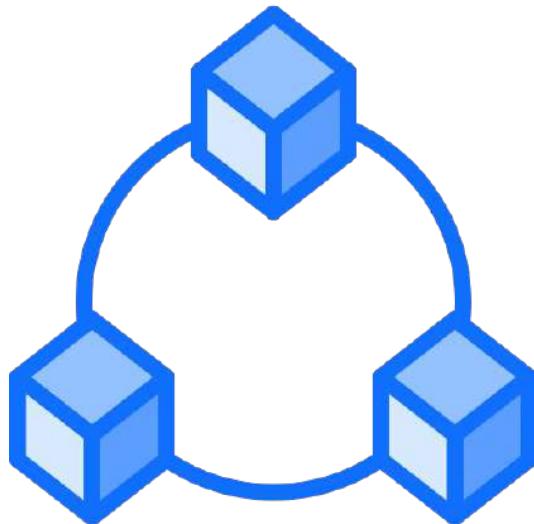
3. leave building



WORKFLOW COLLABORATION

Objectives

**Understand the proper
workflow to collaborate
with GitHub or Gitlab**



How to optimize your workflow with **GitHub or Gitlab?**



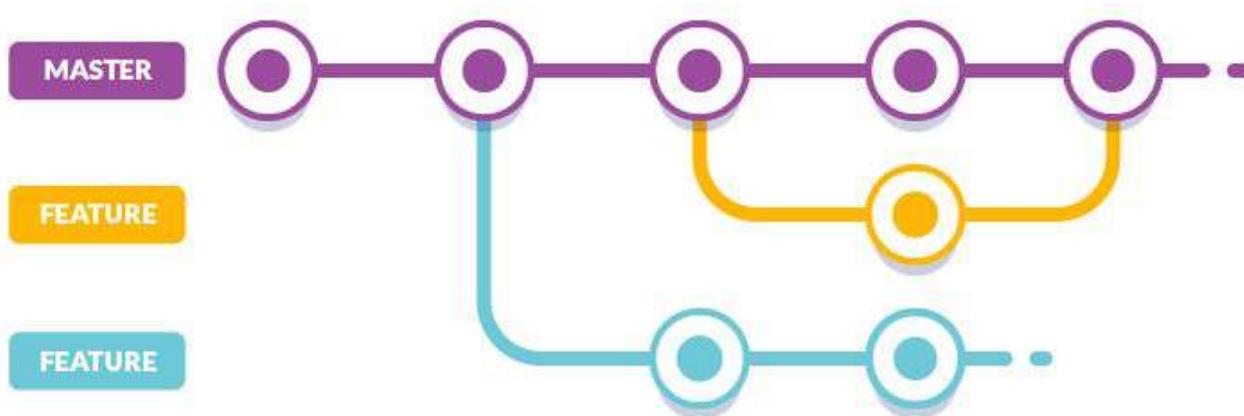
DO YOU WORK LIKE THIS ?

MASTER





OR LIKE THIS ?





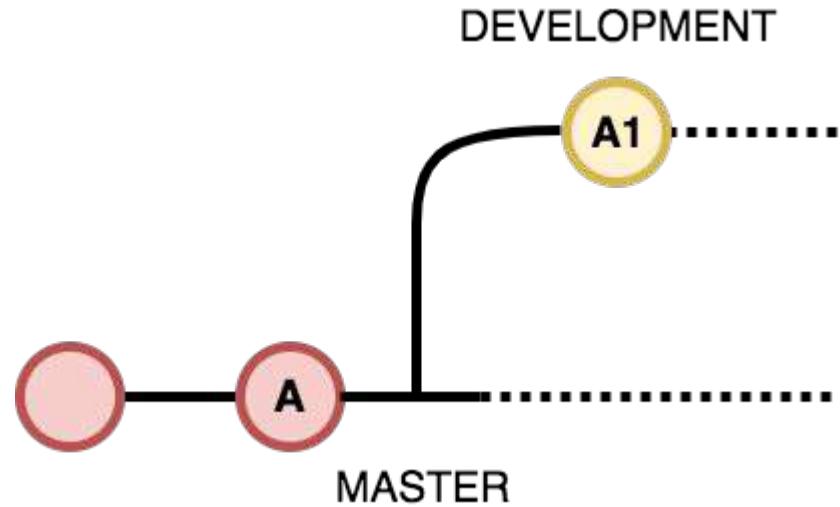
THE BEST WAY LIKE THIS





1

LET
THE MASTER
BRANCH
UNDISTURBED

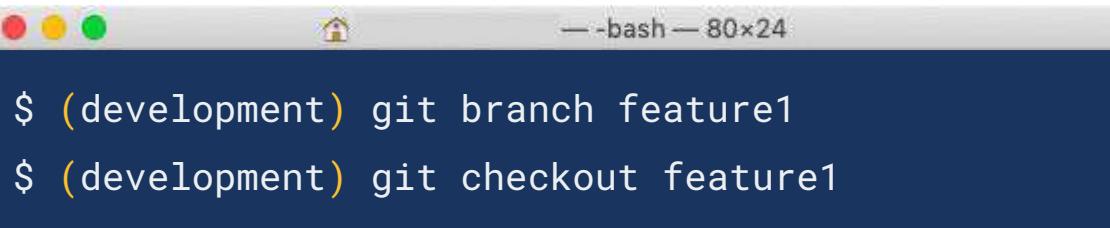
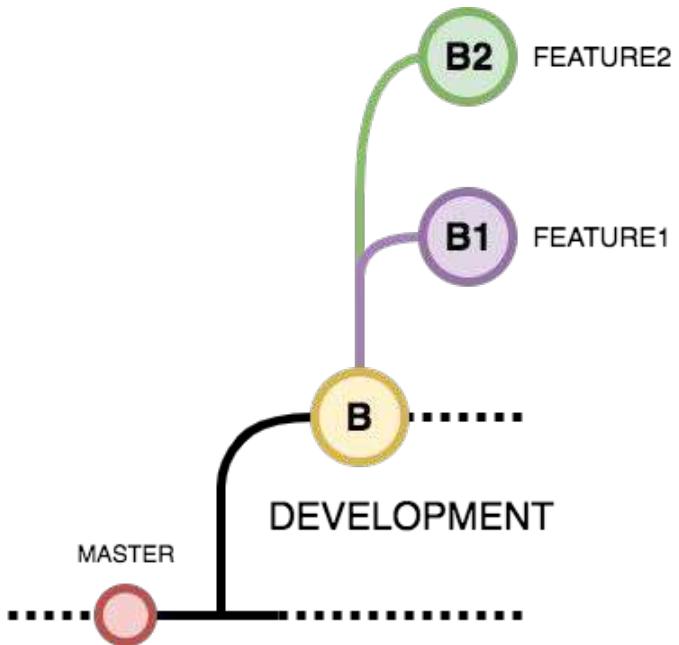


```
$ (master) git branch development
$ (master) git checkout development
```



2

**AVOID
DIRECT EDIT
ON
DEVELOPMENT**



A screenshot of a macOS terminal window. The title bar shows "—bash— 80x24". The terminal content is:

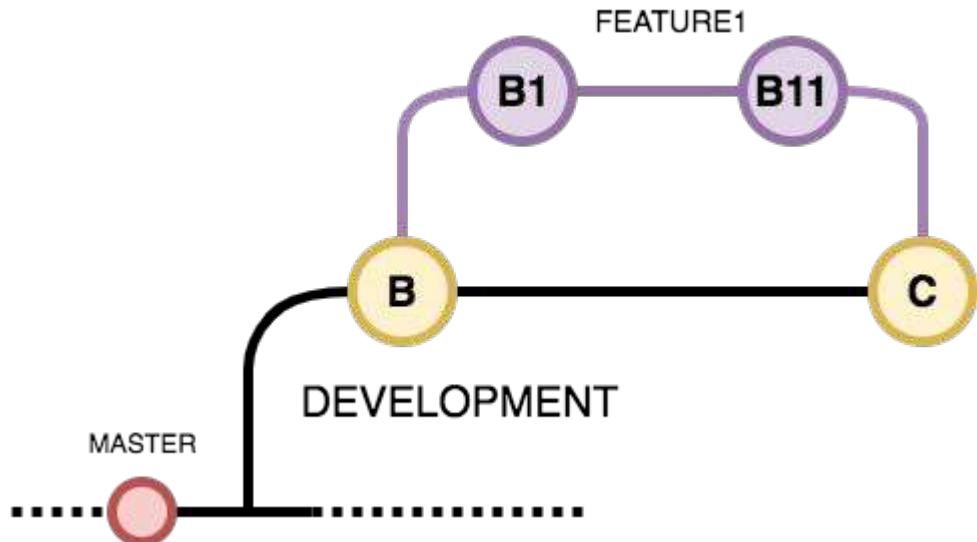
```
$ (development) git branch feature1
$ (development) git checkout feature1
```



3



APPLY THE
FEATURE TO
DEVELOPMENT
ONLY



```
$ (feature1) git checkout development
$ (development) git merge feature1
```

*) git rebase will prevent you from conflicts when merging feature branch to development branch



4

APPLY
DEVELOPMENT
TO MASTER
WHEN IT'S
DONE



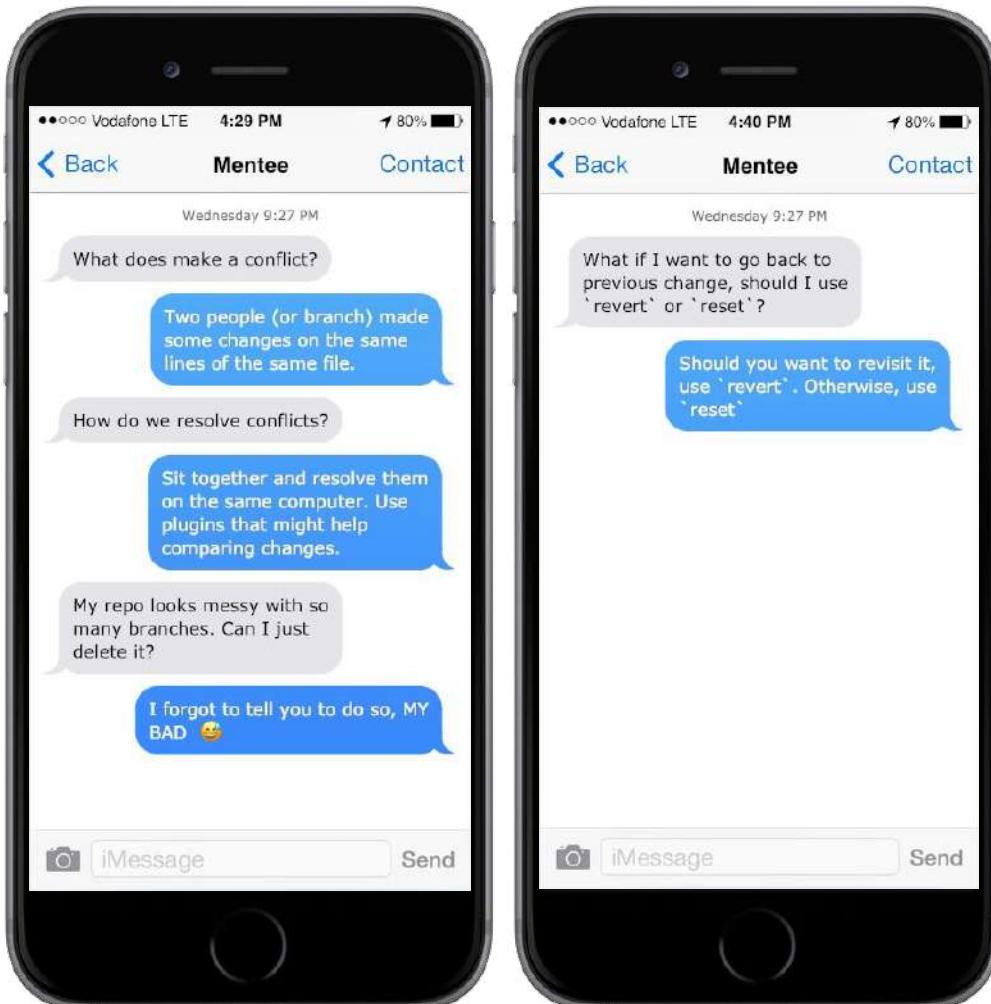
A screenshot of a terminal window with a dark blue background. The window title bar says "— bash — 80x24". The command "\$ (master) git merge development" is typed in the terminal.

```
$ (master) git merge development
```

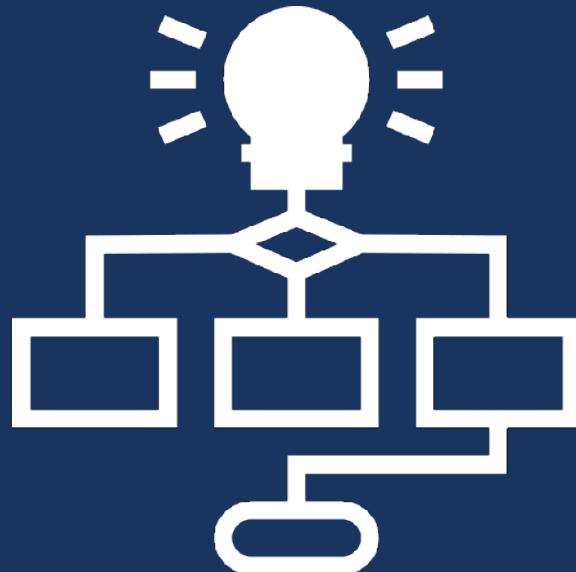
ANY QUESTION



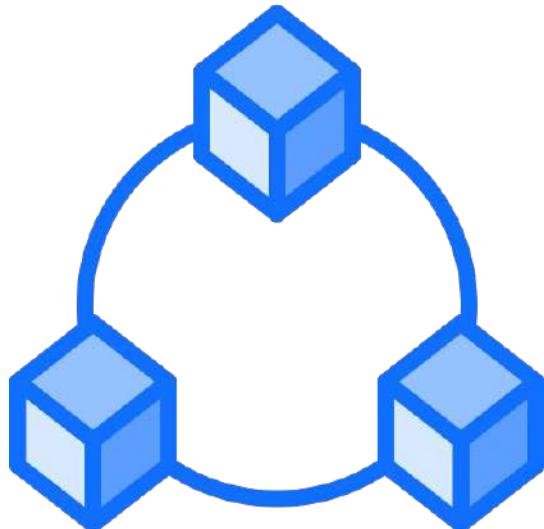
Frequently Asked Question



***The Best Way of
Learning about Anything
is by Doing***



WORKFLOW COLLABORATION



**Bagaimana cara
mengoptimalkan
kolaborasi dalam
development?**

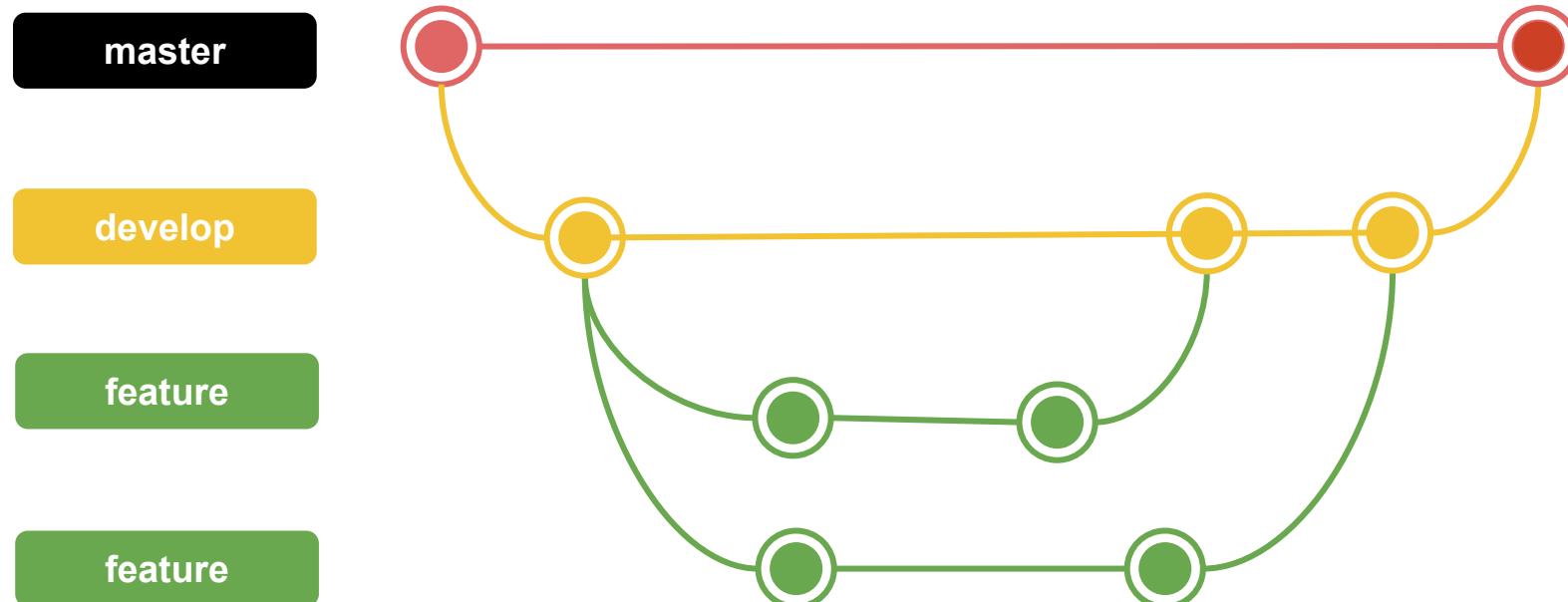


**dalam kolaborasi tim kita tidak bisa
hanya bekerja dalam satu branch**





perlu dibuat beberapa branch agar kolaborasi dapat berjalan dengan optimal





1

**BUAT
BRANCH MASTER
DARI BRANCH
DEVELOPMENT**



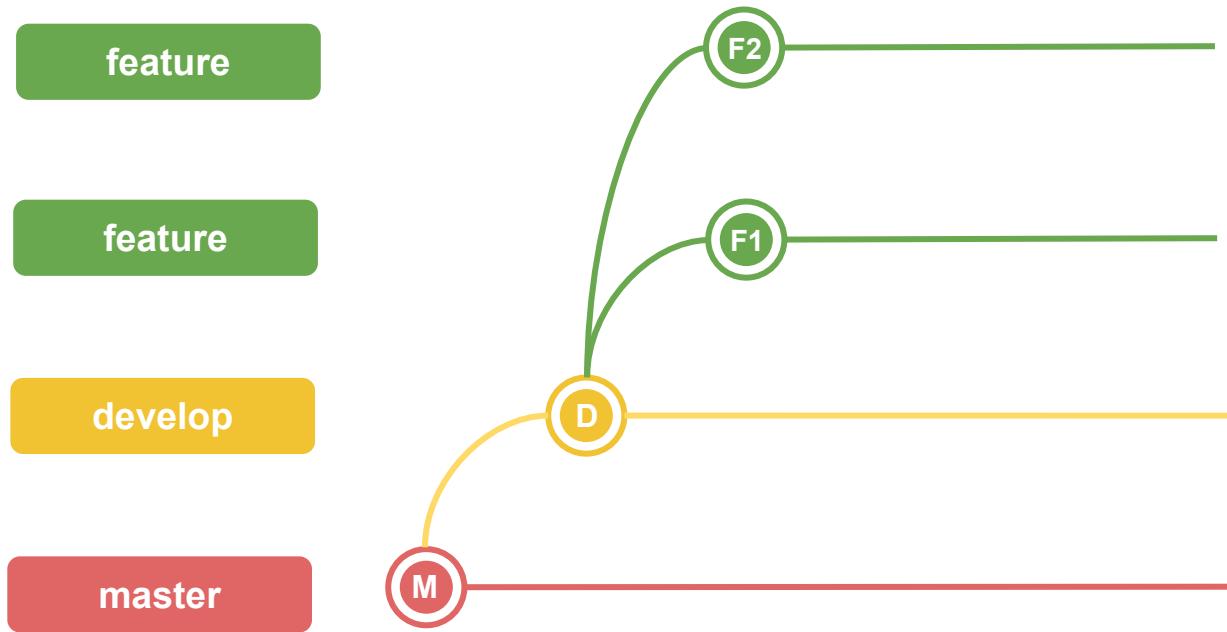
A screenshot of a terminal window titled '-bash- 80x24'. The window contains the following text:

```
$ (master) git branch development
$ (master) git checkout development
```



2

**HINDARI
DIRECT EDIT
KE BRANCH
DEVELOPMENT**



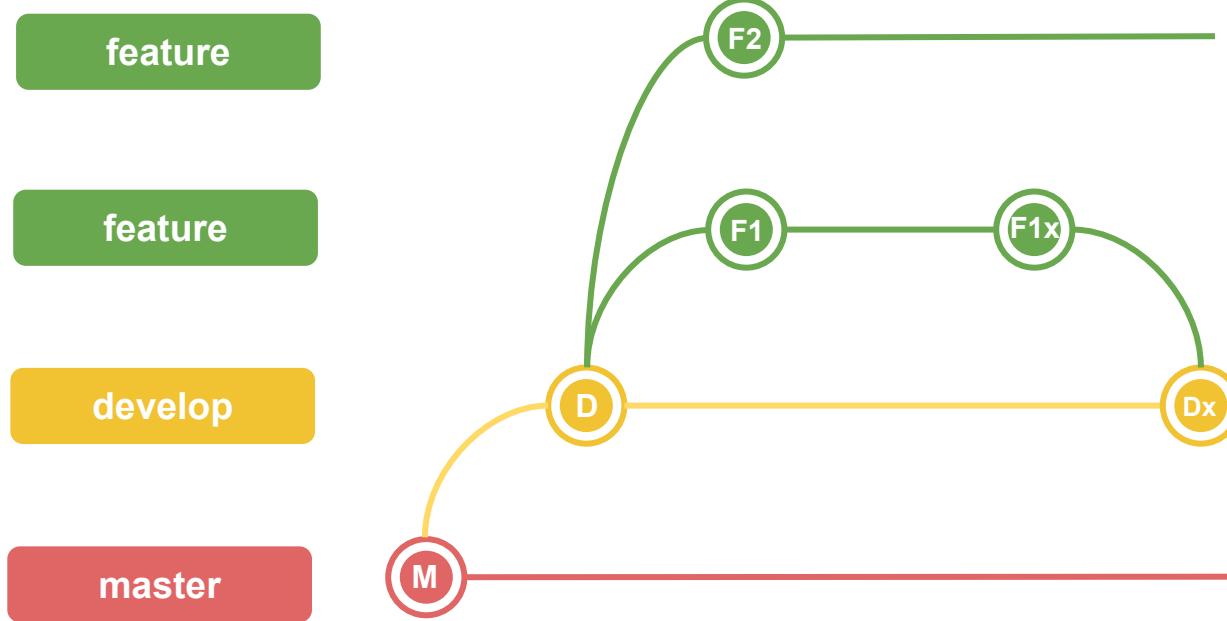
master

```
$ (development) git branch feature1  
$ (development) git checkout feature1
```



3

**MERGE
BRANCH FEATURE
HANYA KE
BRANCH DEVELOPMENT**



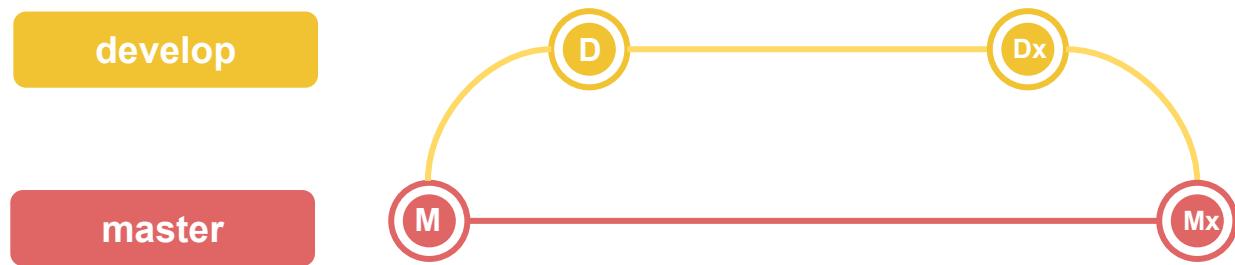
```
$ (feature1) git checkout development
$ (development) git merge feature1
```

*) git rebase akan mencegah konflik ketika merge dari branch feature ke branch development



4

MERGE BRANCH
DEVELOPMENT
KE BRANCH MASTER
JIKA DEVELOPMENT
SELESAI



```
$ (master) git merge development
```

FAQ

Apa yang membuat git conflict?

Ketika ada dua orang yang membuat perubahan di file dan baris yang sama

Bagaimana cara resolve git conflict?

Resolve dengan satu computer, gunakan plugin untuk membandingkan perubahan

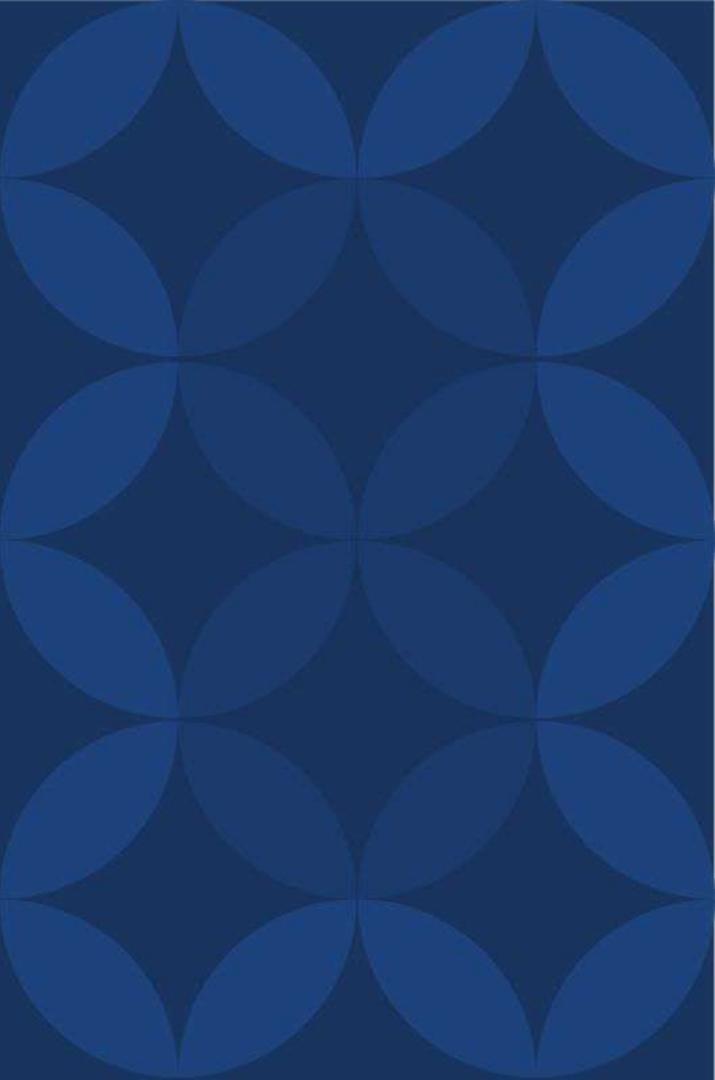
Jika terlalu banyak branch, apakah branch boleh dihapus?

Ya tentu boleh, hapus saja branch yang sudah lama.

***The Best Way of
Learning about Anything
is by Doing***

Tugas

1. Buat sebuah repository di Github
2. Implementasikan penggunaan branching yang terdiri dari master, development, featureA, dan featureB
3. Implementasikan intruksi git untuk push, pull, stash dan merge
4. Implementasikan sebuah penanganan **conflict** di branch developement ketika setelah merge dari branch featureA lalu merge dari branch featureB (Conflict bisa terjadi jika kedua branch mengerjakan di file dan line code yang sama)
5. Gunakan merge no fast forward
6. Kirimkan alamat repository github di :



Any Question



Bahasa Pemrograman Dart

Outline

- Apa dan Mengapa Dart?
- Fungsi Main
- Komentar
- Variabel
- Tipe Data
- Operator

Apa itu Dart?



Bahasa pemrograman yang dirancang untuk membuat aplikasi agar dapat berjalan dengan cepat.



Apa itu Dart? (contd)

Aplikasi yang dibuat dengan Dart adalah aplikasi yang berjalan disisi client (frontend), seperti:



Web
(Javascript)



Desktop
(Dart)



Mobile
(Dart)

Kenapa (harus) Dart?



- **Type safe**
menjamin konsistensi tipe data
- **Null safety**
memberi keamanan dari data bernilai hampa (null)
- **Rich standard library**
hadir dengan banyak dukungan library internal
- **Multiplatform**
mampu berjalan pada berbagai jenis perangkat

Dasar Pemrograman Dart

Menampilkan Teks

Hal sederhana yang dapat dilakukan menggunakan Dart



Selamat Datang di Bahasa Pemrograman Dart!



Program Dart Sederhana

- Fungsi Main
 - Bagian yang dijalankan pertama kali
 - Dapat memiliki tipe data void atau int
 - Memiliki nama **main**
- Perintah **print** digunakan untuk menampilkan data ke layar



```
void main() {
    print('Selamat Datang di Bahasa Pemrograman Dart!');
}
```

Komentar

Deskripsi

Baris kode yang tidak dijalankan dengan tujuan:



memberi catatan
pada kode



mencegah perintah
dijalankan

Menulis Komentar (cara 1)

Diawali dengan tanda garis miring 2 kali.



```
void main() {  
    // ini komentar  
    // selalu diawali tanda garis miring 2 kali  
    print('Selamat Datang di Bahasa Pemrograman Dart!');  
}
```

Menulis Komentar (cara 2)

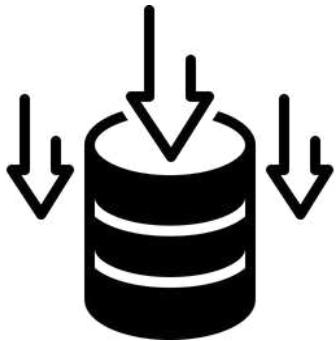
- diawali dengan tanda garis miring dan bintang (/*)
- diakhiri dengan tanda bintang dan garis miring */)



```
void main() {  
    /*  
     ini komentar  
     selalu diawali tanda garis miring 2 kali  
    */  
    print('Selamat Datang di Bahasa Pemrograman Dart!');  
}
```

Variabel

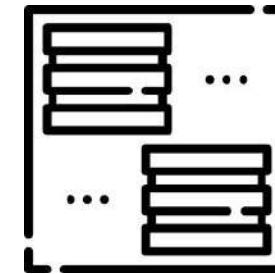
Deskripsi



Digunakan untuk menyimpan data



Memiliki nama

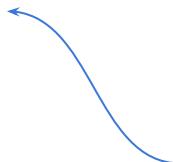


Memiliki tipe data

Deklarasi Variabel

Proses pembuatan variabel

Menentukan tipe data



int score;



Memberi nama

Deklarasi Variabel (contd)

Menggunakan var dan diikuti nama variabel secara otomatis tipe data variabel tersebut adalah Null.



```
void main() {  
    var usia;  
}
```



Memberi Nilai pada Variabel

Menggunakan tanda sama dengan (=).

```
● ● ●  
void main() {  
  
    var usia;  
  
    usia = 18;  
  
}
```

Inisiasi Variabel

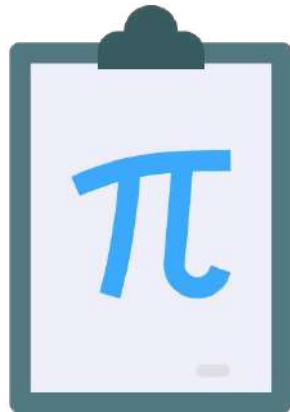
Deklarasi dan memberi nilai variabel secara bersamaan.



```
void main() {  
  
    var usia = 18;  
  
}
```

Konstanta

Deskripsi



- Digunakan untuk menyimpan data
- Memiliki nama
- Memiliki tipe data
- Nilai tetap (tidak dapat diubah)

Membuat Konstanta

Menggunakan final, diikuti nama, dan diberikan nilai secara langsung.



```
void main() {  
  
    final usia = 18;  
  
}
```

Tipe Data

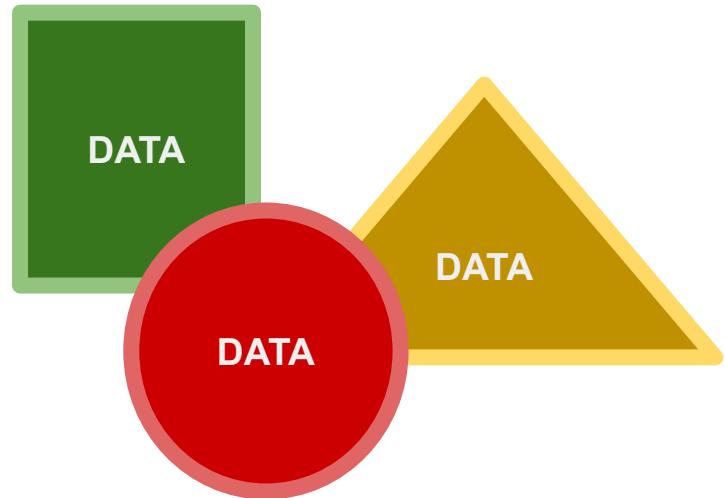
Deskripsi

- Jenis data yang dapat dikelola
- Tipe data sederhana disebut primitive data type



Jenis

- **int**
bilangan bulat
- **double**
bilangan pecahan
- **bool**
true/false
- **string**
teks



int

Menggantikan var dengan int dan data harus bilangan bulat.



```
void main() {  
  
    int usia = 18;  
  
}
```

double

- Menggantikan var dengan double dan data harus bilangan pecahan
- Penulisan pecahan menggunakan tanda titik



```
void main() {  
  
    double beratBadan = 50.5;  
  
}
```

bool

Menggantikan var dengan bool dan data harus true atau false



```
void main() {  
  
    bool pria = false;  
  
}
```

string

- Menggantikan var dengan string dan data berupa teks
- Penulisan teks diawali dan diakhiri tanda kutip

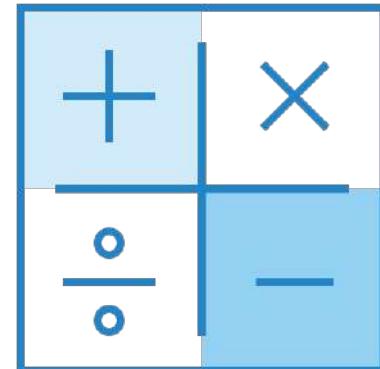


```
void main() {  
  
    String nama = 'Alex';  
  
}
```

Operator

Deskripsi

- Digunakan untuk operasi pengolahan data
- Data yang dikelola disebut operand



Arithmetic

untuk perhitungan matematis
+, -, *, /, %



```
void main() {  
  
    print(1 + 2);  
  
}
```

Assignment

memberi nilai pada variabel
=, +=, -=, *=, /=, %=



```
void main() {  
  
    var usia = 18;  
    usia = 19;  
  
}
```

Comparison

membandingkan kesetaraan nilai

`==, <, <=, >, >=`



```
void main() {  
  
    print(1 == 2);  
  
}
```

Logical

menggabungkan beberapa kondisi
&&, ||, !



```
void main() {  
  
    print(1 == 2 && 2 == 1);  
  
}
```



Terima Kasih

Task

1. Implementasikan rumus luas lingkaran pada program Dart!

 2. Buatlah 3 buah variabel yang berisi string, lalu sambungkan seluruh string tersebut, dan tampilkan pada layar!
- 

A large, solid yellow circle centered on the page, serving as a background element for the title text.

Introduction Data Structures

Outline

- Mengapa Belajar Data Structures?
- Data Structures
- Array
- Set
- Kesimpulan
- Latihan



[Download Material](#)

Mengapa Belajar Data Structures?

Deskripsi

- Mengerti kualitas kode (Pemeliharaan kode)
- Efisiensi kode

Kedua kode fungsi memiliki tujuan yang sama, yaitu menampilkan angka genap dari 2 sampai 100.



```
void printNumberVersionOne() {  
    int number = 2;  
    while (number <= 100) {  
        // Jika number adalah genap, maka akan ditampilkan pada console.  
        if (number % 2 == 0) {  
            print(number);  
        }  
        number += 1;  
    }  
}
```

Looping 100x



```
void printNumberVersionTwo() {  
    int number = 2;  
    while (number <= 100) {  
        print(number);  
        // Number langsung ditambah 2  
        // Otomatis angka berikutnya adalah genap.  
        number += 2;  
    }  
}
```

Looping 51x

Data Structures

Deskripsi

- Data adalah cakupan luas mengacu pada semua tipe informasi yang tersimpan dalam memori komputer (string, number, boolean, dll.)
- Data Structure adalah bagaimana menyimpan dan mengorganisir data/value/element di dalam memori komputer
- Data dapat dikelola dengan berbagai cara

Data string tersimpan dalam 3 variabel x,y,z

```
● ● ●  
String x = 'Halo, ';  
String y = 'Selamat ';  
String z = 'Ulang tahun' ;  
  
print(x + y + z);
```

Data string tersimpan dalam 1 variabel array

```
● ● ●  
List array = ['Halo, ', 'Selamat ', 'Ulang tahun'];  
  
print(array[0] + array[1] + array[2]);
```

Basic Data Structures

- Array
- Set



```
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
array.add('Anggur');

print(array); // console: [Anggur, Belimbing, Ceri, Durian, Erbis, Anggur]
```

Array

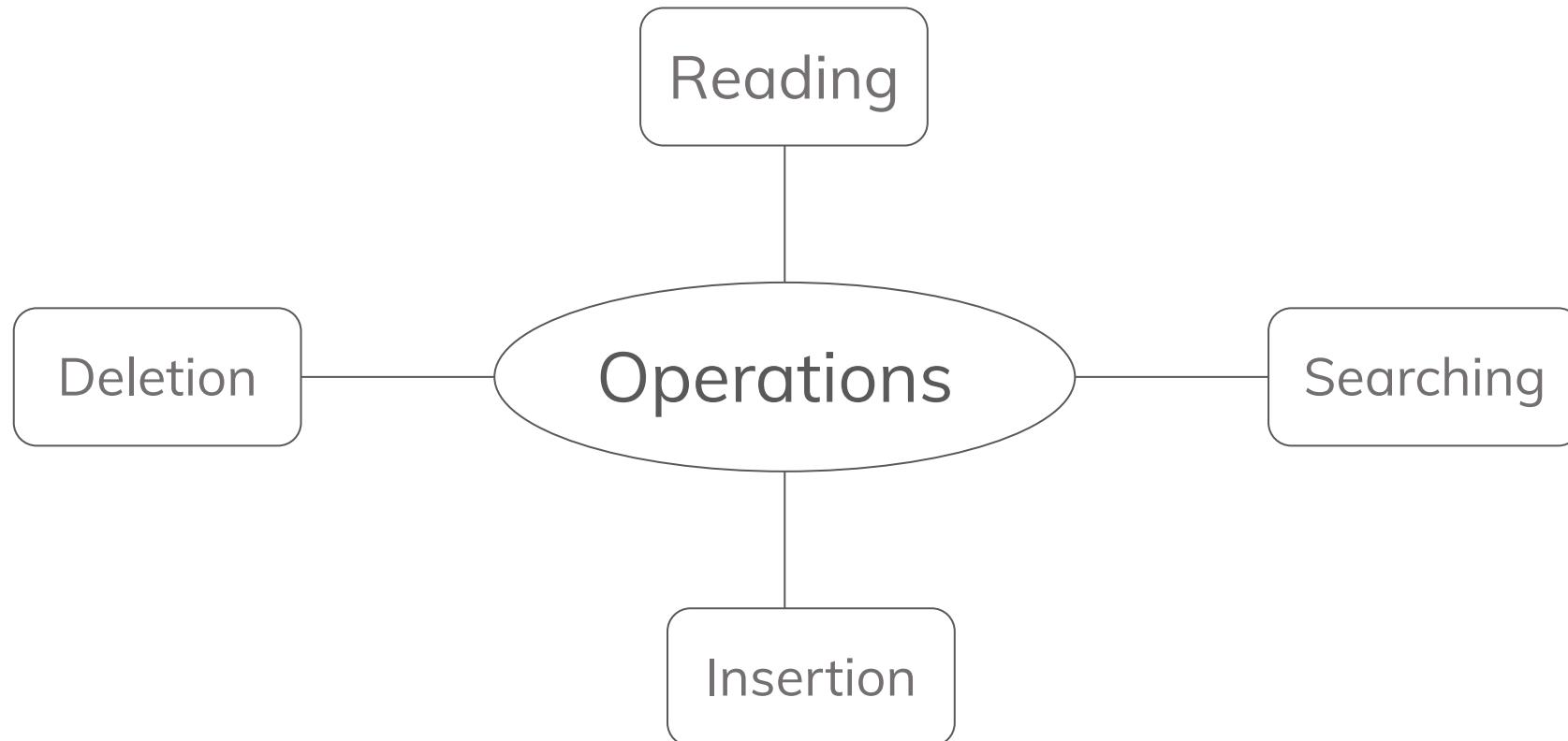


```
var sets = {'Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'};
sets.add('Anggur');

print(sets); // console: {Anggur, Belimbing, Ceri, Durian, Erbis}
```

Set: tidak
mengizinkan
duplikasi

Operasi Data Structure



Mengukur Kecepatan Operasi

- Seberapa banyak langkah atau steps yang diperlukan

Operasi A



5 Steps

Operasi B



50 Steps

- Operasi A lebih cepat

Array

Deskripsi

- Data structure paling dasar
- List dari elemen data
- Serbaguna



```
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
```

Mengandung elemen data string sebanyak 5 value

'Anggur'	'Belimbing'	'Ceri'	'Durian'	'Erbis'
Indeks 0	Indeks 1	Indeks 2	Indeks 3	Indeks 4

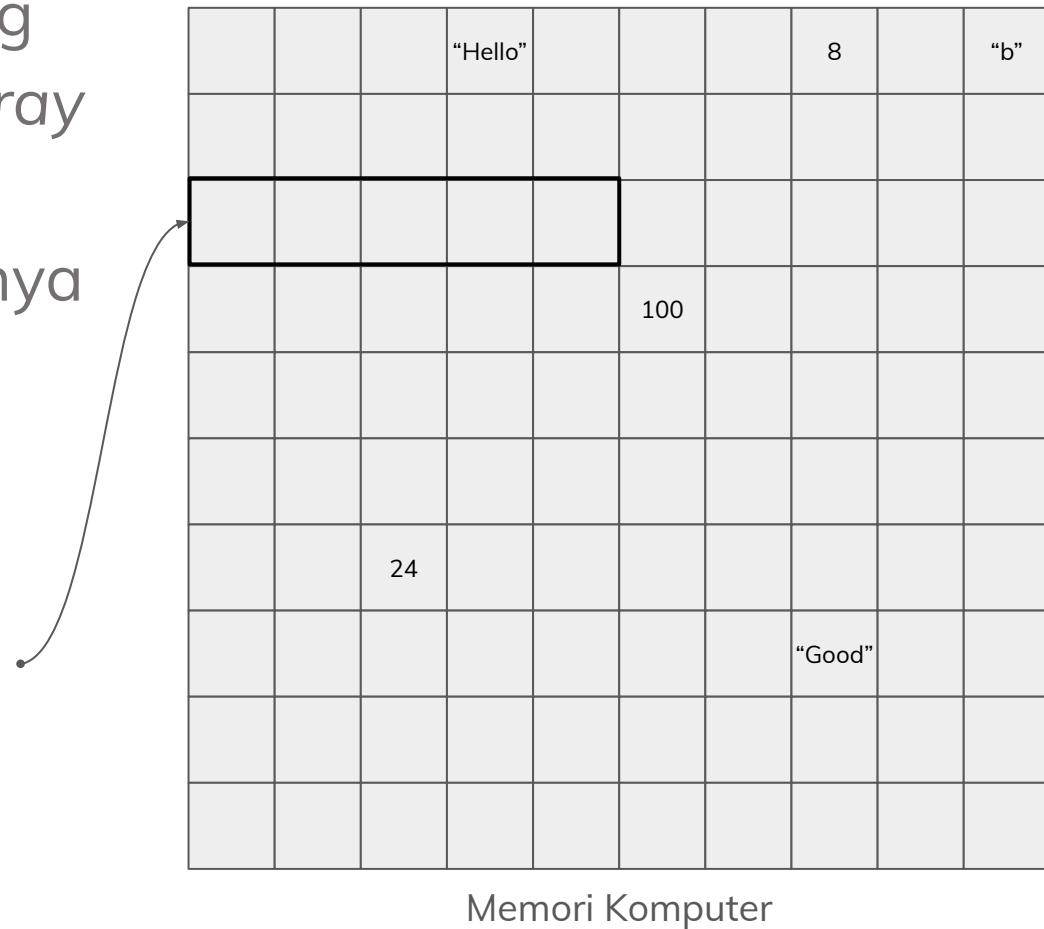
Memiliki indeks yang mengidentifikasi dimana posisi elemen berada

Reading

- Melihat nilai apa yang terkandung pada indeks tertentu di dalam array
- Operasi yang efisien
- Jenis operasi tercepat karena hanya membutuhkan 1 steps

Ketika mendeklarasikan array maka memori komputer akan mengalokasikan cells kosong secara berdekatan.

Disini kita membuat array yang memiliki 5 elemen



Reading

- Setiap cell pada memori memiliki alamat spesifik yang diwakili dengan angka
- Angka yang dimiliki lebih besar dari cell sebelumnya

'Anggur'	'Belimbing'	'Ceri'	'Durian'	'Erbis'
----------	-------------	--------	----------	---------

Indeks:

0 1 2 3 4

Alamat
memori:

1020 1021 1022 1023 1024

1000	1001	1002	1003	1004	1005	1006	1007	1008	1009
1010	1011	1012	1013	1014	1015	1016	1017	1018	1019
1020	1021	1022	1023	1024	1025	1026	1027	1028	1029

1090	1091	1092	1093	1094	1095	1096	1097	1098	1099
------	------	------	------	------	------	------	------	------	------

Memori Komputer

Reading

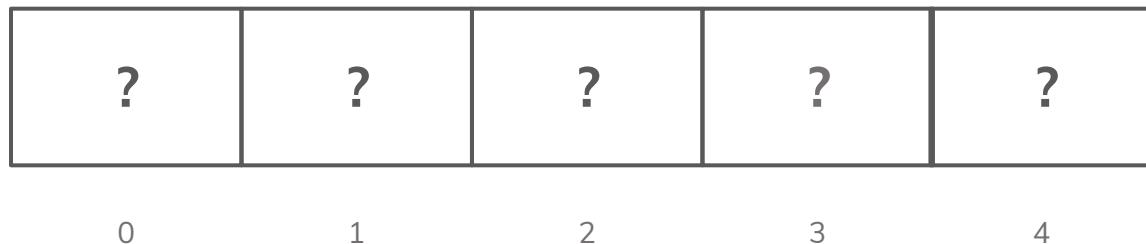
- Ketika melihat sebuah value pada indeks tertentu memori komputer dapat langsung melompat ke indeks tersebut, karena:
 1. Komputer bisa melompat ke alamat memori manapun dalam satu steps
 2. Ketika mengalokasikan sebuah array, komputer tahu dimana alamat memori array dimulai



```
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
var durian = array[3];
print(durian); // console: Durian
```

Searching

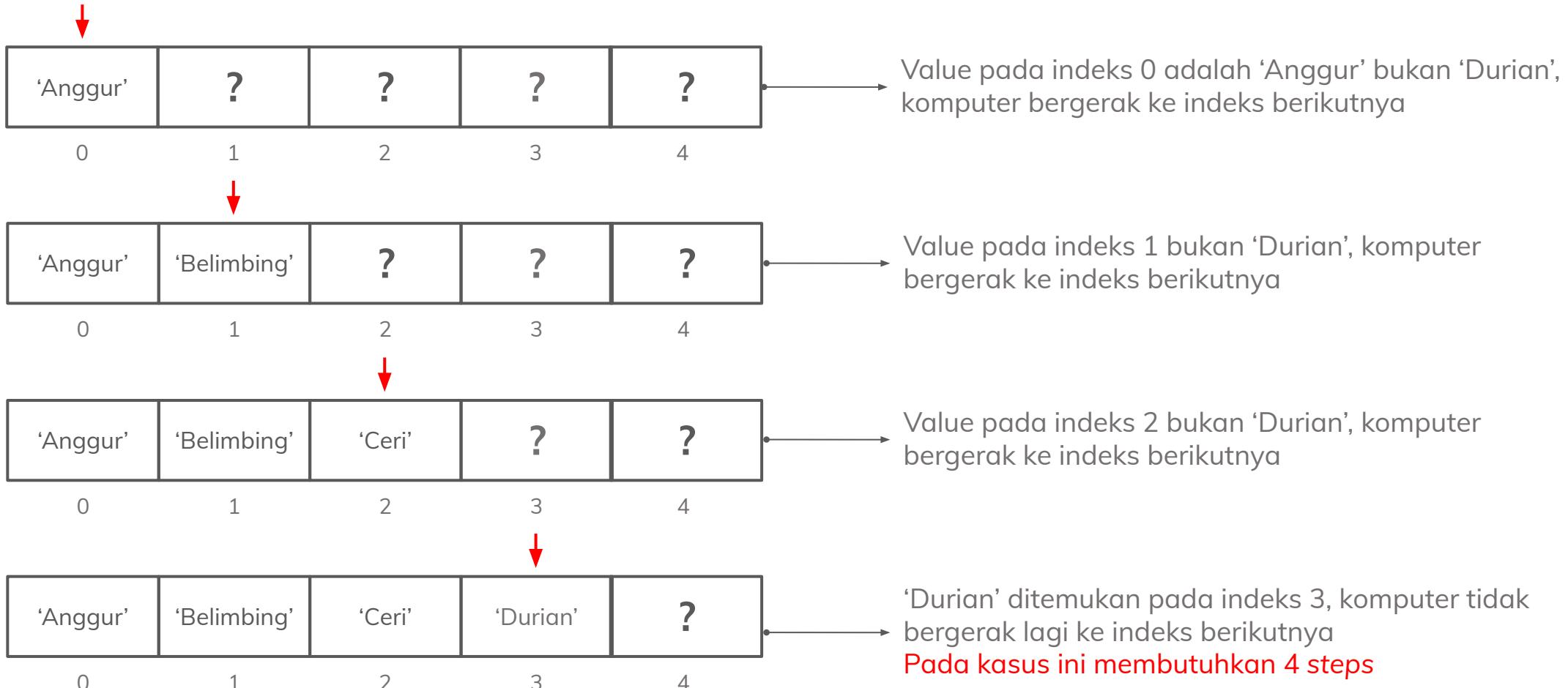
- Mencari apakah nilai tertentu ada didalam array atau tidak
- Kebalikan dari Reading
- Ketika melakukan operasi searching pada array komputer tidak memiliki cara untuk melompat langsung ke nilai tertentu
- Komputer tidak bisa tahu begitu saja nilai apa yang terkandung pada setiap memori



Komputer tidak bisa langsung melihat nilai yang sebenarnya dari setiap cell

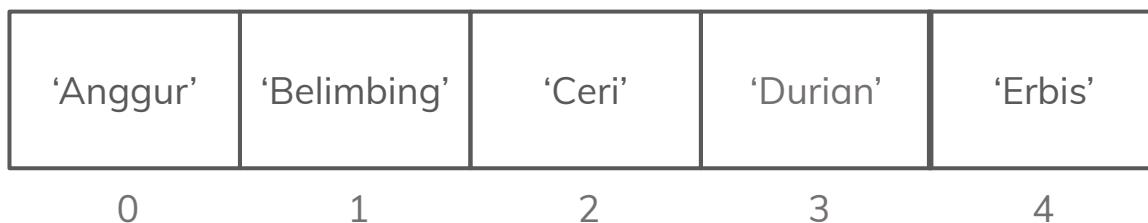
Searching

- Perlu memeriksa setiap cell satu per satu (ex. Mencari value ‘Durian’)



Searching

- Kurang efisien dibanding dengan Reading
- Memeriksa satu per satu cell
- Membutuhkan banyak steps sesuai dengan ukuran array atau disebut juga *linear search*



Untuk jumlah 5 cells maksimum steps yang dibutuhkan adalah 5, jika value yang dicari yaitu 'Erbis'



Untuk jumlah 50 cells maksimum steps yang dibutuhkan adalah 50 atau membutuhkan N steps, dimana N bisa menjadi angka berapapun

Searching

- Mencari indeks dari value ‘Durian’ pada array

```
● ● ●  
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
var indeksDurian = array.indexOf('Durian');
print(indeksDurian); // console: 3
```

Insertion

- Memasukkan/ menambah value baru ke dalam array
- Efisiensi tergantung dimana kita memasukan value nya
- Memasukkan value pada akhir dari array hanya membutuhkan 1 steps

'Anggur'	'Belimbing'	'Ceri'	'Durian'	'Erbis'	'Frambos'
Indeks: 0	1	2	3	4	5

Alamat memori:	1020	1021	1022	1023	1024	1025
-------------------	------	------	------	------	------	------

Jika array dengan ukuran lima value dimulai dari alamat memori 1020, maka akhir alamat memori nya adalah 1024

Ketika kita menambahkan value baru, berarti menambahkannya pada alamat memori berikutnya yaitu 1025

Ini hanya membutuhkan 1 steps



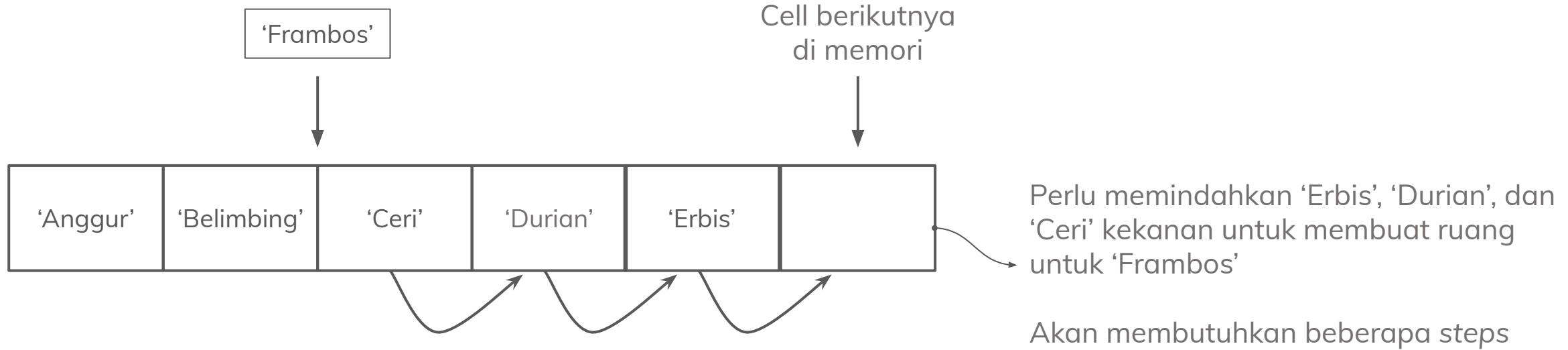
```
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
array.add('Frambos');

print(array); // console: [Anggur, Belimbing, Ceri, Durian, Erbis, Frambos]
```

Insertion

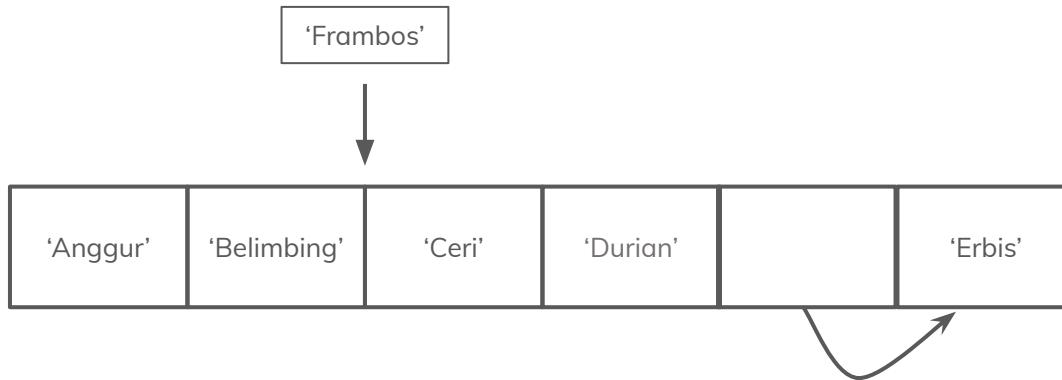
- Memasukan value baru di awal atau tengah array perlu menggeser data yang sudah ada pada array
- Sehingga akan membutuhkan steps tambahan

Kita ingin memasukan ‘Frambos’ disini

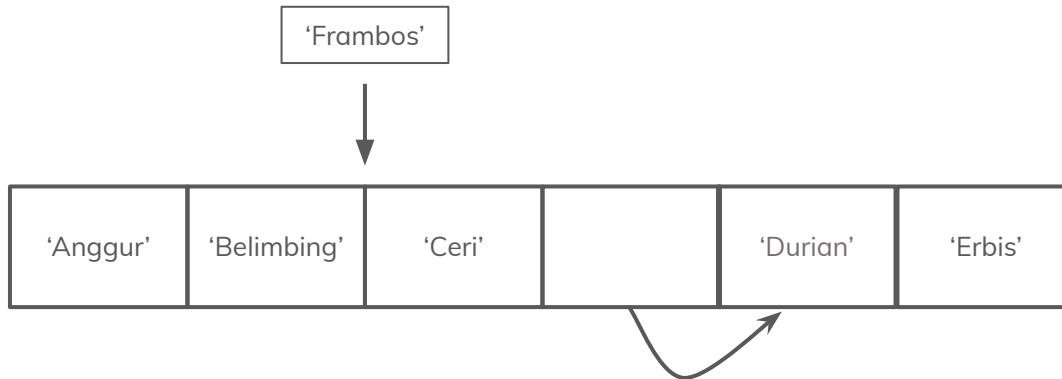


Insertion

- Step 1: Menggeser ‘Erbis’ ke kanan

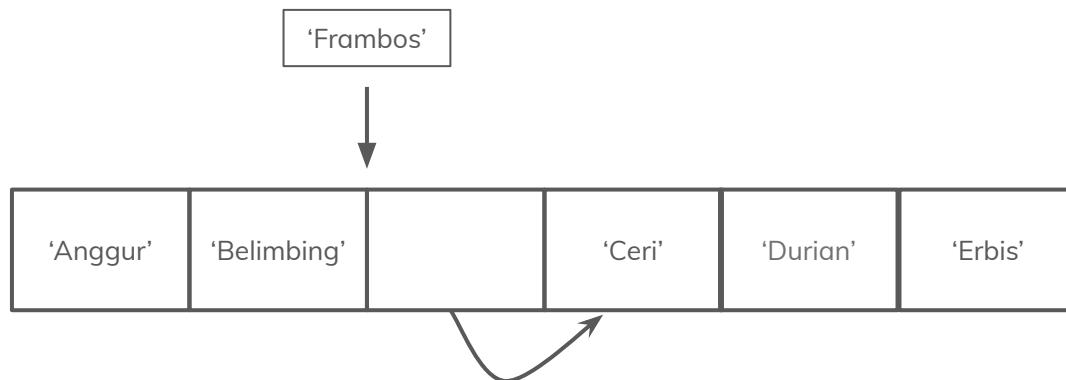


- Step 2: Menggeser ‘Durian’ ke kanan

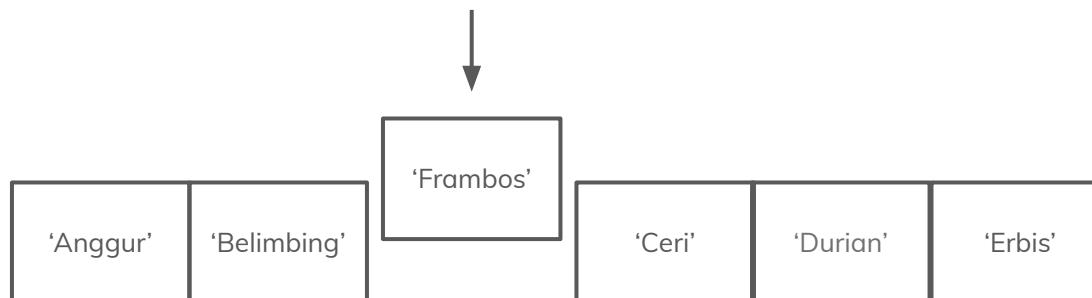


Insertion

- Step 3: Menggeser ‘Ceri’ ke kanan



- Step 4: Akhirnya, ‘Frambos’ bisa dimasukan



Total insertion dalam kasus ini yaitu 4 steps.
 3 steps menggeser data ke kanan
 1 steps merupakan *insertion*

Skenario terburuk *insertion* pada array yaitu melakukannya pada awal dari sebuah array, sehingga skenario tersebut membutuhkan $N + 1$ steps untuk array yang mengandung N elemen. N adalah jumlah steps untuk menggeser dan 1 adalah steps untuk *insertion*.

Insertion

- Menambahkan value di awal array dengan lima elemen (6 steps)



```
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
array.insert(0, 'Frambos');
print(array); // console: [Frambos, Anggur, Belimbing, Ceri, Durian, Erbis]
```

- Menambahkan value di tengah array atau indeks kedua (4 steps)



```
var array = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
array.insert(2, 'Frambos');
print(array); // console: [Anggur, Belimbing, Frambos, Ceri, Durian, Erbis]
```

Deletion

- Operasi menghilangkan value pada indeks tertentu
- Membutuhkan jumlah N steps untuk array yang mengandung N data
- Step 1: Menghilangkan ‘Ceri’ dari array

‘Anggur’	‘Belimbing’	‘Frambos’		‘Durian’	‘Erbis’
----------	-------------	-----------	--	----------	---------

Proses deletion hanya membutuhkan 1 steps. Namun, akibat deletion akan menyebabkan cell bekas ‘Ceri’ menjadi kosong dan menimbulkan gaps. Gaps di tengah pada sebuah array adalah tidak efektif. Untuk menyelesaikan masalah ini, perlu menggeser ‘Durian’ dan ‘Erbis’ ke kiri. Sehingga akan membutuhkan steps tambahan lagi

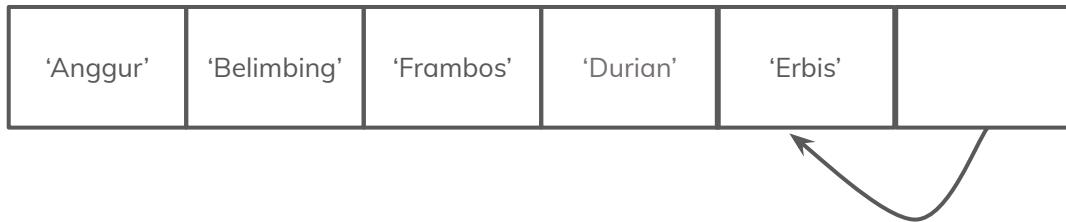
- Step 2: Menggeser ‘Durian’ ke kiri

‘Anggur’	‘Belimbing’	‘Frambos’	‘Durian’		‘Erbis’
----------	-------------	-----------	----------	--	---------



Deletion

- Step 3: Menggeser ‘Erbis’ ke kiri



- Pada kasus ini operasi deletion membutuhkan 3 steps
- Skenario terburuk nya yaitu menghilangkan value pada awal sebuah array
- Jika array mengandung 6 data, 1 step digunakan untuk deletion sendiri lalu 5 steps untuk menggeser data nya
- Begitupun untuk array yang mengandung 50 data, 1 step digunakan untuk deletion sendiri lalu 49 steps untuk menggeser data nya

Deletion

- Menghapus data ‘Ceri’



```
var array = ['Anggur', 'Belimbing', 'Frambos', 'Ceri', 'Durian', 'Erbis'];
array.removeAt(3);
print(array); // console: [Anggur, Belimbing, Frambos, Durian, Erbis]
```

- Menghapus data di awal sebuah array



```
var array = ['Anggur', 'Belimbing', 'Frambos', 'Ceri', 'Durian', 'Erbis'];
array.removeAt(0);
print(array); // console: [Belimbing, Frambos, Ceri, Durian, Erbis]
```

Set

Deskripsi

- Data structure sama seperti array, tetapi perbedaan nya pada set yang tidak mengizinkan duplikat value dimasukan kedalam nya
- Dari keempat operasi reading, searching, insertion, dan deletion, set memiliki perbedaan efisiensi pada operasi insertion. Sedangkan untuk operasi lainnya sama seperti array



```
var sets = {'Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'};  
sets.add('Anggur');  
  
print(sets); // console: {Anggur, Belimbing, Ceri, Durian, Erbis}
```



Komputer tidak mengizinkan menambahkan 'Anggur' lagi, karena 'Anggur' sudah ada didalam set



Operasi pada Set

- Reading: sama seperti array, membutuhkan 1 steps
- Searching: sama seperti array, bisa mencapai N steps
- Deletion: sama seperti array, bisa mencapai N steps



```
var sets = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
var durian = sets.elementAt(3);
print(durian); // console: Durian
```

→ Reading



```
var sets = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
var searchDurian = sets.contains('Durian');
print(searchDurian); // console: true
```

→ Searching, disini akan menghasilkan nilai boolean (true/false)



```
var sets = ['Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'];
sets.remove('Ceri');
print(sets); // console: {'Anggur', 'Belimbing', 'Durian', 'Erbis'}
```

→ Deletion

Insertion pada Set

- Perlu steps tambahan untuk memastikan bahwa value yang ingin dimasukan tidak ada di dalam set
- Maka operasi searching akan dijalankan terlebih dahulu untuk melihat value nya (operasi searching bisa mencapai N steps)
- Contoh, kita sudah mempunyai set yang memiliki 5 value lalu kita ingin menambahkan 1 value ‘Frambos’ di akhir set tersebut.
Berikut step yang akan dilalui:
- Step 1: Memeriksa indeks 0 untuk ‘Frambos’



‘Anggur’	?	?	?	?
----------	---	---	---	---

Insertion pada Set

- Step 2: Memeriksa indeks 1



'Anggur'	'Belimbing'	?	?	?
----------	-------------	---	---	---

- Step 3: Memeriksa indeks 2



'Anggur'	'Belimbing'	'Ceri'	?	?
----------	-------------	--------	---	---

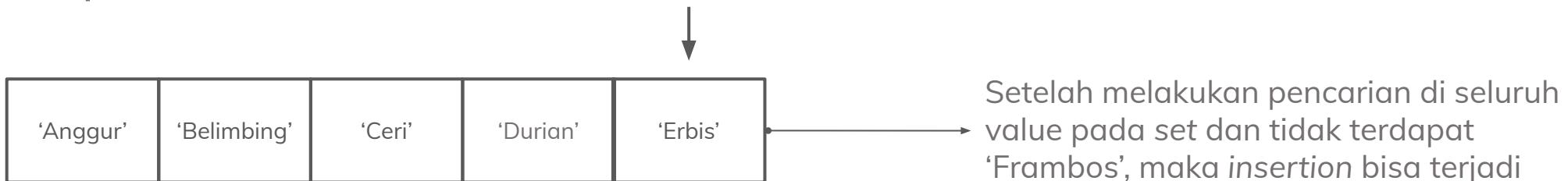
- Step 4: Memeriksa indeks 3



'Anggur'	'Belimbing'	'Ceri'	'Durian'	?
----------	-------------	--------	----------	---

Insertion pada Set

- Step 5: Memeriksa indeks 4



- Step 6: Memasukkan 'Frambos' di akhir set



1

Insertion pada akhir sebuah set adalah skenario terbaik, namun tetap saja membutuhkan 6 steps pada kasus ini dengan jumlah value awal yaitu 5 elemen. Sedangkan untuk array hanya membutuhkan 1 steps.

Skenario terburuk yaitu ketika memasukkan value di awal sebuah set, selain operasi searching yang bisa mencapai N steps, ditambah lagi N steps untuk menggeser data ke kanan, setelah itu value bisa dimasukkan. Sehingga memiliki total $2N + 1$ steps

Insertion pada Set

- Memasukkan ‘Frambos’ di akhir set (6 steps)

```
● ● ●  
  
var sets = {'Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'};  
sets.add('Frambos');  
print(sets); // console: {'Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis', 'Frambos'}
```

- Memasukkan ‘Frambos’ di awal set (11 steps)

```
● ● ●  
  
var sets = {'Anggur', 'Belimbing', 'Ceri', 'Durian', 'Erbis'};  
if (sets.contains('Frambos')) { // Membutuhkan 5 steps  
    print('Nilai Frambos sudah ada');  
} else {  
    var setToList = sets.toList();  
    setToList.insert(0, 'Frambos'); // Membutuhkan 5 steps + 1 steps  
    sets = setToList.toSet();  
    print(sets);  
}
```

Kesimpulan

Deskripsi

- Memahami data structure penting untuk membuat kode yang berkualitas dan berjalan efisien
- Dua jenis data structure dasar yaitu Array dan Set
- Jumlah steps sangat penting untuk mengukur performa data structure
- Terdapat 4 operasi yang umum digunakan untuk berinteraksi dengan data structure reading, searching, insertion, dan deletion
- Reading, searching, dan deletion pada array dan set memiliki efisiensi yang sama kecuali pada operasi insertion
- Pilih Set jika ingin terbebas dari duplikasi
- Menggunakan array lebih disarankan jika tidak ada pemeriksaan duplikasi

Latihan

- Buatlah sebuah array dan set yang mengandung 30 elemen (nama negara, provinsi, hewan, bebas). Buatlah kode untuk masing-masing operasi serta analisa berapa steps yang diperlukan untuk operasi tersebut. Operasi untuk masing-masing array dan set adalah sebagai berikut:
 - a. Reading
 - b. Searching untuk value yang tidak ada
 - c. Insertion di awal
 - d. Insertion di akhir
 - e. Deletion di awal
 - f. Deletion di akhir



Terima Kasih



Dart
Object Oriented Programming

Outline

- Perkenalan Object Oriented Programming
- Class
- Property
- Method

Perkenalan Object Oriented Programming

Deskripsi



- Biasa disebut OOP
- Program disusun dalam bentuk abstraksi object
- Data dan proses diletakkan pada abstraksi tersebut

Keuntungan



Mudah di-troubleshoot



Mudah digunakan ulang

Penggunaan

Dart dan berbagai bahasa pemrograman lainnya, seperti:



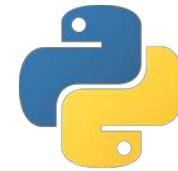
C++



Java



Javascript



Python

Komponen

OOP memiliki beberapa bagian, diantaranya:

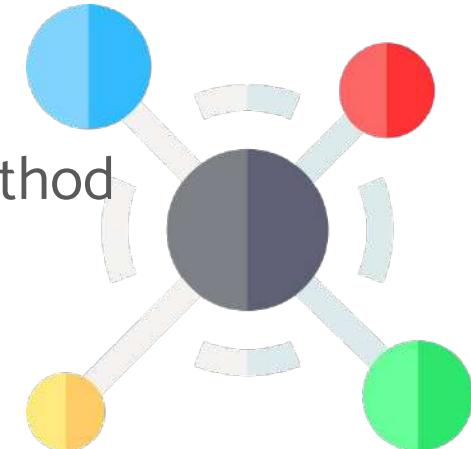
- Class
- Object
- Property
- Method
- Inheritance
- Generics



Class

Deskripsi

- Abstraksi dari sebuah benda (object)
- Memiliki ciri-ciri yang disebut property
- Memiliki sifat dan kemampuan yang disebut method



Membuat Class

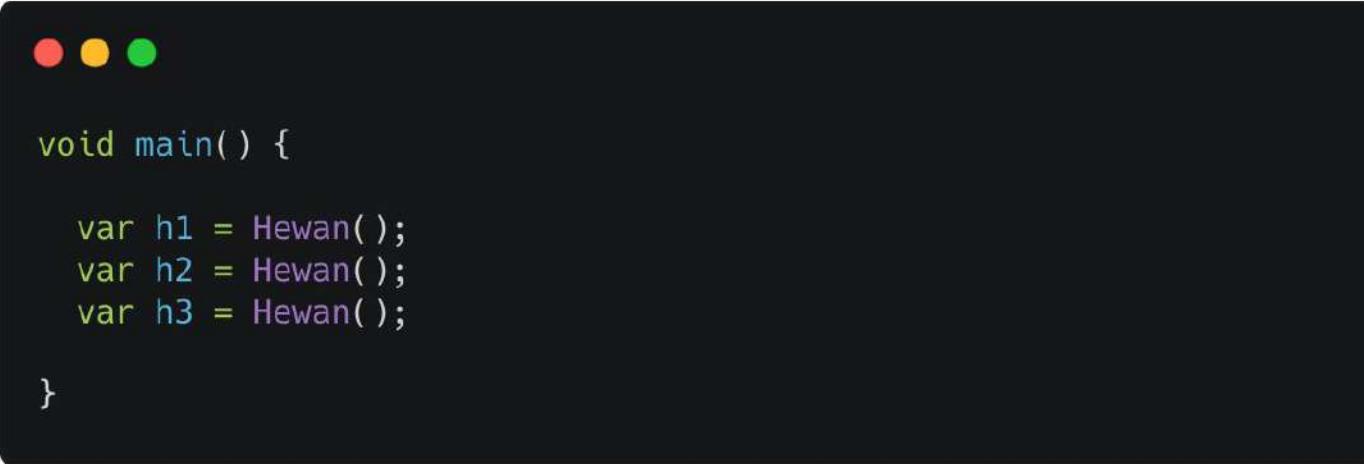
- Menggunakan kata kunci `class`
- Memiliki nama
- Detail class diletakkan dalam kurawal



```
class Hewan {  
    // property  
    // method  
}
```

Membuat Object

- Bentuk sebenarnya dari class
- Disebut juga instance of class
- Diperlukan seperti data

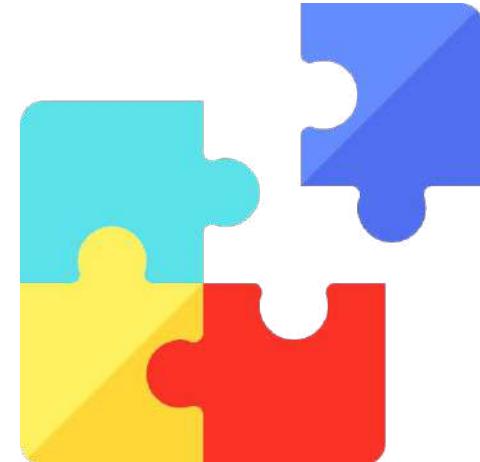


```
void main() {  
  
    var h1 = Hewan();  
    var h2 = Hewan();  
    var h3 = Hewan();  
  
}
```

Property

Deskripsi

- Ciri-ciri suatu class
- Hal-hal yang dimiliki suatu class
- Memiliki sifat seperti variabel



Membuat Property

Seperti variabel tetapi terletak dalam sebuah class.



```
class Hewan {  
    var mata = 0;  
    var kaki = 0;  
}
```

Mengakses Property

Seperti menggunakan variabel tetapi melalui sebuah object.

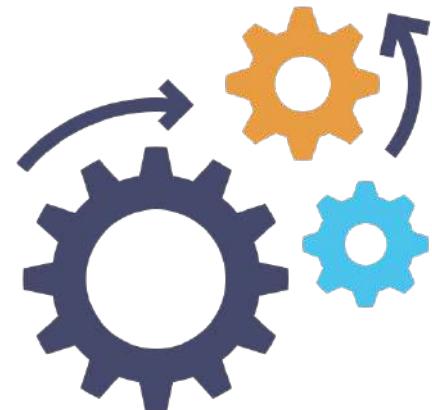


```
void main() {  
  
  var h1 = Hewan();  
  print(h1.mata);  
  
}
```

Method

Deskripsi

- Sifat suatu class
- Aktivitas yang dapat dikerjakan suatu class
- Memiliki sifat seperti fungsi



Membuat Method

Seperti fungsi tetapi terletak dalam sebuah class.



```
class Hewan {  
    void bersuara() {  
        print('');  
    }  
}
```

Menjalankan Method

Seperti menggunakan fungsi tetapi melalui sebuah object.



```
void main() {  
  
    var h1 = Hewan();  
    h1.bersuara();  
  
}
```



Terima Kasih

Task

1. Buatlah beberapa class dengan rangkaian berikut ini!

Class Hewan
Property Berat menyimpan data berat badan hewan

Class Mobil
Property kapasitas total berat maksimum muatan

Property muatan list data hewan yang menjadi muatan
Method tambahMuatan jika kapasitas mencukupi, maka menambah data hewan pada list muatan

Task

2. Pada class Mobil, tambahkan method **totalMuatan** untuk menghitung total muatan yang diangkut oleh mobil tersebut!

Flutter

Outline

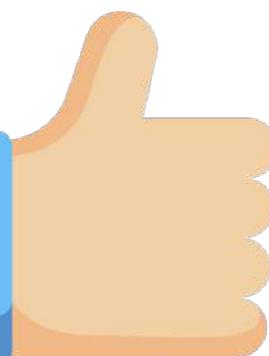
- Perkenalan Flutter
- Stateless Widget
- Stateful Widget
- Built in Widget

Apa itu Flutter?

Alat pengembangan antarmuka pengguna yang digunakan untuk membuat aplikasi mobile, desktop, dan web.



Keunggulan Flutter



Mudah digunakan dan dipelajari

Produktivitas tinggi

Dokumentasi lengkap

Komunitas yang berkembang



Bagian dari Flutter

SDK
(Software Development Kit)

Alat-alat untuk membantu
proses pengembangan aplikasi



Framework
Perlengkapan untuk membentuk
aplikasi yang dapat
dikustomisasi



Membuat Project

Dilakukan dengan menjalankan perintah:

```
flutter create <nama_project>
```

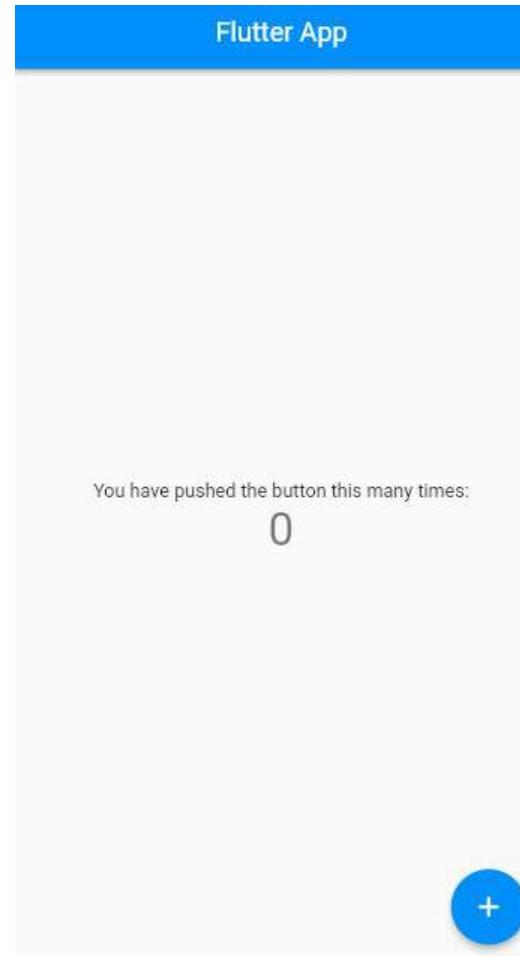


```
flutter create phonebookapp
```

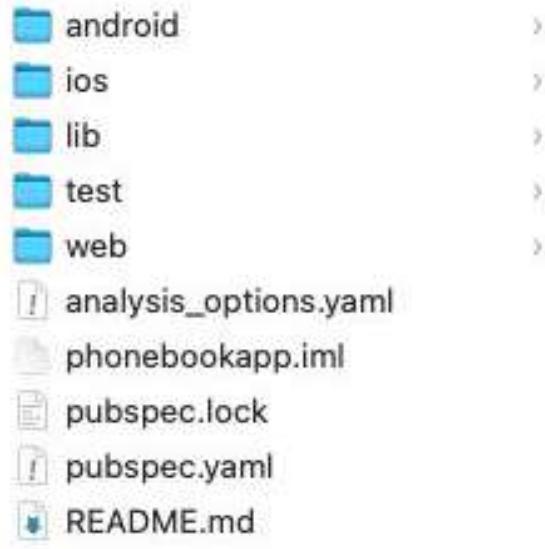


Menjalankan Project

- Masuk ke direktori project
- Jalankan perintah:
`flutter run`



Struktur Direktori



direktori platform:

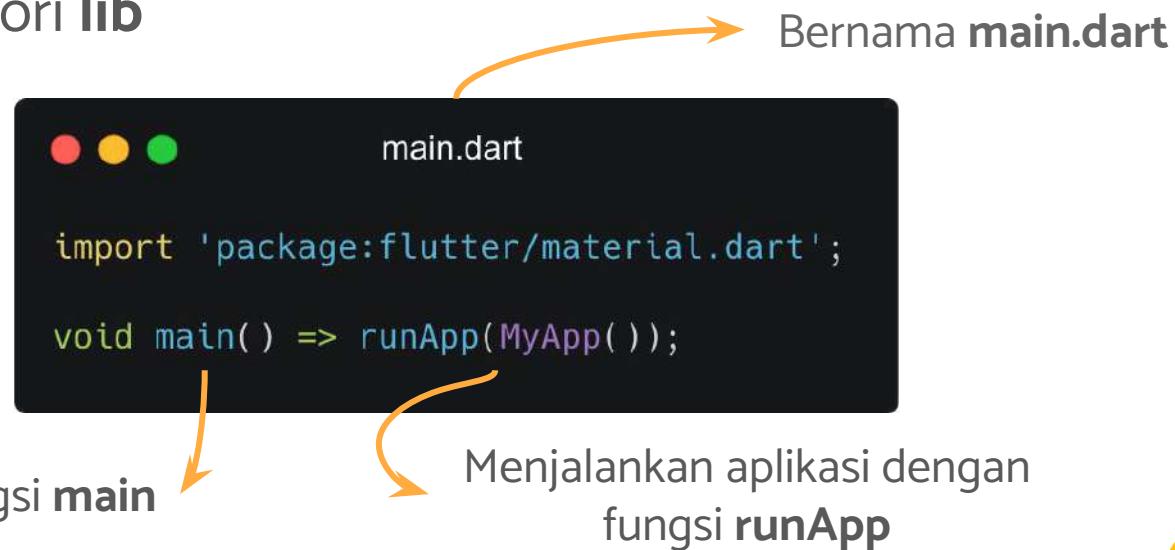
- **android**
- **ios**
- **web**

direktori project:

- **lib** - ruang kerja utama
- **test** - aktivitas pengujian

File Utama

- Pertama kali dibaca dan dijalankan
- Dalam direktori **lib**



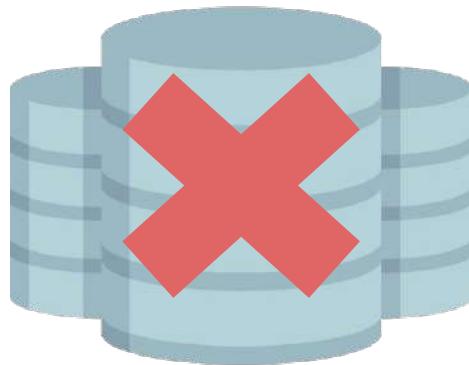
Widget

Deskripsi

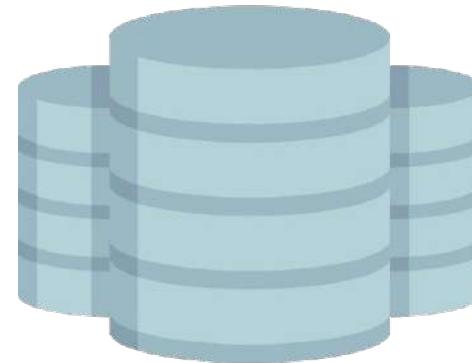
- Digunakan untuk membentuk antarmuka (UI)
- Berupa class
- Dapat terdiri dari beberapa widget lainnya



Jenis



Stateless

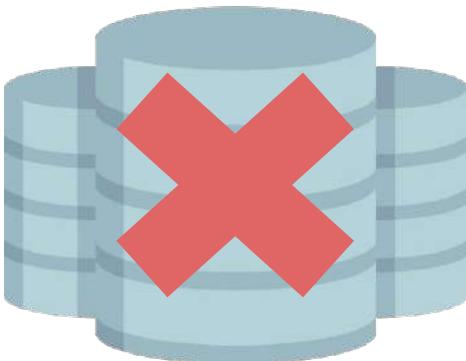


Stateful



Stateless Widget

Deskripsi



- Tidak bergantung pada perubahan data
- Hanya fokus pada tampilan
- Dibuat dengan extends pada class StatelessWidget

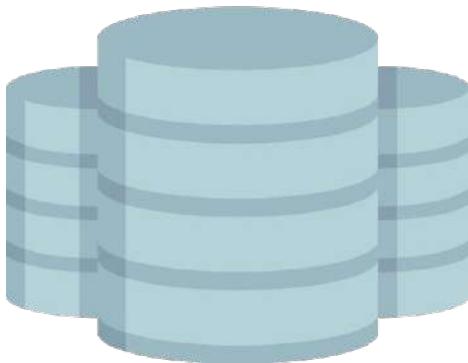
Membuat Stateless Widget



```
class MyWidget extends StatelessWidget {  
    const MyWidget({Key? key}) : super(key: key);  
  
    @override  
    Widget build(BuildContext context) {  
        return const MaterialApp(  
            home: Scaffold(  
                body: Text('Selamat datang di Flutter.'),  
            ),  
        );  
    }  
}
```

Stateful Widget

Deskripsi



- Mementingkan pada perubahan data
- Dibuat dengan extends pada class StatefulWidget
- 1 widget menggunakan 2 class (widget dan state)

Membuat StatefulWidget



```
1 class MyWidget extends StatefulWidget {  
  const MyWidget({Key? key}) : super(key: key);  
  
  @override  
  createState() => MyWidgetState();  
}
```



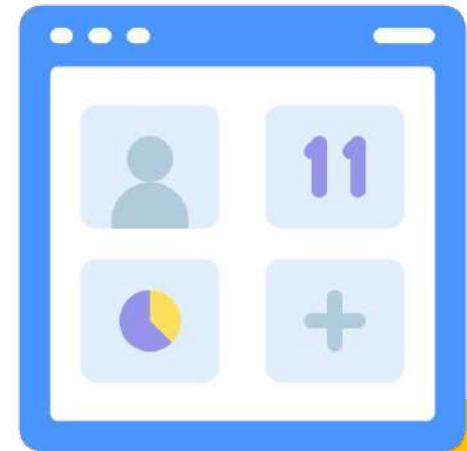
```
2 class MyWidgetState extends State<MyWidget> {  
  @override  
  Widget build(BuildContext context) {  
    return const MaterialApp(  
      home: Scaffold(  
        body: Text('Selamat datang di Flutter.'),  
      ),  
    );  
  }  
}
```



Built in Widget

Deskripsi

- Widget yang dapat langsung digunakan
- Sudah ter-install bersama Flutter



MaterialApp

Membangun aplikasi dengan desain material



```
const MaterialApp();
```



```
const MaterialApp(  
    home: Text('Selamat datang di Flutter.'),  
);
```

Scaffold

Membentuk sebuah halaman



```
const Scaffold();
```



```
const Scaffold(  
    body: Text('Selamat datang di Flutter.'),  
);
```

AppBar

Membentuk application bar yang terletak pada bagian atas halaman



```
AppBar();
```



```
AppBar(  
  title: const Text('Home'),  
);
```

Text

Menampilkan teks



```
const Text(teks);
```



```
const Text('halooo...');
```

Hasil



```
@override  
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      appBar: AppBar(  
        title: const Text('Home'),  
      ),  
      body: const Text('Selamat datang di Flutter.'),  
    ),  
  );  
}
```



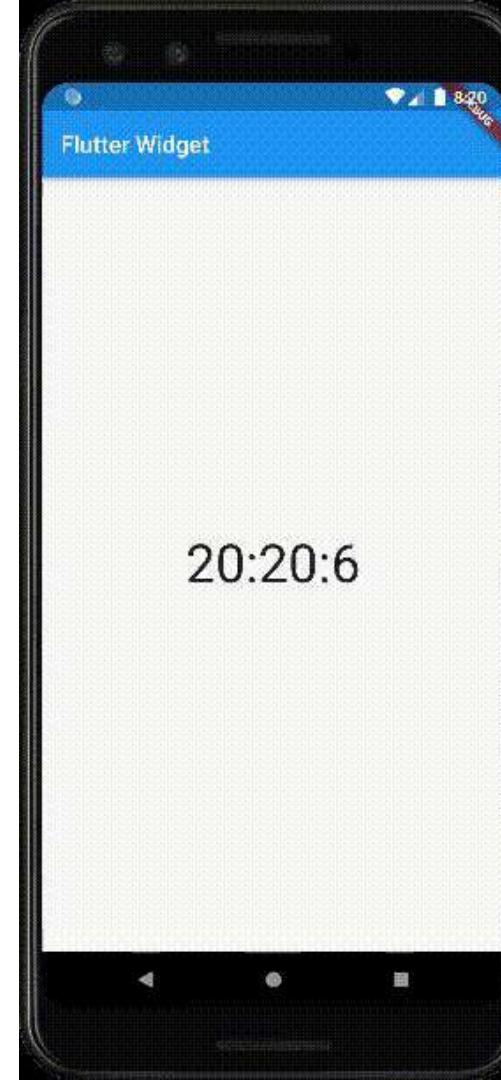


Terima Kasih

Task

1. Buatlah Statefull Widget yang kontennya dapat berubah setiap detik, seperti contoh berikut ini!

20:20:6



Task

2. Dari Widget tersebut, pisahkan menjadi beberapa widget tanpa mengubah tampilannya!

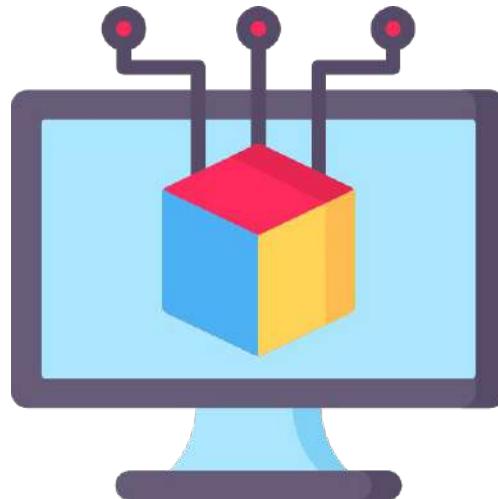
Mobile App Development

Outline

- Pengertian Software
- Jenis Software
- Pengembangan Software
- Mobile App Development

Software

Pengertian



Perangkat lunak yang tersimpan dan dapat dikendalikan pada sistem komputer

Pengertian



Terbentuk dari perintah-perintah dan
tidak memiliki bentuk fisik

Jenis-Jenis Software



Web

- Diakses menggunakan web browser
- Memerlukan internet untuk membukanya



Desktop

- Berjalan pada perangkat desktop
- Umumnya perlu langkah instalasi
- Dapat dibuka tanpa memerlukan internet



Mobile

- Berjalan pada perangkat mobile (smartphone atau tablet)
- Umumnya perlu langkah instalasi
- Instalasi dapat dilakukan melalui store penyedia aplikasi
- Dapat dibuka tanpa memerlukan internet

Pengembangan Software

Bahasa Pemrograman



Rangkaian teks yang digunakan untuk memberi perintah kepada komputer

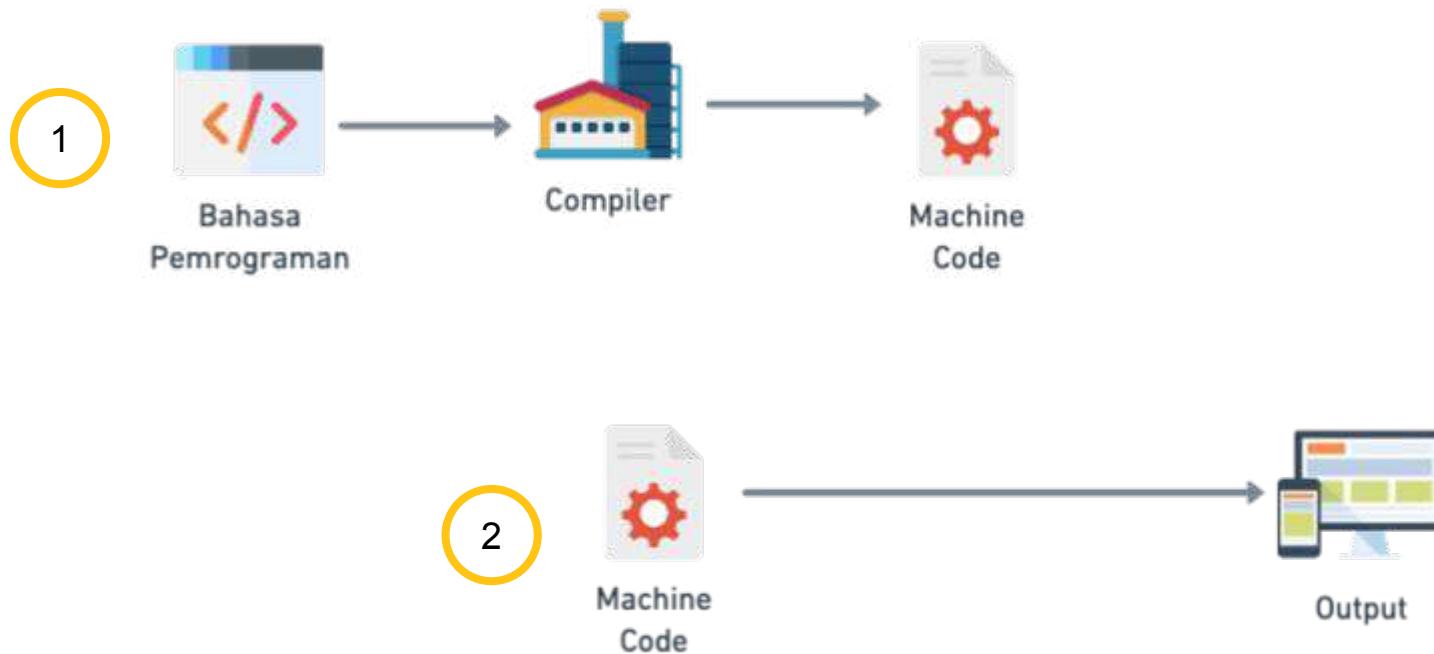
Compiler

Software yang digunakan untuk membaca perintah-perintah dalam bahasa pemrograman dan mengubahnya ke bahasa lain, biasanya ke dalam bentuk yang dikenali oleh komputer, seperti machine code.

* machine code: kumpulan instruksi yang hanya dikenali oleh mesin

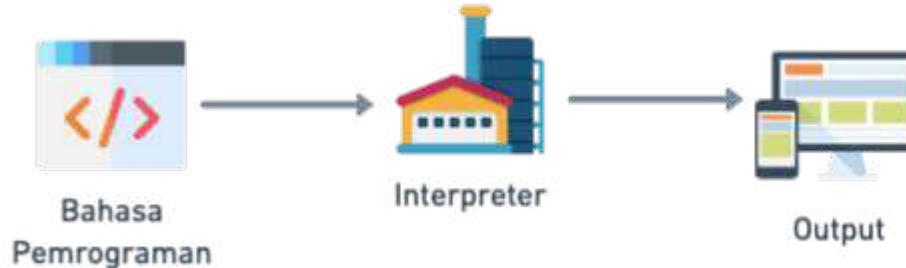


Compiler



Interpreter

Software yang digunakan untuk membaca perintah-perintah dalam bahasa pemrograman dan menjalankan langsung perintah-perintah tersebut.





**Jika interpreter lebih ringkas,
kenapa compiler masih populer?**

Compiler vs Interpreter

Compiler

- Mengubah kode menjadi file yang dapat dijalankan
- Prosesnya lebih panjang
- Program dapat dijalankan dengan cepat karena tidak membaca ulang kode

Interpreter

- Membaca kode tanpa mengubah ke bentuk lain
- Prosesnya lebih singkat
- Seringkali lebih lambat karena harus membaca ulang keseluruhan kode

Mobile App Development

Proses pengembangan aplikasi untuk perangkat mobile

iOS

- Berjalan pada sistem operasi iOS.
- Perangkat hanya dikembangkan oleh Apple.

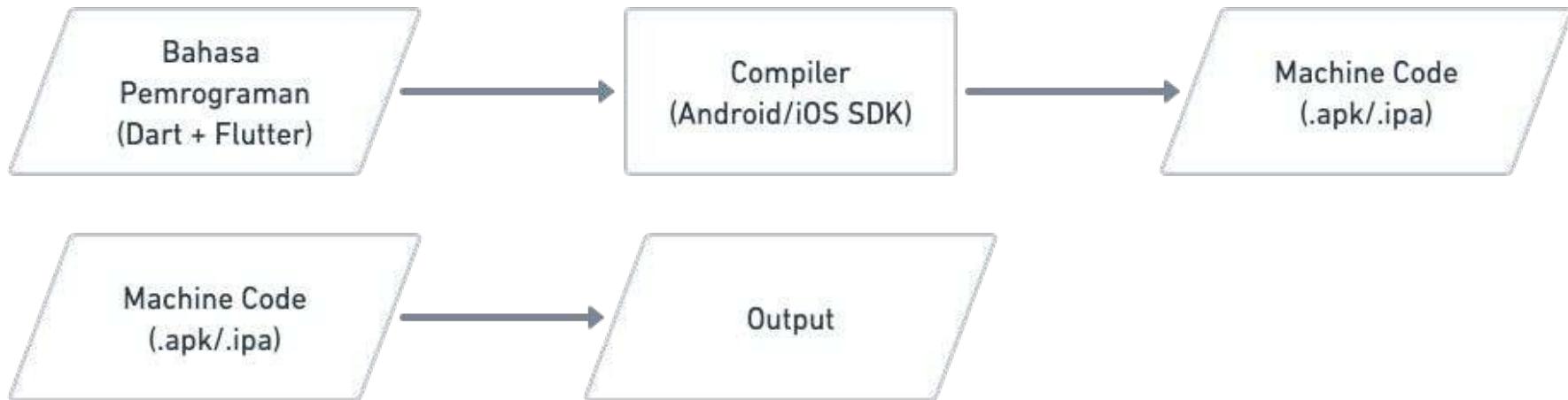


Android

- Berjalan pada sistem operasi Android.
- Perangkat berkembang luas karena open source.



Tahapan





Terima Kasih

Task

Buatlah sebuah document Google Docs, lalu tuliskan hasil pekerjaan Anda dari tugas-tugas berikut ini.

1. Kumpulkan 5 bahasa pemrograman untuk membuat aplikasi mobile!
Sertakan platform dari aplikasi yang dihasilkan (contoh pada desktop: C# untuk Windows)!
2. Kumpulkan 3 aplikasi yang dapat berjalan di iOS tetapi tidak di Android, juga 3 aplikasi yang dapat berjalan di Android tetapi tidak di iOS!



Install Flutter and Tools Part I

Outline

- System Requirements
- Install Flutter SDK
- Install Android Studio
- Install Android Toolchain

System Requirements

System Requirements

- Sistem operasi Windows 7 SP1 or later (64-bit)
- Space kosong 1.64 GB (Tidak termasuk space tool lainnya)
- Tools pendukung [Windows PowerShell 5.0](#) dan [Git for Windows](#)
- RAM 8 GB (rekomendasi 12 GB)
- SSD 120GB (rekomendasi untuk menjalankan emulator Android Studio)



Install Flutter SDK

Langkah - langkah

- Step 1: Download [Flutter SDK](#)
 - Step 2: Ekstrak zip file hasil download di **C:\src**
 - Step 3: Update path
 - Step 4: Pergi ke Environment Variables
 - Step 5: Pilih **Path** yang di User variables
 - Step 6: Tambahkan lokasi **flutter\bin**
 - Step 7: Tekan OK
 - Step 8: Jalankan perintah **flutter** di terminal
 - Step 9: Jalankan perintah **flutter doctor** di terminal
- 

Step 1

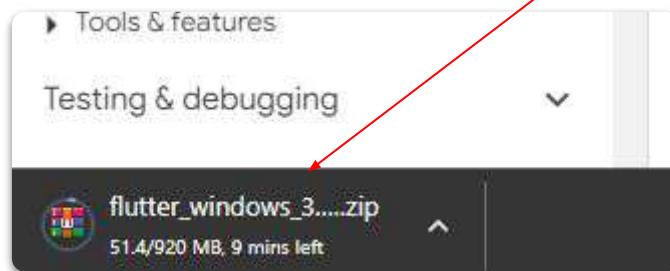
- Download Flutter SDK

Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

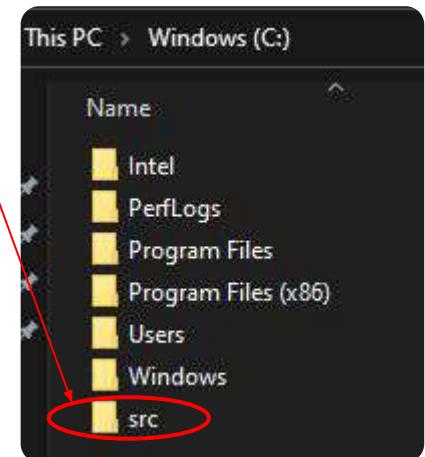
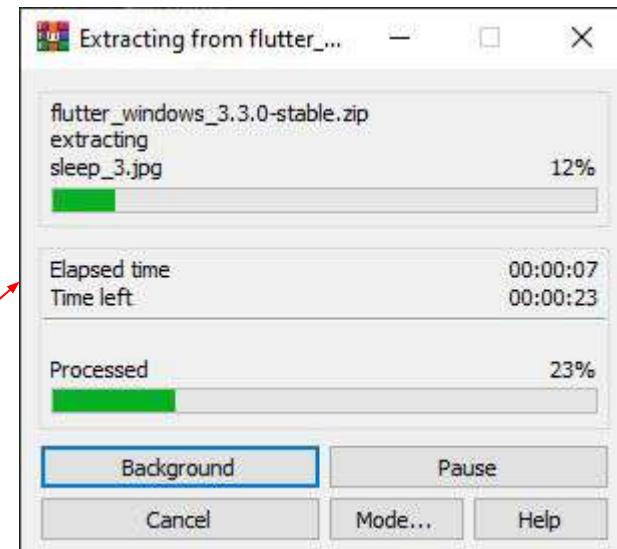
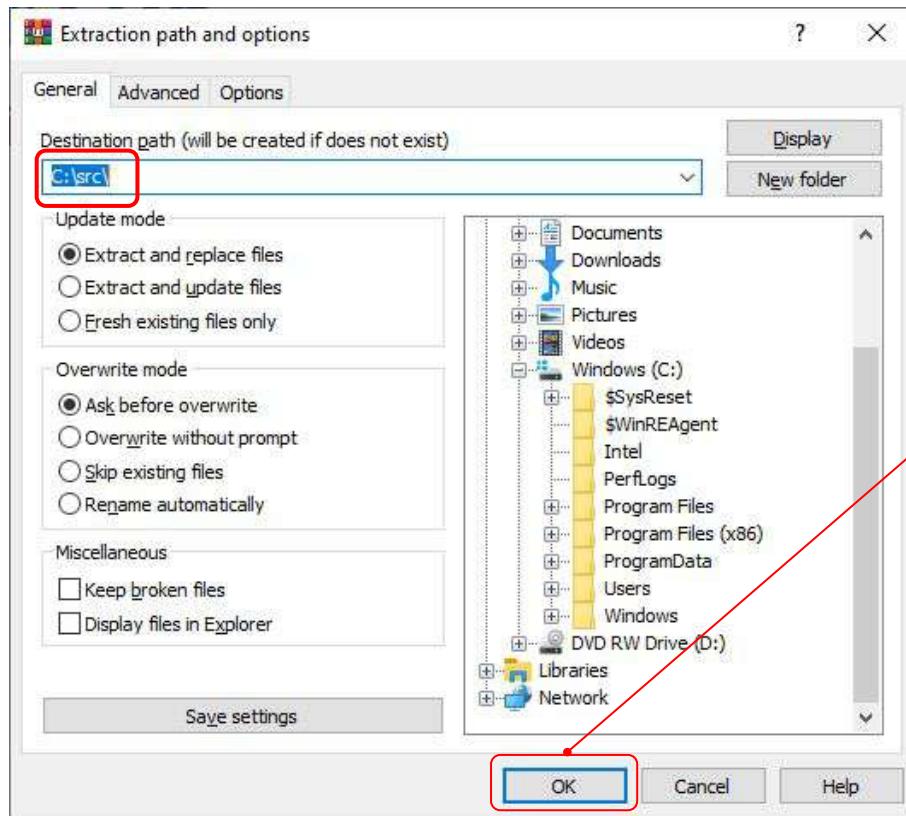
flutter_windows_3.3.0-stable.zip

For other release channels, and older builds, see the [SDK releases](#) page.



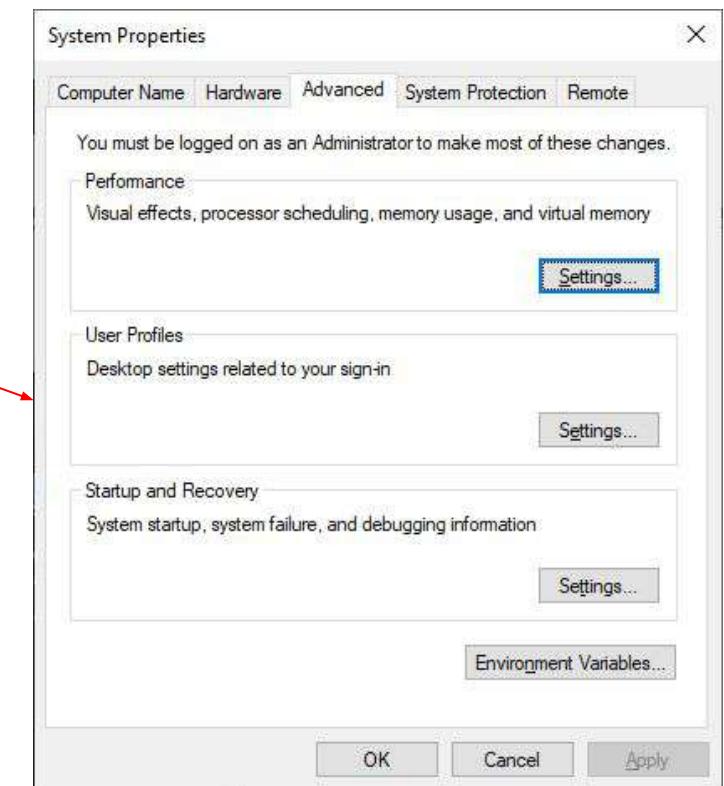
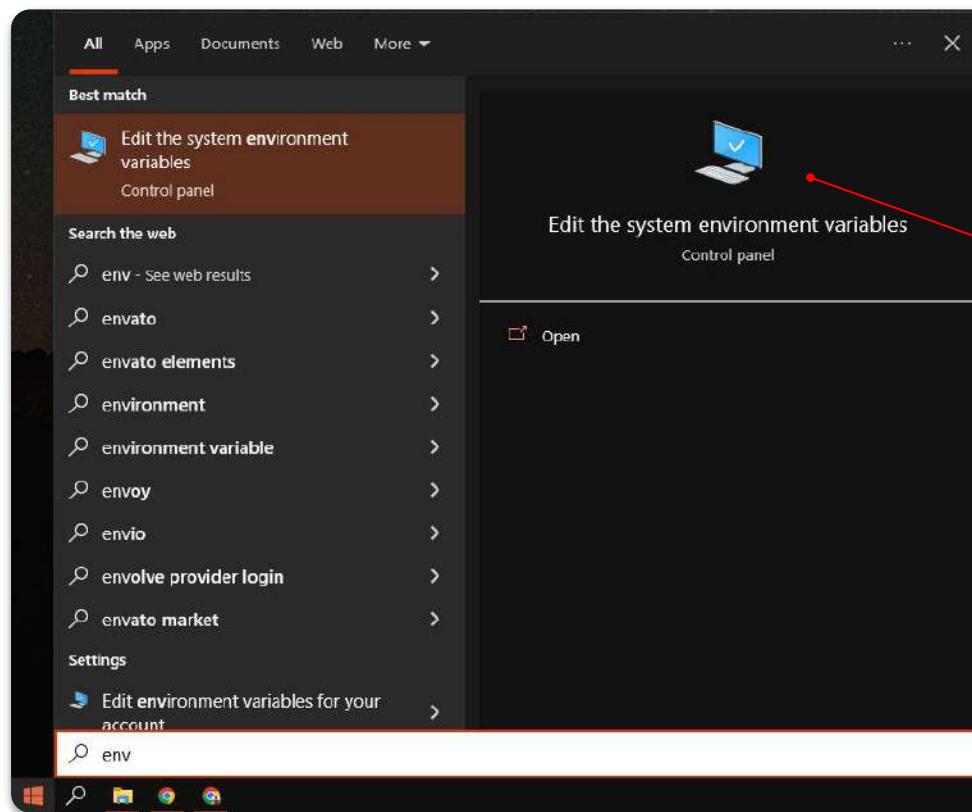
Step 2

- Ekstrak zip file hasil download di C:\src\flutter



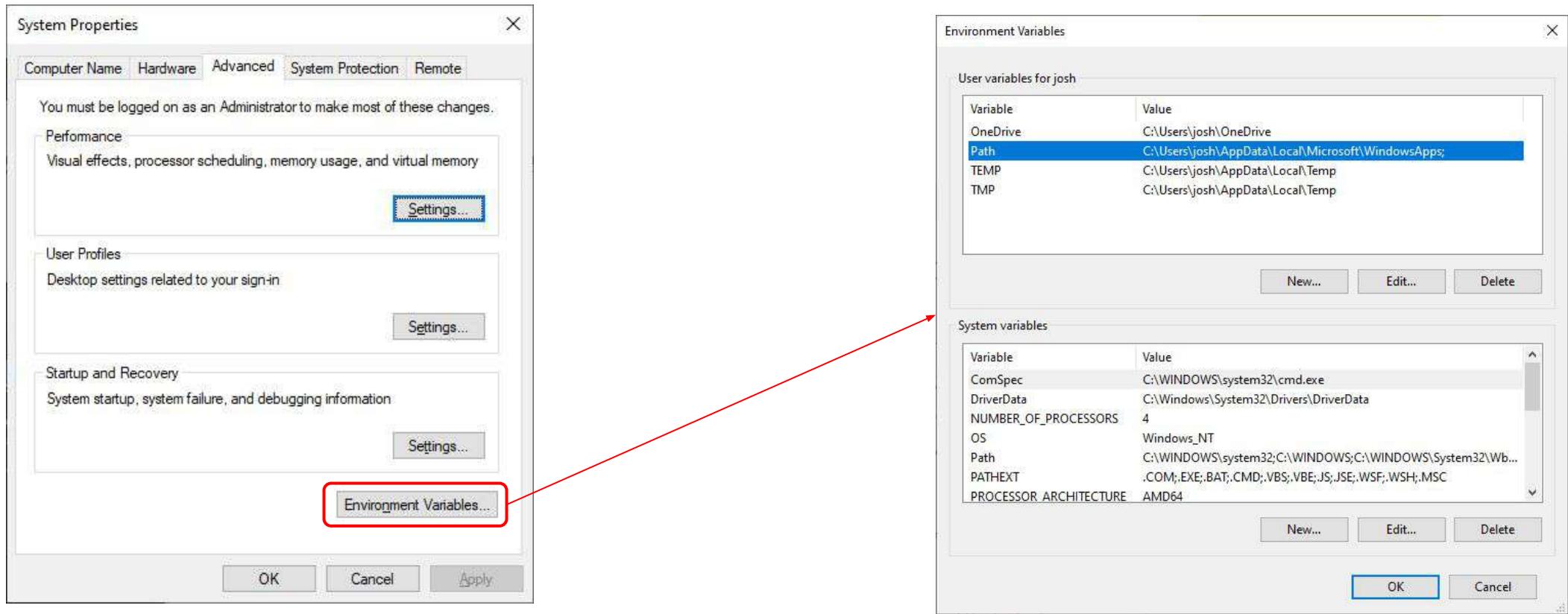
Step 3

- Update path, pergi ke System Properties (tekan windows lalu ketik env)



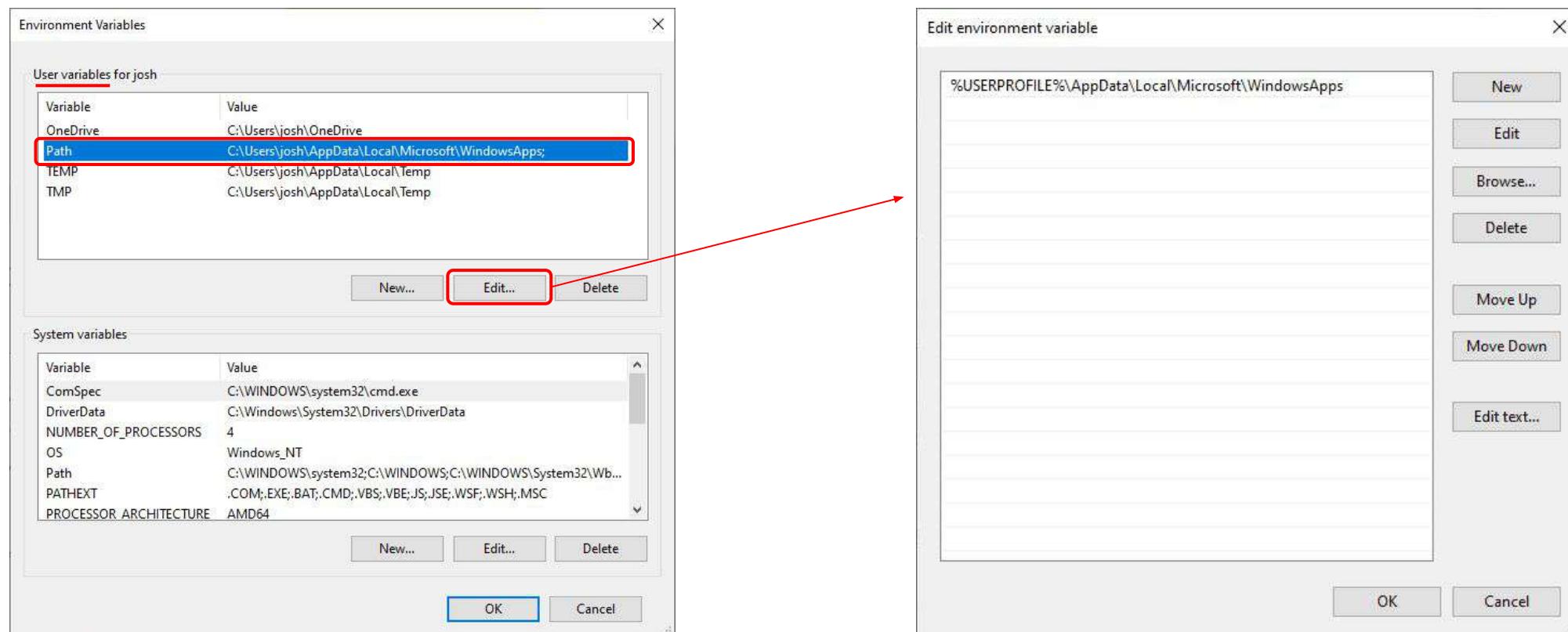
Step 4

- Pergi ke Environment Variables, klik **Environment Variables...**



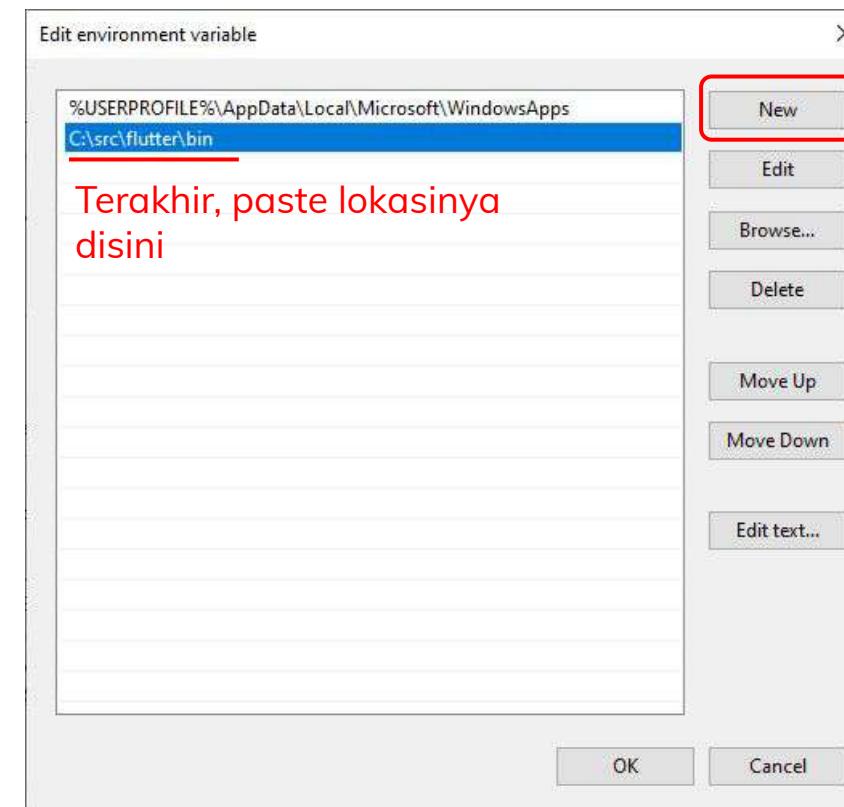
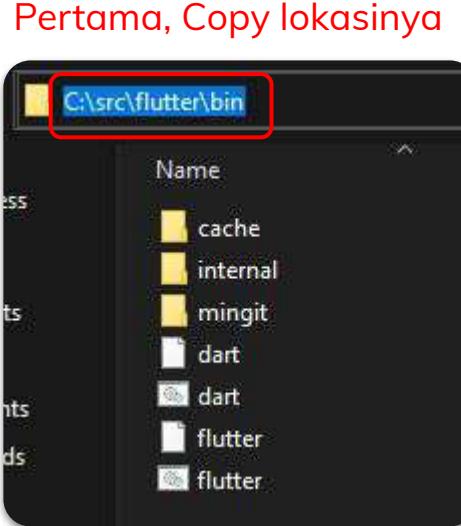
Step 5

- Pilih Path yang di User variables, lalu Klik Edit



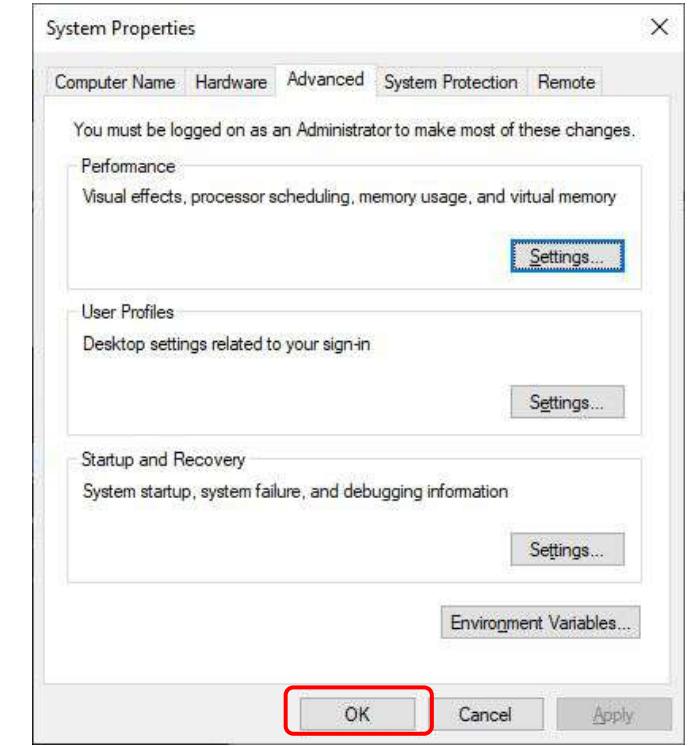
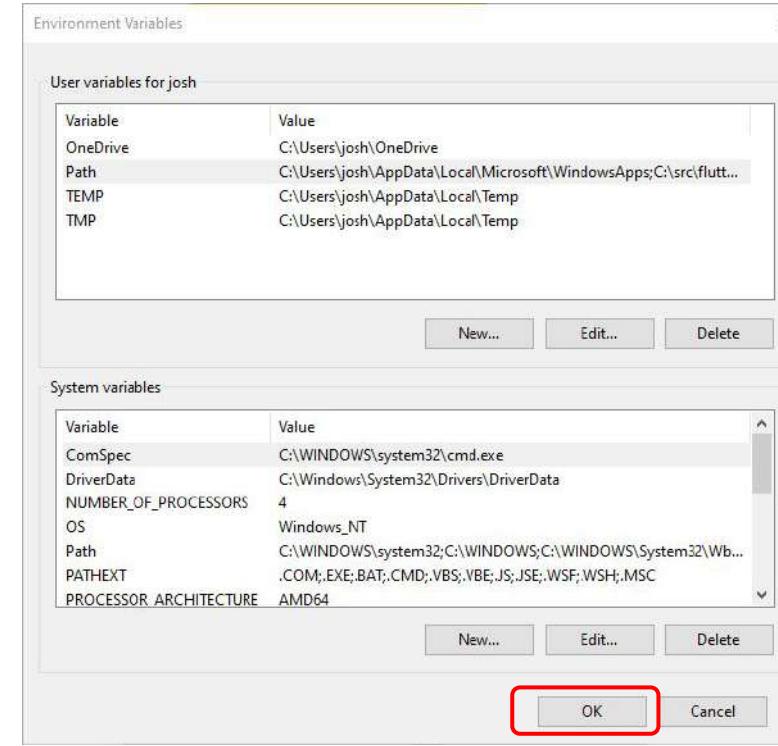
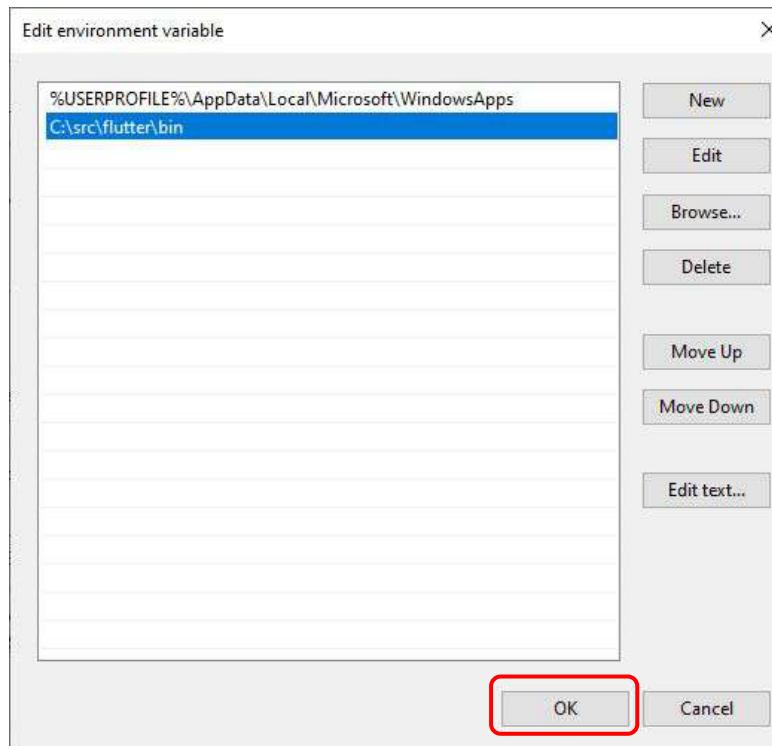
Step 6

- Tambahkan lokasi **flutter\bin** ke dalam environment variables
(Copy: C:\src\flutter\bin)



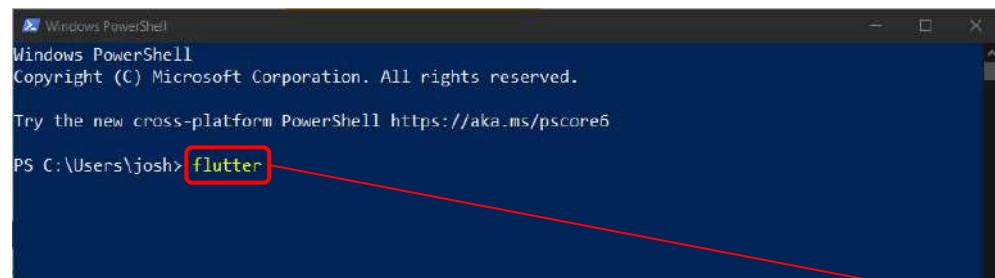
Step 7

- Tekan OK untuk menutup seluruh environment variable nya

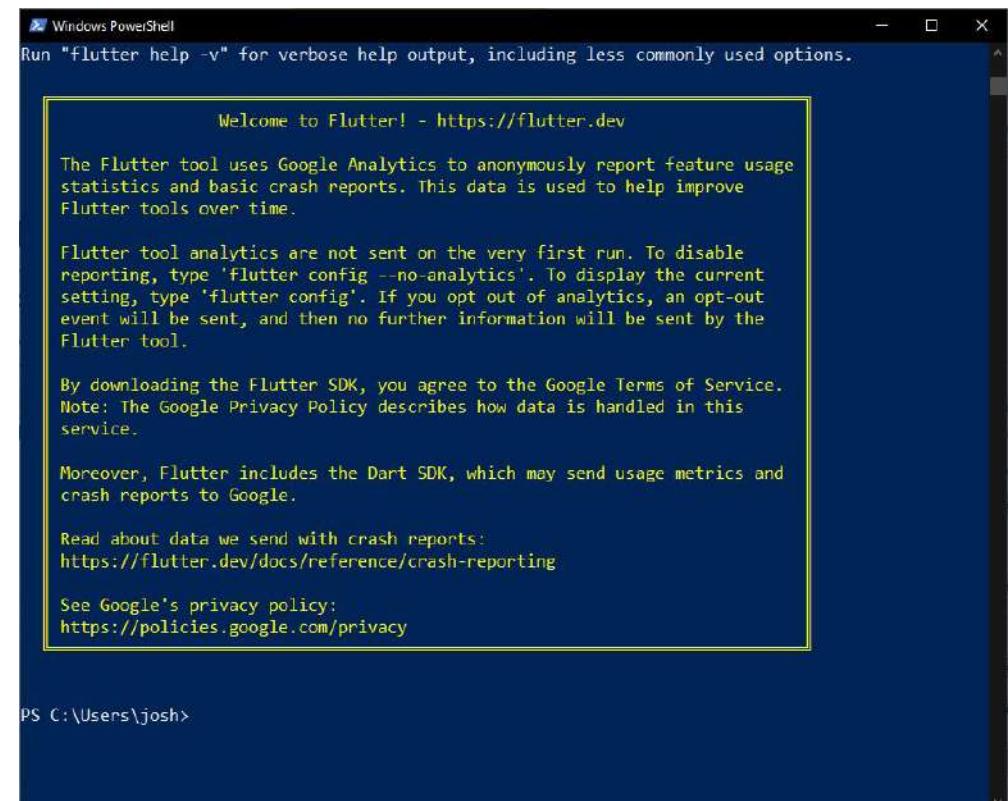


Step 8

- Untuk memastikan flutter sdk sudah berhasil terinstall, buka Windows PowerShell lalu jalankan perintah **flutter**



Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell <https://aka.ms/pscore6>
PS C:\Users\josh> **flutter**



Run "flutter help -v" for verbose help output, including less commonly used options.

Welcome to Flutter! - <https://flutter.dev>

The Flutter tool uses Google Analytics to anonymously report feature usage statistics and basic crash reports. This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, type 'flutter config --no-analytics'. To display the current setting, type 'flutter config'. If you opt out of analytics, an opt-out event will be sent, and then no further information will be sent by the Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service. Note: The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and crash reports to Google.

Read about data we send with crash reports:
<https://flutter.dev/docs/reference/crash-reporting>

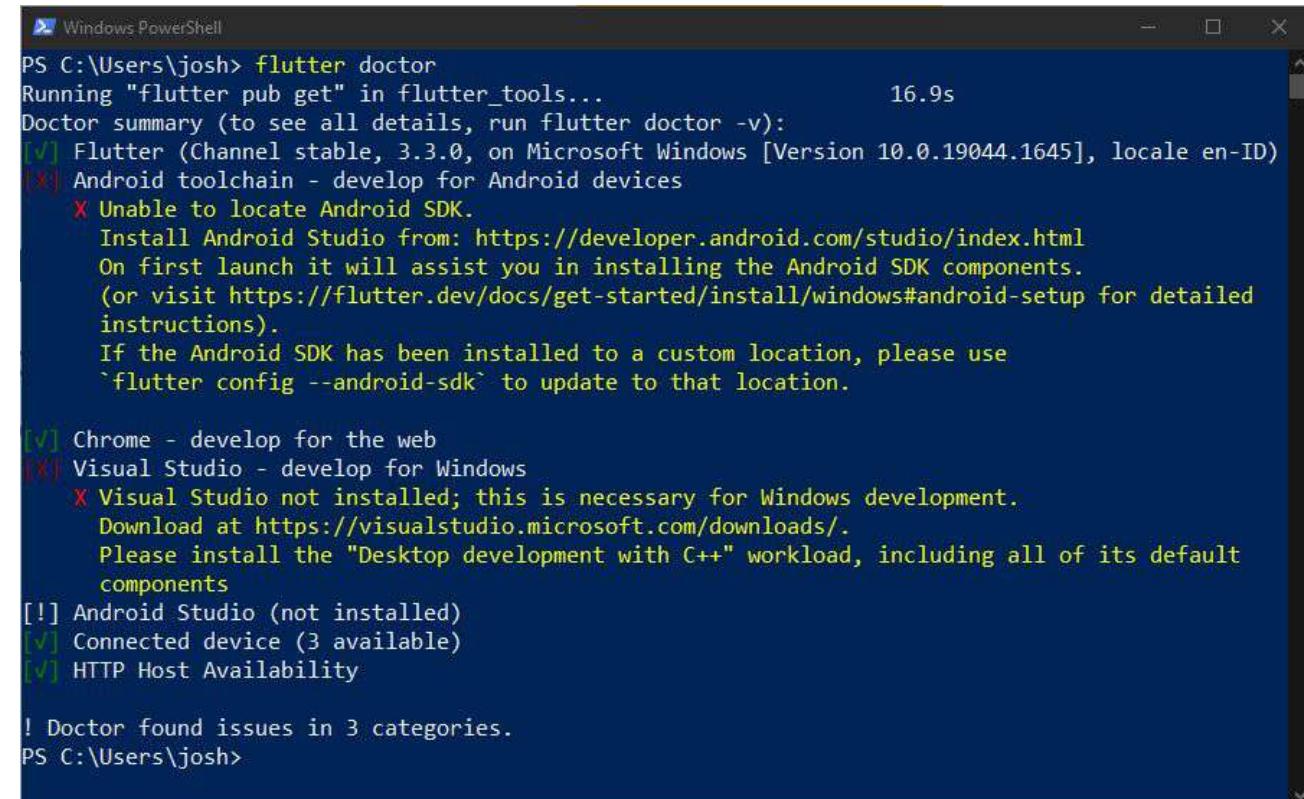
See Google's privacy policy:
<https://policies.google.com/privacy>

PS C:\Users\josh>

Jika muncul
seperti ini, flutter
berhasil terinstall

Step 9

- Menjalankan perintah **flutter doctor** pada Windows PowerShell



```
PS C:\Users\josh> flutter doctor
Running "flutter pub get" in flutter_tools...                                16.9s
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.3.0, on Microsoft Windows [Version 10.0.19044.1645], locale en-ID)
[✗] Android toolchain - develop for Android devices
    ✗ Unable to locate Android SDK.
        Install Android Studio from: https://developer.android.com/studio/index.html
        On first launch it will assist you in installing the Android SDK components.
        (or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed
        instructions).
        If the Android SDK has been installed to a custom location, please use
        `flutter config --android-sdk` to update to that location.

[✓] Chrome - develop for the web
[✗] Visual Studio - develop for Windows
    ✗ Visual Studio not installed; this is necessary for Windows development.
        Download at https://visualstudio.microsoft.com/downloads/.
        Please install the "Desktop development with C++" workload, including all of its default
        components
[!] Android Studio (not installed)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

! Doctor found issues in 3 categories.
PS C:\Users\josh>
```

Flutter doctor akan memeriksa tools apa saja yang belum terinstall:

- Android Studio
- Visual Studio (Tidak perlu)
- Android toolchain

Install Android Studio

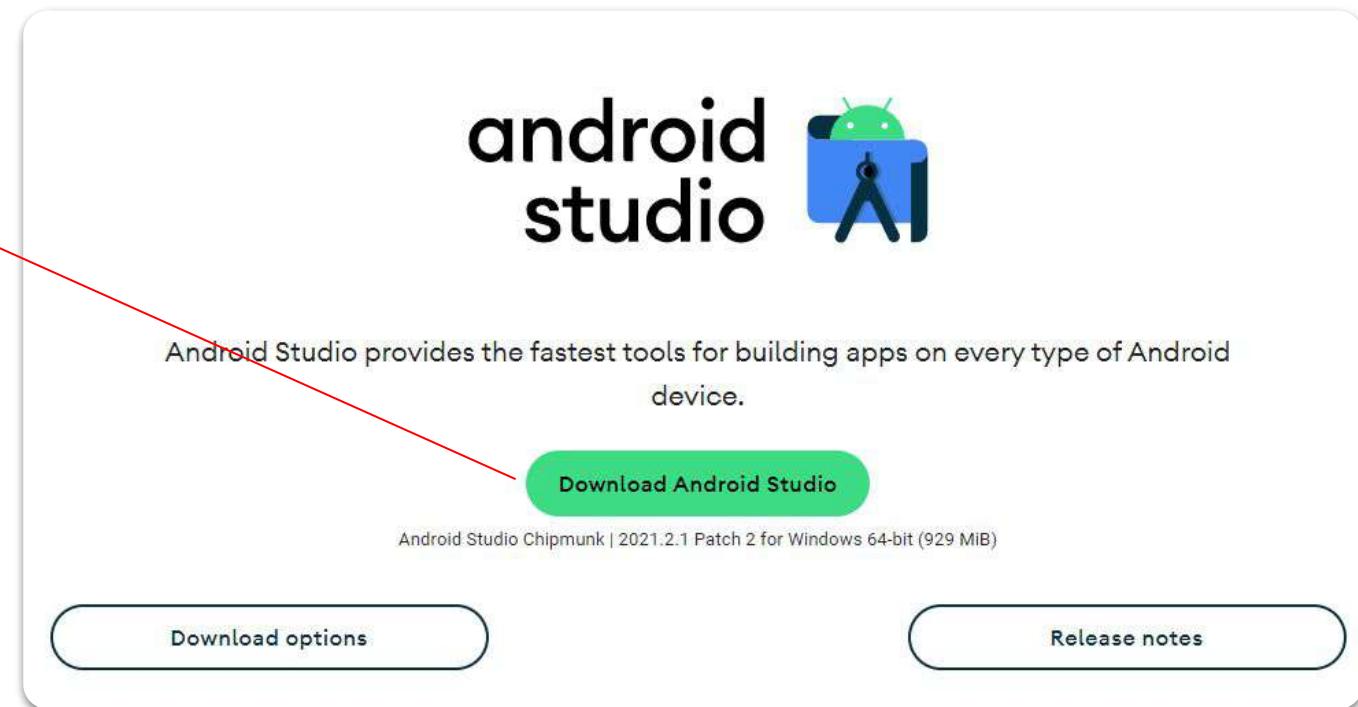
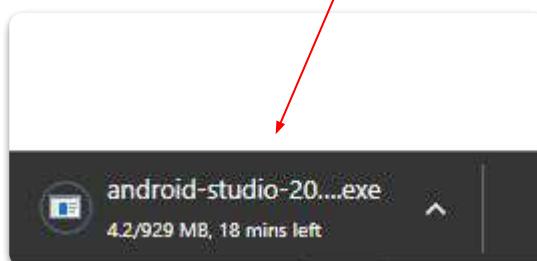
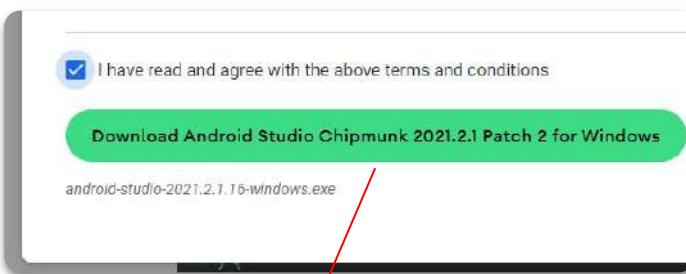
Langkah - langkah

- Step 1: Download [Android Studio](#)
 - Step 2: Double klik file installer Android Studio
 - Step 3: Klik **Next**
 - Step 4: Klik **Install**
 - Step 5: Klik **Finish**
 - Step 6: Pilih **Do not import settings**
 - Step 7: Klik **Next**
 - Step 8: Pilih UI Theme
 - Step 9: Klik **Next**
 - Step 10: Pilih **Accept** di persetujuan licenses
 - Step 11: Tunggu proses Download Components selesai
 - Step 12: Klik **Finish**
 - Step 13: Jalankan perintah flutter doctor
- 

Step 1

- Step 1: Download Android Studio

Klik checkbox



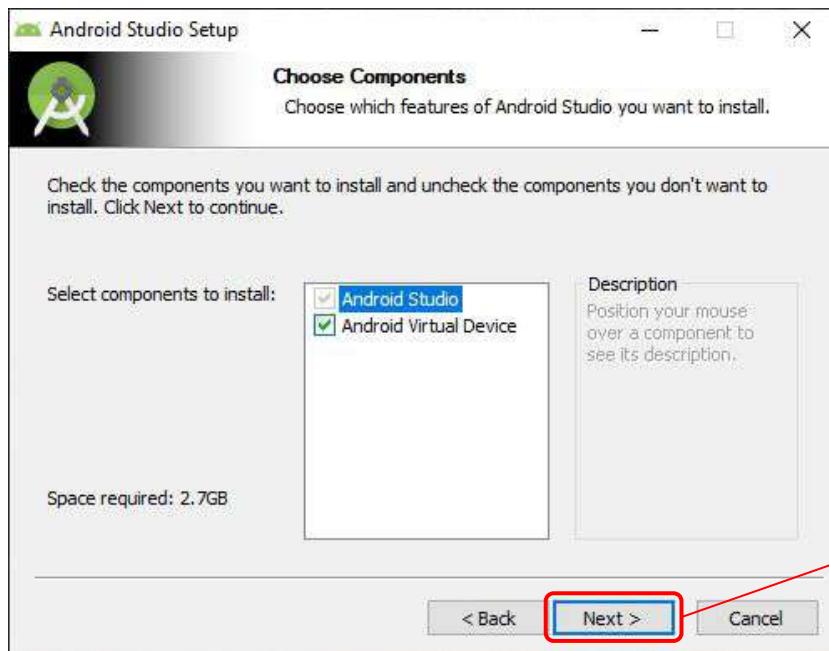
Step 2

- Double klik file installer Android Studio yang sudah di download, lalu klik **Next**



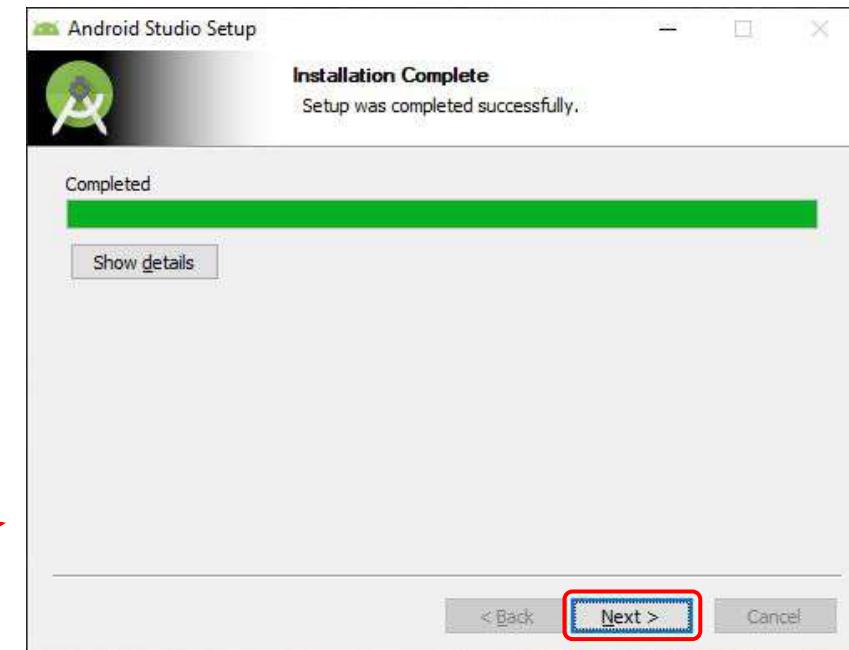
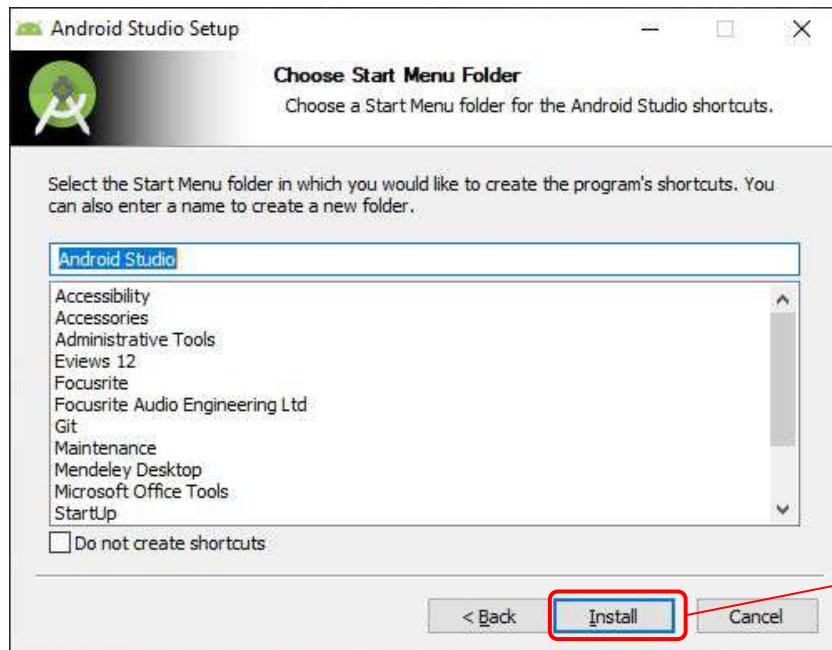
Step 3

- Klik **Next**, biarkan seluruh settingan dan lokasi penyimpanannya secara default. Klik sampai tombol Next berubah menjadi **Install**



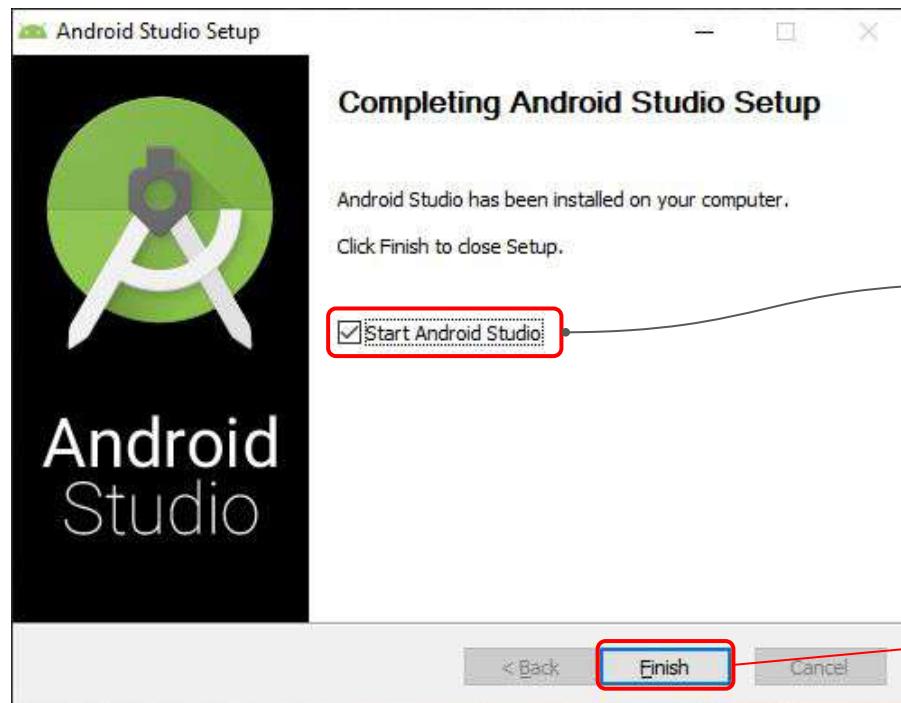
Step 4

- Klik **Install**, tunggu sampai instalasi **Completed**. Setelah itu klik **Next**

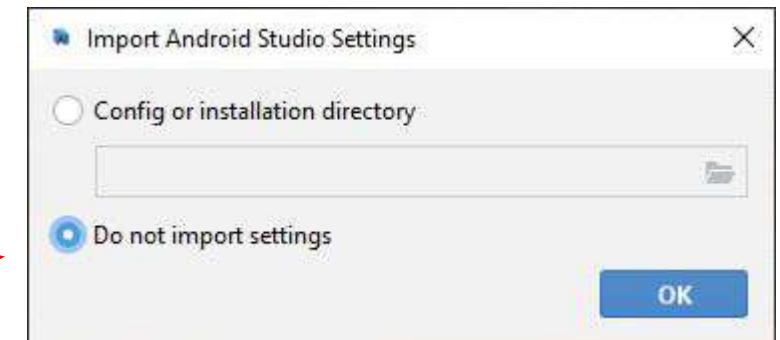


Step 5

- Klik **Finish**, pastikan checkbox Start Android Studio tercentang

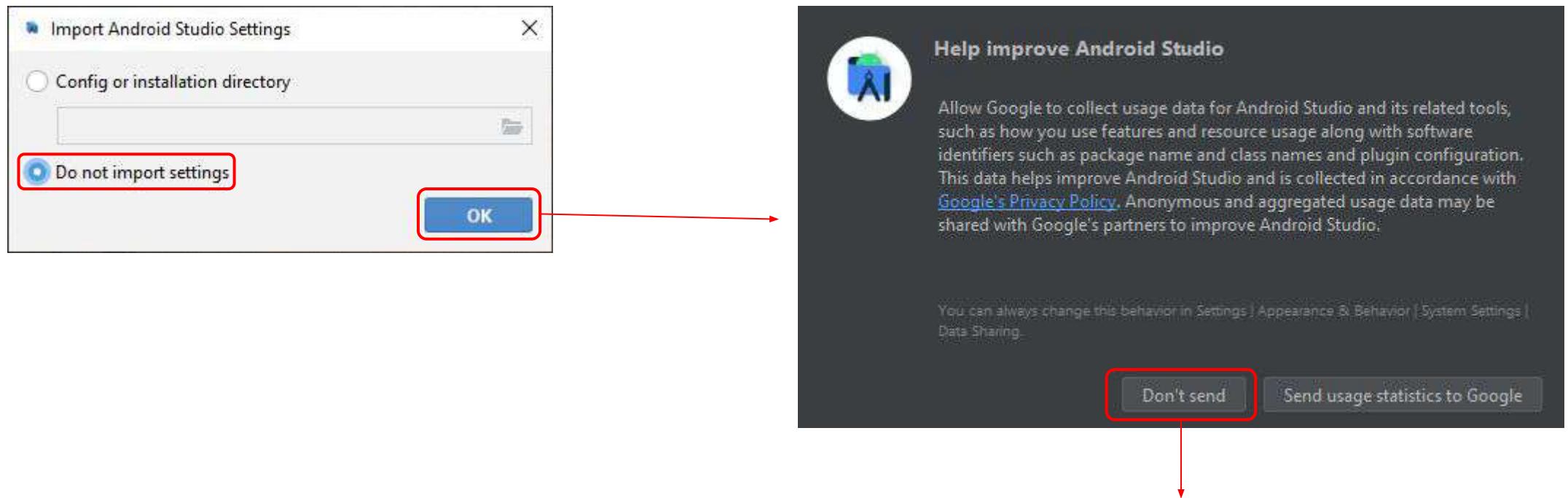


Pastikan
checkbox ini
tercentang



Step 6

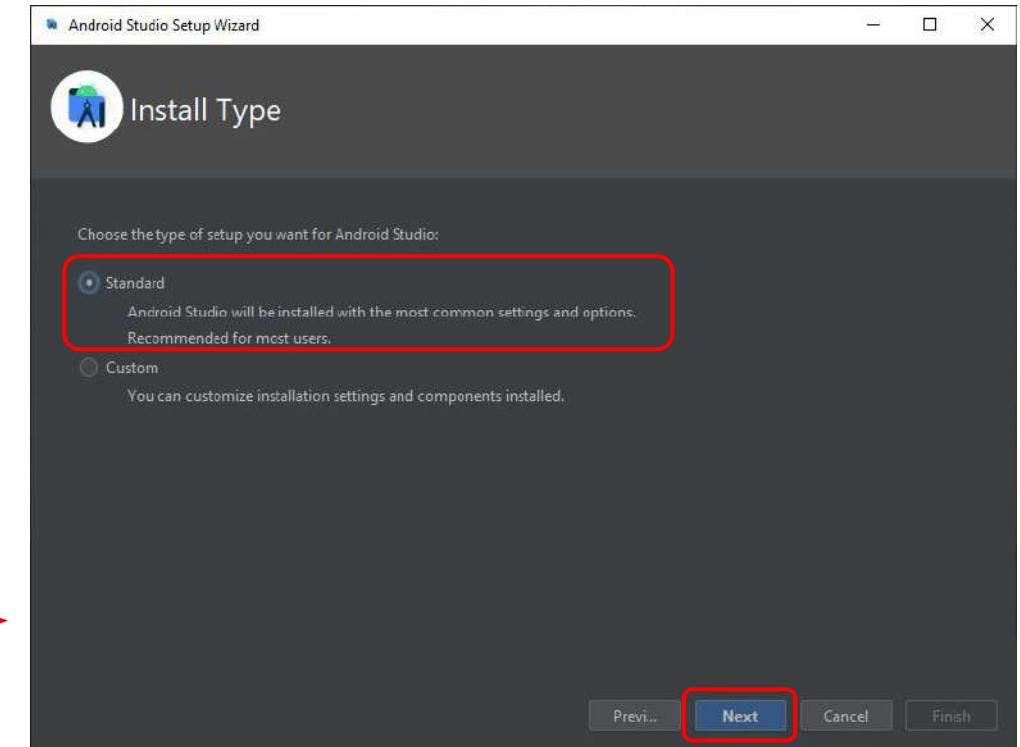
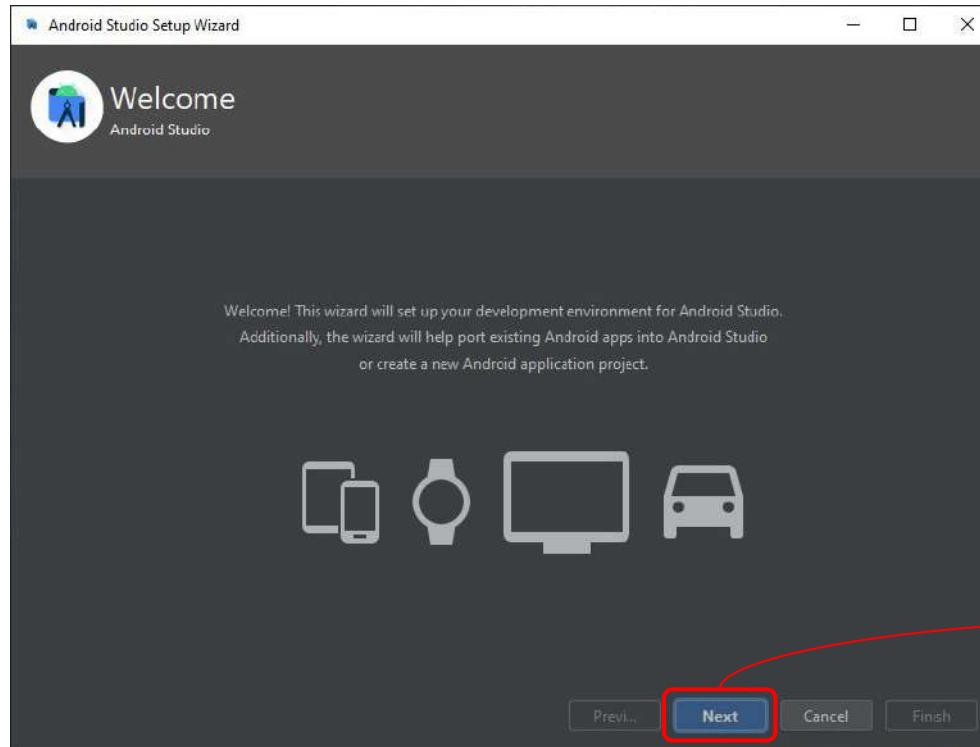
- Pilih **Do not import settings**, lalu tekan **OK**



Untuk data sharing,
klik **Don't send**

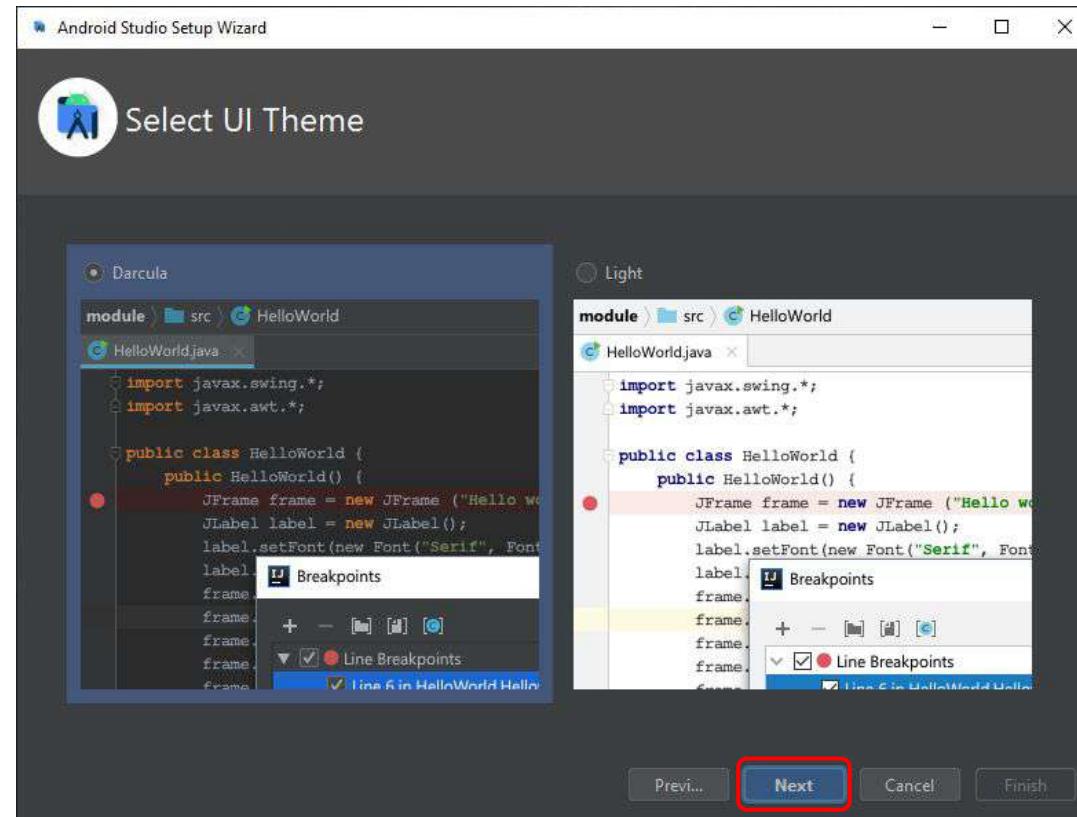
Step 7

- Di Setup Wizard ini, biarkan settingannya secara default. Klik **Next**



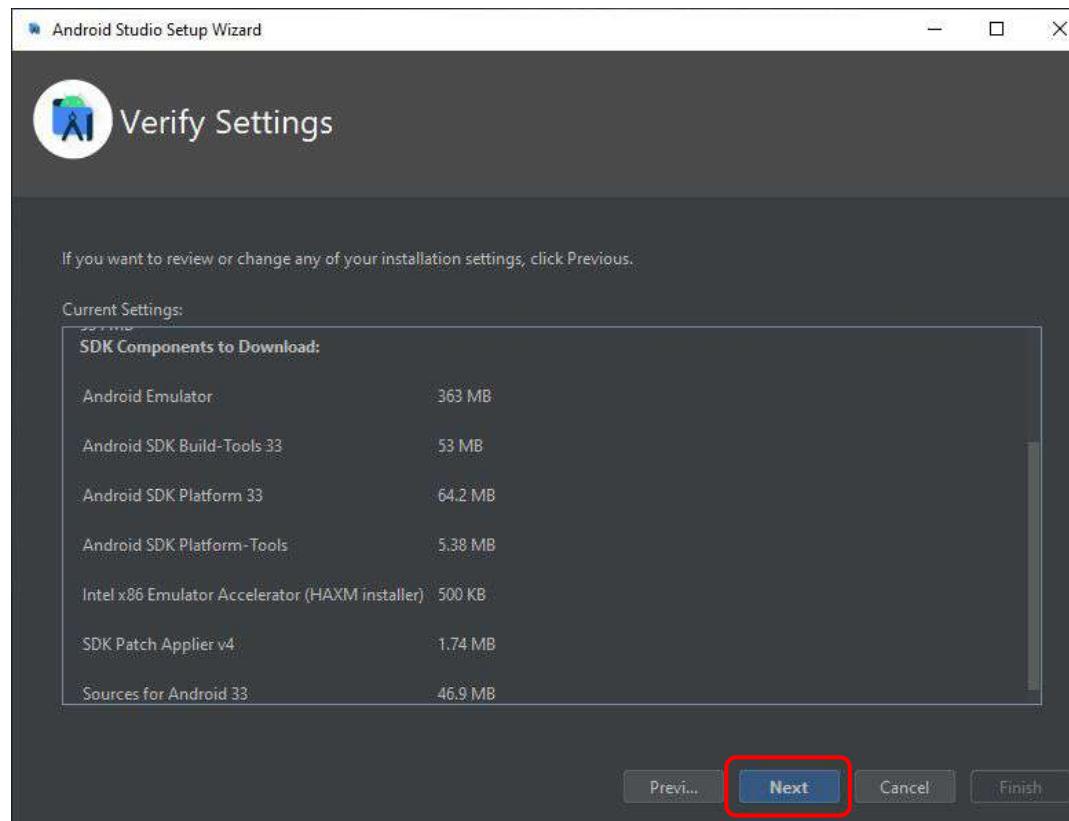
Step 8

- Pilih UI Theme, Light atau Dark (Darcula) setelah itu klik **Next**



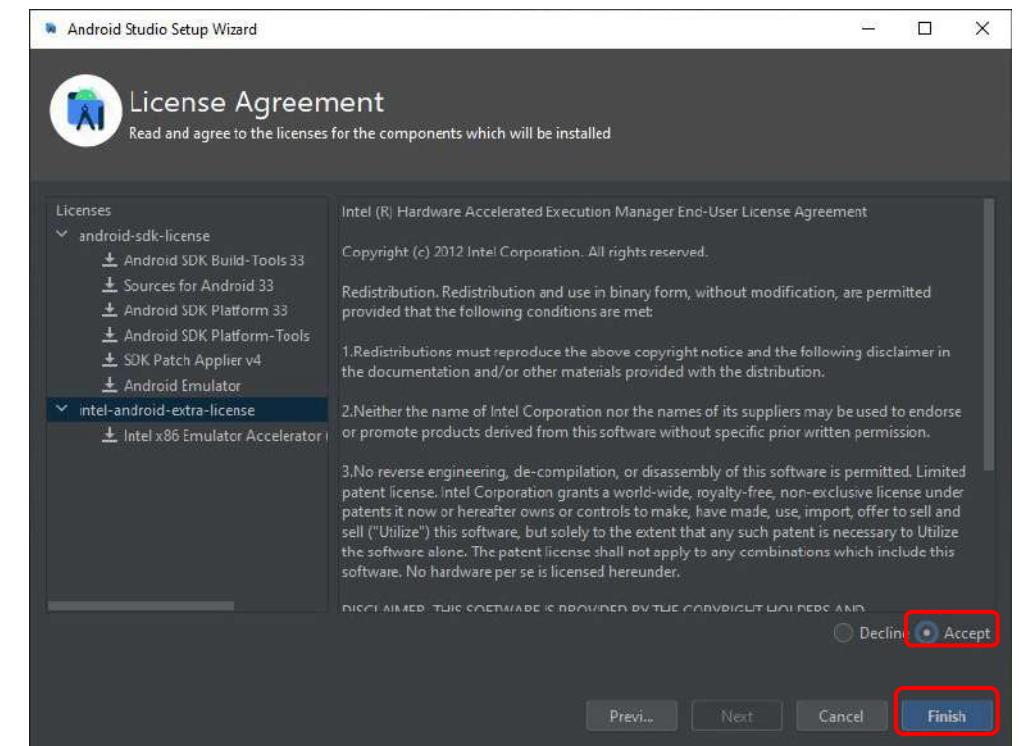
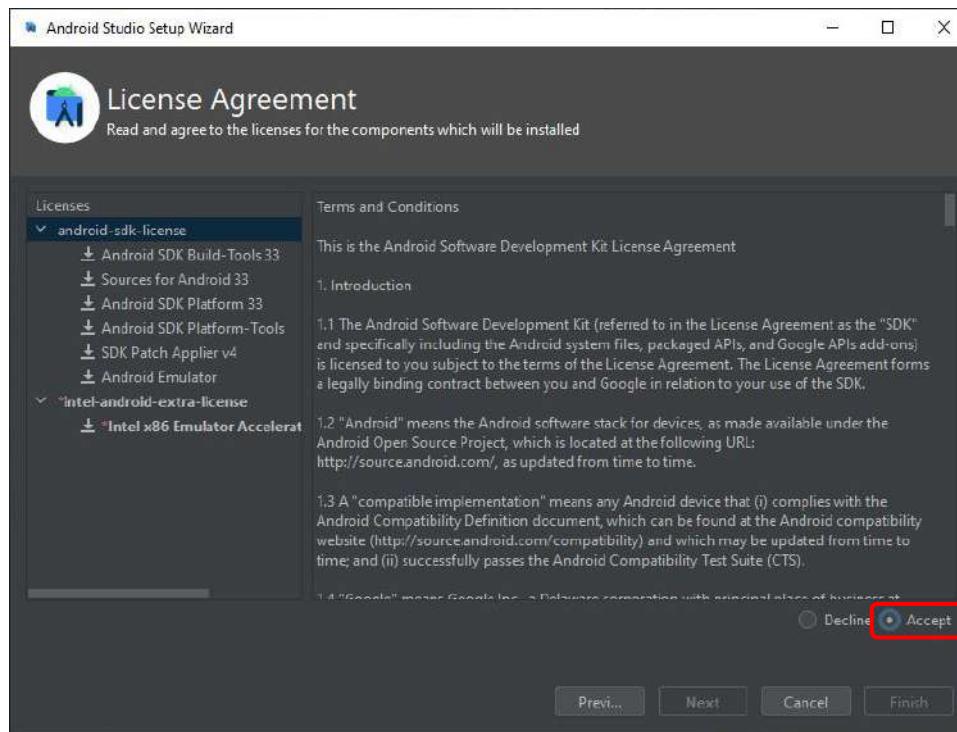
Step 9

- Klik **Next**, Disini ditampilkan komponen Android SDK apa saja yang akan di download



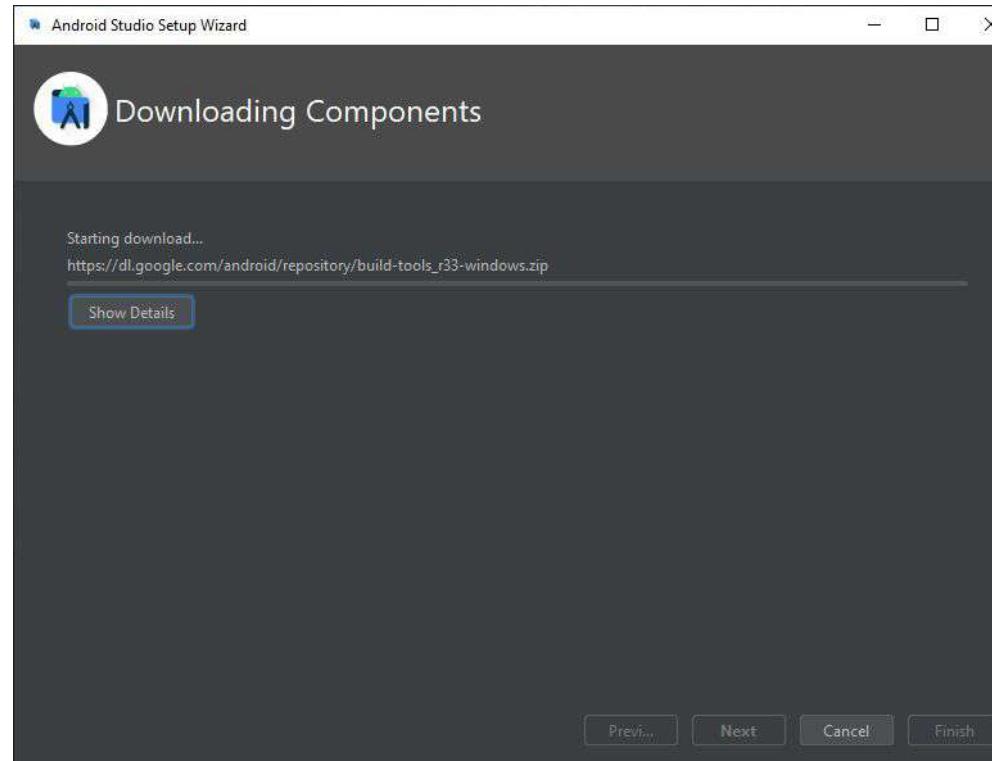
Step 10

- Pilih **Accept** di persetujuan android-sdk-license dan intel-android-extra-license, setelah itu klik **Finish**



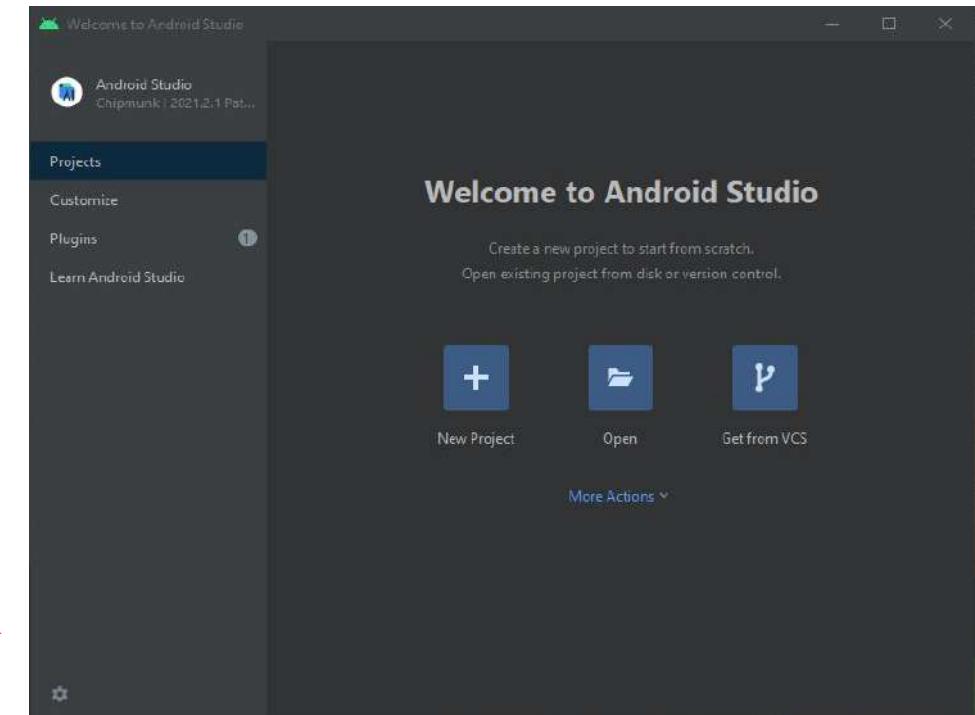
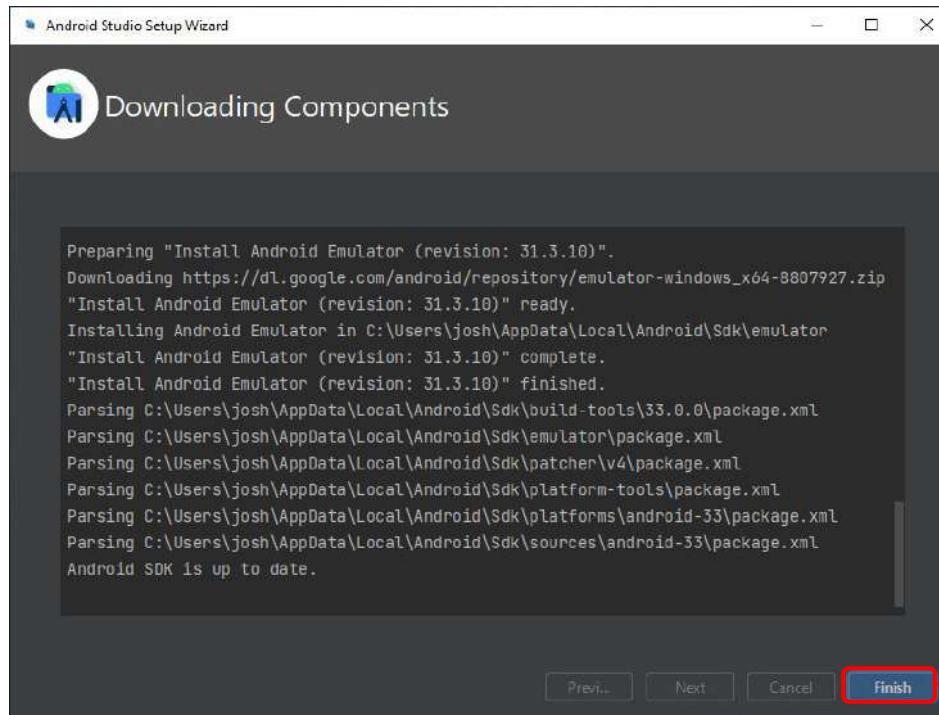
Step 11

- Tunggu proses Download Components selesai karena pada bagian ini memerlukan waktu yang agak lama, pastikan selalu terhubung dengan internet



Step 12

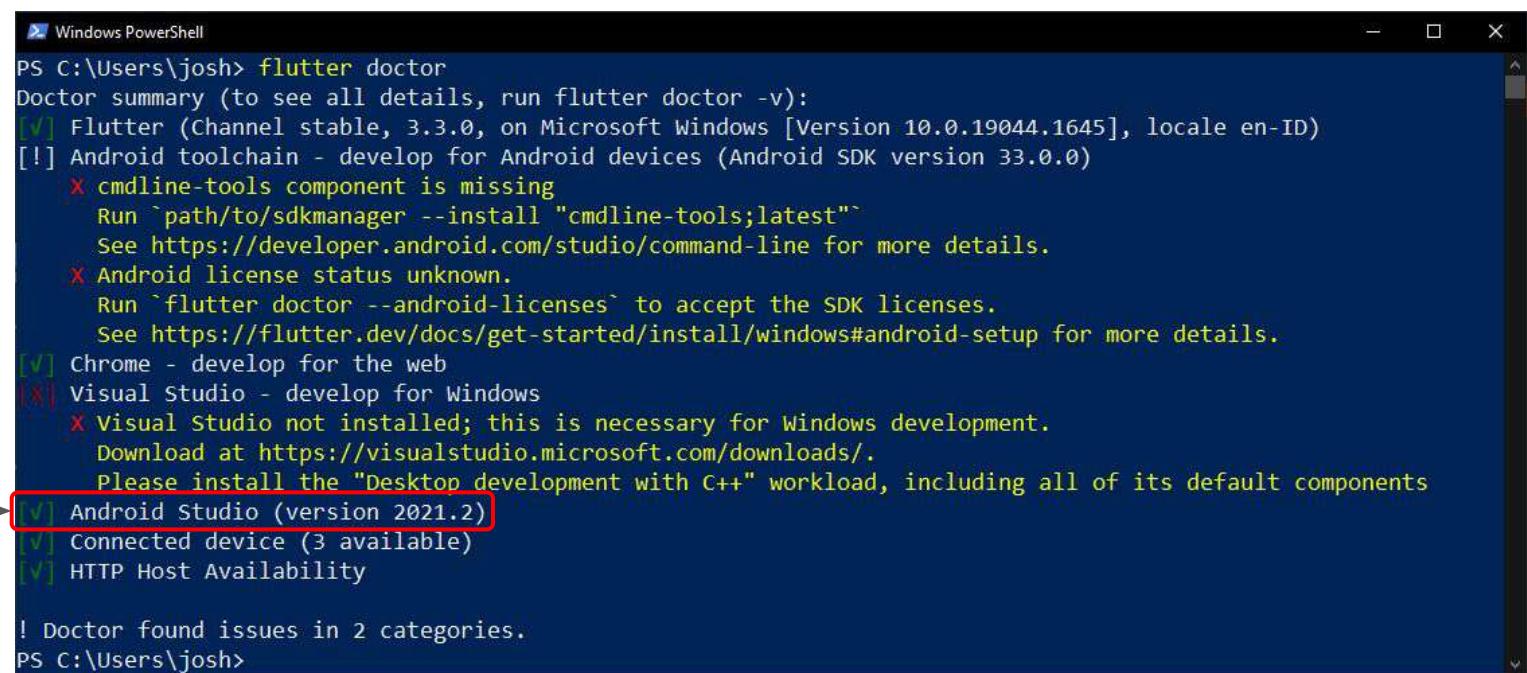
- Setelah selesai, Klik **Finish** maka tampilan **Welcome to Android Studio** akan terlihat



Step 13

- Buka kembali Windows PowerShell. Jalankan kembali **flutter doctor**

Android Studio
berhasil ter install



```
Windows PowerShell
PS C:\Users\josh> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Flutter (Channel stable, 3.3.0, on Microsoft Windows [Version 10.0.19044.1645], locale en-ID)
[!] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
    ✘ cmdline-tools component is missing
      Run `path/to/sdkmanager --install "cmdline-tools;latest"`
      See https://developer.android.com/studio/command-line for more details.
    ✘ Android license status unknown.
      Run `flutter doctor --android-licenses` to accept the SDK licenses.
      See https://flutter.dev/docs/get-started/install/windows#android-setup for more details.
[!] Chrome - develop for the web
[!] Visual Studio - develop for Windows
    ✘ Visual Studio not installed; this is necessary for Windows development.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (version 2021.2)
[!] Connected device (3 available)
[!] HTTP Host Availability

! Doctor found issues in 2 categories.
PS C:\Users\josh>
```

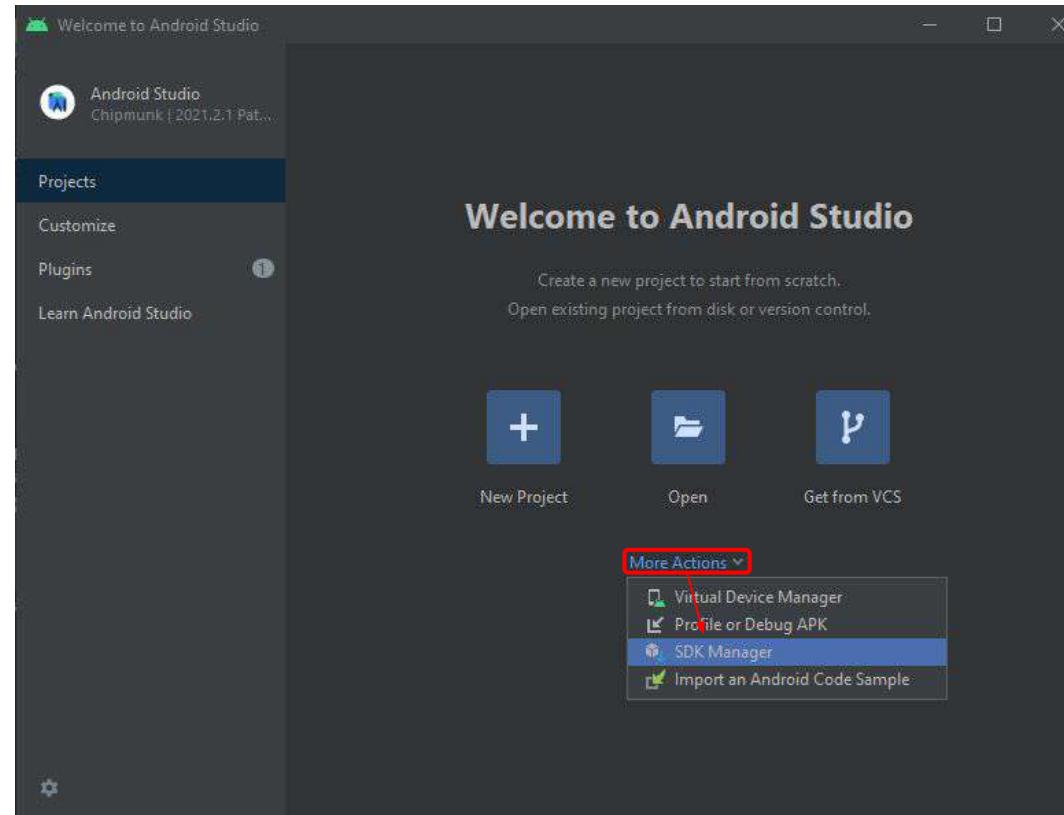
Install Android Toolchain

Langkah - langkah

- Step 1: Klik **More Actions** di Android Studio
 - Step 2: Klik SDK Tools
 - Step 3: Pastikan beberapa SDK Tools sudah terinstall atau belum
 - Step 4: Centang Tools yang diperlukan, klik **Apply** dan **OK**
 - Step 5: Pilih **Accept** untuk permintaan persetujuan dan klik **Next** untuk menginstal tools nya
 - Step 6: Klik **Next**, emulator settings biarkan secara default
 - Step 7: Klik **Finish**
 - Step 8: Klik **Apply** dan klik **OK**
 - Step 9: Jalankan perintah flutter doctor
 - Step 10: Jalankan perintah **flutter doctor --android-licenses**
 - Step 11: Jalankan kembali **flutter doctor**
- 

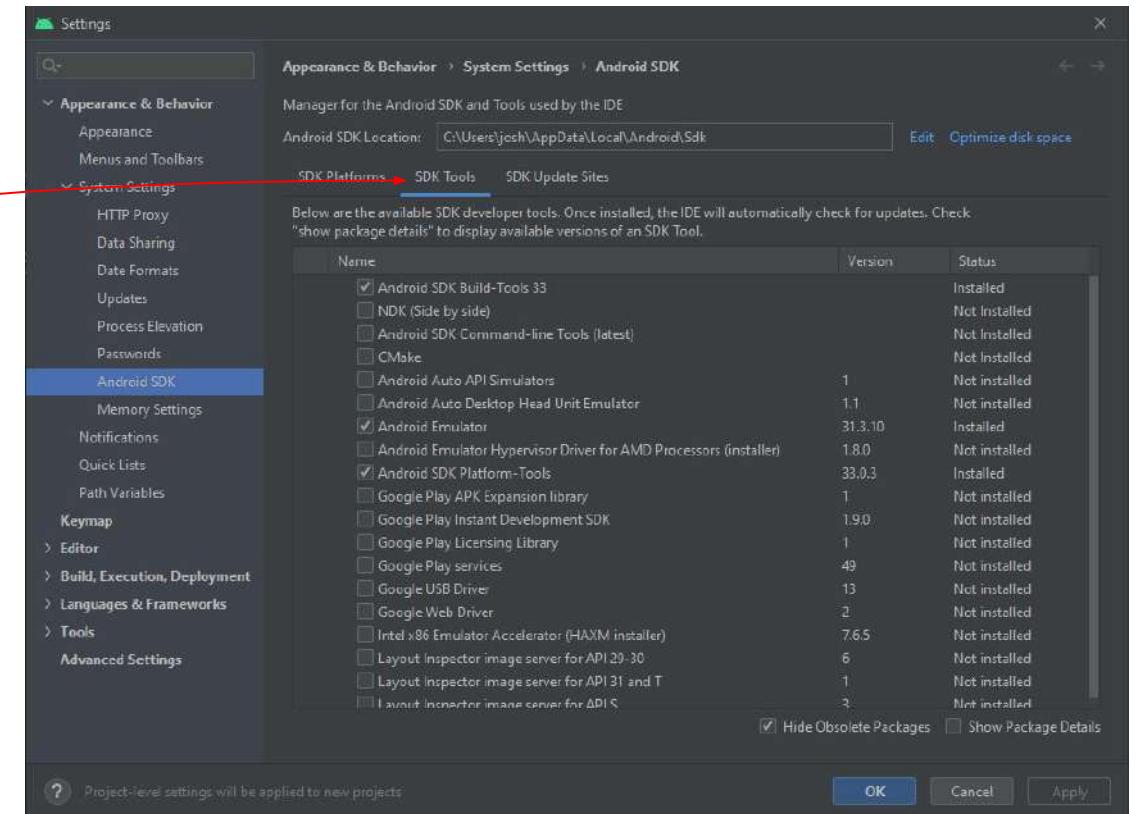
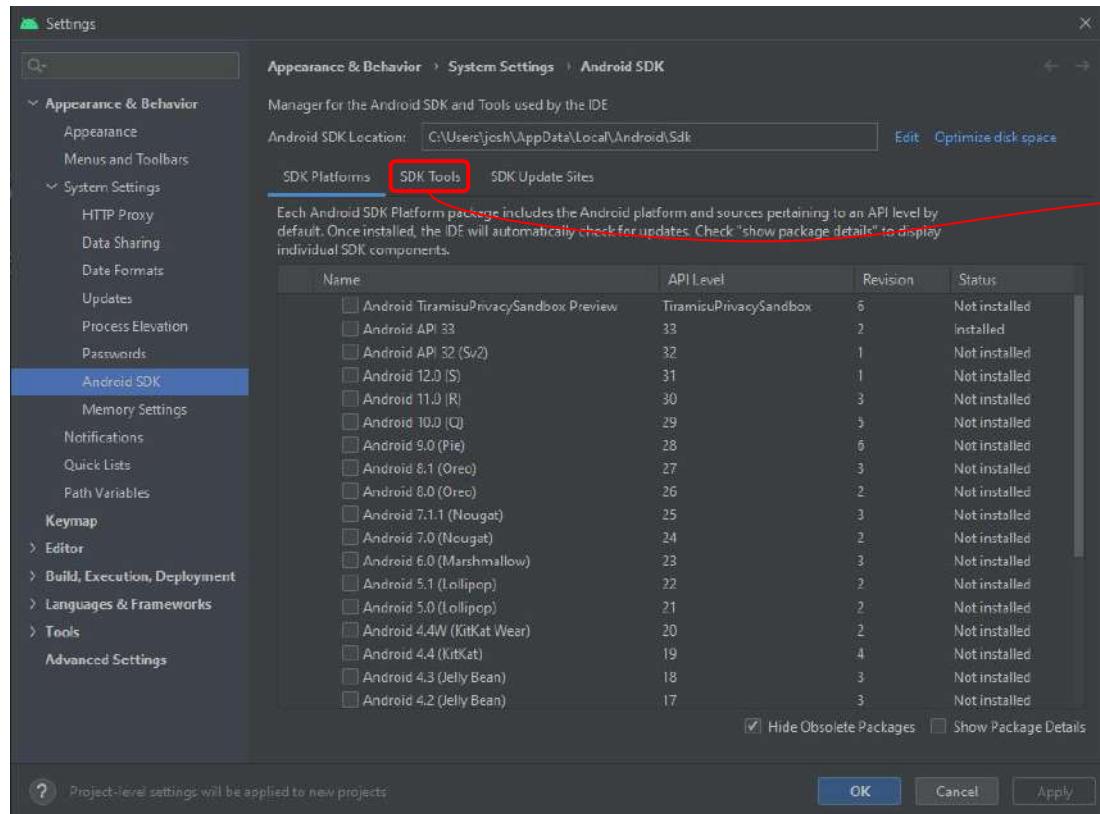
Step 1

- Klik **More Actions**, lalu klik **SDK Manager**



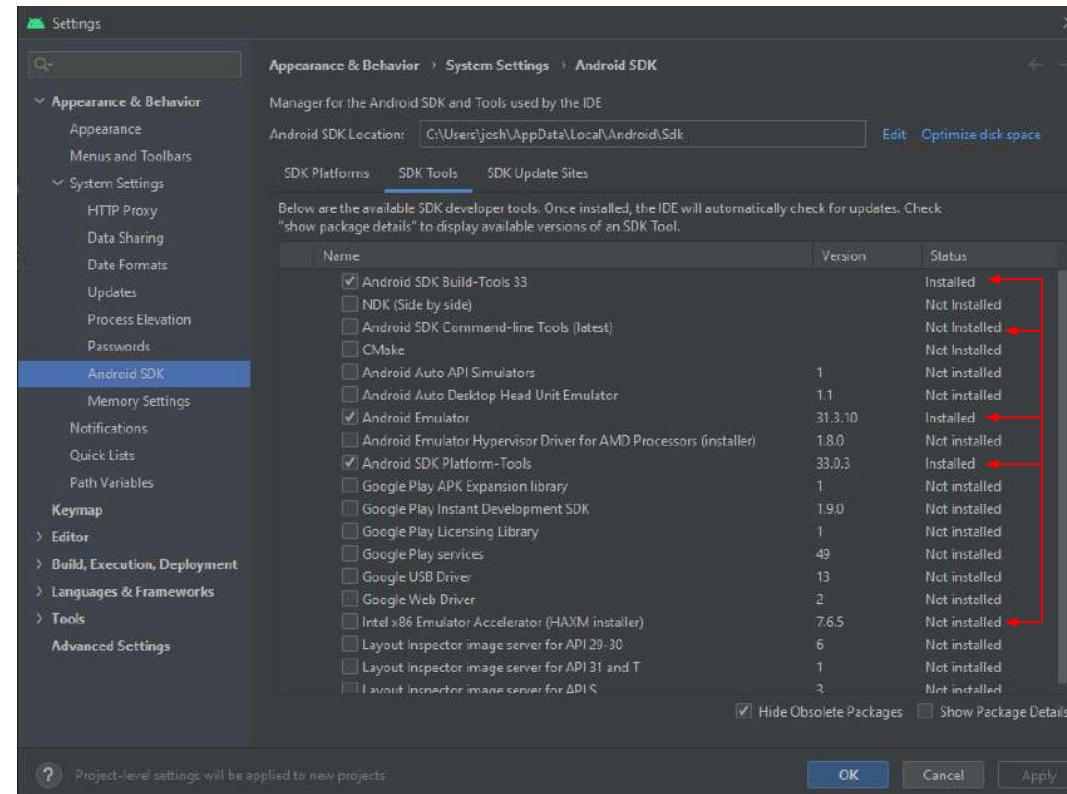
Step 2

- Klik SDK Tools



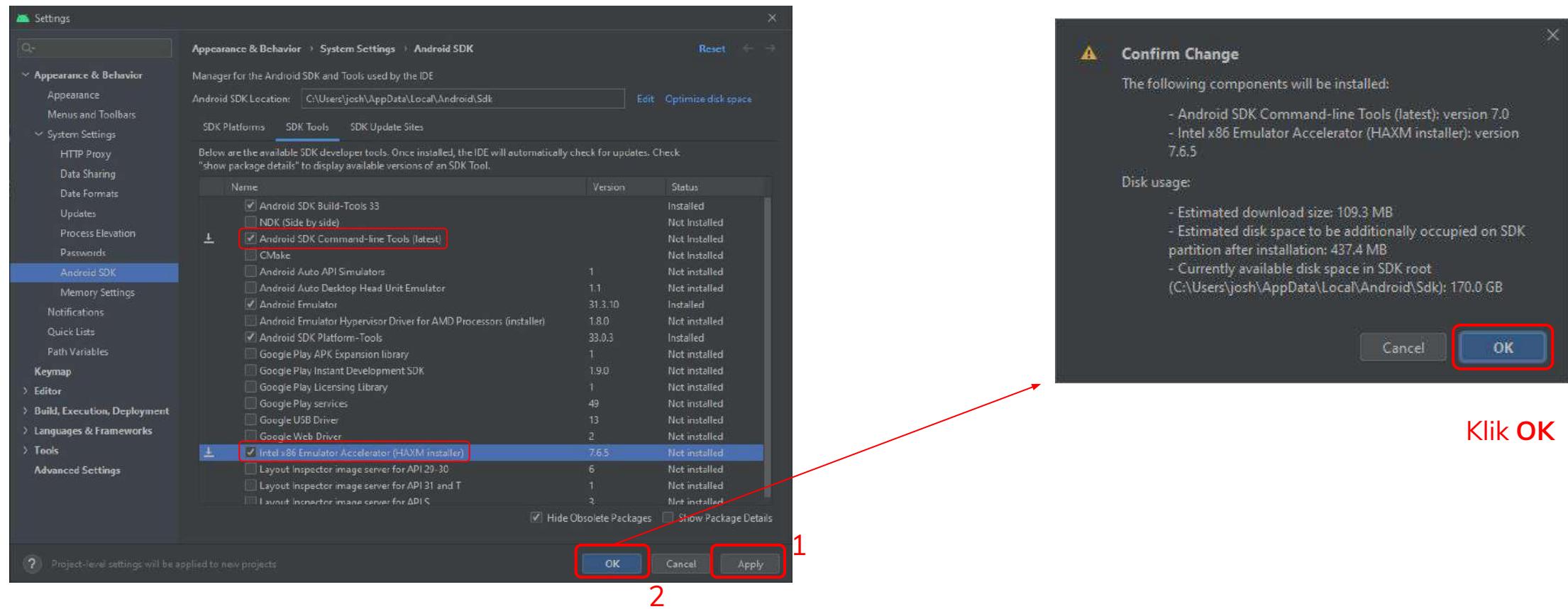
Step 3

- Pastikan Tools yang ditunjuk panah merah di bawah sudah tercentang dan terinstall, jika belum maka masuk ke Step selanjutnya



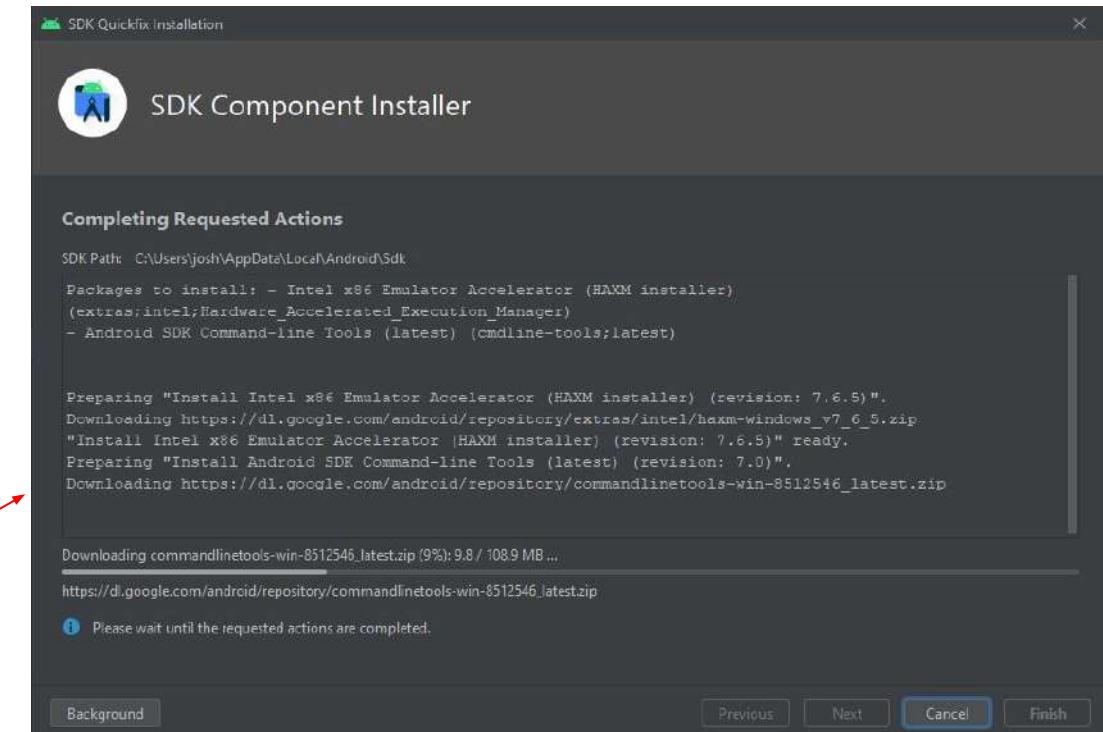
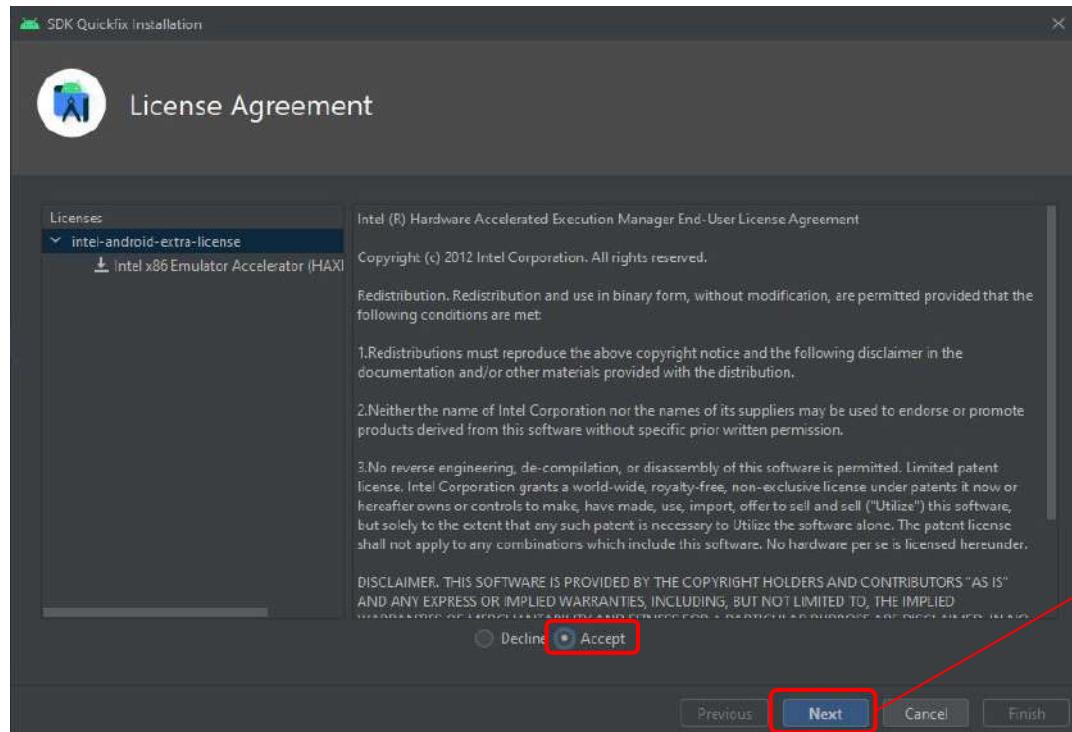
Step 4

- Centang Tools yang diperlukan, lalu klik **Apply** dan klik **OK**



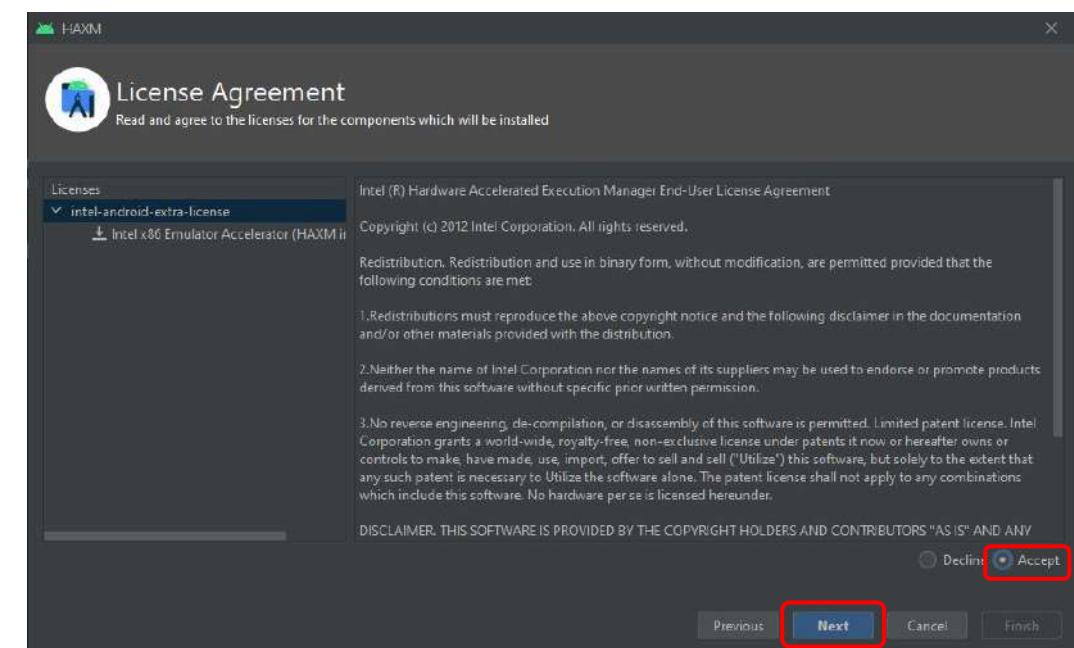
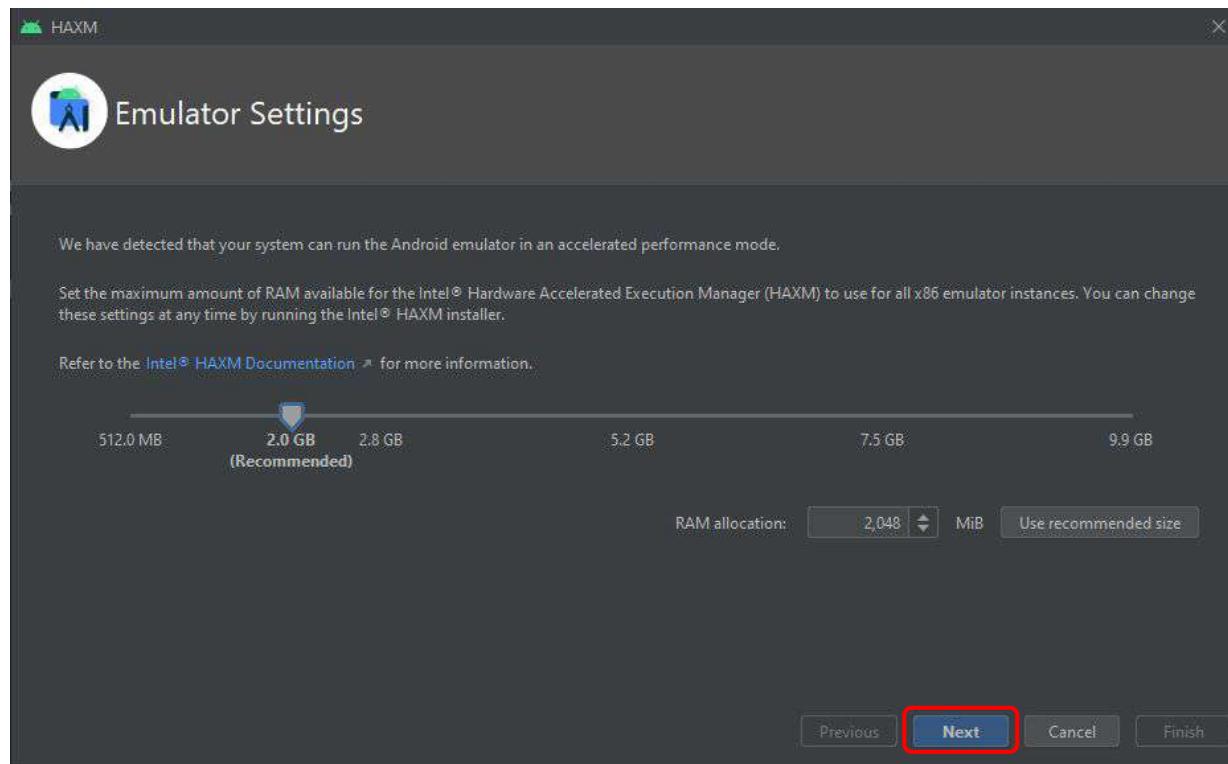
Step 5

- Pilih **Accept** untuk permintaan persetujuan, lalu klik **Next**



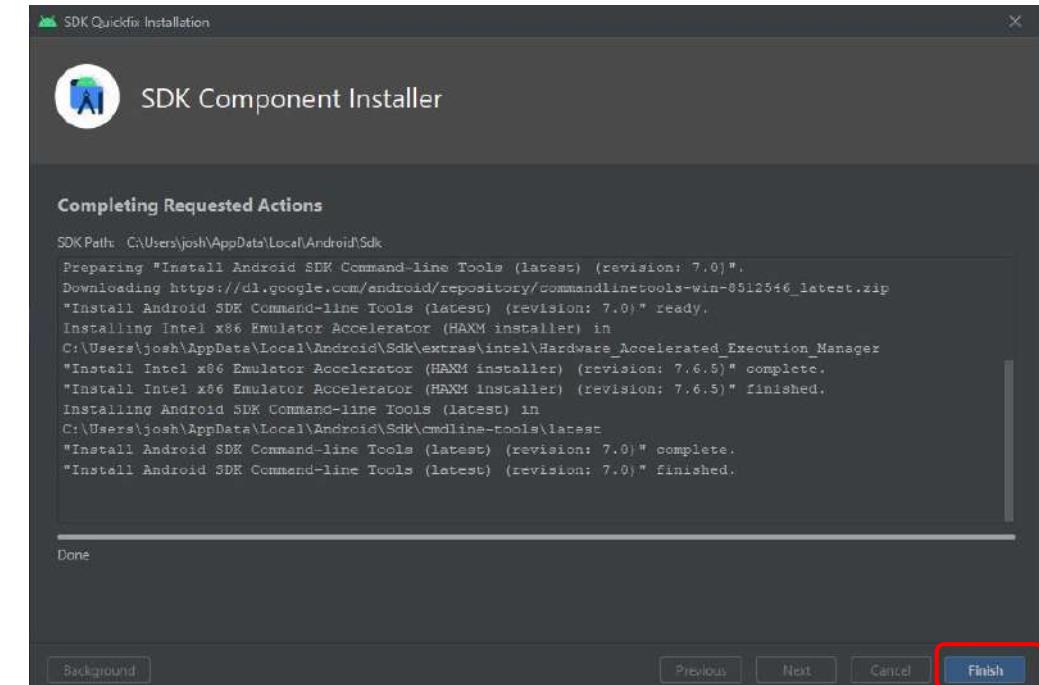
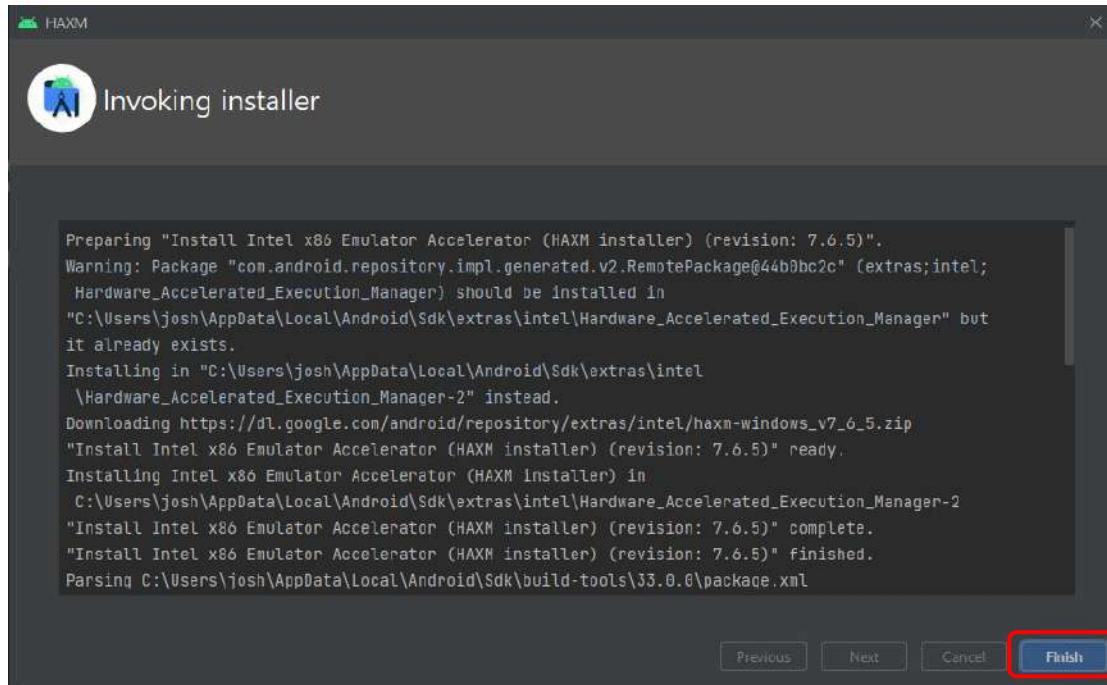
Step 6

- Klik **Next**, biarkan secara default. Jika ada permintaan persetujuan pilih **Accept**, lalu klik **Next**



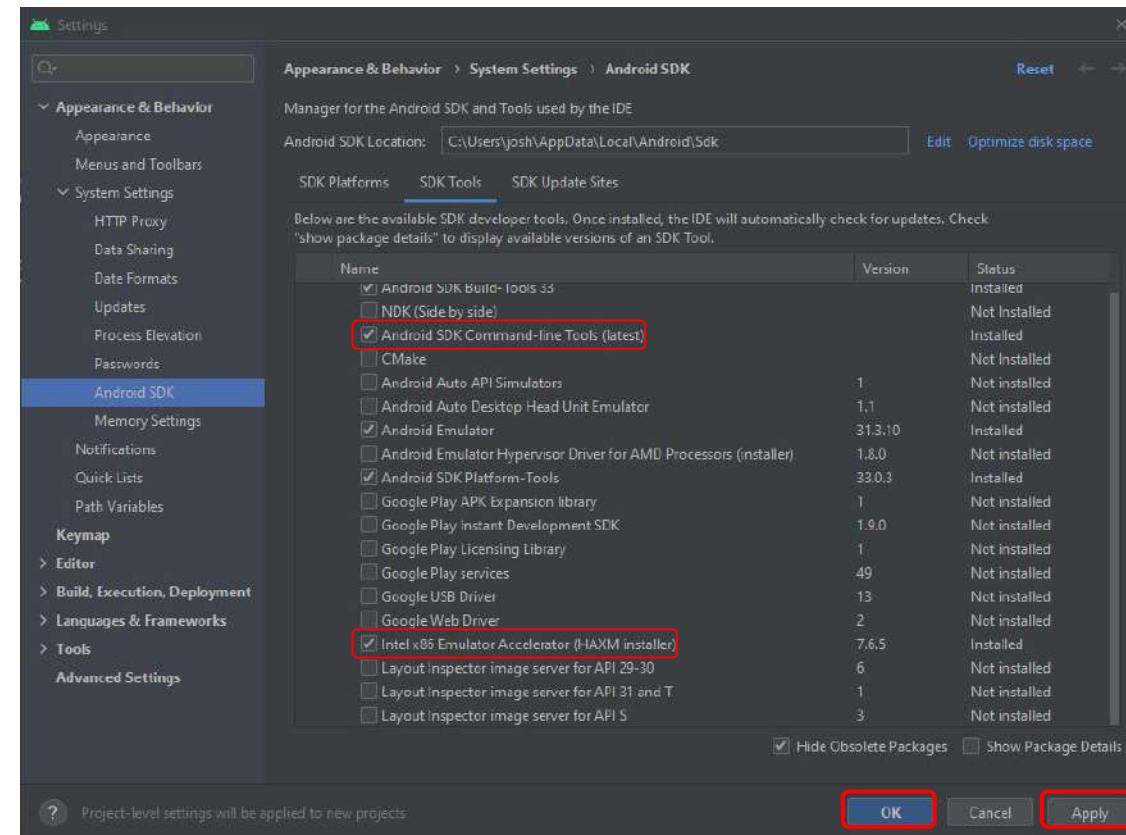
Step 7

- Klik **Finish**



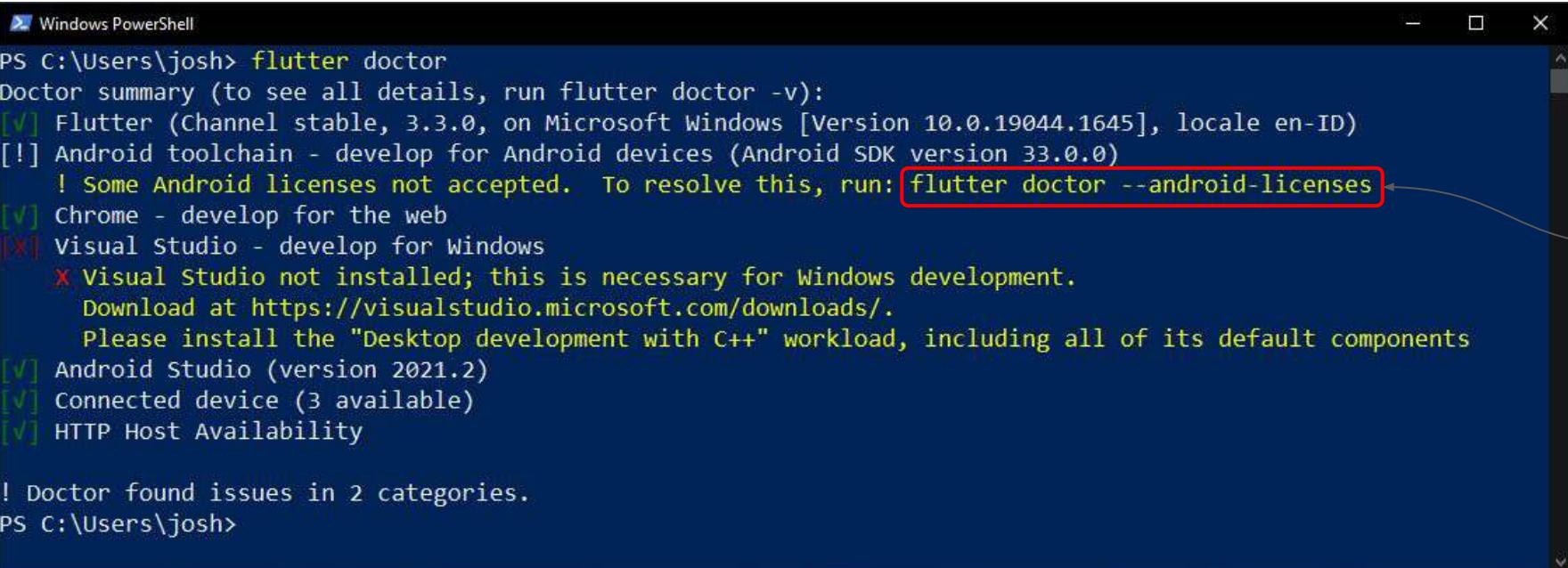
Step 8

- Jika sudah berhasil terinstall, lalu klik **Apply** dan klik **OK**



Step 9

- Buka kembali Windows PowerShell, jalankan kembali perintah **flutter doctor**



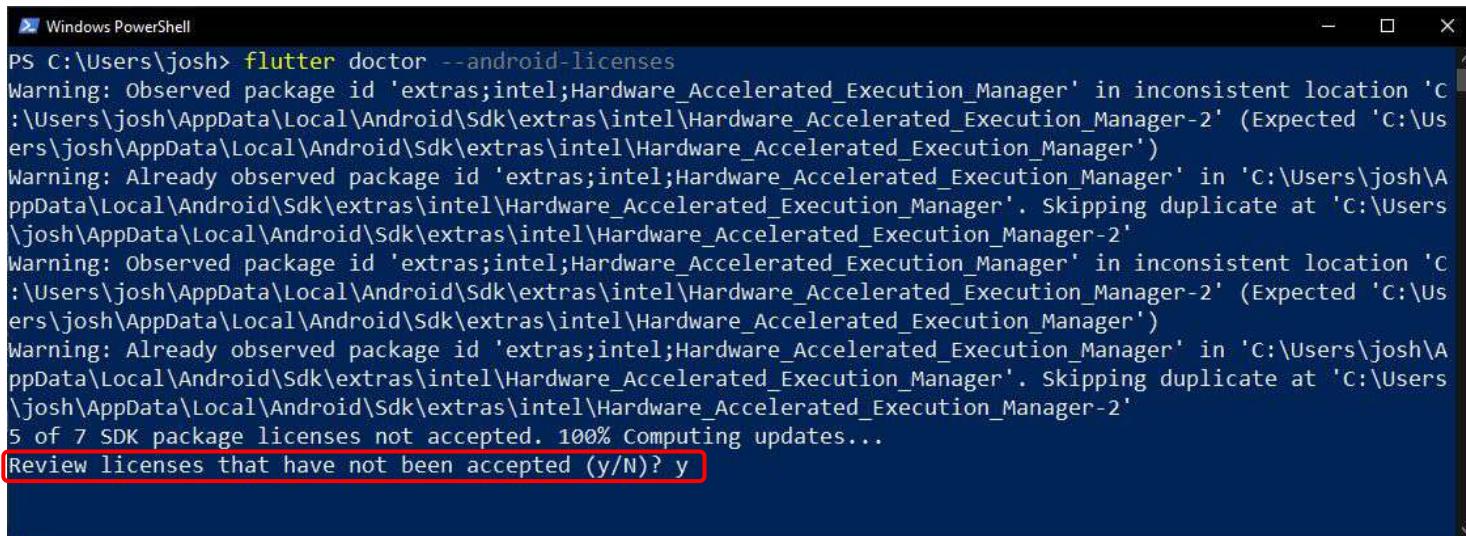
```
PS C:\Users\josh> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[!] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
    ! Some Android licenses not accepted. To resolve this, run: flutter doctor --android-licenses
[!] Chrome - develop for the web
[!] Visual Studio - develop for Windows
    X Visual Studio not installed; this is necessary for Windows development.
        Download at https://visualstudio.microsoft.com/downloads/.
        Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (version 2021.2)
[!] Connected device (3 available)
[!] HTTP Host Availability

! Doctor found issues in 2 categories.
PS C:\Users\josh>
```

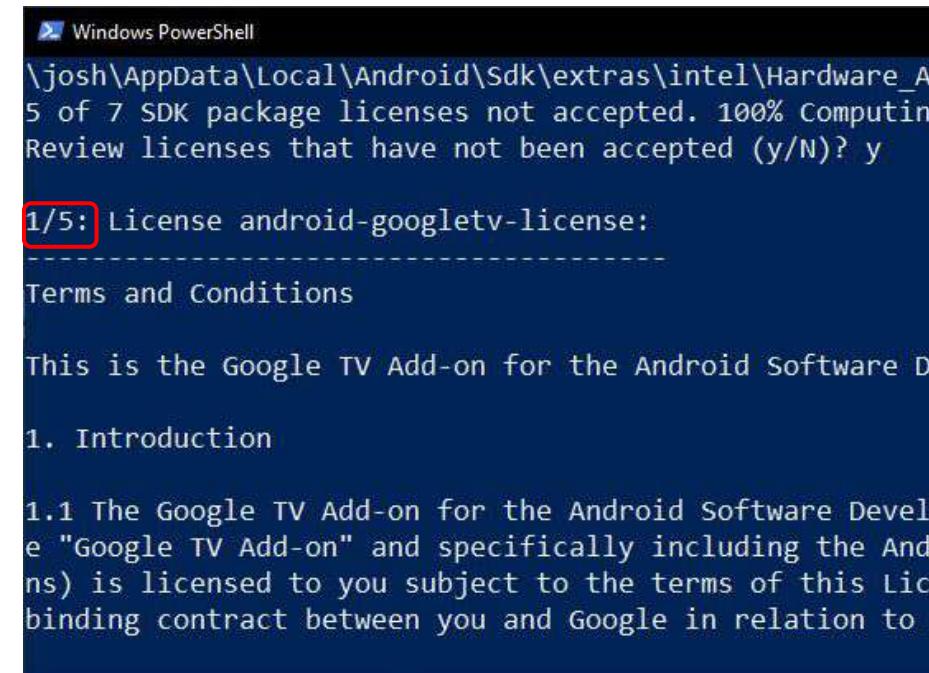
Jalankan perintah ini di Step berikutnya

Step 10

- Jalankan perintah **flutter doctor --android-licenses**, setelah itu akan muncul 6 pertanyaan terkait licenses. Balas semua dengan y dan tekan **Enter**



```
PS C:\Users\josh> flutter doctor --android-licenses
Warning: Observed package id 'extras;intel;Hardware_Accelerated_Execution_Manager' in inconsistent location 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager-2' (Expected 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager')
Warning: Already observed package id 'extras;intel;Hardware_Accelerated_Execution_Manager' in 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager'. Skipping duplicate at 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager-2'
Warning: Observed package id 'extras;intel;Hardware_Accelerated_Execution_Manager' in inconsistent location 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager-2' (Expected 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager')
Warning: Already observed package id 'extras;intel;Hardware_Accelerated_Execution_Manager' in 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager'. Skipping duplicate at 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager-2'
5 of 7 SDK package licenses not accepted. 100% Computing updates...
Review licenses that have not been accepted (y/N)? y
```



```
\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager-2' (Expected 'C:\Users\josh\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager')
5 of 7 SDK package licenses not accepted. 100% Computing updates...
Review licenses that have not been accepted (y/N)? y
1/5: License android-googletv-license:
-----
Terms and Conditions
This is the Google TV Add-on for the Android Software Development Kit (SDK). The "Google TV Add-on" and specifically including the Android TV (ATV) version of the "Google TV Add-on" (the "Product") is licensed to you subject to the terms of this License Agreement (the "License Agreement"), which is binding between you and Google in relation to the Product.
```

Step 11

- Jalankan kembali **flutter doctor**

Android
Toolchain
berhasil ter
install



```
PS C:\Users\josh> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.3.0, on Microsoft Windows [Version 10.0.19044.1645], locale en-ID)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
[✓] Chrome - develop for the web
[✗] Visual Studio - develop for Windows
  X Visual Studio not installed; this is necessary for Windows development.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components.
[✓] Android Studio (version 2021.2)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

! Doctor found issues in 1 category.
PS C:\Users\josh>
```



Terima Kasih

Flutter Platform Widget

Abstract

Memanfaatkan widget dengan gaya berbeda pada Android dan iOS.



Outline



MaterialApp



CupertinoApp

MaterialApp

Pengertian

- Widget dasar yang **mengemas seluruh widget** dalam aplikasi
- Widget yang digunakan pada sistem **Android**
- Di-import dari **package:flutter/material.dart**

MaterialApp



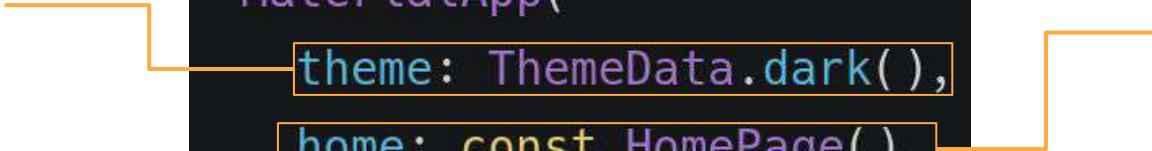
Struktur

Widget yang pertama kali dibuka, diletakkan pada bagian **home**.

Mengatur
tema aplikasi

```
MaterialApp(  
    theme: ThemeData.dark(),  
    home: const HomePage(),  
);
```

Halaman
utama



Struktur

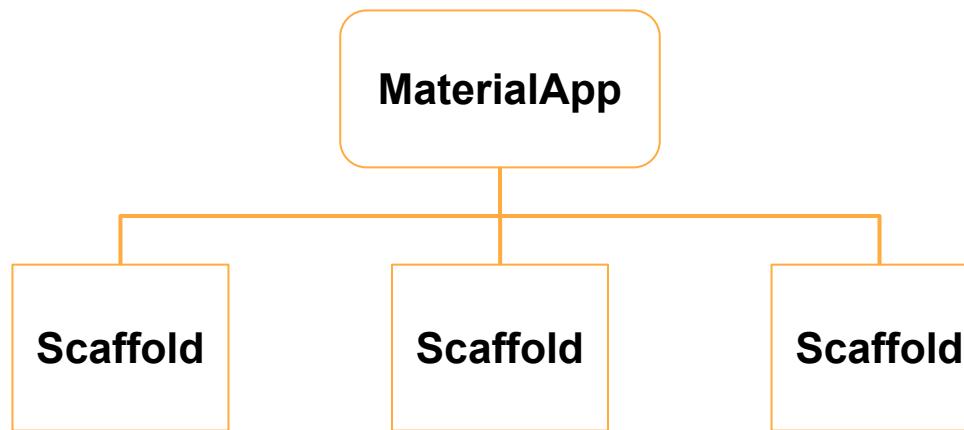
Mengatur halaman juga dapat dilakukan dengan menggunakan **routes** dan **initialRoute**.



Scaffold

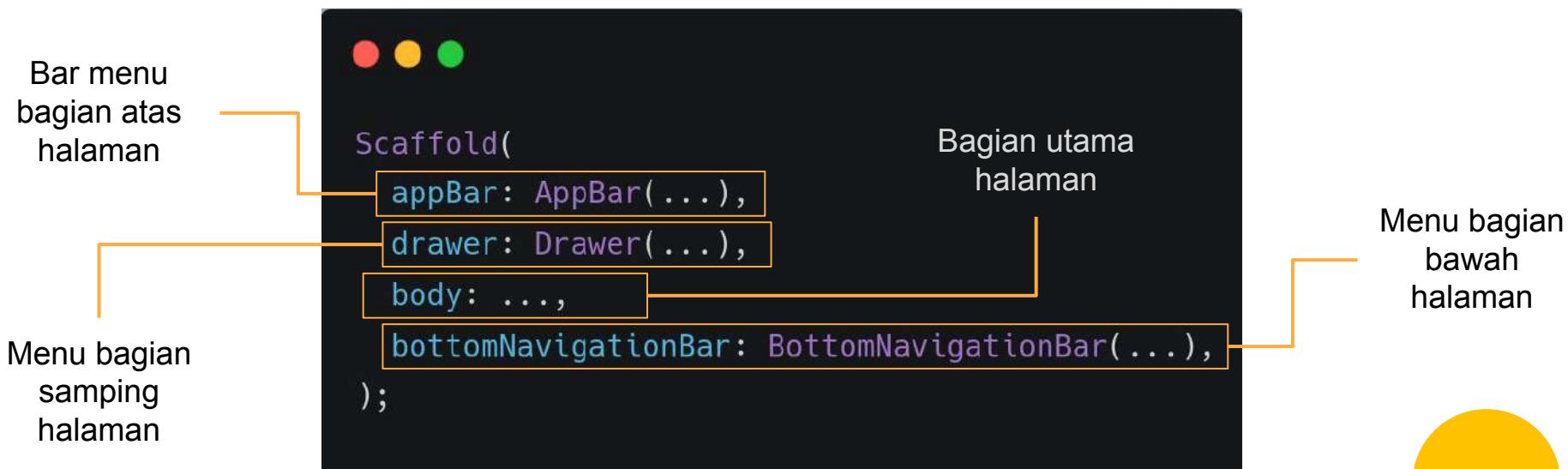
Pengertian

Widget dasar untuk membangun sebuah halaman pada MaterialApp

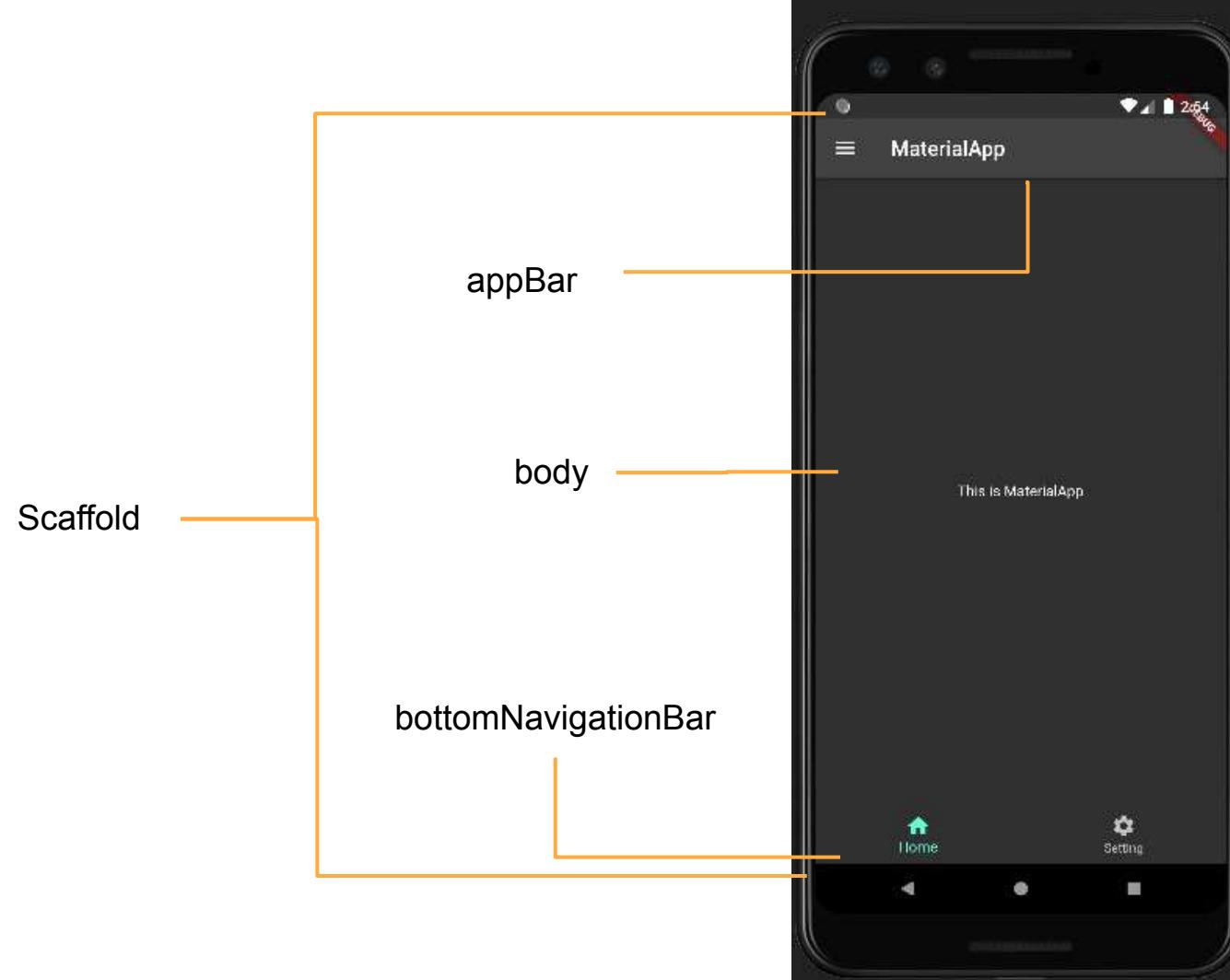


Struktur

Membentuk tata letak dasar sebuah halaman yang ditulis melalui properti-properti

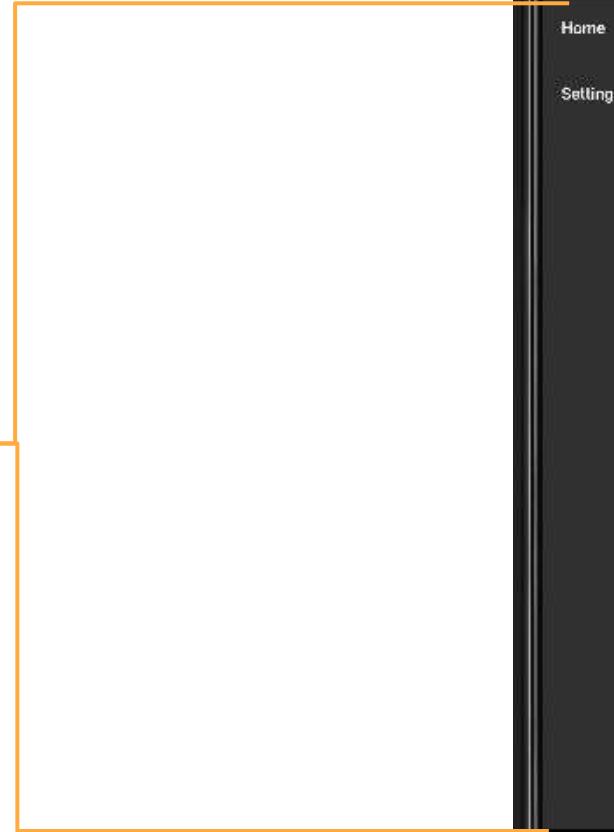


Struktur



Struktur

drawer



CupertinoApp

Pengertian

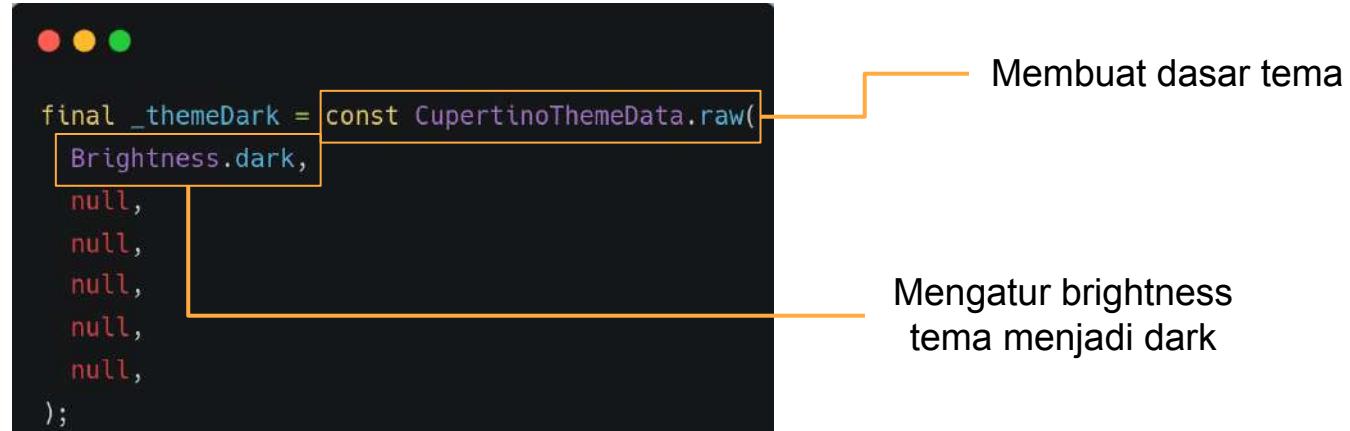
- Widget dasar yang **mengemas seluruh widget** dalam aplikasi
- Widget yang digunakan pada sistem **iOS**
- Di-import dari **package:flutter/cupertino.dart**

CupertinoApp



Struktur

- Variabel `_themeDark` dibuat untuk meyimpan tema
- Diperlukan karena Cupertino **tidak menyediakan** `ThemeData.dark()` seperti pada Material



Struktur

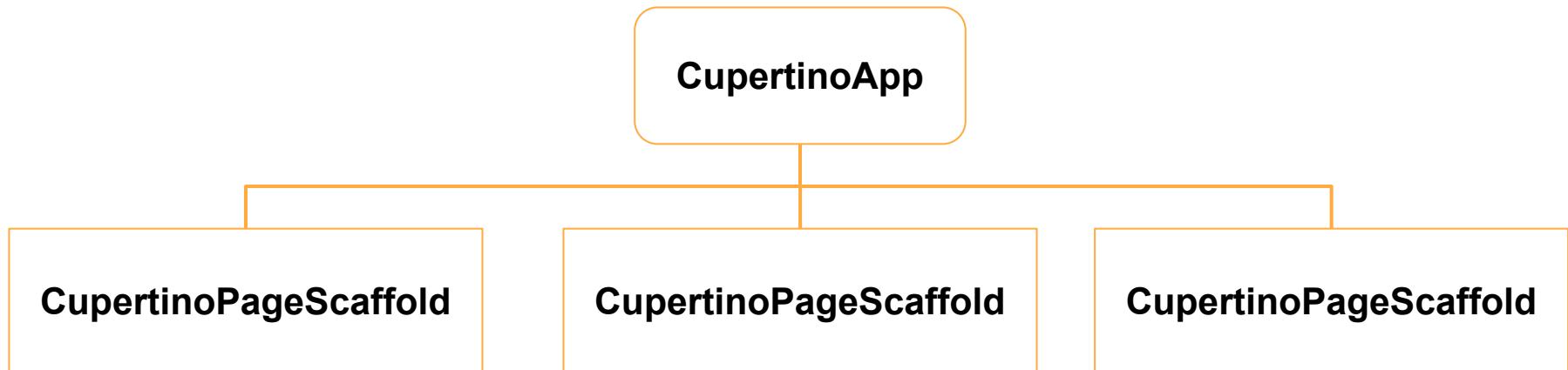
Widget yang pertama kali dibuka, diletakkan pada bagian **home**.



CupertinoPageScaffold

Pengertian

Widget dasar untuk membangun sebuah halaman pada CupertinoApp



Struktur

Membentuk tata letak dasar sebuah halaman yang ditulis melalui properti-properti

Bar menu
bagian atas
halaman

```
CupertinoPageScaffold(  
    navigationBar: CupertinoNavigationBar(...),  
    child: ...,  
);
```

Bagian utama
halaman

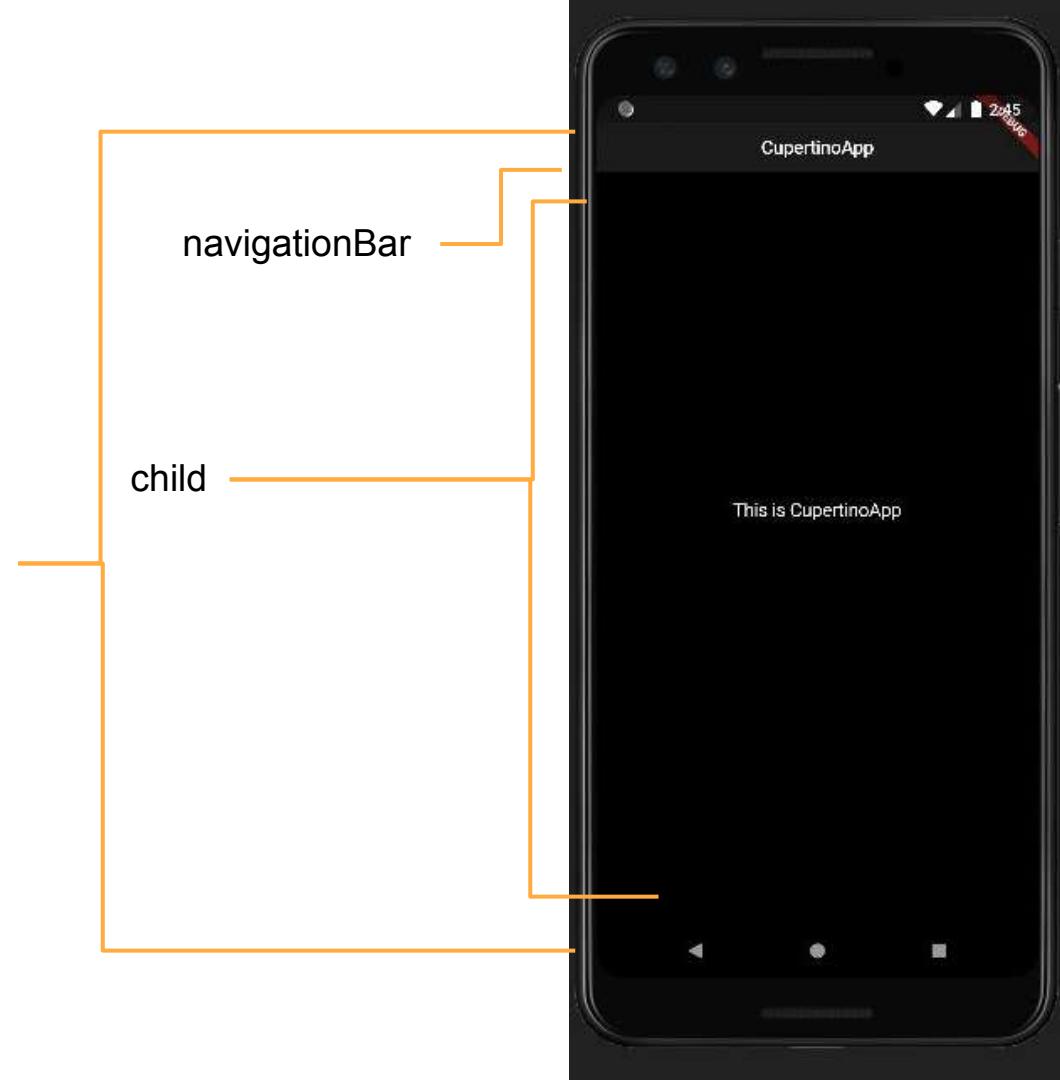
Struktur

CupertinoPageScaffold

navigationBar

child

This is CupertinoApp

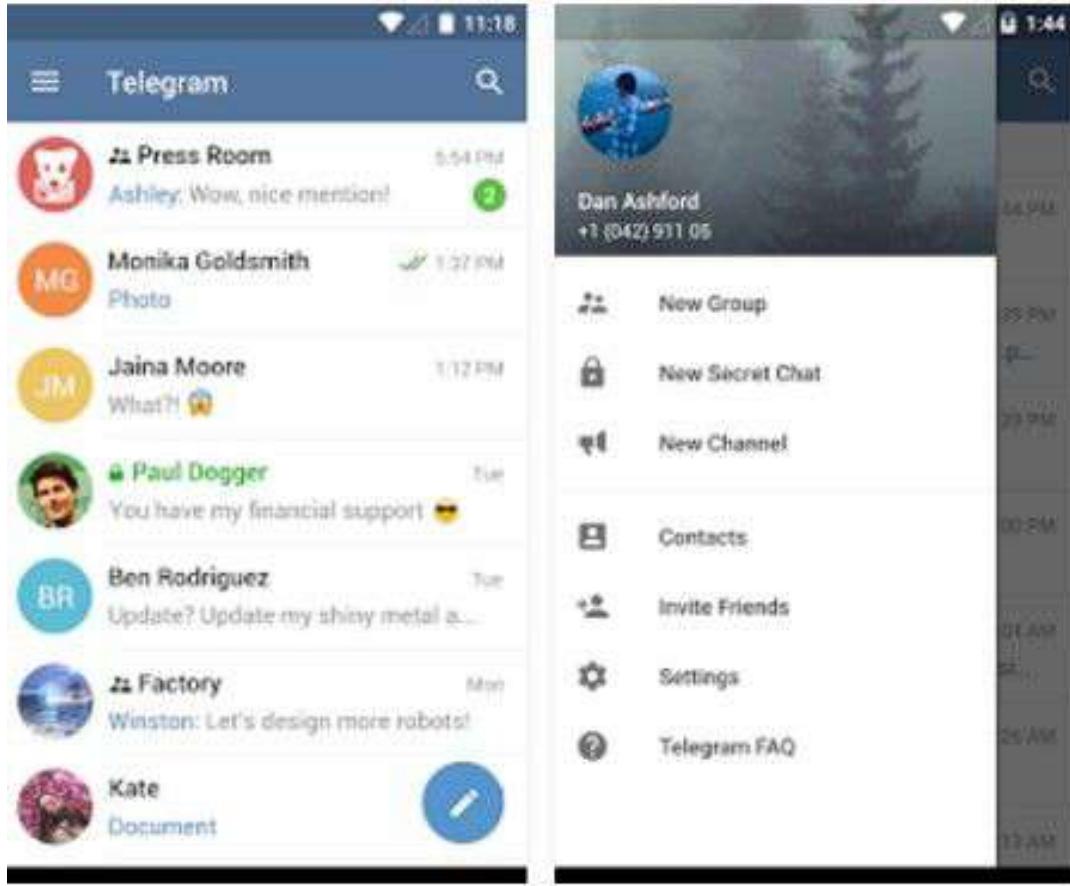




Terima Kasih

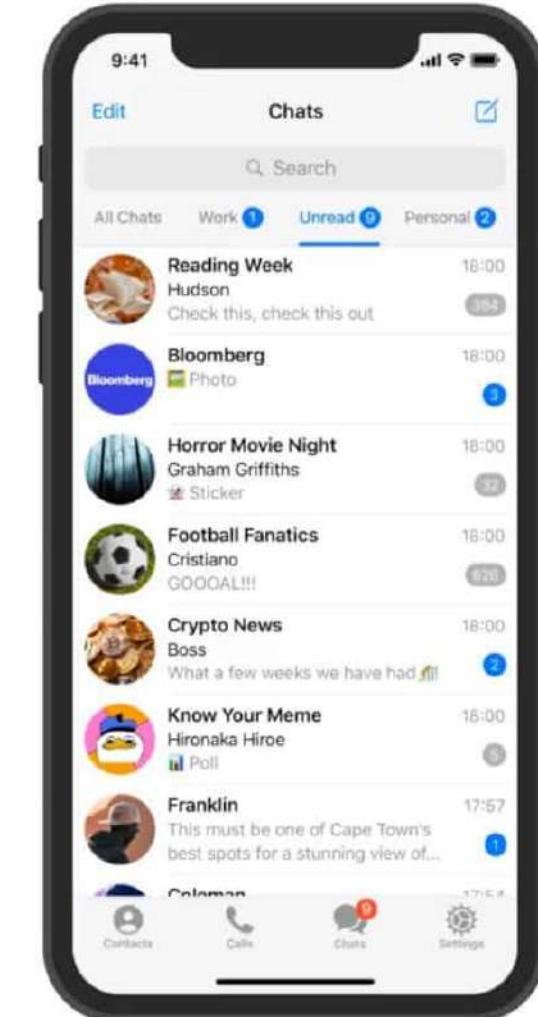
Task

1. Buatlah tampilan berikut ini menggunakan Flutter Material!



Task

2. Buatlah tampilan berikut ini menggunakan Flutter Cupertino!



Flutter Layout

Outline

- Apa itu Layout?
- Single-child Layout
- Multi-child Layout

Apa itu Layout?

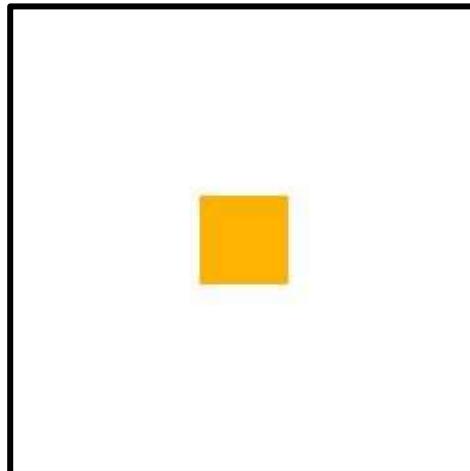
- Untuk mengatur tata letak
- Layout berbentuk widget yang mengatur widget di dalamnya



Single-child Layout

Container

- Membuat sebuah box
- Membungkus widget lain
- Box memiliki margin, padding, dan border



```
Container(  
    child: Text(teks),  
);
```

Penggunaan Container



```
Container(  
    margin: const EdgeInsets.all(10),  
    padding: const EdgeInsets.all(10),  
    decoration: BoxDecoration(  
        border: Border.all(),  
    ),  
    child: const Text('HALO'),  
),  
);
```



HALO

Center

- Membuat sebuah box
- Membungkus widget lain
- Memenuhi lebar dan tinggi ruang di luarnya
- Meletakkan widget berada di bagian tengan



```
Center(  
    child: Text(teks),  
);
```

SizedBox

- Membuat sebuah box
- Membungkus widget lain
- Box dapat diatur lebar dan tingginya
- Lebih sederhana dari container



```
SizedBox(  
    width: 100,  
    height: 150,  
    child: Text(teks),  
);
```

Multi-child Layout

Column

- Mengatur widgets secara vertikal



```
Column(  
    children: const [  
        Text(teks),  
        Text(teks),  
        Text(teks),  
        Text(teks),  
    ],  
);
```

Penggunaan Column

```
Column(  
  children: [  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('H'),  
    ),  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('A'),  
    ),  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('L'),  
    ),  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('O'),  
    ),  
  ],  
);
```



H



A



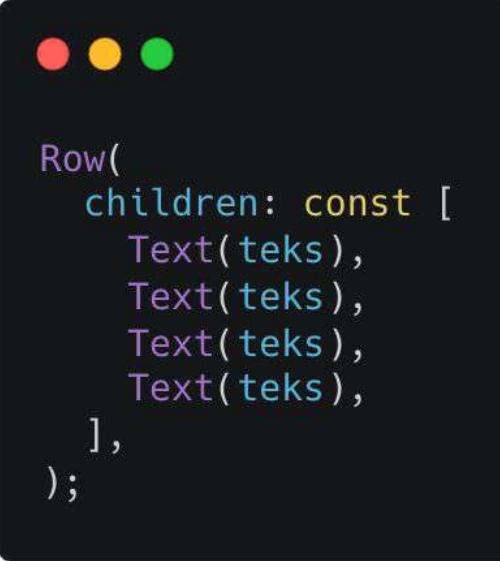
L



O

Row

- Mengatur widgets secara horizontal



```
Row(  
    children: const [  
        Text(teks),  
        Text(teks),  
        Text(teks),  
        Text(teks),  
    ],  
);
```

Penggunaan Row

```
Row(  
  children: [  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('H'),  
    ),  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('A'),  
    ),  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('L'),  
    ),  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('O'),  
    ),  
  ],  
);
```



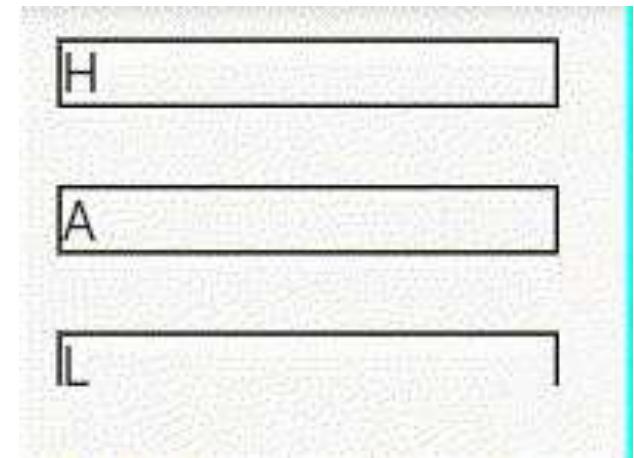
H A L O

ListView

- Mengatur widgets dalam bentuk list
- Memiliki kemampuan scroll



```
ListView(  
  children: [  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('H'),  
    ),  
  );
```

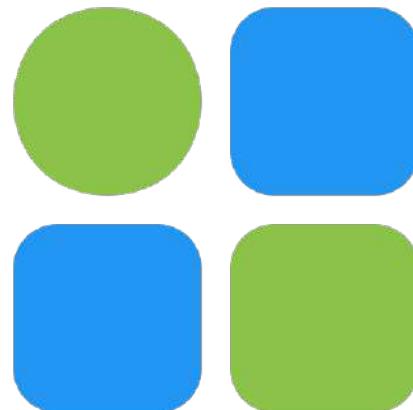


GridView

Mengatur widgets dalam bentuk galeri

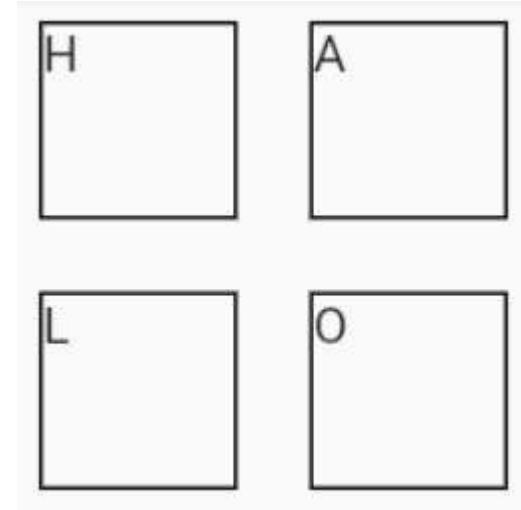


```
GridView.count(  
    crossAxisCount: angka,  
    children: [],  
);
```



Penggunaan GridView

```
GridView.count(  
  crossAxisCount: 2,  
  children: [  
    Container(  
      margin: const EdgeInsets.all(10),  
      decoration: BoxDecoration(border: Border.all()),  
      child: const Text('H'),  
    ),  
    l,  
  ],  
);
```





Terima Kasih

Task

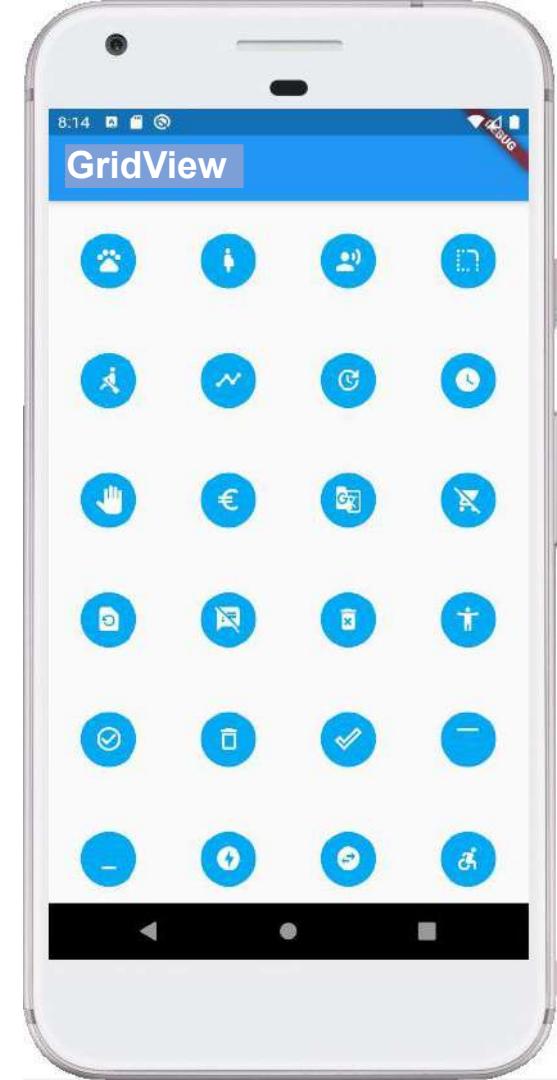
1. Buatlah tampilan berikut ini menggunakan ListView!

The screenshot shows a mobile application interface titled "JSON ListView in Flutter". The top status bar indicates the time as 7:08 and shows a "DEBUG" indicator. The main content area displays a list of eight contacts, each consisting of a green circular icon containing a white letter (L, E, C, P, C, M, K) and the contact's name and phone number. The names and numbers are displayed in a light blue font.

Initial	Name	Phone Number
L	Leanne Graham	1-770-736-8031 x56442
E	Ervin Howell	010-692-6593 x09125
C	Clementine Bauch	1-463-123-4447
P	Patricia Lebsack	493-170-9623 x156
C	Chelsey Dietrich	(254)954-1289
M	Mrs. Dennis Schulist	1-477-935-8478 x6430
K	Kurtis Weissnat	210.067.6132

Task

2. Buatlah tampilan berikut ini menggunakan GridView!





Dialog & Bottom Sheet

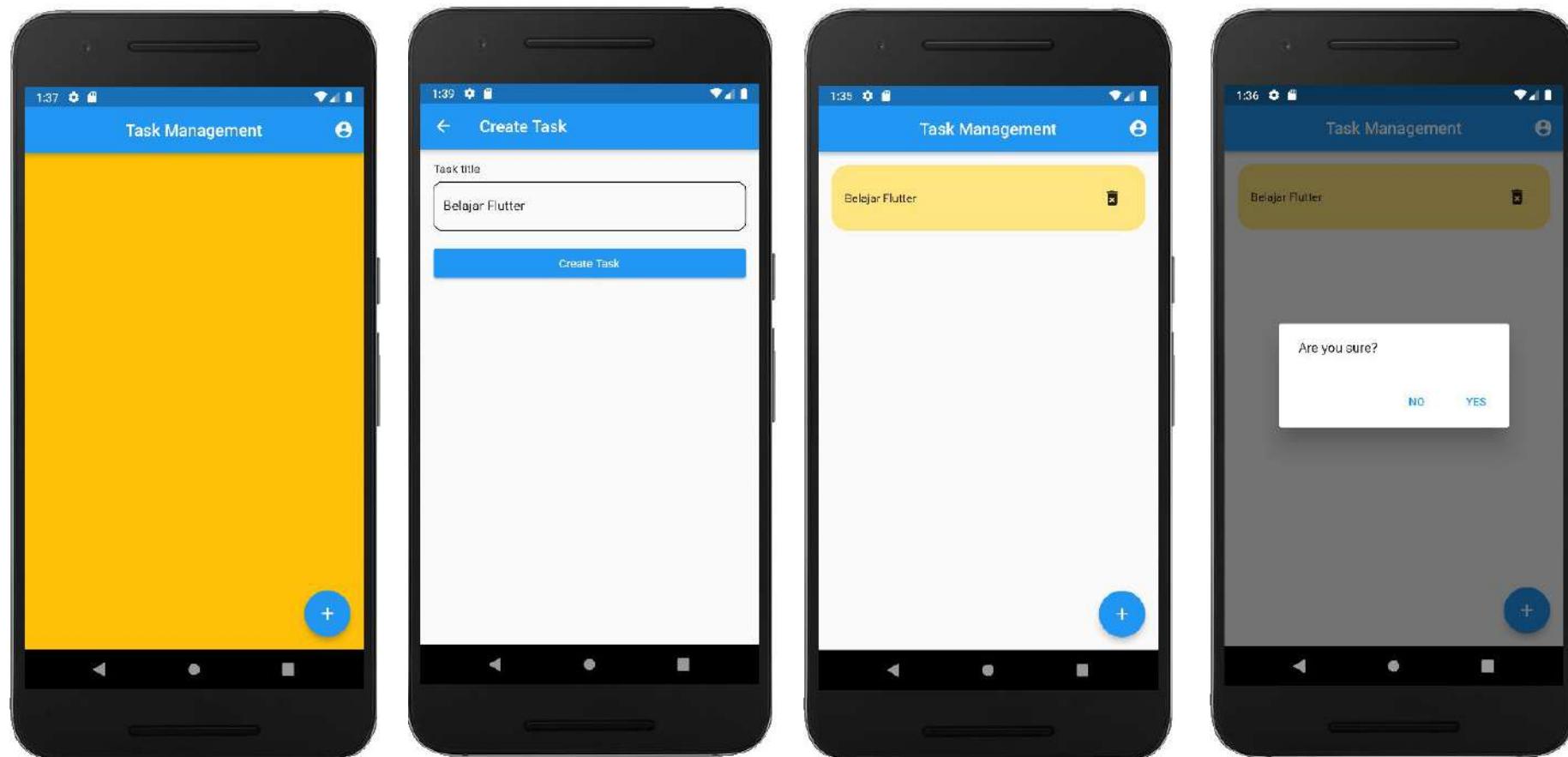
Outline

- Aplikasi Task Management
- AlertDialog
- Bottom Sheet

Aplikasi Task Management

Deskripsi

- Menambahkan dan menghapus kegiatan



Aplikasi Task Management

- TODO 1: Membuat model untuk informasi yang mau disimpan, buat folder models dan di dalam nya buat file dart baru task_model.dart

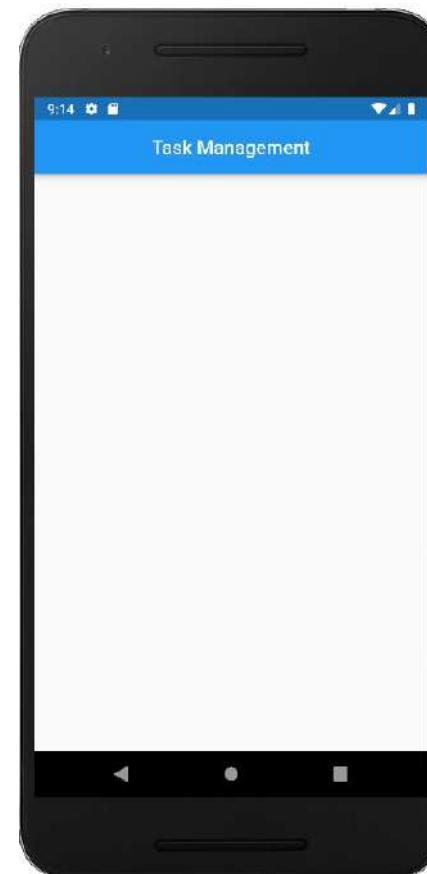
```
class TaskModel {  
    final String id;  
    final String taskName;  
  
    TaskModel({  
        required this.id,  
        required this.taskName,  
    });  
}
```

Aplikasi Task Management

- TODO 2: Membuat Task Screen, buat folder screens di dalam nya
buat file dart baru task_screen.dart

```
class TaskScreen extends StatelessWidget {
    const TaskScreen({Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: const Text('Task Management'),
                centerTitle: true,
            ),
        );
    }
}
```



Aplikasi Task Management

- TODO 3: Membuat Empty Screen, buat file dart baru empty_task_screen.dart di folder screens

```
class EmptyTaskScreen extends StatelessWidget {  
    const EmptyTaskScreen({Key? key}) : super(key: key);  
  
    @override  
    Widget build(BuildContext context) {  
        return Container(color: Colors.amber);  
    }  
}
```

Aplikasi Task Management

- TODO 4: Tambahkan package provider di pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  provider: ^6.0.2
```

Aplikasi Task Management

- TODO 5: Membuat Task Manager, buat file dart baru `task_manager.dart` di dalam folder `models`. Tambahkan method `deleteTask` dan `addTask`

```
class TaskManager extends ChangeNotifier {
    final _taskModels = <TaskModel>[];
    List<TaskModel> get taskModels => List.unmodifiable(_taskModels);

    void deleteTask(int index) {
        _taskModels.removeAt(index);
        notifyListeners();
    }

    void addTask(TaskModel task) {
        _taskModels.add(task);
        notifyListeners();
    }
}
```

Aplikasi Task Management

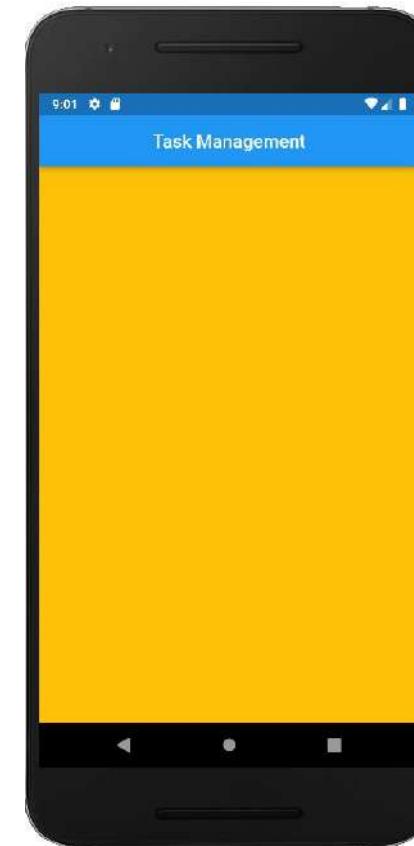
- TODO 6: (main.dart) menambahkan TaskManager sebagai provider

```
Widget build(BuildContext context) {
  return MultiProvider(
    providers: [
      ChangeNotifierProvider(
        create: (context) => TaskManager(),
      ), // ChangeNotifierProvider
    ],
    child: const MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Task Management',
      home: TaskScreen(),
    ), // MaterialApp
  ); // MultiProvider
}
```

Aplikasi Task Management

- TODO 7: (task_screen.dart) membuat fungsi buildTaskScreen, akan me-return antara Task Screen yang ada data nya atau yang kosong dan panggil fungsi nya di body

```
body: buildTaskScreen(),  
  
Widget buildTaskScreen() {  
  return Consumer<TaskManager>(  
    builder: (context, manager, child) {  
      if (manager.taskModels.isNotEmpty) {  
        return Container();  
      } else {  
        return const EmptyTaskScreen();  
      }  
    },  
  ); // Consumer  
}
```



Aplikasi Task Management

- TODO 8: Membuat Task Item Screen, buat file dart baru task_item_screen.dart di dalam folder screens dan tambahkan properti onCreate

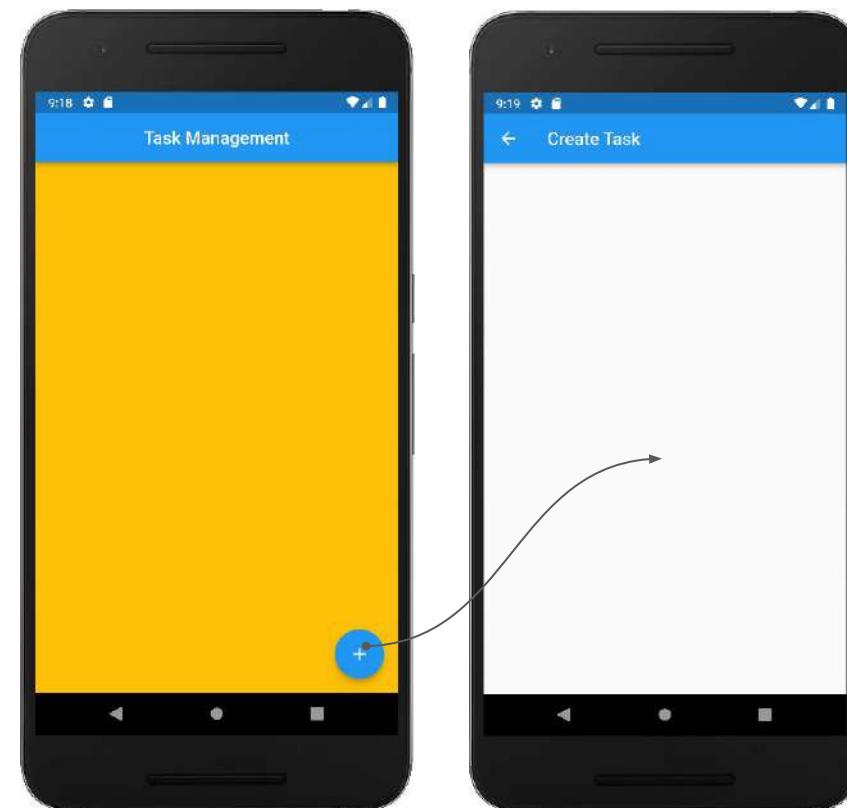
```
class _TaskItemScreenState extends State<TaskItemScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Create Task'),
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.all(16),
        child: ListView(
          children: [
            // buildTaskNameField
            // buildButton
            ],
        ), // ListView
      ), // Padding
    ); // Scaffold
  }
}
```

```
final Function(TaskModel) onCreate;
const TaskItemScreen({
  Key? key,
  required this.onCreate,
}) : super(key: key);
```

Aplikasi Task Management

- TODO 9: (task_screen.dart) buat FloatingActionButton, untuk navigasi ke TaskItemScreen

```
floatingActionButton: FloatingActionButton(  
    child: const Icon(Icons.add),  
    onPressed: () {  
        final manager = Provider.of<TaskManager>(context, listen: false);  
        Navigator.push(  
            context,  
            MaterialPageRoute(  
                builder: (context) => TaskItemScreen(  
                    onCreate: (task) {  
                        manager.addTask(task);  
                        Navigator.pop(context);  
                    },  
                ), // TaskItemScreen  
            ), // MaterialPageRoute  
        );  
    },  
, // FloatingActionButton
```



Aplikasi Task Management

- TODO 10: (task_item_screen.dart) menambahkan state properti, initState, dan dispose

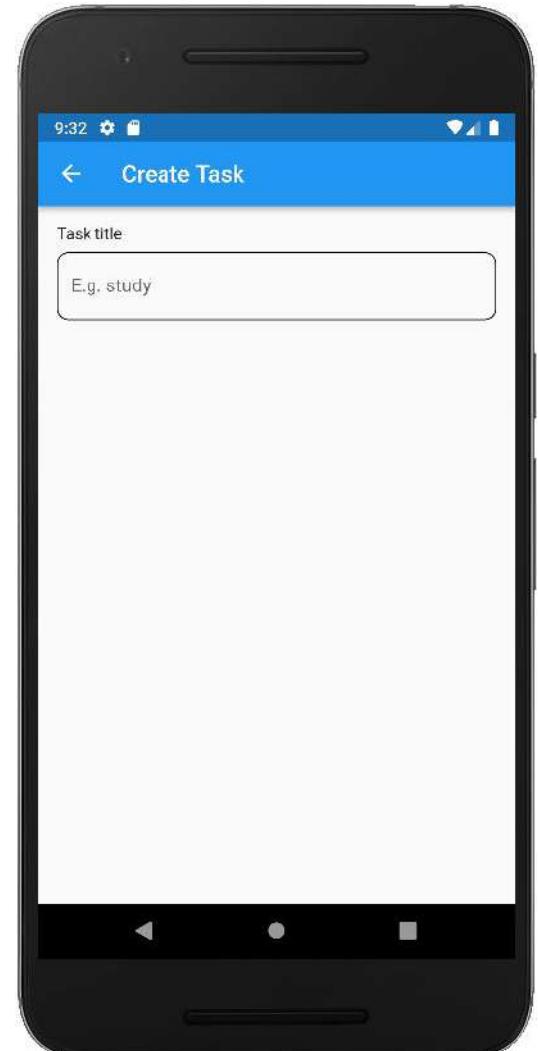
```
final _taskNameController = TextEditingController();
String _taskName = '';
@Override
void initState() {
    super.initState();
    _taskNameController.addListener(() {
        setState(() {
            _taskName = _taskNameController.text;
        });
    });
}

@Override
void dispose() {
    _taskNameController.dispose();
    super.dispose();
}
```

Aplikasi Task Management

- TODO 11:
(task_item_screen.dart)
membuat buildNameField
dan panggil fungsi nya di
dalam ListView

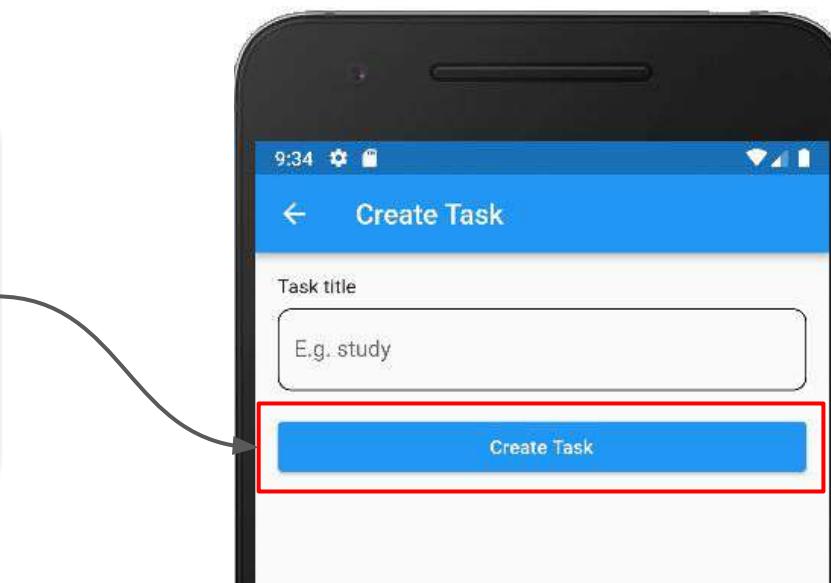
```
Widget buildNameField() {  
    return Column(  
        crossAxisAlignment: CrossAxisAlignment.start,  
        children: [  
            const Text('Task title'),  
            const SizedBox(  
                height: 8,  
            ), // SizedBox  
            TextField(  
                controller: _taskNameController,  
                cursorColor: Colors.black,  
                decoration: const InputDecoration(  
                    hintText: 'E.g. study',  
                    enabledBorder: OutlineInputBorder(  
                        borderSide: BorderSide(color: Colors.black),  
                        borderRadius: BorderRadius.all(  
                            Radius.circular(10),  
                        ), // BorderRadius.all  
                    ), // OutlineInputBorder  
                    focusedBorder: OutlineInputBorder(  
                        borderSide: BorderSide(color: Colors.black),  
                        borderRadius: BorderRadius.all(  
                            Radius.circular(10),  
                        ), // BorderRadius.all  
                    ), // OutlineInputBorder  
                ), // InputDecoration  
            ) // TextField  
        ],  
    ); // Column  
}
```



Aplikasi Task Management

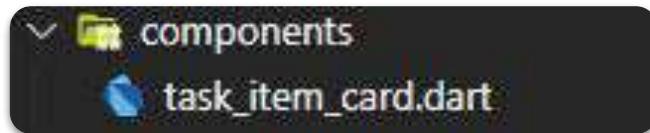
- TODO 12: (task_item_screen.dart) membuat buildButton dan panggil fungsi nya di dalam ListView

```
Widget buildButton() {
  return ElevatedButton(
    child: const Text('Create Task'),
    onPressed: () {},
  ); // ElevatedButton
}
```



Aplikasi Task Management

- TODO 13: Membuat Task Item Card, buat folder baru components lalu didalam nya buat file dart baru task_item_card.dart



```
class TaskItemCard extends StatelessWidget {
    final TaskModel task;
    const TaskItemCard({
        Key? key,
        required this.task,
    }) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return Container(
            padding: const EdgeInsets.all(16),
            decoration: BoxDecoration(
                color: Colors.amberAccent[100],
                borderRadius: BorderRadius.circular(
                    20,
                ), // BorderRadius.circular
            ), // BoxDecoration
            child: Row(
                children: [
                    Text(task.taskName),
                    IconButton(
                        onPressed: () {},
                        icon: const Icon(Icons.delete_forever_rounded),
                    ), // IconButton
                ],
            ), // Row
        ); // Container
    }
}
```

Aplikasi Task Management

- TODO 14: (task_item_screen.dart) menambahkan call back handler di ElevatedButton pada buildButton, dan menambahkan package uuid

```
Widget buildButton() {
  return ElevatedButton(
    child: const Text('Create Task'),
    onPressed: () {
      final taskItem = TaskModel(
        id: const Uuid().v1(),
        taskName: _taskNameController.text,
      ); // TaskModel
      widget.onCreate(taskItem);
    },
  );
}
```

```
dependencies:
  flutter:
    sdk: flutter
  provider: ^6.0.2
  uuid: ^3.0.5
```

Aplikasi Task Management

- TODO 15: Membuat Task List Screen, dalam folder screens buat file dart baru task_list_screen.dart. Dan tambahkan di buildTaskScreen (task_screen.dart)

```
class TaskListScreen extends StatelessWidget {
  final TaskManager manager;
  const TaskListScreen({
    Key? key,
    required this.manager,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    final taskItems = manager.taskModels;
    return Padding(
      padding: const EdgeInsets.all(16),
      child: ListView.separated(
        itemCount: taskItems.length,
        itemBuilder: (context, index) {
          final item = taskItems[index];
          return TaskItemCard();
        },
        separatorBuilder: (context, index) {
          return const SizedBox(
            height: 16,
          ); // SizedBox
        },
      ), // ListView.separated
    );
}
```

```
Widget buildTaskScreen() {
  return Consumer<TaskManager>(
    builder: (context, manager, child) {
      if (manager.taskModels.isNotEmpty) {
        return TaskListScreen(
          manager: manager,
        ); // TaskListScreen
      } else {
        return const EmptyTaskScreen();
      }
    },
  ); // Consumer
}
```

Aplikasi Task Management

- TODO 16: (task_list_screen.dart) menambahkan properti yang dibutuhkan pada TaskItemCard untuk melihat task yang sudah dibuat.

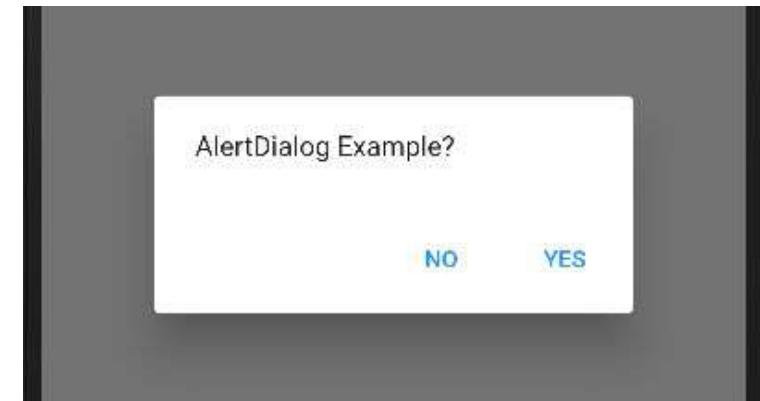
```
return TaskItemCard(  
    key: Key(item.id),  
    task: item,  
); // TaskItemCard
```



AlertDialog

Deskripsi

- Untuk tampilan android, dari material design
- Menginformasikan pengguna tentang situasi tertentu
- Bisa digunakan untuk mendapatkan input dari user
- Membutuhkan helper method showDialog



Cara Membuat AlertDialog

- TODO 17: (task_item_card.dart) menambahkan Alert Dialog
- Memanggil fungsi `showDialog` di dalam fungsi `onPressed` pada `IconButton`
- `showDialog` membutuhkan `context` dan `builder`
- Di `builder` nya akan me-return `AlertDialog`

```
 onPressed: () {
    showDialog(
        context: context,
        builder: (context) => AlertDialog(
            content: const Text('Are you sure?'),
            actions: [
                TextButton(
                    onPressed: () {
                        Navigator.pop(context);
                    },
                    child: const Text('NO'),
                ), // TextButton
                TextButton(
                    onPressed: onPressed,
                    child: const Text('YES'),
                ), // TextButton
            ],
        ), // AlertDialog
    );
},
```

Cara Membuat AlertDialog

- AlertDialog menyediakan properti seperti content dan actions
- Content bisa dimasukan widget text, gambar dan animasi gambar
- Actions bisa ditambahkan button untuk menerima respon dari user
- TODO 18: (task_item_card.dart) membuat properti onPressed

```
onPressed: () {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      content: const Text('Are you sure?'),
      actions: [
        TextButton(
          onPressed: () {
            Navigator.pop(context);
          },
          child: const Text('NO'),
        ), // TextButton
        TextButton(
          onPressed: onPressed,
          child: const Text('YES'),
        ), // TextButton
      ],
    ), // AlertDialog
);
```

```
class TaskItemCard extends StatelessWidget {
  final TaskModel task;
  final Function() onPressed;
  const TaskItemCard({
    Key? key,
    required this.task,
    required this.onPressed,
  }) : super(key: key);
```

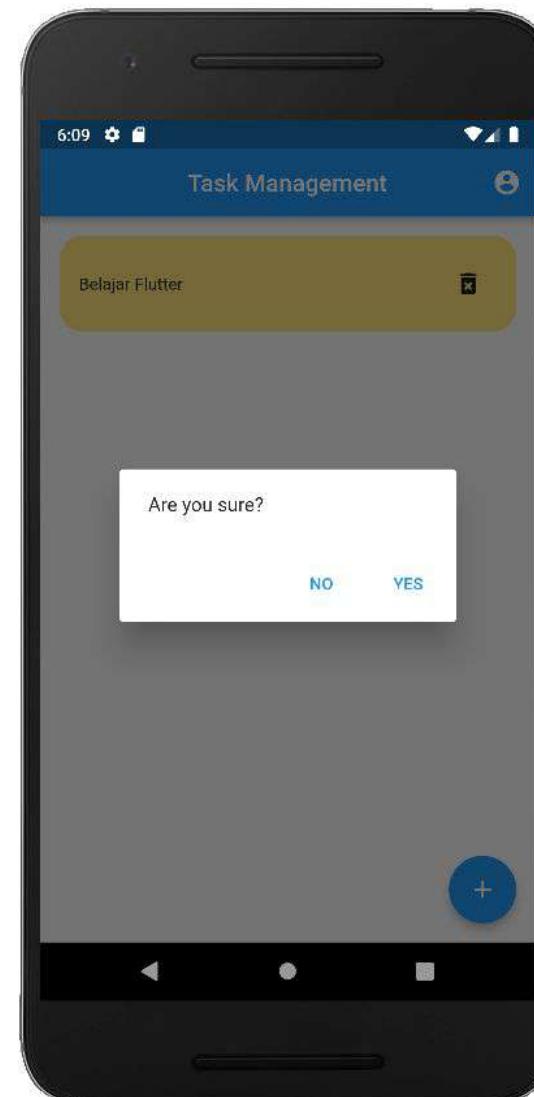
Cara Membuat AlertDialog

- TODO 19: (task_list_screen.dart) menambahkan fungsi onPressed pada TaskItemCard, pada body fungsi nya tambahkan method deleteTask, Navigator.pop(context), dan ScaffoldMessenger

```
return TaskItemCard(
  key: Key(item.id),
  task: item,
  onPressed: () {
    manager.deleteTask(index);
    Navigator.pop(context);
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('${item.taskName} Deleted')),
    );
  },
); // TaskItemCard
```

Hasil Akhir

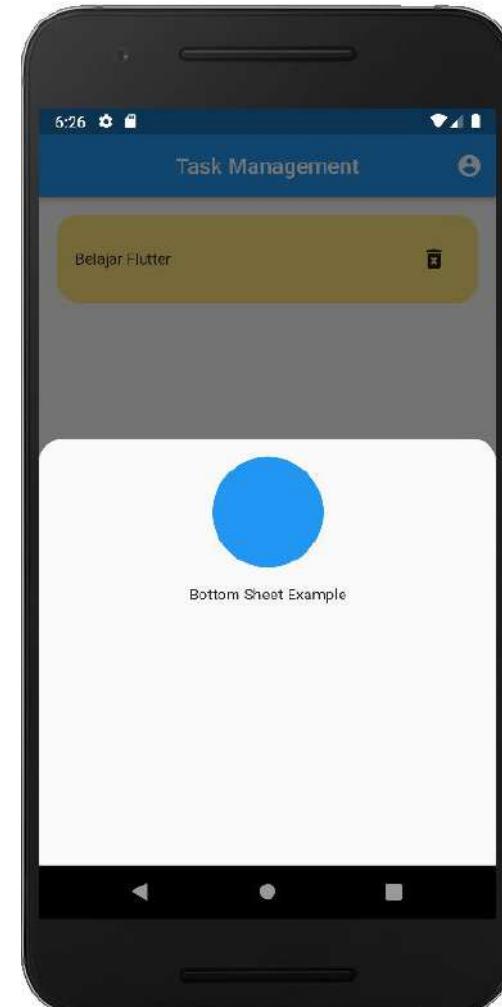
- Ketika icon delete nya ditekan maka akan menampilkan alert dialog seperti gambar disamping ini



Bottom Sheet

Deskripsi

- Seperti dialog tetapi muncul dari bawah layar aplikasi
- Menggunakan fungsi bawaan flutter `showModalBottomSheet`
- Membutuhkan dua properti, yaitu context dan builder



Cara Membuat Bottom Sheet

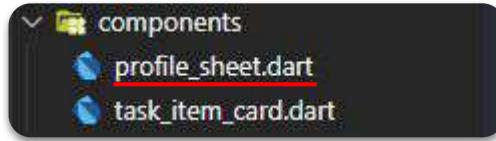
- TODO 20: (task_screen.dart) membuat Bottom Sheet

```
actions: [
    IconButton(
        onPressed: () {
            showModalBottomSheet(
                shape: const RoundedRectangleBorder(
                    borderRadius: BorderRadius.vertical(
                        top: Radius.circular(20),
                    ), // BorderRadius.vertical
                ), // RoundedRectangleBorder
                context: context,
                builder: (context) => Container(),
            );
        },
        icon: const Icon(Icons.account_circle),
    ), // IconButton
],
```



Cara Membuat Bottom Sheet

- TODO 21: buat file dart baru profile_sheet.dart di dalam folder components dan buat konten-konten nya



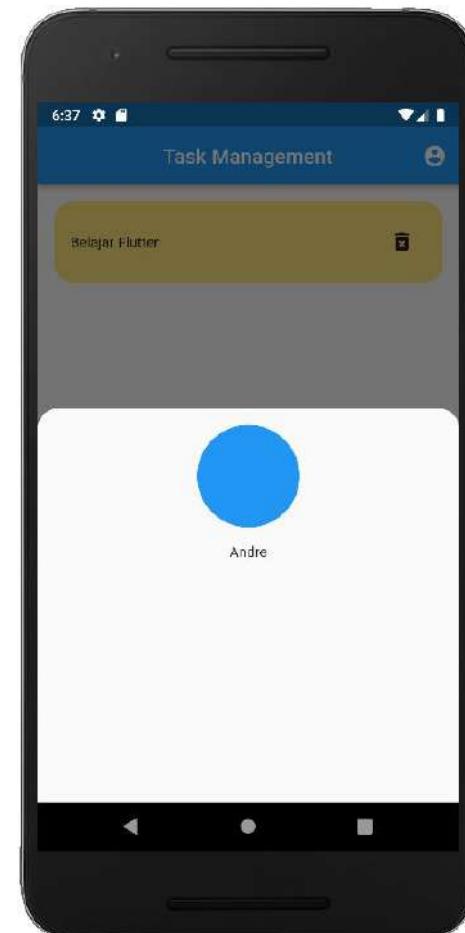
```
class ProfileSheet extends StatelessWidget {
  const ProfileSheet({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(16),
      child: Column(
        children: [
          Container(
            height: 100,
            width: 100,
            decoration: const BoxDecoration(
              shape: BoxShape.circle,
              color: Colors.blue,
            ), // BoxDecoration
          ), // Container
          const SizedBox(
            height: 16,
          ), // SizedBox
          const Text('Andre'),
        ],
      ), // Column
    );
  }
}
```

Cara Membuat Bottom Sheet

- TODO 22: (task_screen.dart)
mengganti Container pada
`builder showModalBottomSheet`
menjadi `ProfileSheet()`

```
builder: (context) => const ProfileSheet(),
```



Full Source Code:
<https://bit.ly/3HtkP09>



Terima Kasih

Task

1. Buatlah gallery yang menampilkan gambar-gambar!
2. Buatlah agar tiap gambar dapat ditampilkan pada bottom sheet saat ditekan!
3. Buatlah agar tiap gambar dapat ditampilkan pada dialog saat ditekan!

Assets

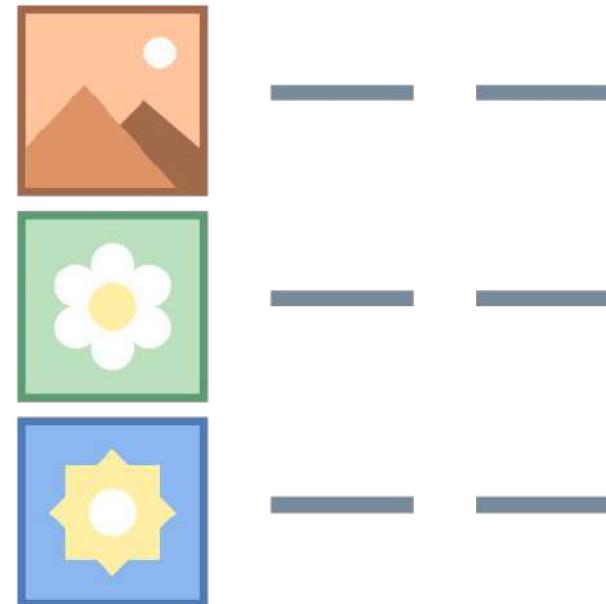
Outline

- Assets
- Image
- Font
- Aplikasi Task Management
(continue)

Assets

Deskripsi

- File yang di bundled dan di deployed bersamaan dengan aplikasi
- Tipe-tipe assets, seperti: static data (JSON files), icons, images, dan font file (ttf)



Menentukan Assets

- Flutter menggunakan pubspec.yaml
- Pubspec.yaml terletak pada root project, untuk mengidentifikasi assets yang dibutuhkan aplikasi
- Gunakan karakter “ / ” untuk memasukan semua assets dibawah satu directory name

```
assets:  
  - assets/my_icon.png  
  - assets/background.jpg
```

```
assets:  
  - assets/
```

Image

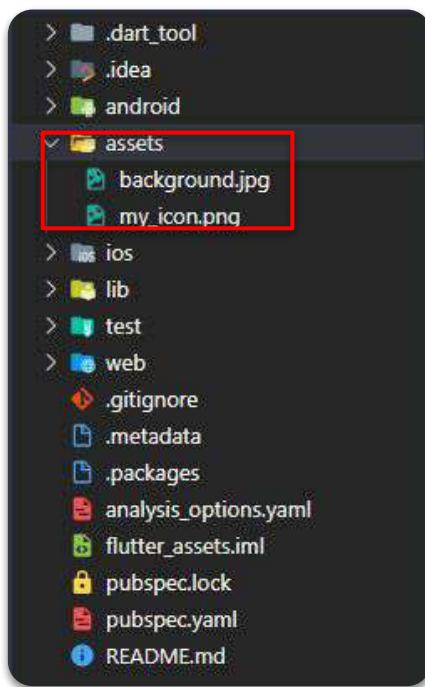
Deskripsi

- Image atau gambar akan membuat tampilan aplikasi menjadi lebih menarik
- Flutter mendukung format gambar seperti JPEG, WebP, GIF, Animated Web/GIF, PNG, BMP, dan WBMP.
- Menampilkan gambar dari project asset dan internet



Loading Images

- Gunakan widget Image
- Membutuhkan properti *image* dengan nilai class AssetImage()



```
body: Column(  
    children: const [  
        Image(  
            image: AssetImage('assets/background.jpg'),  
        ), // Image  
        Image(  
            image: AssetImage('assets/my_icon.png'),  
        ), // Image
```



Loading Images

- Menggunakan method `Image.asset`, mendapatkan image yang sudah ditambahkan dalam project
- Menggunakan method `Image.network`, mendapatkan data image melalui internet dengan menggunakan string url nya

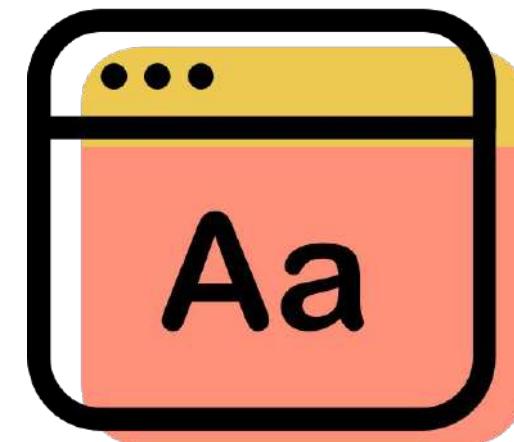
```
body: Column(  
  children: [  
    Image.asset('assets/background.jpg'),  
    Image.network('https://picsum.photos/id/1/200/300'),  
  ],
```



Font

Deskripsi

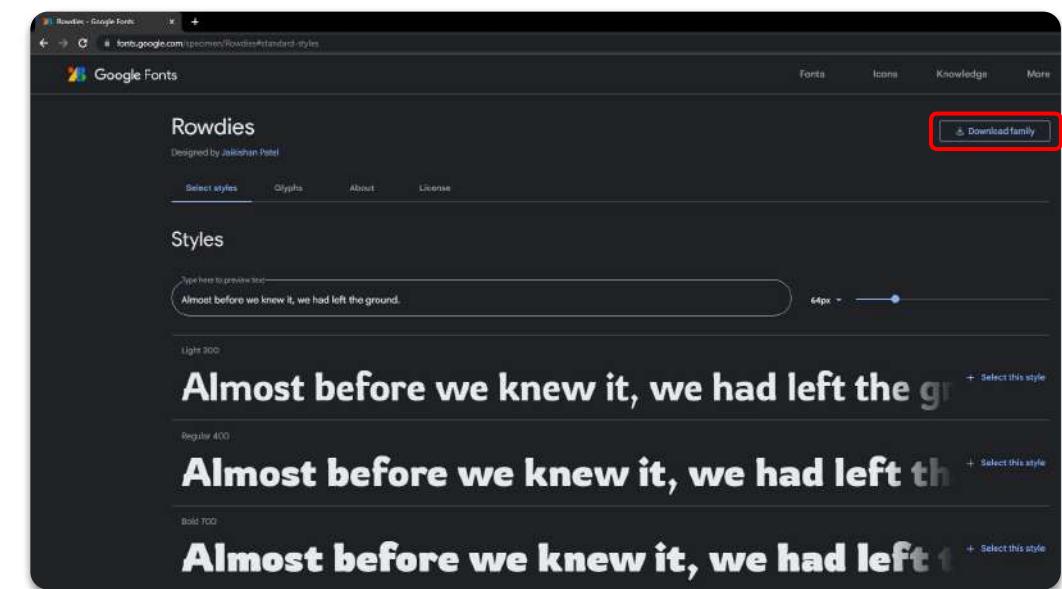
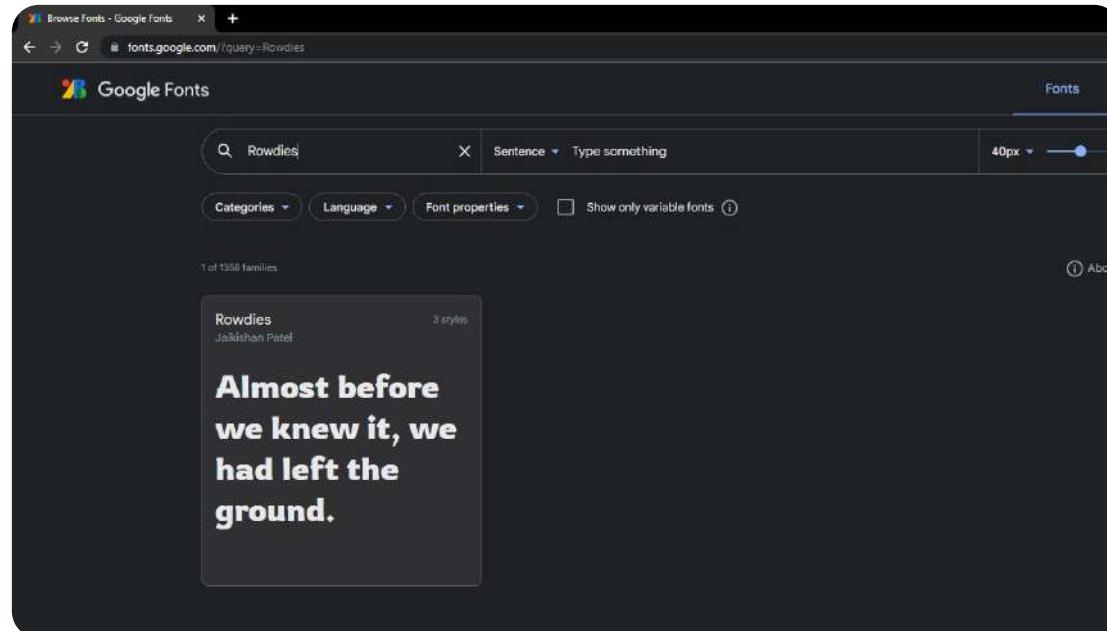
- Penggunaan font dengan style tertentu akan menjadi keunikan pada aplikasi
- Penentuan font yang mau dipakai biasanya oleh UI designer
- Penerapan font menggunakan custom font atau dari package



Custom Font

Cara menggunakan custom font:

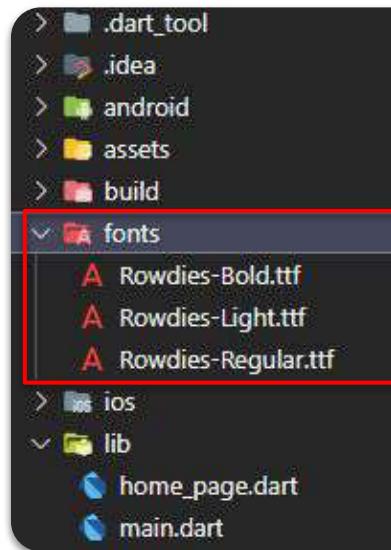
- Step 1: Cari dan download font (Ex. <https://fonts.google.com/>)



Custom Font

Cara menggunakan custom font:

- Step 2: Import file .ttf
- Step 3: Daftarkan font di pubspec.yaml



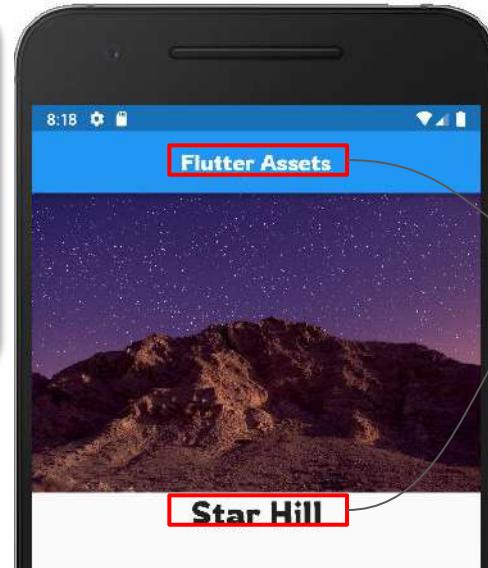
```
fonts:  
  - family: Rowdies  
    fonts:  
      - asset: fonts/Rowdies-Bold.ttf  
      - asset: fonts/Rowdies-Light.ttf  
      - asset: fonts/Rowdies-Regular.ttf
```

Custom Font

Cara menggunakan custom fonts:

- Step 4: Mengatur font sebagai default

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    debugShowCheckedModeBanner: false,  
    title: 'Flutter Assets',  
    theme: ThemeData(fontFamily: 'Rowdies'),  
    home: const HomePage(),  
  ); // MaterialApp  
}
```



Semua text memiliki
style font family
Rowdies

Custom Font

Cara menggunakan custom fonts:

- Step 5: Gunakan font di spesifik widget

```
Text(  
    'Star Hill',  
    style: TextStyle(  
        fontSize: 30,  
        fontFamily: 'Rowdies',  
    ), // TextStyle  
, // Text
```



Hanya text yang
diberikan style
font family

Font dari Package

Cara menggunakan font dari package:

- Step 1: Tambahkan package google_fonts di dependencies

```
dependencies:  
  flutter:  
    sdk: flutter  
  google_fonts: ^2.2.0
```

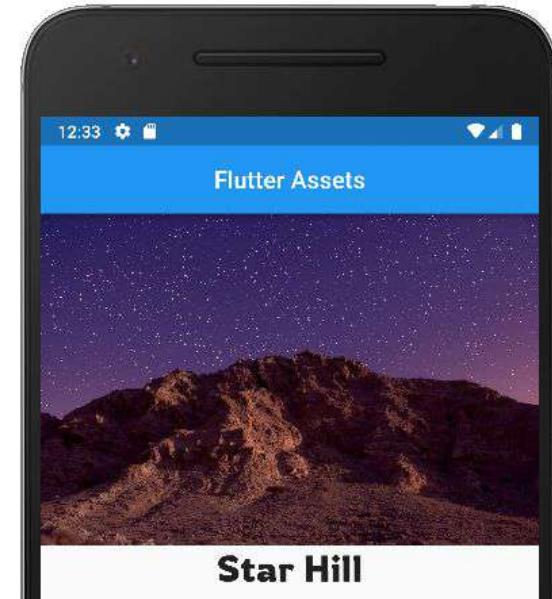
Font dari Package

Cara menggunakan font dari package:

- Step 2: Import package di file dart
- Step 3: Gunakan font dengan memanggil GoogleFonts.namaFont()

```
import 'package:google_fonts/google_fonts.dart';
```

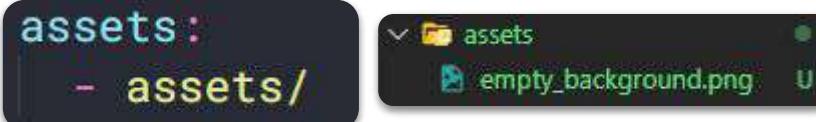
```
Text(  
  'Star Hill',  
  style: GoogleFonts.rowdies(fontSize: 30),  
, // Text
```



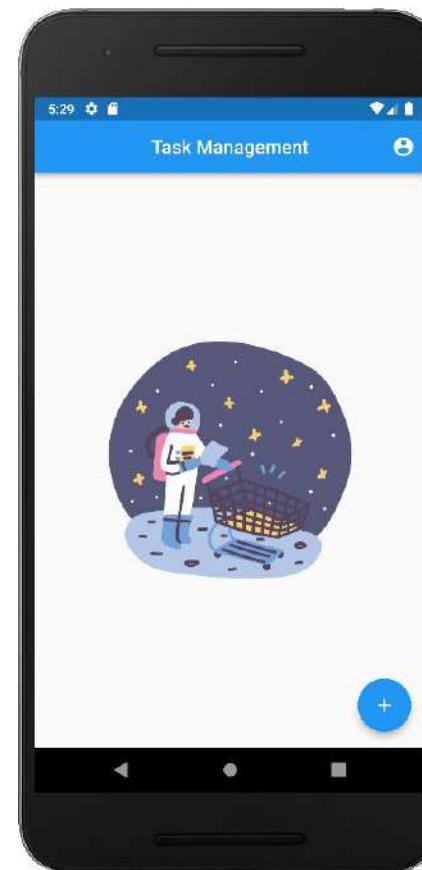
Aplikasi Task Management (Continue)

Menambahkan Image

TODO 23: Menambahkan image pada empty_task_screen.dart

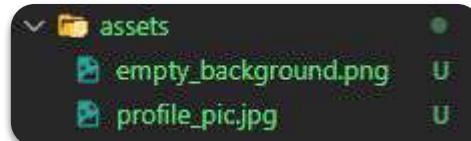


```
assets:  
  - assets/  
  
return Container(  
  alignment: Alignment.center,  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: const [  
      Image(  
        image: AssetImage(  
          'assets/empty_background.png',  
        ), // AssetImage  
        height: 250,  
      ), // Image  
    ],  
  ), // Column  
) // Container
```

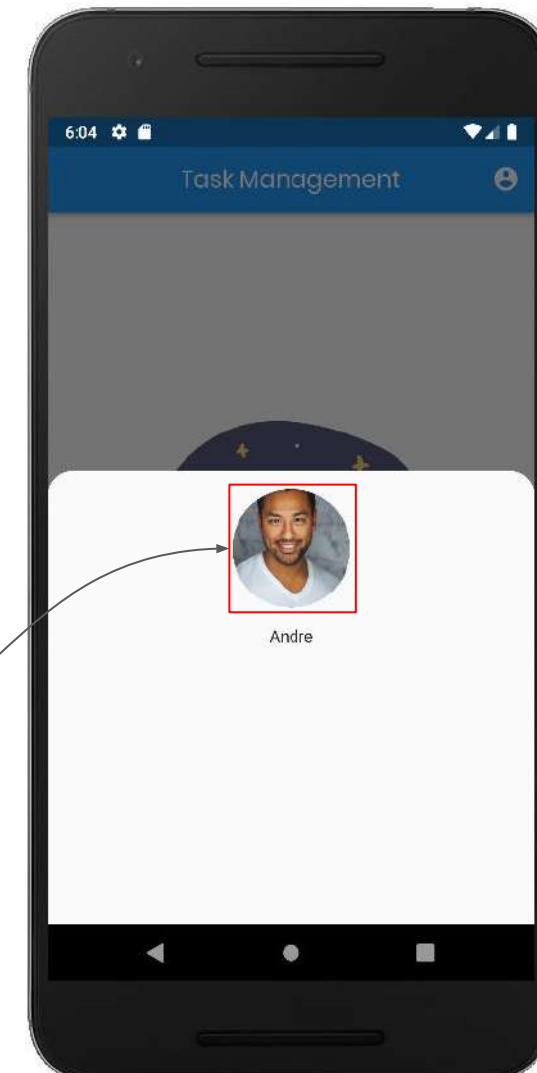


Menambahkan Image

TODO 24: Membuat profile picture di `profile_sheet.dart`, menggunakan widget `CircleAvatar`



```
CircleAvatar(  
    radius: 50,  
    backgroundImage: AssetImage('assets/profile_pic.jpg'),  
, // CircleAvatar
```



Menambahkan Font

TODO 25: Menambahkan font pada setiap text

```
dependencies:  
  flutter:  
    sdk: flutter  
  provider: ^6.0.2  
  uuid: ^3.0.5  
  google_fonts: ^2.2.0
```

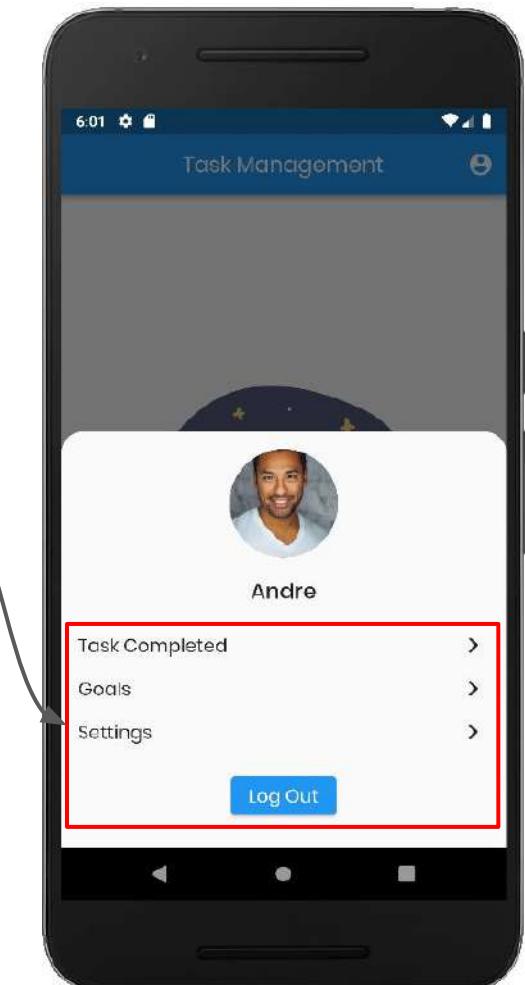
```
Text(  
  'Andre',  
  style: GoogleFonts.poppins(  
    fontSize: 20,  
    fontWeight: FontWeight.w500,  
  ),  
, // Text
```



Menambahkan Konten

TODO 26:
(profile_sheet.dart)
menambahkan konten lain

```
SizedBox(  
  child: Column(  
    children: [  
      Row(  
        mainAxisAlignment: MainAxisAlignment.spaceBetween,  
        children: [  
          Text(  
            'Task Completed',  
            style: GoogleFonts.poppins(fontSize: 17),  
          ), // Text  
          const Icon(  
            Icons chevron_right_rounded,  
            size: 30,  
          ) // Icon  
        ],  
      ), // Row  
      const SizedBox( // SizedBox ...  
      Row( // Row ...  
      const SizedBox( // SizedBox ...  
      Row( // Row ...  
      const SizedBox( // SizedBox ...  
        ElevatedButton(  
          onPressed: () {},  
          child: Text(  
            'Log Out',  
            style: GoogleFonts.poppins(fontSize: 17),  
          ), // Text  
        ), // ElevatedButton  
      ],  
    ), // Column  
  ) // SizedBox
```



Menambahkan Konten

TODO 27: (empty_task_screen.dart) menambahkan text di bawah image empty screen

```
Text(  
  'task masih kosong...',  
  style: GoogleFonts.poppins(  
    fontSize: 30,  
    fontWeight: FontWeight.w500,  
  ),  
) // Text
```



Full Source Code:
<https://bit.ly/3sgGr90>



Terima Kasih

Task

1. Carilah minimum 3 gambar dari internet, lalu tampilkan menggunakan GridView!
2. Tambahkan fungsionalitas berikut:
 - a. ketika sebuah gambar ditekan, akan terbuka halaman baru
 - b. halaman tersebut menampilkan gambar yang ditekan
 - c. gambar ditampilkan memenuhi halaman

Flutter Command Line Interface & Flutter Package Management

Outline

- Flutter CLI
- Important CLI Commands
- Packages Management

Flutter CLI

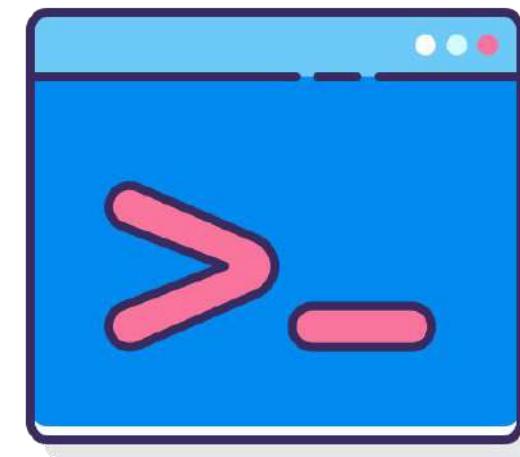
Deskripsi

- Alat yang digunakan untuk berinteraksi dengan Flutter SDK
- Perintah dijalankan dalam terminal



Important CLI Commands

- Flutter Doctor
- Flutter Create
- Flutter Run
- Flutter Emulator
- Flutter Channel
- Flutter Pub
- Flutter Build
- Flutter Clean



Flutter Doctor

Perintah untuk menampilkan informasi software yang dibutuhkan flutter

➡ flutter doctor

```
PS C:\Users\USER> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 2.8.1, on Microsoft Windows [Version 10.0 19043.1469], locale en-ID)
[✓] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
[✓] Chrome - develop for the web
[✓] Android Studio (version 2020.3)
[✓] VS Code (version 1.63.2)
[✓] Connected device (1 available)

• No issues found!
PS C:\Users\USER>
```

Flutter Create

Perintah untuk membuat project aplikasi flutter baru di directory tertentu

▶▶▶ flutter create <APP_NAME>

```
PS P:\> flutter create aplikasi_flutter
Creating project P:\aplikasi_flutter...
  P:\aplikasi_flutter\lib\main.dart (created)
  P:\aplikasi_flutter\pubspec.yaml (created)
  P:\aplikasi_flutter\README.md (created)
  P:\aplikasi_flutter\test\widget_test.dart (created)
  P:\aplikasi_flutter\.gitignore (created)
  P:\aplikasi_flutter\.idea\libraries\Dart_SDK.xml (created)
  P:\aplikasi_flutter\.idea\libraries\KotlinJavaRuntime.xml (created)
```

```
  P:\aplikasi_flutter\web\manifest.json (created)          2,320ms
Running "flutter pub get" in aplikasi_flutter...
Wrote 81 files.

All done!
In order to run your application, type:

$ cd P:\aplikasi_flutter
$ flutter run

Your application code is in P:\aplikasi_flutter\lib\main.dart.

PS P:\>
```

Flutter Run

Perintah untuk menjalankan project aplikasi di device yang tersedia

▶ flutter run <dart_file>

```
PS P:\aplikasi_flutter> flutter run lib/main.dart
Launching lib/main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...          20.7s
This app is linked to the debug service: ws://127.0.0.1:58927/NpHd8gC-2XQ=/ws
Debug service listening on ws://127.0.0.1:58927/NpHd8gC-2XQ=/ws

Running with sound null safety

To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".

An Observatory debugger and profiler on Chrome is available at: http://127.0.0.1:58927/NpHd8gC-2XQ=
The Flutter DevTools debugger and profiler on Chrome is available at:
http://127.0.0.1:9100?uri=http://127.0.0.1:58927/NpHd8gC-2XQ=
```

Flutter Emulator

Perintah untuk menampilkan daftar emulator yang terinstall dan menampilkan option untuk membuka emulator atau membuat emulator baru

- ||||> flutter emulators
- ||||> flutter emulators --launch <EMULATOR_ID>
- ||||> flutter emulators --create [--name xyz]

```
PS P:\aplikasi_flutter> flutter emulators
1 available emulator:

Pixel_4_XL_API_30 • Pixel 4 XL API 30 • Google • android

To run an emulator, run 'flutter emulators --launch <emulator id>'.
To create a new emulator, run 'flutter emulators --create [--name xyz]'.

You can find more information on managing emulators at the links below:
  https://developer.android.com/studio/run/managing-avds
  https://developer.android.com/studio/command-line/avdmanager
PS P:\aplikasi_flutter>
```

Flutter Channel

Perintah untuk menampilkan daftar flutter channel yang tersedia dan menunjukan channel yang digunakan saat ini.

||||> flutter channel

```
PS P:\aplikasi_flutter> flutter channel
Flutter channels:
  master
  dev
  beta
* stable
PS P:\aplikasi_flutter>
```

Flutter Pub

Ada dua syntax yang bisa kita gunakan, yaitu:

- Flutter pub add, untuk menambahkan packages ke dependencies yang ada di pubspec.yaml
 - ➡ flutter pub add <package_name>
- Flutter pub get, untuk mendownload semua packages atau dependencies yang ada di pubspec.yaml
 - ➡ flutter pub get

```
PS P:\aplikasi_flutter> flutter pub add http
Resolving dependencies...
+ http 0.13.4
+ http_parser 4.0.0
  material_color_utilities 0.1.3 (0.1.4 available)
  source_span 1.8.1 (1.8.2 available)
Changed 2 dependencies!
PS P:\aplikasi_flutter>
```

```
PS P:\aplikasi_flutter> flutter pub get
Running "flutter pub get" in aplikasi_flutter... 704ms
PS P:\aplikasi_flutter>
```

Flutter Build

Perintah untuk memproduksi sebuah file aplikasi untuk keperluan deploy atau publish ke AppStore, PlayStore, dll.

▶ flutter build <DIRECTORY>

```
PS P:\aplikasi_flutter> flutter build apk
Building with sound null safety
Running Gradle task 'assembleRelease'...
  Built build\app\outputs\flutter-apk\app-release.apk (15.5MB).
328.8s
PS P:\aplikasi_flutter>
```

Flutter Clean

- Perintah untuk menghapus folder build serta file lainnya yang dihasilkan saat kita menjalankan aplikasi di emulator
- Perintah ini akan memperkecil ukuran project tersebut.

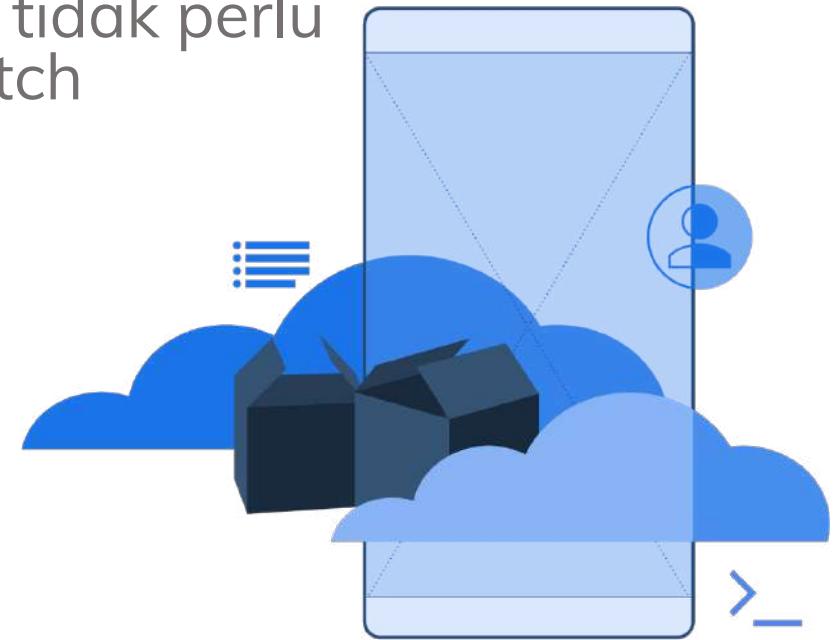
➤ flutter clean

```
PS P:\aplikasi_flutter> flutter clean
Deleting build...                                2,269ms
Deleting .dart_tool...                            214ms
Deleting .packages...                           1ms
Deleting Generated.xcconfig...                  0ms
Deleting flutter_export_environment.sh...      0ms
PS P:\aplikasi_flutter> code .
PS P:\aplikasi_flutter>
```

Packages Management

Deskripsi

- Flutter mendukung sharing packages
- Packages dibuat developers lain
- Mempercepat pengembangan aplikasi karena tidak perlu membuat semuanya dari awal atau from scratch
- Mendapatkan packages di website pub.dev



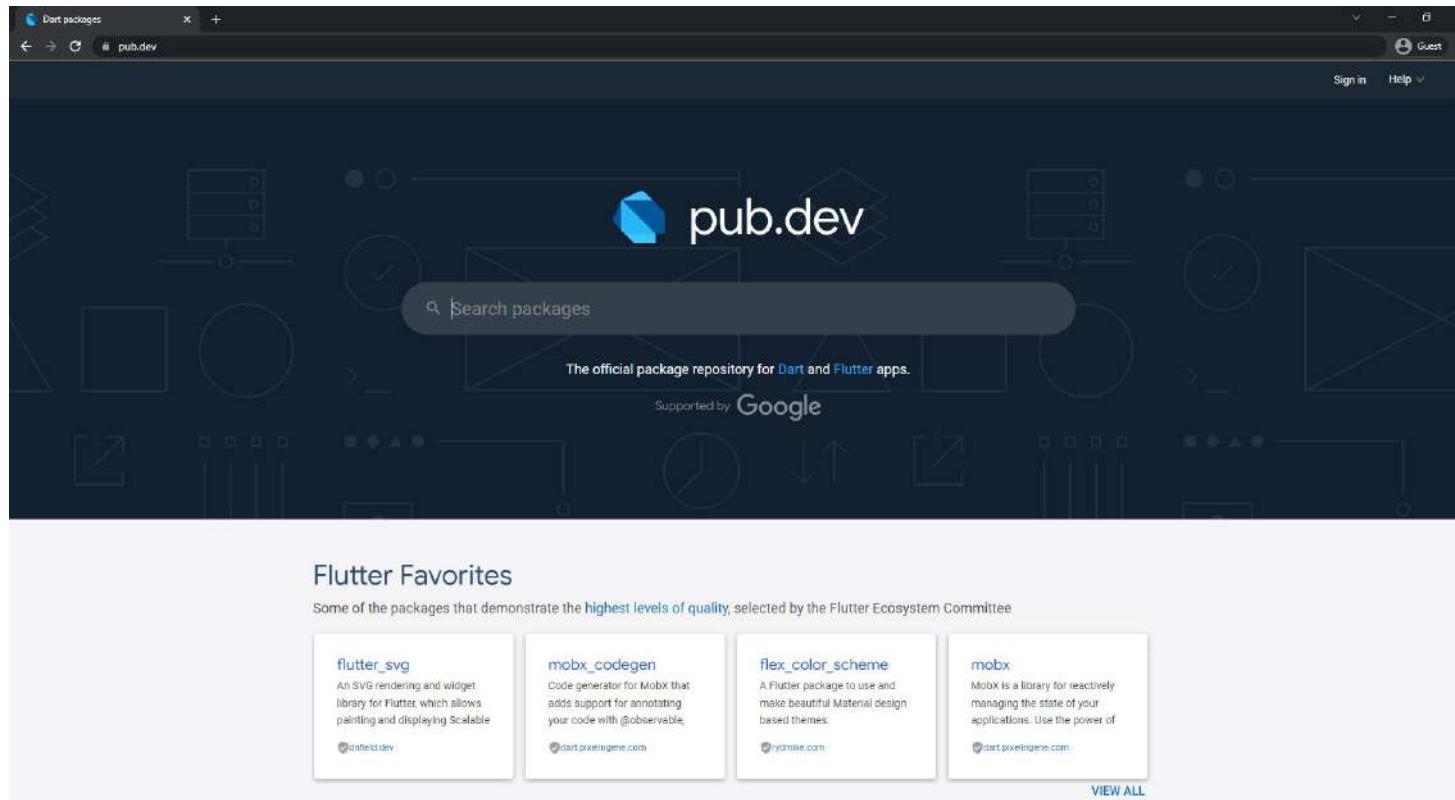
Cara Menambahkan Packages

- Cari package di pub.dev
- Copy baris dependencies yang ada di bagian installing
- Buka pubspec.yaml
- Paste barisnya dibawah dependencies pubspec.yaml
- Run `flutter pub get` di terminal
- Import package di file dart agar bisa digunakan
- Stop atau restart aplikasi jika dibutuhkan



Cara Menambahkan Packages

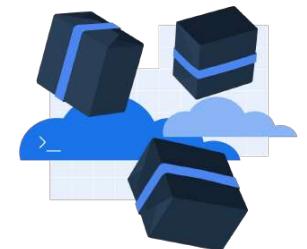
Cari package di pub.dev



The screenshot shows the pub.dev homepage. At the top, there's a search bar with the placeholder "Search packages". Below it, a banner reads "The official package repository for Dart and Flutter apps. Supported by Google". Underneath the banner, there's a section titled "Flutter Favorites" which lists several popular packages:

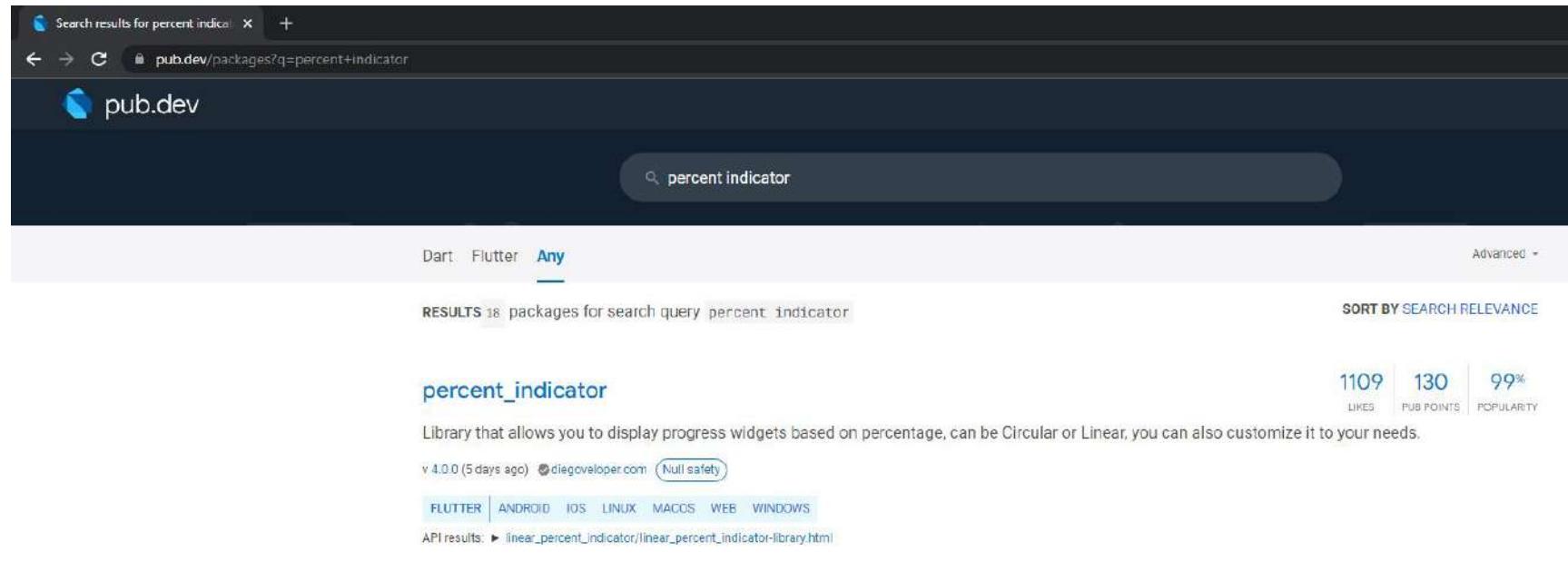
- flutter_svg**: An SVG rendering and widget library for Flutter, which allows painting and displaying Scalable Vector Graphics.
- mobx_codegen**: Code generator for MobX that adds support for annotating your code with `@observable`.
- flex_color_scheme**: A Flutter package to use and make beautiful Material design based themes.
- mobx**: MobX is a library for reactively managing the state of your applications. Use the power of

At the bottom of the list, there's a "VIEW ALL" button.



Cara Menambahkan Packages

Ex. Menambahkan package Percent Indicator



Search results for percent indicator

pub.dev/packages?q=percent+indicator

percent_indicator 4.0.0

Published 5 days ago · diegveloper.com · Null safety

Flutter | Android | iOS | Linux | macOS | Web | Windows

1.11K

Percent Indicator

pub v4.0.0

Circular and Linear percent indicators

Linear Percent Indicators

Circular Percent Indicators

40 hours

70.0%

Sales this week

100%

10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

1109 LIKES 130 PUB POINTS 99% POPULARITY

SORT BY SEARCH RELEVANCE

RESULTS 18 packages for search query percent indicator

Dart | Flutter | Any

Advanced

percent_indicator

Library that allows you to display progress widgets based on percentage, can be Circular or Linear, you can also customize it to your needs.

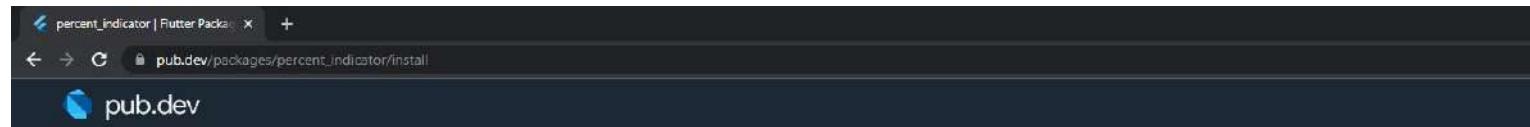
v 4.0.0 (5 days ago) · diegveloper.com · Null safety

FLUTTER | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

API results: linear_percent_indicator/library.html

Cara Menambahkan Packages

Copy baris dependencies yang ada di bagian installing



percent_indicator 4.0.0

Published 5 days ago · diegodeveloper.com Null safety

FLUTTER | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

1.11K

Readme Changelog Example **Installing** Versions Scores

1109 | 130 | 99%

Likes Pub Points Popularity

Publisher diegodeveloper.com

Metadata

Library that allows you to display progress widgets based on percentage, can be Circular or Linear, you can also customize it to your needs.

Homepage Repository (GitHub) View/report issues

Documentation

Use this package as a library

Depend on it

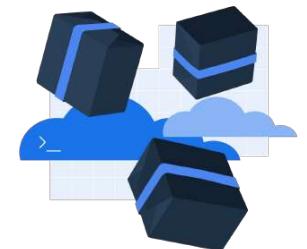
Run this command:

With Flutter:

```
$ flutter pub add percent_indicator
```

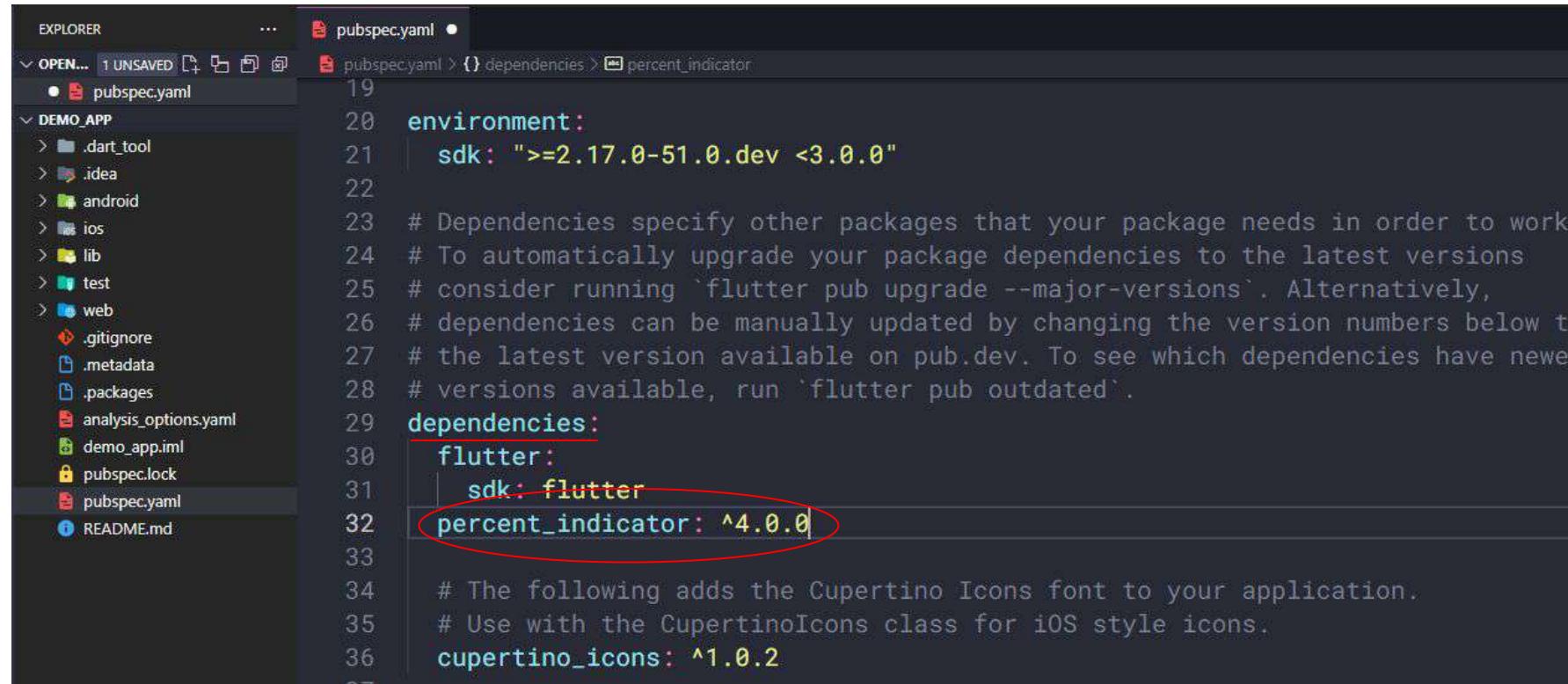
This will add a line like this to your package's pubspec.yaml (and run an implicit flutter pub get):

```
dependencies:  
  percent_indicator: ^4.0.0
```



Cara Menambahkan Packages

Buka pubspec.yaml dan Paste barisnya dibawah dependencies



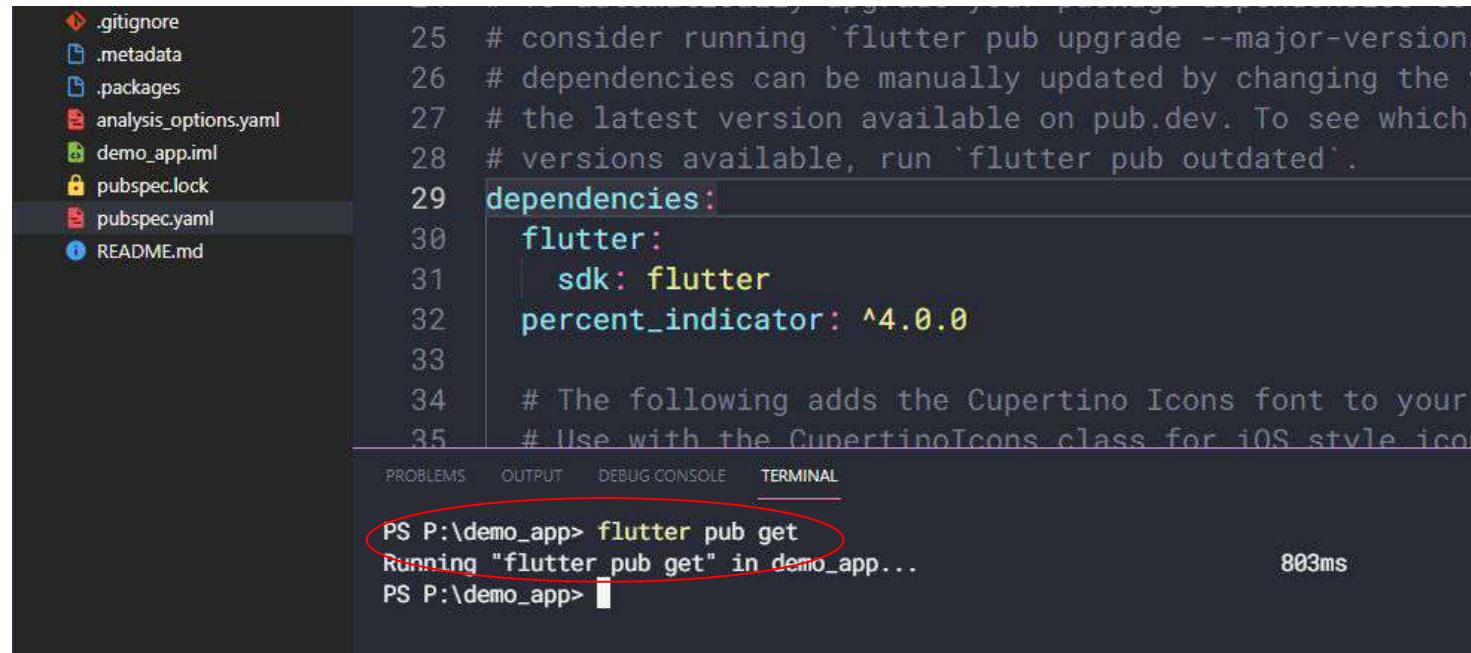
```
EXPLORER          ...  pubspec.yaml •
OPEN...  1 UNSAVED  🔍  🗑  🗃  🔍
pubspec.yaml
DEMO_APP
  .dart_tool
  .idea
  android
  ios
  lib
  test
  web
    .gitignore
    .metadata
    .packages
    analysis_options.yaml
    demo_app.iml
    pubspec.lock
    pubspec.yaml
  README.md

19
20 environment:
21   sdk: ">=2.17.0-51.0.dev <3.0.0"
22
23 # Dependencies specify other packages that your package needs in order to work.
24 # To automatically upgrade your package dependencies to the latest versions
25 # consider running `flutter pub upgrade --major-versions`. Alternatively,
26 # dependencies can be manually updated by changing the version numbers below to
27 # the latest version available on pub.dev. To see which dependencies have newer
28 # versions available, run `flutter pub outdated`.
29 dependencies:
30   flutter:
31     sdk: flutter
32     percent_indicator: ^4.0.0
33
34 # The following adds the Cupertino Icons font to your application.
35 # Use with the CupertinoIcons class for iOS style icons.
36/cupertino_icons: ^1.0.2
37
```



Cara Menambahkan Packages

Run flutter pub get di terminal atau CTRL + S

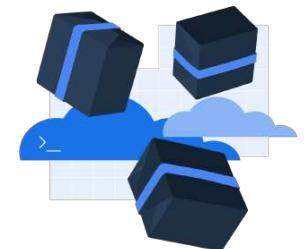


The screenshot shows a code editor interface with a dark theme. On the left, there's a sidebar displaying project files: .gitignore, .metadata, .packages, analysis_options.yaml, demo_app.iml, pubspec.lock, pubspec.yaml, and README.md. The main area shows a portion of a pubspec.yaml file with the following content:

```
25 # consider running `flutter pub upgrade --major-versions`
26 # dependencies can be manually updated by changing the ve
27 # the latest version available on pub.dev. To see which c
28 # versions available, run `flutter pub outdated`.
29 dependencies:
30   flutter:
31     sdk: flutter
32     percent_indicator: ^4.0.0
33
34   # The following adds the Cupertino Icons font to your a
35   # Use with the CupertinoIcons class for iOS style icons
```

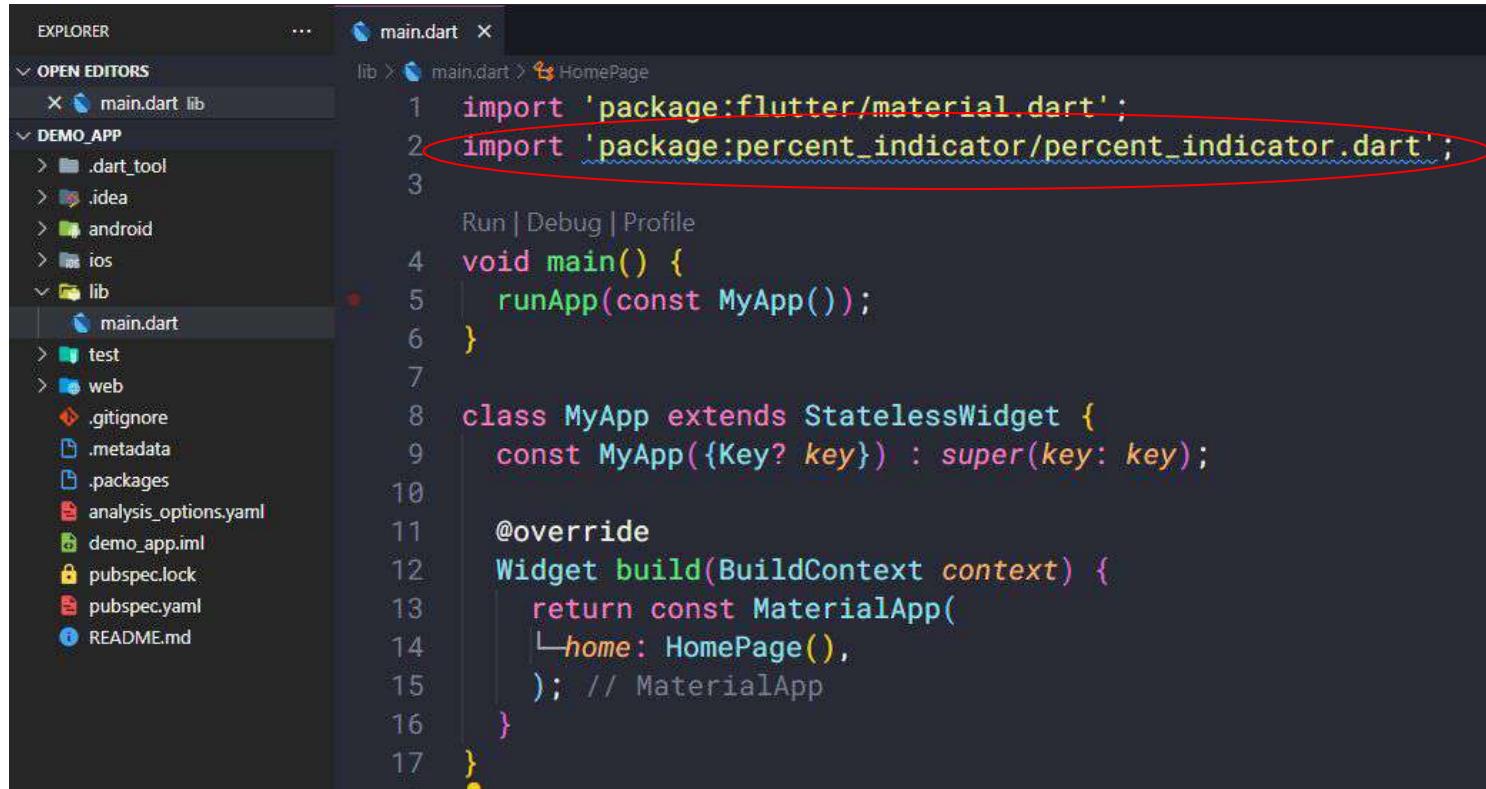
Below the code editor is a terminal window with the following output:

```
PS P:\demo_app> flutter pub get
Running "flutter pub get" in demo_app...
803ms
PS P:\demo_app>
```



Cara Menambahkan Packages

Import package di file dart agar bisa digunakan



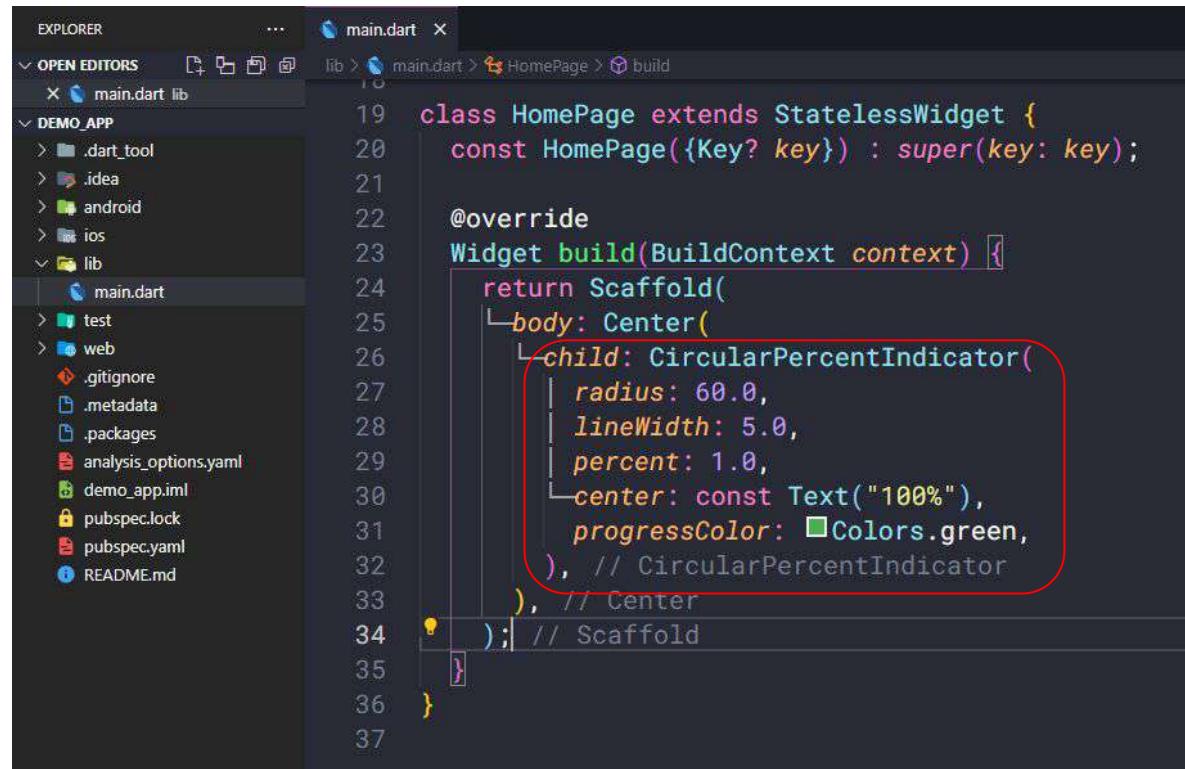
```
EXPLORER          ... main.dart x
OPEN EDITORS      lib X main.dart > HomePage
DEMO_APP          .dart_tool
                  .idea
                  android
                  ios
lib              lib
                  main.dart
test
web
.gitignore
.metadata
.packages
analysis_options.yaml
demo_app.iml
pubspec.lock
pubspec.yaml
README.md

main.dart
1 import 'package:flutter/material.dart';
2 import 'package:percent_indicator/percent_indicator.dart';
3
4 void main() {
5     runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9     const MyApp({Key? key}) : super(key: key);
10
11     @override
12     Widget build(BuildContext context) {
13         return const MaterialApp(
14             home: HomePage(),
15         ); // MaterialApp
16     }
17 }
```



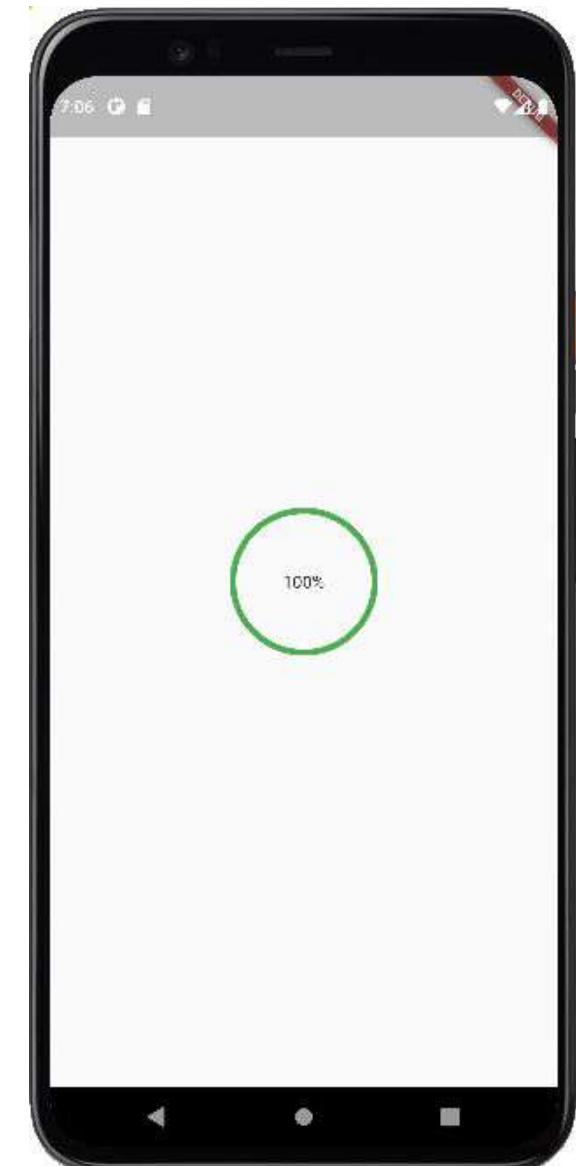
Menggunakan Packages

Hanya membutuhkan beberapa baris kode untuk menghasilkan UI yang kompleks



```
EXPLORER ... main.dart x
OPEN EDITORS lib > main.dart > HomePage > build
main.dart lib
DEMO_APP .dart_tool
.idea
android
ios
lib
main.dart
test
web
.gitignore
.metadata
.packages
analysis_options.yaml
demo_app.iml
.pubspec.lock
.pubspec.yaml
README.md

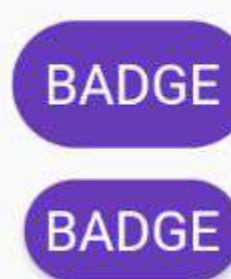
19 class HomePage extends StatelessWidget {
20   const HomePage({Key? key}) : super(key: key);
21
22   @override
23   Widget build(BuildContext context) {
24     return Scaffold(
25       body: Center(
26         child: CircularPercentIndicator(
27           radius: 60.0,
28           lineWidth: 5.0,
29           percent: 1.0,
30           center: const Text("100%"),
31           progressColor: Colors.green,
32         ), // CircularPercentIndicator
33       ), // Center
34     ); // Scaffold
35   }
36 }
37 }
```



Terima Kasih

Task

- 1A. Menggunakan package <https://pub.dev/packages/badges>
- 1B. Buatlah tampilan berikut ini!



Task

- 2A. Menggunakan package https://pub.dev/packages/barcode_widget
- 2B. Buatlah tampilan barcode dengan data berikut ini!
- a. Alterra Academy
 - b. Flutter Asik
 - c. <nama lengkap Anda>

Flutter Navigation

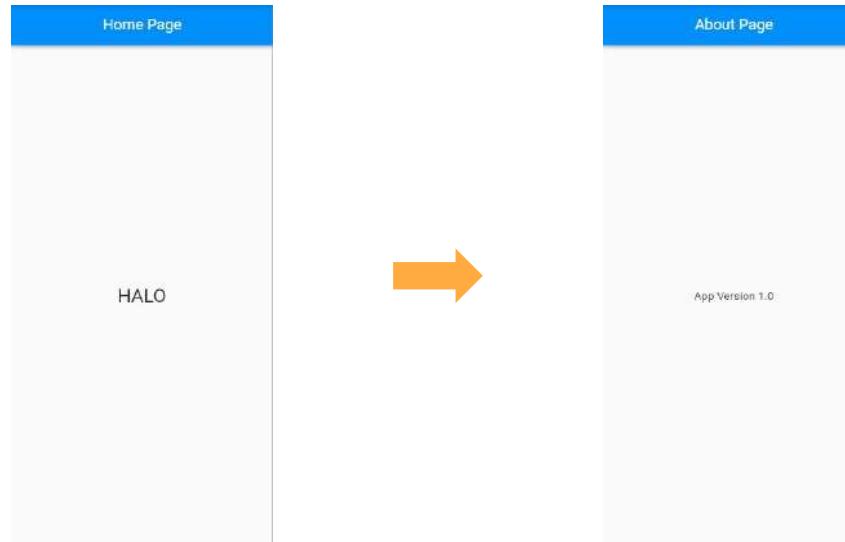
Outline

- Apa itu Navigation?
- Navigation Dasar
- Navigation dengan Named Routes

Apa itu Navigation?

Deskripsi

Berpindah dari halaman satu ke halaman lain



Navigation Dasar

Deskripsi

Perpindahan halaman menggunakan
Navigator.push()



GO



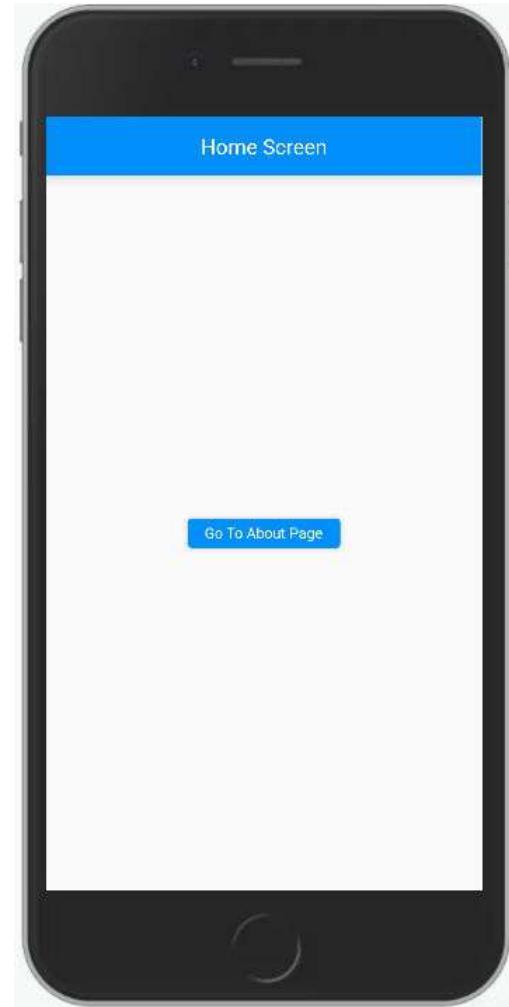
Back

Kembali ke halaman sebelumnya menggunakan
Navigator.pop()

Perpindahan Halaman

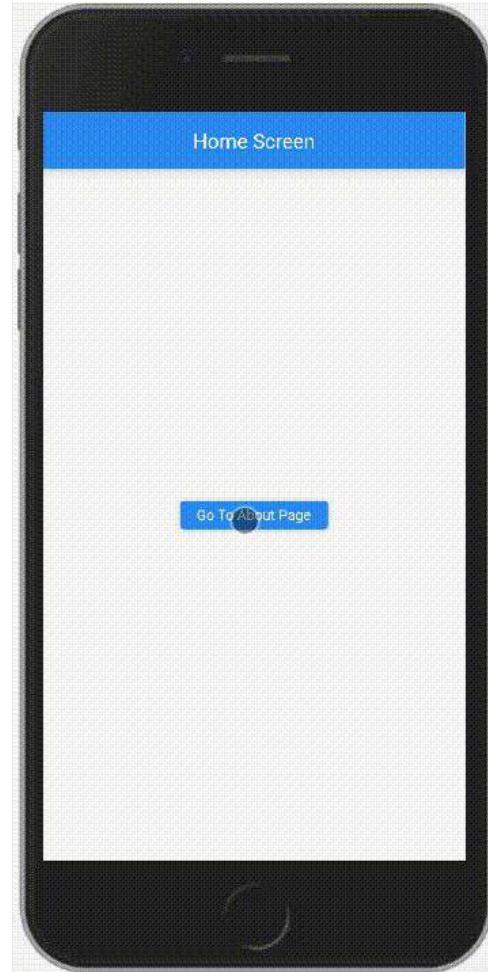
```
class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Home Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          child: const Text('Go To About Page'),
          onPressed: () {
            // saat tombol di tekan
            // pindah ke halaman AboutScreen
            Navigator.of(context).push(
              MaterialPageRoute(builder: (_) => const AboutScreen()),
            );
          },
        ),
      ),
    );
  }
}
```



Perpindahan Halaman

```
•••  
  
class AboutScreen extends StatelessWidget {  
  const AboutScreen({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('About Screen'),  
      ),  
    );  
  }  
}
```



Mengirim Data ke Halaman Baru

Menggunakan parameter pada constructor halaman



```
ElevatedButton(  
    child: const Text('Go To About Page'),  
    onPressed: () {  
  
        // saat tombol di tekan  
        // pindah ke halaman AboutScreen dengan parameter  
        Navigator.of(context).push(  
            MaterialPageRoute(  
                builder: (_) => const AboutScreen(parameter: 'Home'),  
            ),  
        );  
    },  
);
```



```
class AboutScreen extends StatelessWidget {  
  
    // property yang akan menerima parameter  
    final String parameter;  
    const AboutScreen({Key? key, required this.parameter}) : super(key: key);  
  
    @override  
    Widget build(BuildContext context) {  
        return ...;  
    }  
}
```

Navigation dengan Named Routes

Deskripsi

// Tiap halaman memiliki alamat yang disebut route

Perpindahan halaman menggunakan
Navigator.pushNamed()



GO



Back

Kembali ke halaman sebelumnya menggunakan
Navigator.pop()

Mendaftarkan Route

- Tambahkan initialRoute dan routes pada MaterialApp
- Tiap route adalah fungsi yang membentuk halaman

```
MaterialApp(  
    debugShowCheckedModeBanner: false,  
  
    // menentukan halaman yang pertama kali dibuka  
    initialRoute: '/',  
  
    // daftar halaman yang dapat dibuka  
    routes: {  
        '/': (_) => const HomeScreen(),  
        '/about': (_) => const AboutScreen(),  
    },  
);
```

Perpindahan Halaman

```
ElevatedButton(  
    child: const Text('Go To About Page'),  
    onPressed: () {  
  
        // saat tombol di tekan  
        // pindah ke halaman dengan nama /about  
        Navigator.of(context).pushNamed('/about');  
    },  
);
```

Mengirim Data ke Halaman Baru

Menggunakan arguments saat melakukan pushNamed

```
ElevatedButton(  
    child: const Text('Go To About Page'),  
    onPressed: () {  
  
        // saat tombol di tekan  
        // pindah ke halaman AboutScreen dengan parameter  
        Navigator.of(context).pushNamed(  
            '/about',  
            arguments: 'Hore',  
        );  
    },  
);
```

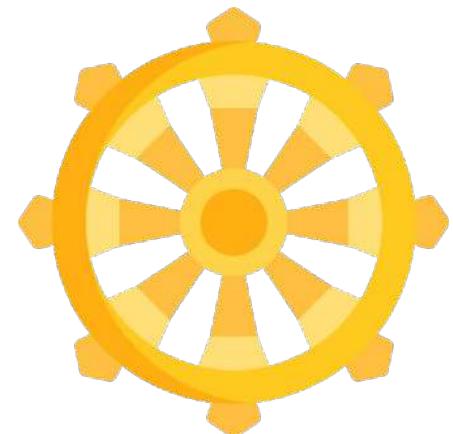
```
class AboutScreen extends StatelessWidget {  
  
    // ...  
  
    @override  
    Widget build(BuildContext context) {  
  
        // mengambil parameter yang dikirim melalui arguments  
        final parameter = ModalRoute.of(context)!.settings.arguments as String;  
  
        return ...;  
    }  
}
```



Terima Kasih

Task

1. Buatlah halaman **Contacts** yang menampilkan data contacts menggunakan List View dan letakkan button untuk membuka halaman **Create Contact**!
2. Lakukan navigation di atas dengan dan tanpa named routes!





Flutter **Global State Management**

Outline

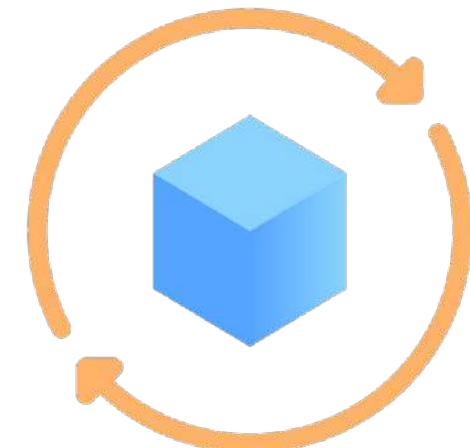
- Pengertian State
- Global State



State

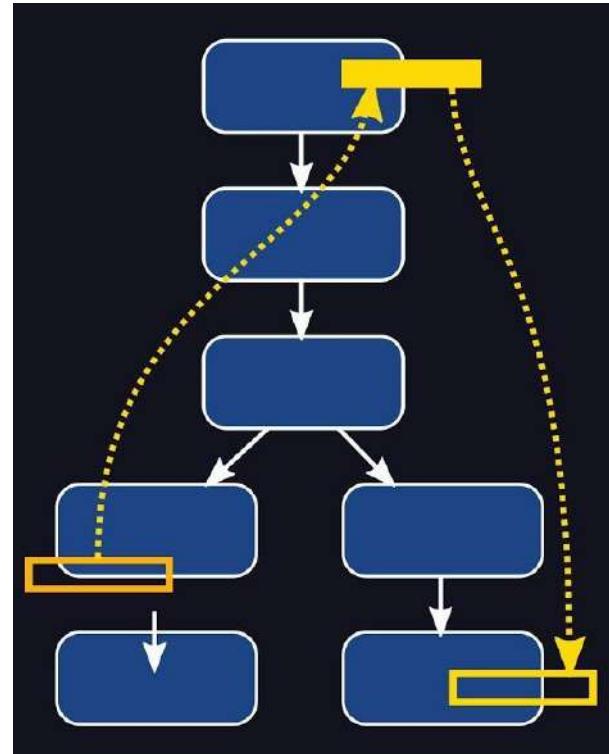
Deskripsi

- Data yang dapat dibaca saat pembuatan widget
- Dapat berubah saat widget sedang aktif
- Hanya dimiliki oleh **StatefulWidget**



Kenapa Perlu Global State?

Agar antara widget dapat
memanfaatkan state yang sama
dengan mudah



Memanfaatkan State



```
// property  
var number = 0;  
  
// build  
Text( '$number' );
```

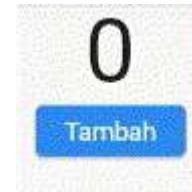
- Dibuat sebagai property dari class
- Digunakan pada widget saat build

Mengubah State

- Menggunakan method **setState**



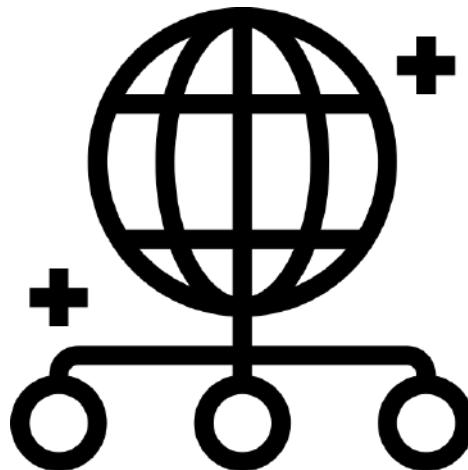
```
ElevatedButton(  
    child: const Text('Tambah'),  
    onPressed: () {  
        setState(() {  
            number = number + 1;  
        });  
    },  
);
```



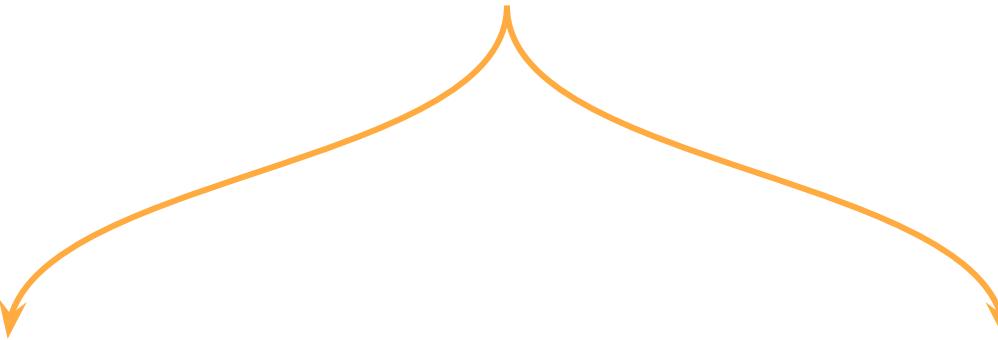
Global State

Deskripsi

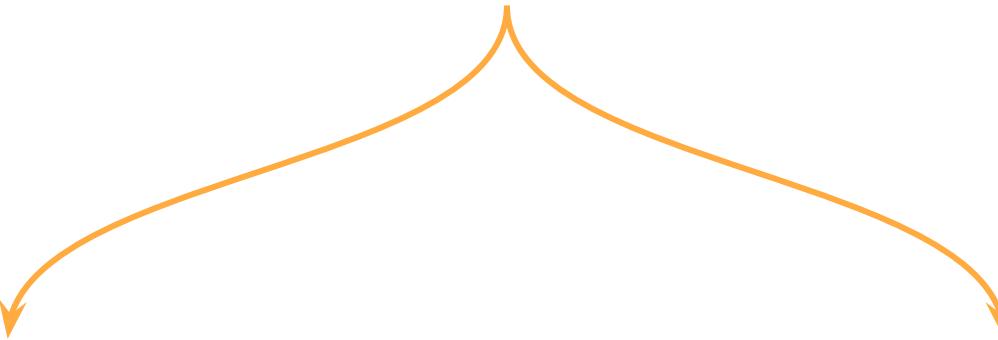
State biasa yang dapat digunakan pada seluruh widget



Provider



State management



Perlu di-install agar dapat
digunakan

Instalasi Provider

- Menambah package provider pada bagian dependencies dalam file pubspec.yaml
- Jalankan perintah **flutter pub get**



```
dependencies:  
  provider:
```



```
$ flutter pub get  
Running "flutter pub get" in phonebookapp...
```

1,024ms

Membuat State Provider

- Buat file bernama **contact.dart**
- Definisikan state dalam bentuk class

```
import 'package:flutter/material.dart';

class Contact with ChangeNotifier {
  List<Map<String, String>> _contacts = [];

  List<Map<String, String>> get contacts => _contacts;

  void add(Map<String, String> contact) {
    _contacts.add(contact);
    notifyListeners();
  }
}
```

Mendaftarkan State Provider

- Import dalam file main.dart
- Daftarkan pada runApp dengan MultiProvider

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:phonebookapp/stores/contact.dart' as contact_store;

void main() {
  runApp(
    MultiProvider(
      providers: [
        ChangeNotifierProvider(create: (_) => contact_store.Contact())
      ],
      child: MyWidget(),
    ),
  );
}
```

Menggunakan State dari Provider

- Simpan provider dalam variable
- Ambil data dari provider melalui getter

Alex Under
0823878273

```
// build
final contactProvider = Provider.of<contact_store.Contact>(context);

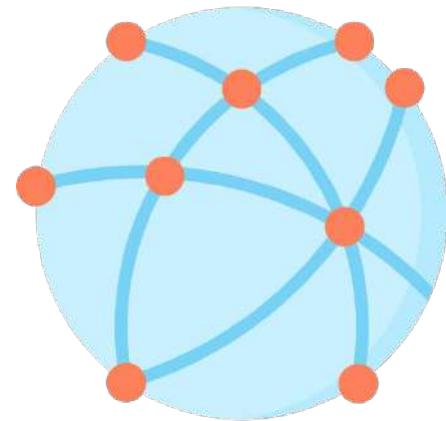
// return
ListView.builder(
    itemCount: contactProvider.contacts.length,
    itemBuilder: (context, index) {
        return ListTile(
            title: Text(contactProvider.contacts[index]['name'] ?? ''),
            subtitle: Text(contactProvider.contacts[index]['phone'] ?? ''),
        );
    },
);
```



Terima Kasih

Task

1. Buatlah state yang menyimpan data list contacts pada halaman Contacts!
2. Pindahkan state tersebut ke dalam global state!



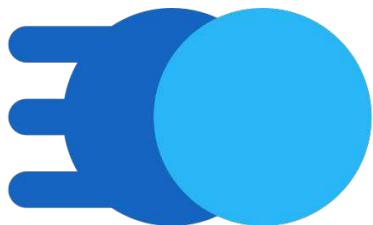
Flutter Animation

Outline

- Animation
- Implicit Animation
- Transition
- Transition Umum

Animation

Deskripsi



- Animasi membuat aplikasi terlihat hidup
- Widget yang bergerak menambah daya tarik
- Pergerakan dari kondisi A ke kondisi B

Implicit Animation

Deskripsi

Versi animasi dari widget yang sudah ada

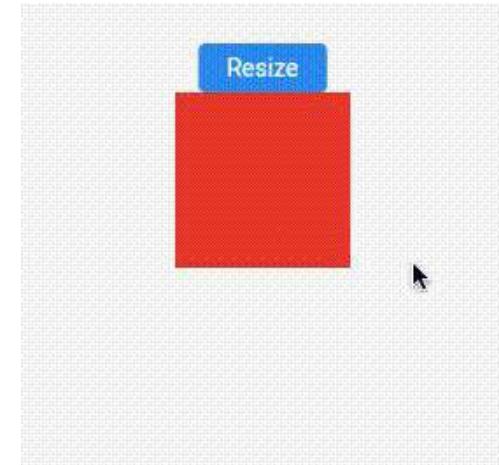


Tanpa Animation

Perubahan ukuran container tampak kaku

```
// state
bool isBig = false;

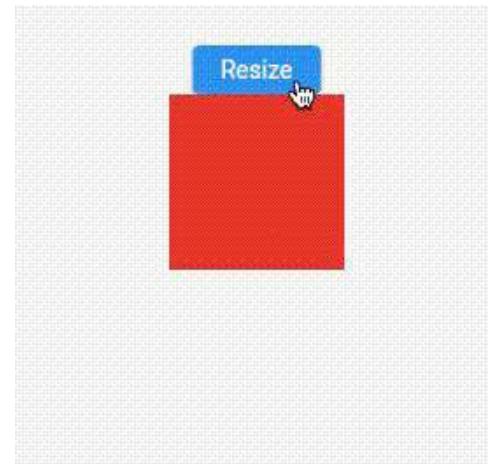
// container berubah ukuran
// jika isBig = true, tinggi dan lebar menjadi 200
// jika isBig = false, tinggi dan lebar menjadi 100
Container(
  width: isBig ? 200 : 100,
  height: isBig ? 200 : 100,
  color: Colors.red,
);
```



Dengan Animation

Perubahan ukuran container menjadi lebih hidup dan nyaman dilihat

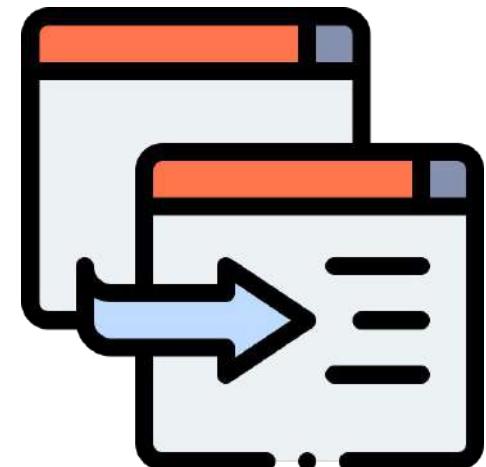
```
● ● ●  
  
// state  
bool isBig = false;  
  
// container berubah ukuran dengan animasi  
// jika isBig = true, tinggi dan lebar menjadi 200  
// jika isBig = false, tinggi dan lebar menjadi 100  
AnimatedContainer(  
  width: isBig ? 200 : 100,  
  height: isBig ? 200 : 100,  
  color: Colors.red,  
  duration: const Duration(milliseconds: 300),  
);
```



Transition

Deskripsi

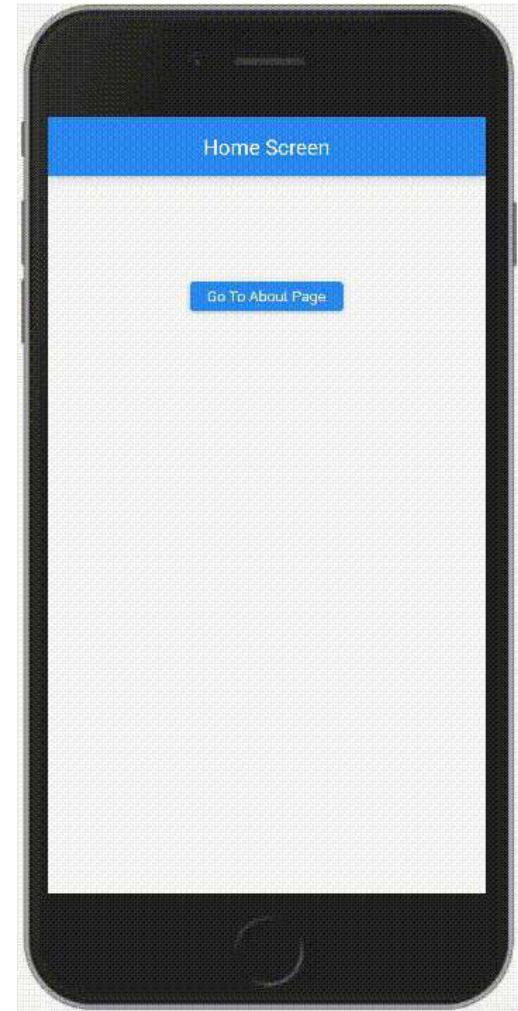
- Animasi yang terjadi saat perpindahan halaman
- Dilakukan pada bagian `Navigator.push()`



Melakukan Transition



```
Navigator.of(context).push(  
  
    // mengatur proses Navigasi terjadi  
    PageRouteBuilder(  
  
        // proses pembuatan halaman AboutScreen  
        pageBuilder: (context, animation, secondaryAnimation) {  
            return const AboutScreen();  
        },  
  
        // proses mengatur transition  
        // halaman AboutScreen muncul dari tengah halaman  
        // menuju ke halaman penuh dengan transisi bergeser (slide)  
        transitionsBuilder: (context, animation, secondaryAnimation, child) {  
            final tween = Tween(begin: const Offset(0, .5), end: Offset.zero);  
  
            return SlideTransition(  
                position: animation.drive(tween),  
                child: child,  
            );  
        },  
    ),  
);
```



Transition Yang Umum

FadeTransition

Halaman muncul dengan efek redup
menuju ke tampak jelas secara penuh



```
final tween = Tween(begin: 0.0, end: 1.0);
return FadeTransition(
  opacity: animation.drive(tween),
  child: child,
);
```

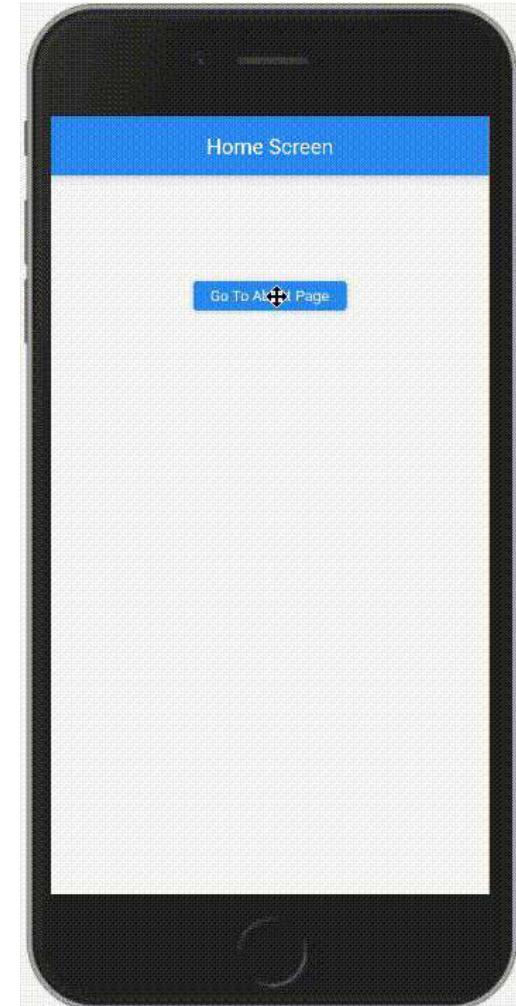


ScaleTransition

Halaman muncul dengan efek berubah ukuran dari tidak penuh menuju tampil keseluruhan



```
final tween = Tween(begin: 0.0, end: 1.0);
return ScaleTransition(
  scale: animation.drive(tween),
  child: child,
);
```





Terima Kasih

Task

1. Buatlah sebuah halaman dengan spesifikasi berikut:
 - a. menampilkan sebuah gambar kecil di posisi tengah,
 - b. jika gambar diklik, akan membesar,
 - c. jika diklik kembali, akan mengecil (ukuran semula),
 - d. proses perubahan gambar menggunakan animasi.
2. Tambahkan transisi saat perpindahan halaman **Contact** ke **Create Contact!**

