

BAB 1

DASAR-DASAR FLUTTER

IT CLUB – Android Development

Fasilitator: Alfiki Diastama Afan Firdaus

Menyiapkan Android Device

Pastikan peranti yang akan dipakai minimal Android 8 (Oreo) agar bisa menyesuaikan Flutter terbaru

01

Kemudian tekan Build number sebanyak 7 kali.

03

Masuk ke dalam menu Developer Options dan pastikan opsi USB Debugging Mode sudah dalam keadaan On.

05

02

Buka setting dan masuk ke dalam menu About.

04


Kembali ke menu setting di awal dan akan muncul menu baru di bawah about yaitu Developer Options.

06

Hubungkan laptop/PC dengan smartphone Android melalui kabel data.

Project Wizard



 New Project ✕

Project name:

flutter_app

Project location:

D:\Project\Private\Flutter\flutter_app

...

Description:

A new Flutter project.

Project type:

Application ▾

Organization:

com.dicoding

Android language:

☐ Java

☒ Kotlin

iOS language:

☐ Objective-C

☒ Swift

Platforms:

☒ Android

☒ iOS

☐ Linux

☐ MacOS

☒ Web

☒ Windows

Platform availability might depend on your Flutter SDK channel,
and which desktop platforms have been enabled.

Additional desktop platforms can be enabled by, for example, running
"flutter config --enable-linux-desktop" on the command line.

When created, the new project will run on the selected platforms (others can be added later).

☐ Create project offline

Struktur Proyek Flutter

Setelah membuat project Flutter pertama kali, flutter akan membuatkan struktur project. Ketika membuka folder project Flutter pada berkas explorer, kita mendapati folder-folder seperti berikut:

- .dart_tool
- .idea
- android
- ios
- lib
- test

- .gitignore
- .metadata
- .packages
- eudeka.iml
- pubspec.lock
- pubspec.yaml
- README.md

Folder android
tempat untuk mengatur konfigurasi untuk aplikasi android.

1

2

Folder ios
tempat konfigurasi untuk aplikasi iOS.

3

Folder build
Ketika kita melakukan build project flutter, hasil build akan ada pada folder ini.

4

Folder lib
Seluruh source code flutter akan berada pada folder ini.

5

Folder test
tempat menyimpan source code testing. Untuk sekarang belum dipakai.

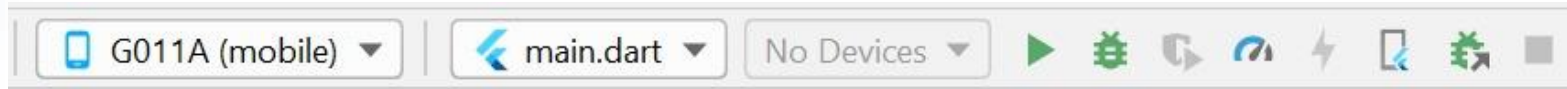
```
1.import 'package:flutter/material.dart';
2.
3.void main() {
4.  runApp(MyApp());
5.}
6.
7.class MyApp extends StatelessWidget {
8.  @override
9.  Widget build(BuildContext context) {
10.   return MaterialApp(
11.     title: 'IT CLUB Android Dev Demo',
12.     theme: ThemeData(
13.       primarySwatch: Colors.blue,
14.     ),
15.     home: Scaffold(
16.       appBar: AppBar(
17.         title: Text('Ini kode pertamaku!'),
18.       ),
19.       body: Center(
20.         child: Text('Semangat teman-teman!'),
21.       ),
22.     ),
23.   );
24. }
25. }
```

Hello
World App

Menjalankan Proyek Pertama

1

Temukan toolbar seperti di bawah ini:



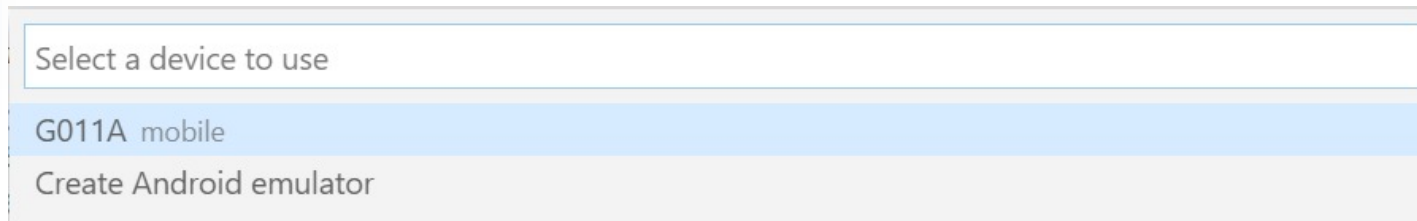
Android Studio/IntelliJ



Visual Studio Code

2

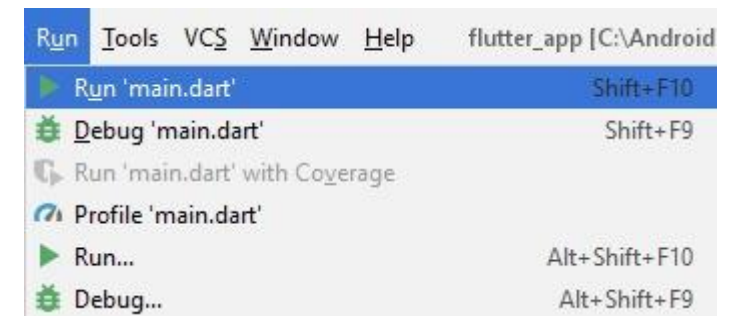
Pada target selector, pilih perangkat Android kamu



3

Klik ikon run pada toolbar, atau buka item menu Run > Run 'main.dart'. (Android Studio/IntelliJ)

Pilih Run > Start Debugging (VSCode)



Fitur Hot Reload, Hot Restart, dan Full Restart

Hot Reload

Memuat perubahan kode ke dalam VM dan menjalankan kembali metode build yang ada di dalam widget. Ini akan membangun kembali widget-widget yang ada, dan mempertahankan status terakhir aplikasi.

Hot Restart

Memuat perubahan kode ke dalam VM dan mengulang kembali aplikasi dari awal dan akan kehilangan state aplikasi (kembali ke nilai awal).

Full Restart

Mengkompilasi kode dari awal, tidak ada pintasan keyboard untuk ini, Anda harus menghentikan dan menjalankan kembali project (klik stop kemudian run kembali).

Melakukan Hot Reload

1

Buka lib/main.dart.

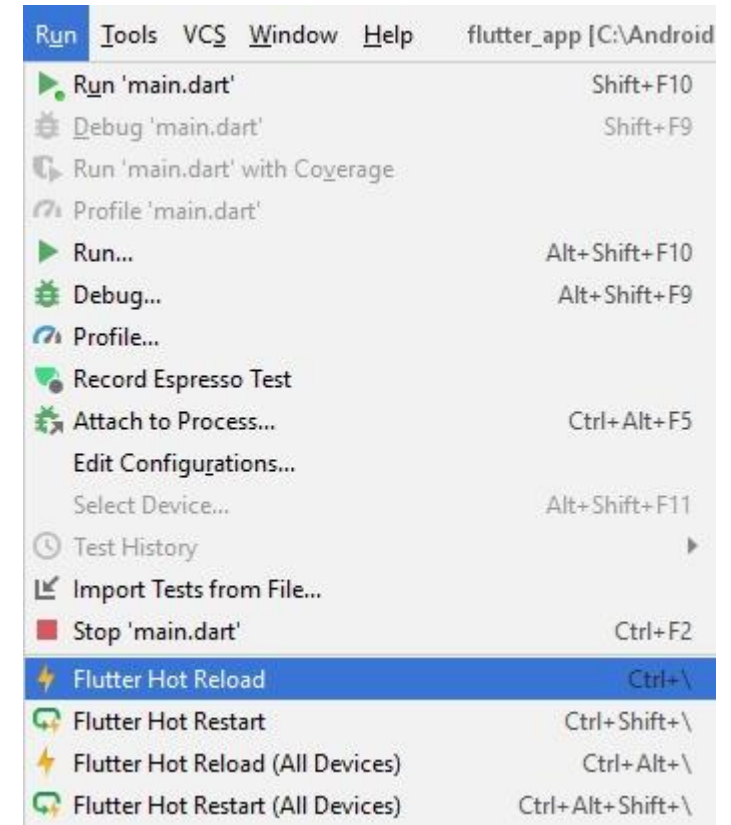
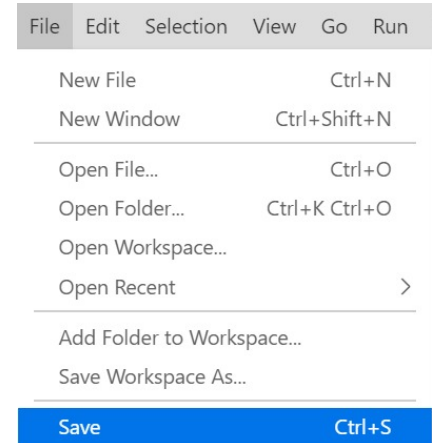
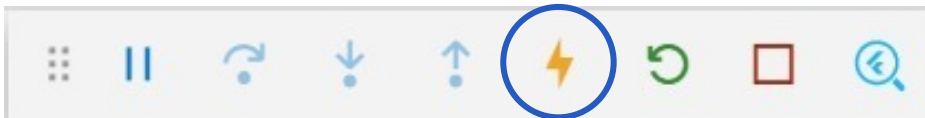
2

Ubah string Semangat teman-teman' menjadi Saya suka belajar Android'.

3

Simpan perubahan baris perintah dengan cara : Save All, atau klik Hot Reload. (IntelliJIDEA/Android Studio)

Pilih File > Save All, atau klik Hot Reload (tombol petir kuning).



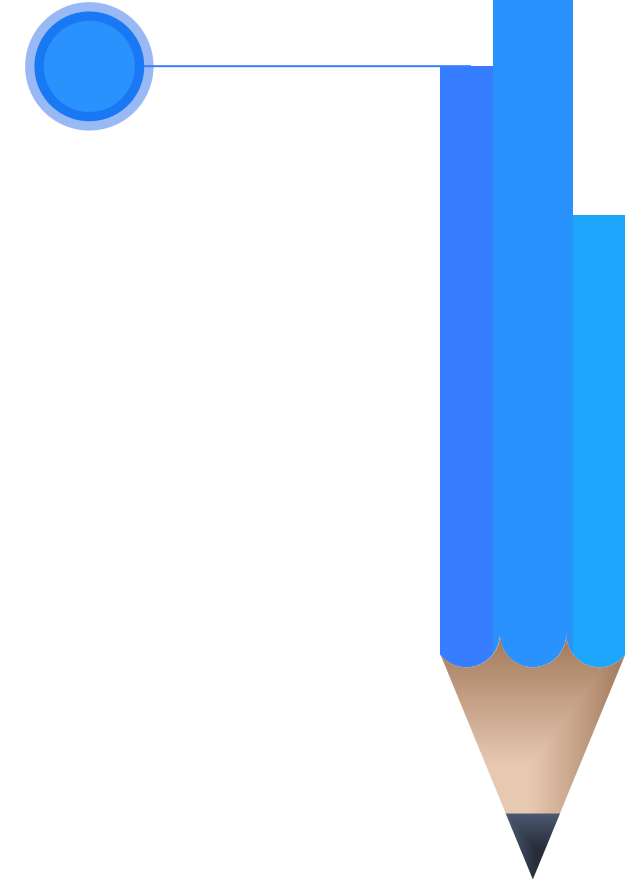
Package Dependencies

Sekumpulan package yang dibutuhkan dalam pengembangan aplikasi.

Bagaimana kita menggunakan pub pada project Flutter kita? Untuk mengatur package-package yang akan kita gunakan, cukup buka berkas pubspec.yaml yang ada pada folder project.

Name	Date modified	Type	Size
.dart_tool	15/01/2020 22:55	File folder	
.idea	16/01/2020 23:09	File folder	
android	15/01/2020 22:48	File folder	
build	15/01/2020 22:55	File folder	
ios	15/01/2020 22:44	File folder	
lib	15/01/2020 23:53	File folder	
test	15/01/2020 22:44	File folder	
.gitignore	15/01/2020 22:44	Text Document	1 KB
.metadata	15/01/2020 22:44	METADATA File	1 KB
.packages	15/01/2020 23:56	PACKAGES File	3 KB
first_app.iml	15/01/2020 22:44	IML File	1 KB
pubspec.lock	15/01/2020 22:44	LOCK File	5 KB
pubspec.yaml	15/01/2020 22:44	YAML File	3 KB
README.md	15/01/2020 22:44	Markdown docu...	1 KB

Dart Pub



- 1.dependencies:
2. flutter:
3. sdk: flutter
- 4.
5. # The following adds the Cupertino Icons font to your application.
6. # Use with the CupertinoIcons class for iOS style icons.
7. cupertino_icons: ^0.1.2
- 8.
- 9.dev_dependencies:
10. flutter_test:
- 11.sdk: flutter

Kode di atas merupakan package-package yang digunakan pada project Flutter kita. Fungsi dev_dependencies digunakan untuk package-package yang berkaitan ketika proses pengembangan aplikasi Flutter, contohnya seperti flutter_test yang digunakan untuk testing.

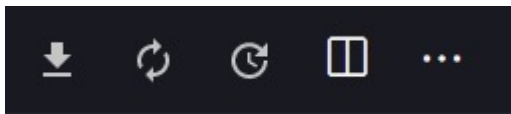
Untuk mendaftarkan package yang dibutuhkan kita cukup menulis seperti di bawah ini pada bagian dependencies:



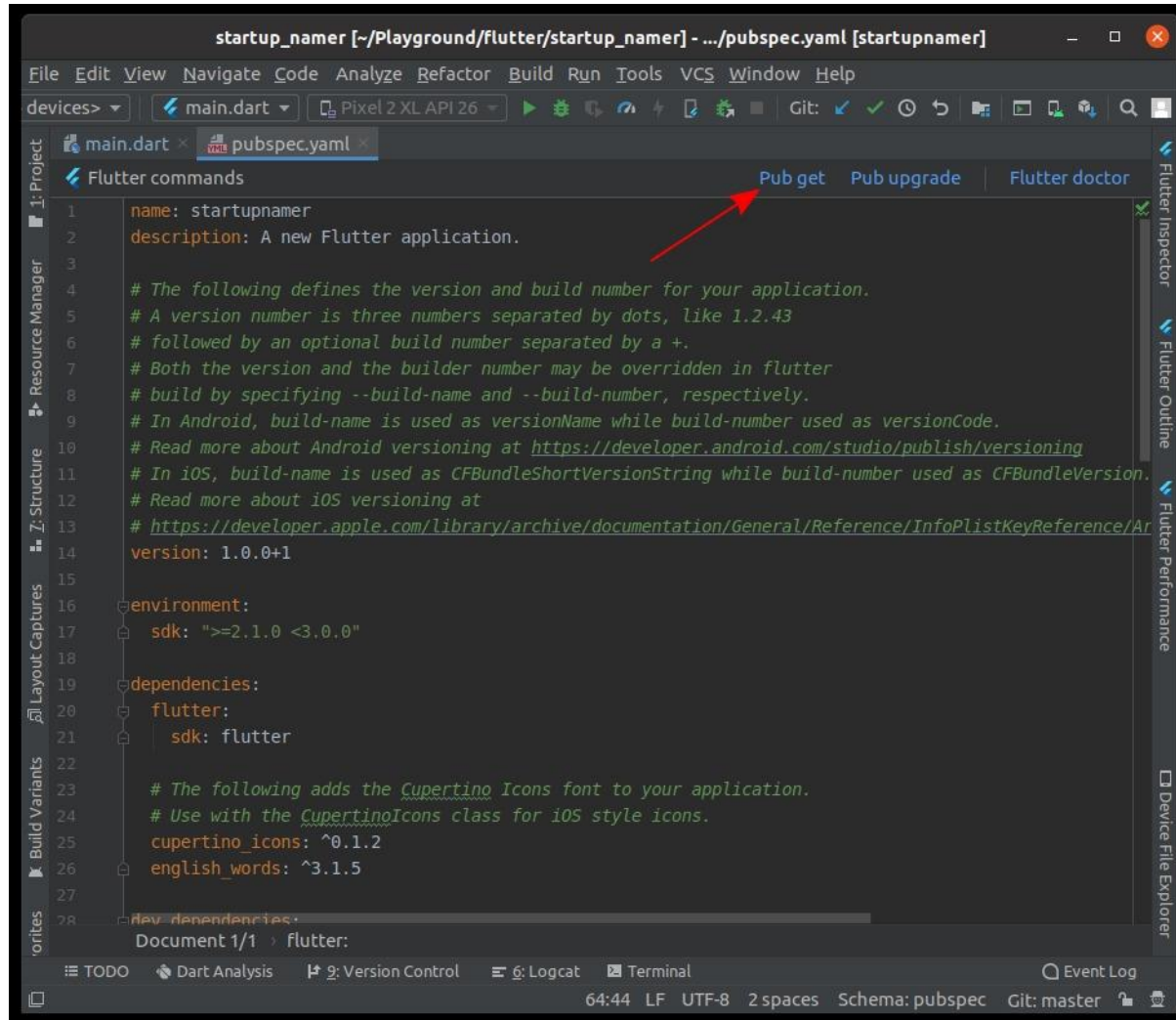
- 1.dependencies:
2. flutter:
3. sdk: flutter
- 4.
5. cupertino_icons: ^0.1.2
- 6.provider: ^4.0.1

Kita menambahkan dependencies provider dengan versi 4.0.1, maka penulisannya seperti di atas. Perhatikan indentasinya.

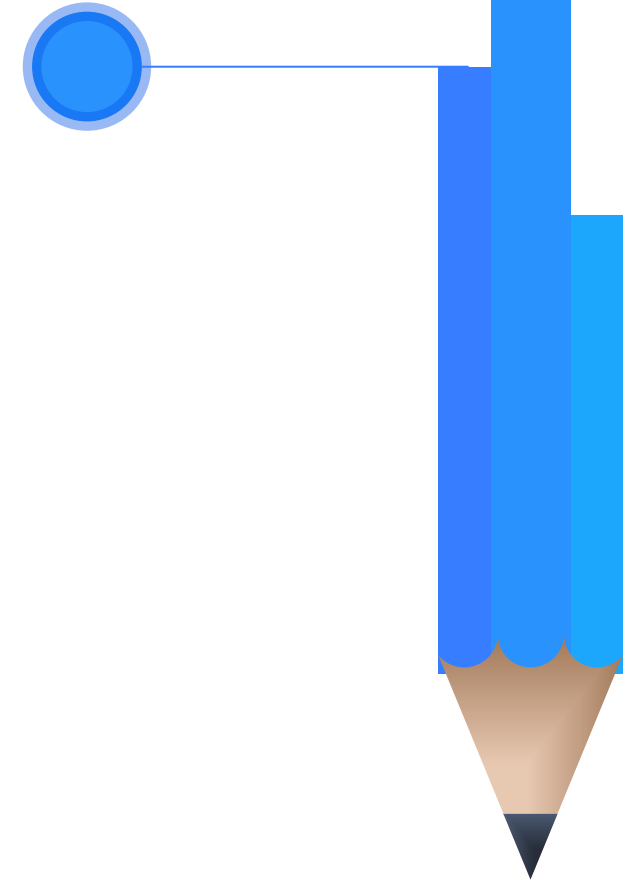
Setelah menambahkan package yang dibutuhkan, kita dapat melakukan get package tersebut. Jika Anda menggunakan visual studio code cukup simpan berkas pubspec.yaml, atau bisa klik tombol unduh di bawah ini:



Bila menggunakan Android Studio/IntelliJ Anda cukup menekan tombol "Pub get" pada Android Studio/IntelliJ



Dart Pub



Tempat-tempat mencari package flutter

1

<https://pub.dev>

Website ini merupakan *web official* untuk mencari *package*.

2

<https://flutterawesome.com>

Berisi *package-package* yang dibuat oleh komunitas, di sini banyak sekali *package* UI keren yang dapat Anda coba.

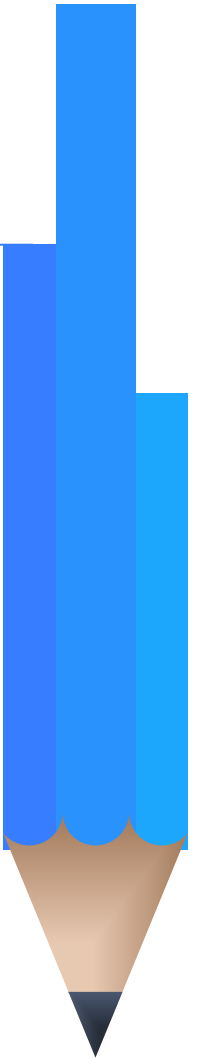
Private Packages

Teman-teman juga bisa menggunakan package yang tidak dipublikasikan pada pub.dev tersebut dengan cara menggunakan url git package tersebut:

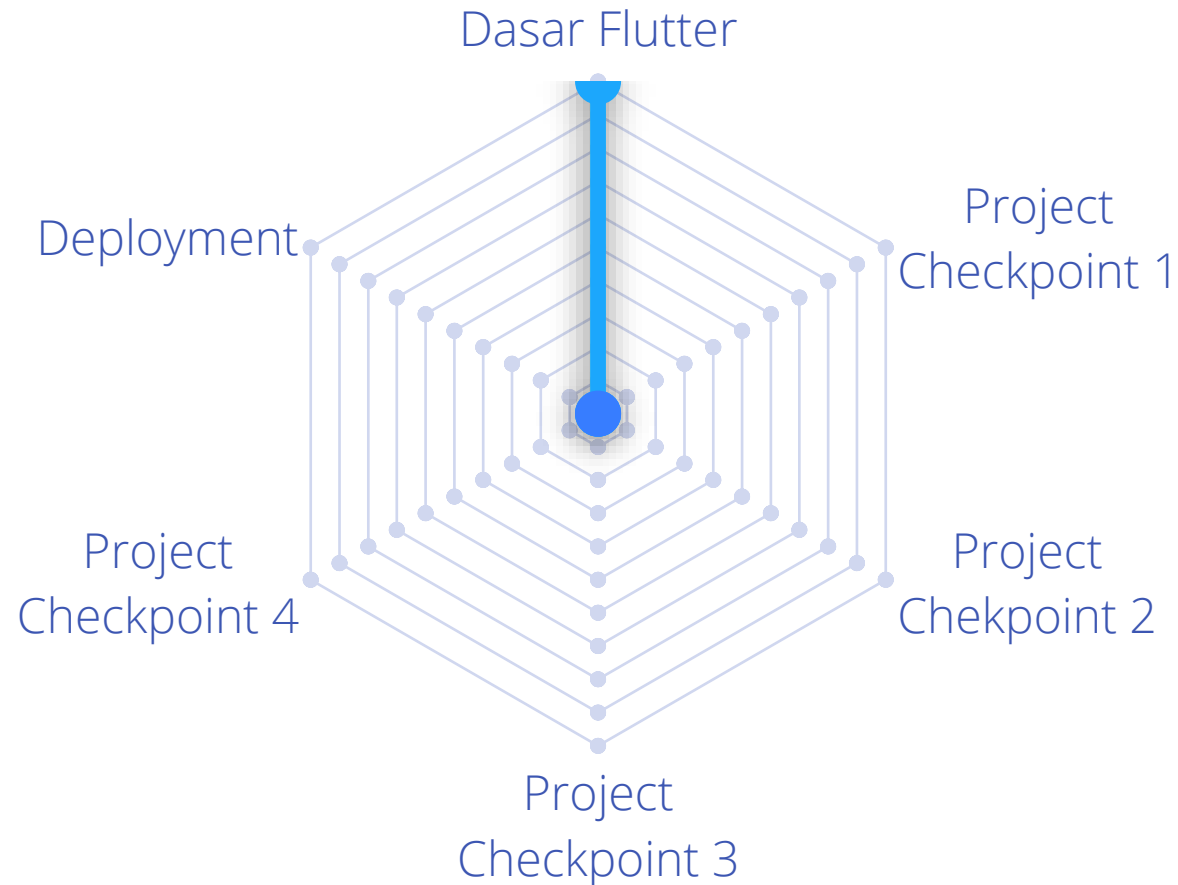
1. **dependencies:**
2. **plugin1:**
3. **git:**
4. **url:** `git://github.com/flutter/plugin1.git`

Atau path direktori package tersebut yang tersimpan secara offline di komputer Anda.

1. **dependencies:**
2. **plugin1:**
3. **path:** `../plugin1/`



Selamat! Kita telah sampai di tahap pertama Android Development Menggunakan Flutter



FLUTTER

TERIMA KASIH

Sampai jumpa di pertemuan berikutnya!

IT CLUB – Android Development