

Pertemuan I - Pengenalan Mobile Programming dan Setup Lingkungan

1. Ringkasan

Komponen	Uraian
Tujuan Praktikum	<ol style="list-style-type: none">1. Menjelaskan Flutter sebagai framework mobile cross-platform.2. Menginstal Flutter SDK, Android Studio, Android SDK, serta plugin pendukung.3. Menjalankan aplikasi Flutter pertama di emulator dan smartphone Android.
Dasar Teori	<ol style="list-style-type: none">1. Flutter = framework UI Google berbasis Dart.2. Mendukung Android, iOS, Web,3. Desktop dengan 1 basis kode.4. Komponen inti: Flutter SDK, Android Studio/VS Code, Android SDK, emulator/perangkat nyata.5. Perintah penting: flutter doctor, flutter --version, flutter create.
Alat & Bahan	<ol style="list-style-type: none">1. OS: Windows 10/terbaru2. Flutter SDK (± 2 GB)3. Android Studio (± 1 GB)4. Android SDK + Command-line Tools5. Kabel USB untuk uji di perangkat nyata6. AVD (emulator) di Android Studio7. Koneksi internet stabil

2. Langkah Kerja

2.1. Instalasi Flutter SDK (Windows)

- a. Unduh Flutter SDK dari situs resmi (versi stabil).

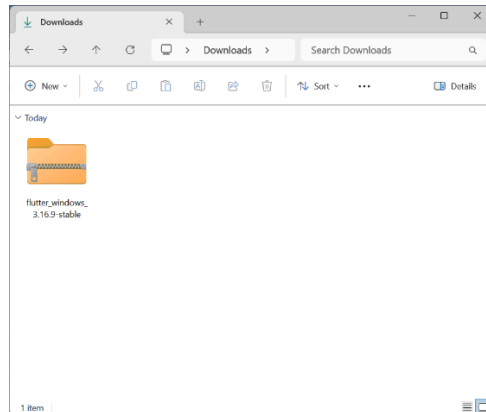
Stable channel (Windows)

Select from the following scrollable list:

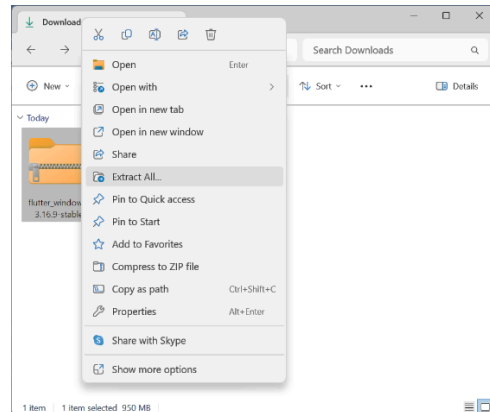
Flutter version	Architecture	Ref	Release Date	Dart version	Provenance
3.16.9	x64	4145645	1/26/2024	3.2.6	Attestation bundle
3.16.8	x64	67457e6	1/18/2024	3.2.5	Attestation bundle
3.16.7	x64	ef1af02	1/12/2024	3.2.4	Attestation bundle
3.16.6	x64	46787ee	1/11/2024	3.2.3	Attestation bundle
3.16.5	x64	78666c8	12/21/2023	3.2.3	Attestation bundle

Gambar 1. Link download Flutter

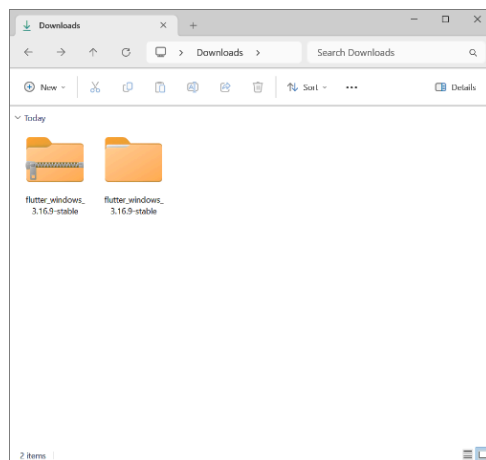
- b. File instalasi yang terkompres perlu diekstrak terlebih dahulu dengan memilih *Extract All* melalui klik kanan pada *file*. Selanjutnya, ubah nama *folder* hasil ekstraksi menjadi “flutter” menggunakan menu *Rename* (Gambar 2.2).



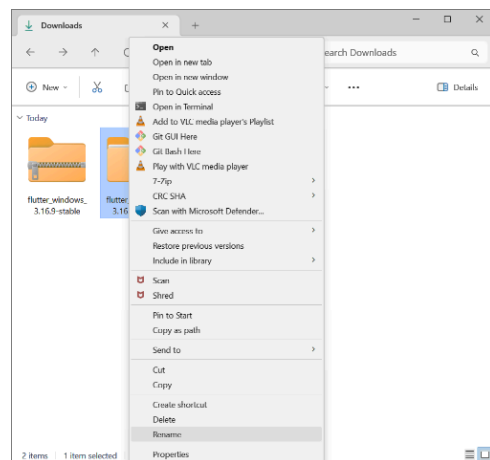
(a) File framework Flutter



(b) Ekstrak framework Flutter



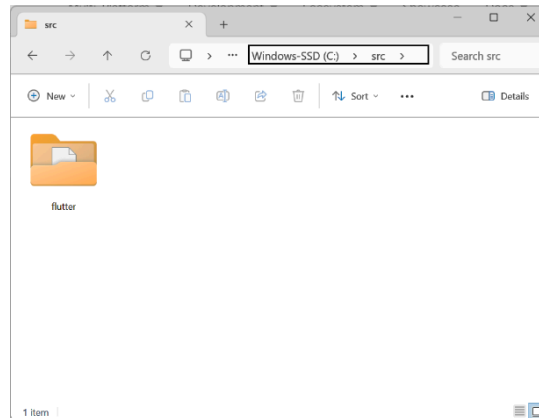
(c) Folder hasil ekstrak



(d) Rename folder

Gambar 2. Ekstrak file instalasi dan rename folder hasil ekstrak

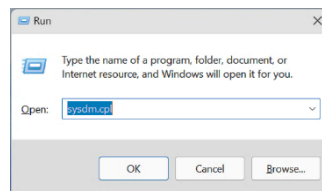
- c. Pindahkan folder flutter ke *C:/src/flutter* (buat folder *C:/src* jika belum ada).



Gambar 3. Hirarki folder Flutter

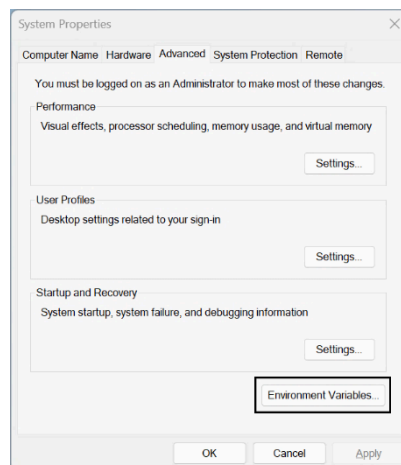
d. Tambahkan Path ke Environment Variables:

- Gunakan kombinasi `• + R` pada keyboard untuk membuka Run. Ketik perintah `“sysdm.cpl”` tanpa tanda petik, lalu tekan Enter atau klik OK (Gambar 2.4).



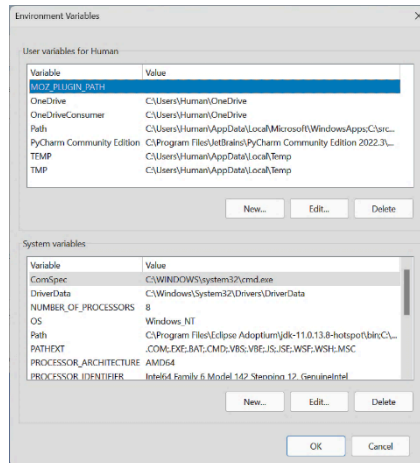
Gambar 4. Membuka system properties

- Kotak dialog *System Properties* akan muncul. Lanjutkan dengan memilih tab *Advanced* agar tampilannya sesuai Gambar 2.5.



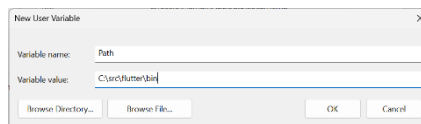
Gambar 5. Pengaturan environment variables

- Tekan tombol *Environment Variables*, maka jendela konfigurasi akan ditampilkan (Gambar 2.6). Pada bagian *User variables for [username]*, pastikan tersedia variabel dengan nama *Path*.



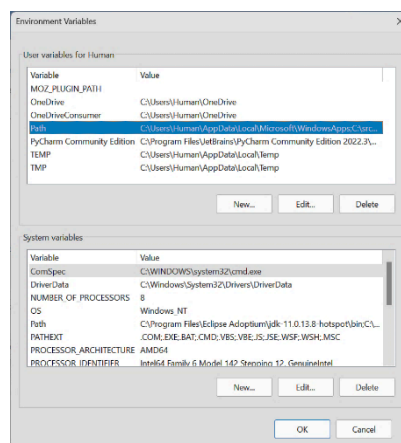
Gambar 6. Environment variables

- Jika variabel *Path* tidak ditemukan, tambahkan variabel baru melalui tombol *New*. Masukkan *Variable name* = *Path* dan *Variable value* = *C:\src\flutter\bin*, sebagaimana ditunjukkan pada Gambar 2.7. Simpan dengan klik *OK*, kemudian tutup jendela *Environment Variables* dan *System Properties* dengan menekan *OK*.



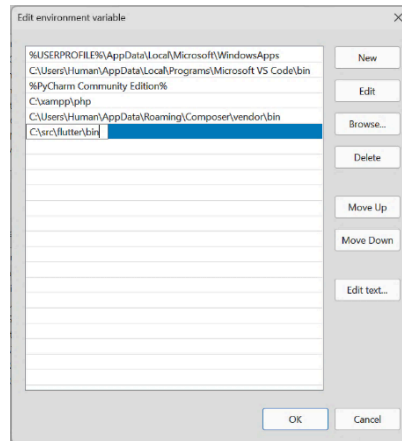
Gambar 7. Nama variabel dan value baru

- Apabila variabel *Path* sudah tersedia, cukup tambahkan nilai baru dengan memilih variabel tersebut, lalu klik tombol *Edit* (Gambar 2.8).



Gambar 8. Menambahkan value variable “Path”

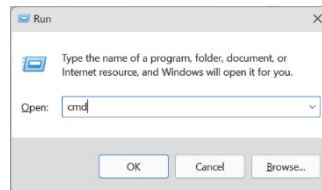
- Selanjutnya muncul jendela *Edit Environment Variable*. Tambahkan nilai baru dengan memilih *New*, lalu masukkan *C:\src\flutter\bin* pada kolom input (Gambar 2.9). Akhiri dengan klik *OK*, lalu klik *OK* kembali untuk menutup jendela *Environment Variables* dan *System Properties*.



Gambar 9. Menambahkan value pada variabel yang telah ada

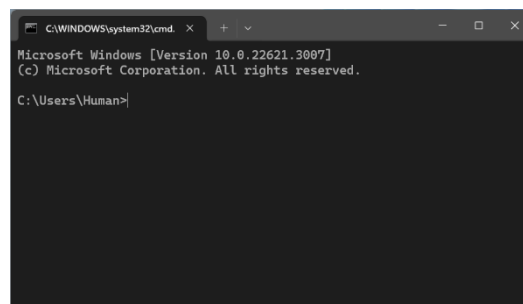
e. Langkah berikutnya adalah menguji hasil instalasi framework Flutter.

- Buka Command Prompt dengan menekan kombinasi tombol `• + R`, lalu ketik `cmd` (Gambar 2.10)..



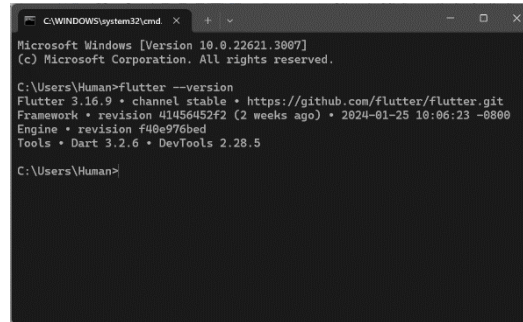
Gambar 10. Membuka DOS prompt

- Tekan *Enter* atau klik tombol *OK*, maka jendela *Command Prompt* akan terbuka sebagaimana ditunjukkan pada Gambar 2.11.



Gambar 11. Jendela DOS prompt

- Di dalam *Command Prompt*, jalankan perintah `flutter --version` dan tekan *Enter*. Jika hasilnya menampilkan informasi versi Flutter seperti Gambar 2.12, berarti instalasi telah berhasil.



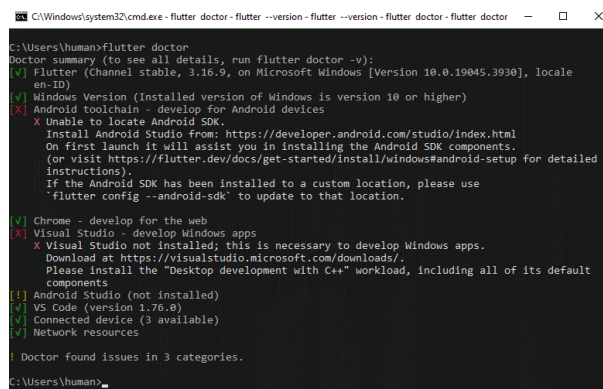
```
C:\WINDOWS\system32\cmd
Microsoft Windows [Version 10.0.22621.3807]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Human>flutter --version
Flutter 3.16.9 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 4155452f2 (2 weeks ago) • 2024-01-25 10:06:23 -0800
Engine • revision f40e976bed
Tools • Dart 3.2.6 • DevTools 2.28.5

C:\Users\Human>
```

Gambar 12. Tampilan versi Flutter

- Untuk memastikan seluruh perangkat lunak pendukung Flutter telah terpasang, ketik perintah flutter doctor pada Command Prompt (Gambar 2.13).



```
C:\Windows\system32\cmd.exe - flutter doctor - flutter --version - flutter --version - flutter doctor - flutter doctor
C:\Users\human>Flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.9, on Microsoft Windows [Version 10.0.19045.3930], locale en-ID)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[X] Android toolchain - develop for Android devices
    X 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
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      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)
[✓] VS Code (version 1.76.0)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 3 categories.

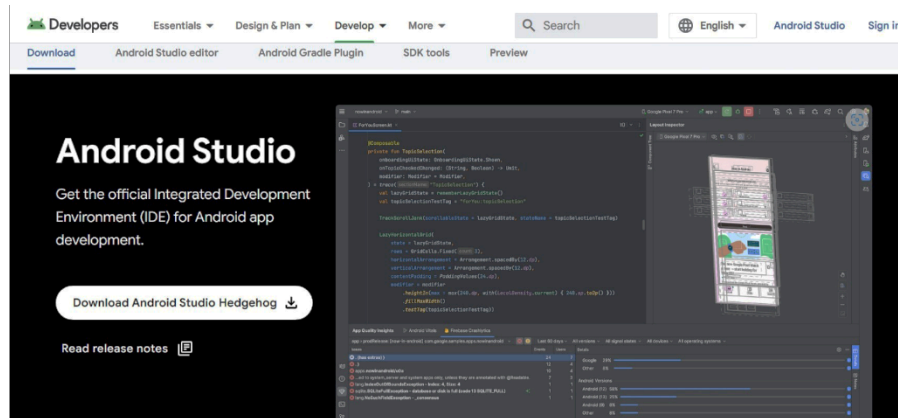
C:\Users\human>
```

Gambar 13. Daftar keperluan software

Gambar tersebut memperlihatkan bahwa Flutter masih memerlukan Android SDK dan Android Studio untuk mendukung pengembangan aplikasi mobile. Sedangkan, Visual Studio tidak wajib diinstal karena pengembangan aplikasi mobile dapat dilakukan dengan Android Studio. Instalasi Visual Studio dengan pustaka C++ hanya diperlukan untuk pengembangan aplikasi desktop berbasis Windows.

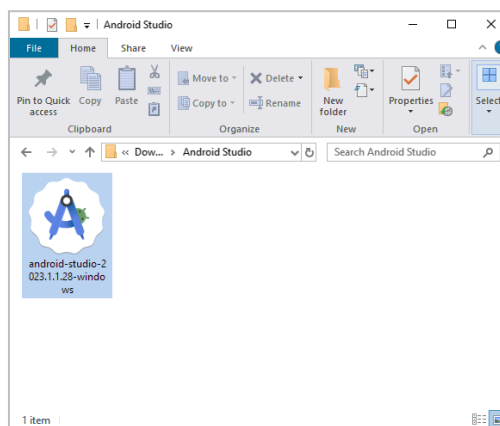
2.2. Instalasi Android Studio & SDK

- a. Unduh file instalasi Android Studio melalui website resminya (Gambar 2.14). Klik Download, setuju syarat penggunaan, lalu tunggu hingga selesai. Ukuran file sekitar 1 GB.



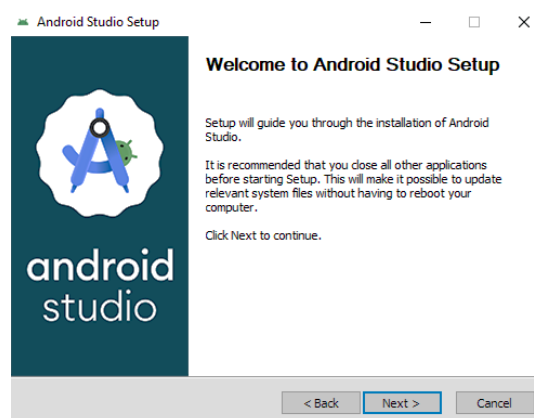
Gambar 14. Halaman web Android Studio

- Klik ganda pada ikon file instalasi (Gambar 2.15), lalu ikuti instruksi yang ditampilkan pada layar.



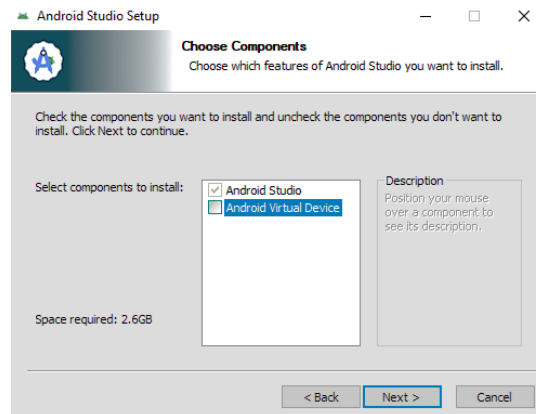
Gambar 15. Icon file instalasi

- Program instalasi menampilkan jendela sambutan “Selamat Datang di Pemasangan Android Studio” sebagaimana Gambar 2.16. Klik *Next* untuk melanjutkan ke tahap berikutnya.



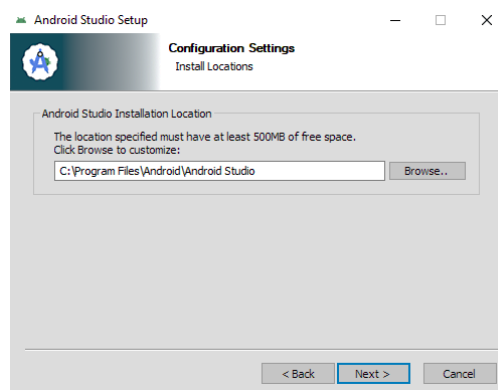
Gambar 16. Jendela sambutan instalasi

- Hilangkan centang pada opsi Android Virtual Device, kemudian lanjutkan dengan Next (lihat Gambar 2.17).



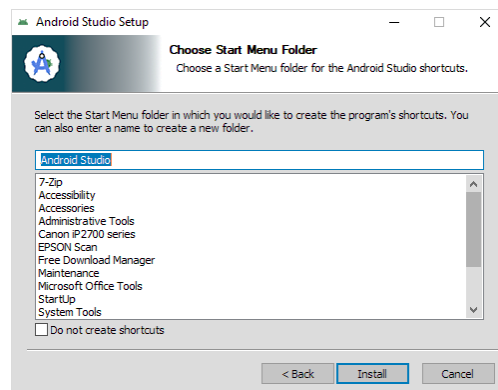
Gambar 17. Penentuan komponen yang di-install

- Pada jendela lokasi instalasi (lihat Gambar 2.18), gunakan pengaturan *default* dan lanjutkan dengan *Next*.



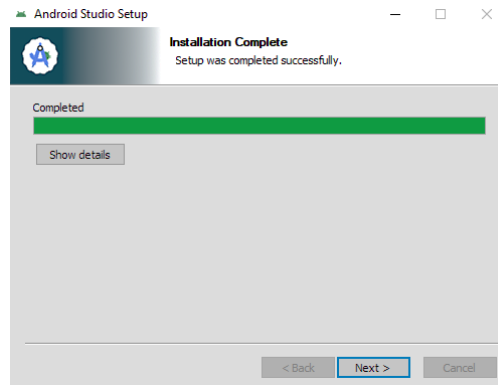
Gambar 18. Penentuan lokasi instalasi

- Pada jendela *Start Menu* (lihat Gambar 2.19), gunakan pengaturan standar dan tekan *Next*.



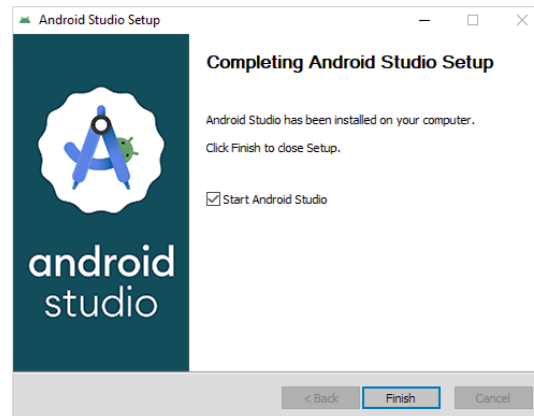
Gambar 19. Pengaturan folder menu Start

- Proses instalasi ditunjukkan melalui *progress bar* (lihat Gambar 2.20). Setelah penuh dan berstatus “*Completed*”, tekan *Next*.



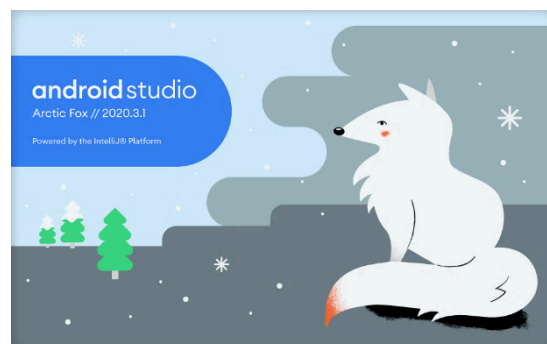
Gambar 20. Proses instalasi

- Pada jendela penyelesaian instalasi (lihat Gambar 2.21), aktifkan opsi *Start Android Studio* dan akhiri dengan menekan *Finish*.



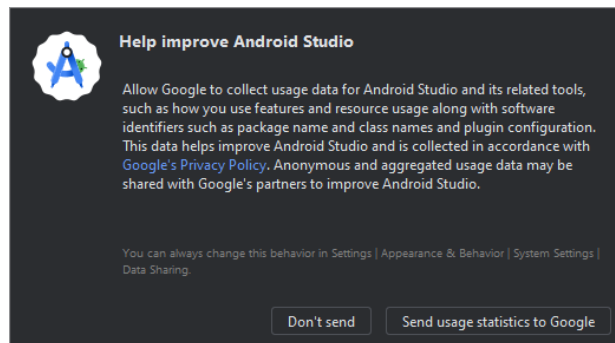
Gambar 21. Mengakhiri instalasi

- Instalasi Android Studio telah selesai, tetapi *Android SDK* masih perlu ditambahkan sebagai komponen pengembang. Setelah tombol *Finish* ditekan, Android Studio langsung dijalankan dengan *splash screen* sebagaimana Gambar 2.22.



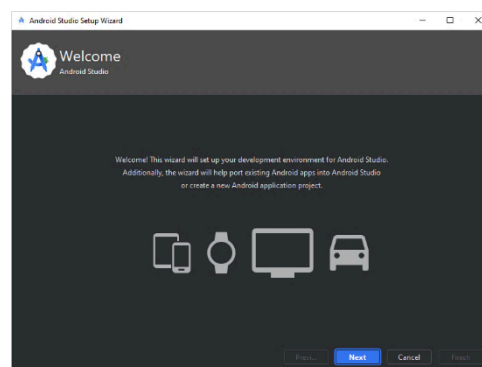
Gambar 22. Splash screen Android Studio

- c. Selanjutnya muncul dialog (Gambar 2.23) yang menawarkan pengiriman data penggunaan Android Studio. Pilih *Don't Send* bila tidak setuju, atau *Send usage statistics to Google* bila setuju.



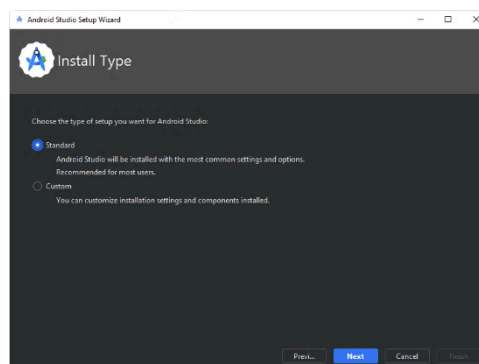
Gambar 23. Persetujuan pengiriman data penggunaan

- d. Muncul jendela sambutan *Welcome to Android Studio* (Gambar 2.24). Klik *Next* untuk melanjutkan instalasi Android SDK dengan tahapan berikut:



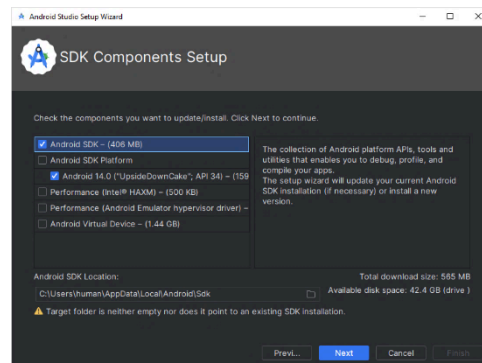
Gambar 24. *Welcome screen*

- Setelah klik tombol *Next* pada dialog *Welcome*, akan muncul jendela *Install Type* (Gambar 2.25). Pilih opsi *Custom*, lalu klik *Next* untuk melanjutkan.



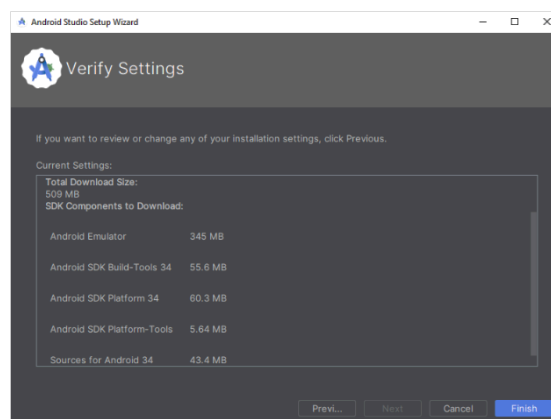
Gambar 25. Jenis Instalasi

- Kemudian ditampilkan jendela *SDK Components Setup* sebagaimana Gambar 2.26. Pastikan hanya Android SDK dan Android 14.0 yang dicentang, kemudian tekan *Next*.

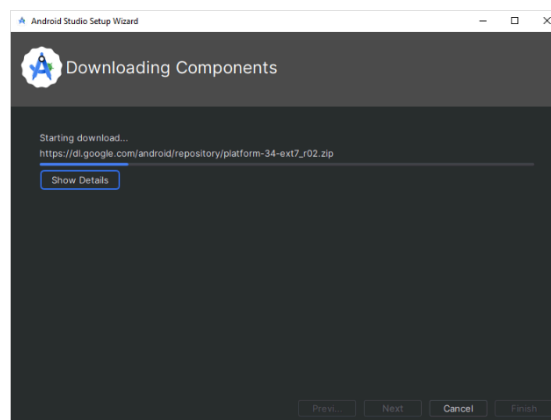


Gambar 26. Menentukan komponen yang di-install

- Berikutnya muncul jendela verifikasi konfigurasi instalasi (lihat Gambar 2.27). Tekan *Finish* untuk memulai pengunduhan sekaligus instalasi komponen, dengan progres yang terlihat pada Gambar 2.28.

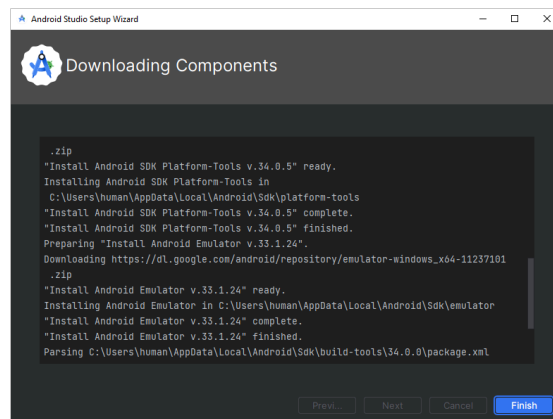


Gambar 27. Verifikasi pengaturan instalasi



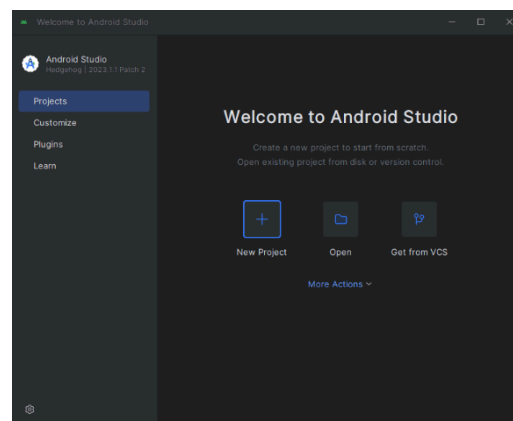
Gambar 28. Proses download dan instalasi

- Jika instalasi telah selesai, akhiri dengan menekan tombol *Finish* (lihat Gambar 2.29).”



Gambar 29. Instalasi selesai

Pada tahap ini, Android SDK telah berhasil terinstall, kemudian akan ditampilkan jendela *Welcome to Android Studio* sebagaimana Gambar 2.30.



Gambar 30. Jendela “Selamat Datang di Android Studio”

Setelah Android Studio dan Android SDK selesai diinstal, lakukan pengecekan kelengkapan software melalui *Command Prompt* dengan perintah “*flutter doctor*” (lihat Gambar 2.31). Hasil pemeriksaan menunjukkan bahwa Android Studio dan Android SDK telah terpasang, namun masih terdapat persyaratan yang belum terpenuhi, yaitu *cmdline-tools* dan lisensi Android.

```

C:\Windows\system32\cmd.exe - flutter doctor
C:\Users\human>Flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.16.9, on Microsoft Windows [Version 10.0.19045.3930], locale en-ID)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 34.0.0)
[!] X cmdline-tools component is missing
    Run 'path/to/sdkmanager --install "cmdline-tools;latest"'
    See https://developer.android.com/studio/command-line for more details.
[✗] X 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
[✗] X Visual Studio - develop Windows apps
    Visual Studio not installed; this is necessary to develop Windows apps.
    Download at https://visualstudio.microsoft.com/downloads/.
    Please install the "Desktop development with C++" workload, including all of its default components
[✓] Android Studio (version 2023.1)
[✓] VS Code (version 1.76.0)
[✓] Connected device (3 available)
[✓] Network resources

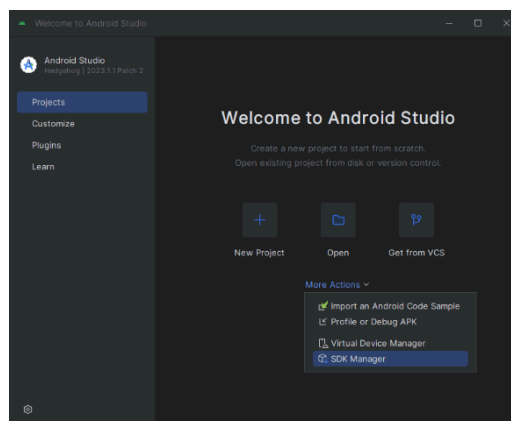
Doctor found issues in 2 categories.
C:\Users\human>

```

31. Memeriksa kelengkapan software

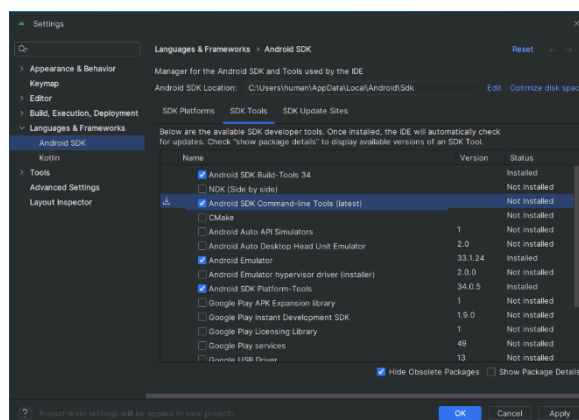
Untuk memenuhi kebutuhan *cmdline-tools*, instalasi dapat dilakukan melalui SDK Manager dengan langkah-langkah berikut:

- a. Pada jendela *Welcome to Android Studio*, klik *More Actions* kemudian pilih opsi SDK Manager (lihat Gambar 2.32).



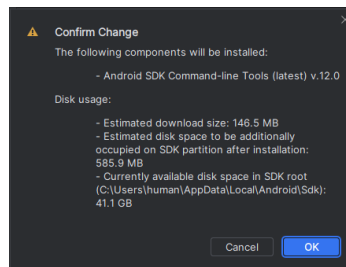
Gambar 32. Pilih menu SDK Manager

- b. Kemudian dari dialog *Settings*, buka tab *SDK Tools*, lalu aktifkan opsi *Android SDK Command-line Tools (Latest)* dengan memberi tanda cek, sebagaimana ditunjukkan pada Gambar 2.33. Setelah itu klik tombol *Apply* untuk memulai instalasi.



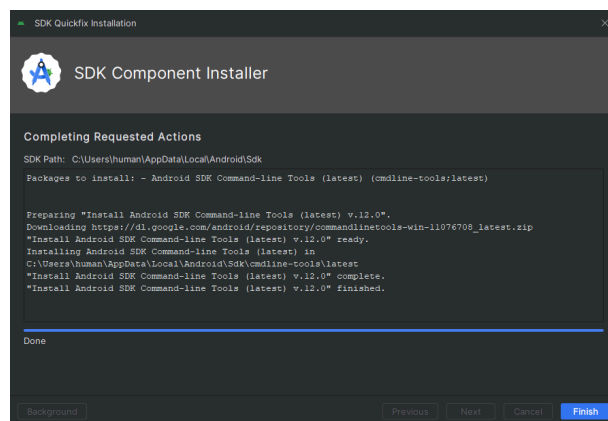
Gambar 33. Beri cek pada Android SDK Command-line Tools

- c. Jika ditampilkan dialog konfirmasi sebagaimana Gambar 2.34, tekan tombol *OK* agar proses instalasi berlanjut.



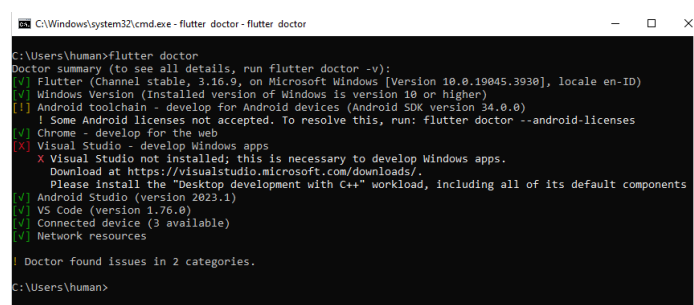
34. Dialog konfirmasi perubahan

- d. Tunggu hingga proses *download* dan *instalasi* selesai sebagaimana ditunjukkan pada Gambar 2.35, kemudian akhiri dengan menekan tombol *Finish*.



Gambar 35. Instalasi selesai

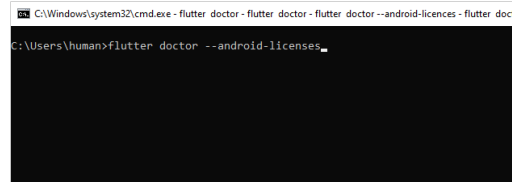
- e. Lakukan pengecekan ulang melalui perintah *flutter doctor* pada prompt DOS sebagaimana ditunjukkan pada Gambar 2.36. Hasil pemeriksaan menunjukkan bahwa kendala terkait *cmdline-tools* telah terselesaikan.



Gambar 36. Permasalahan cmdline-tools terselesaikan

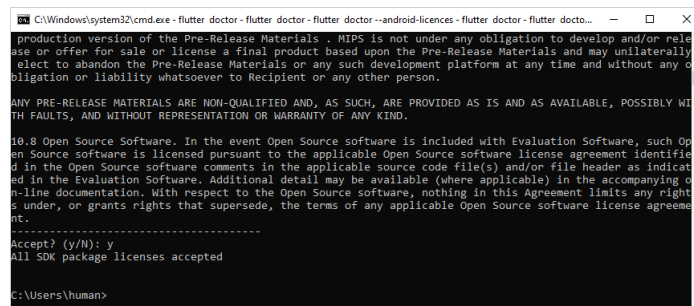
Untuk mengatasi permasalahan lisensi Android, lakukan langkah-langkah berikut:

- a. Jalankan perintah *flutter doctor --android-licenses* pada prompt DOS sebagaimana ditunjukkan pada Gambar 2.37.



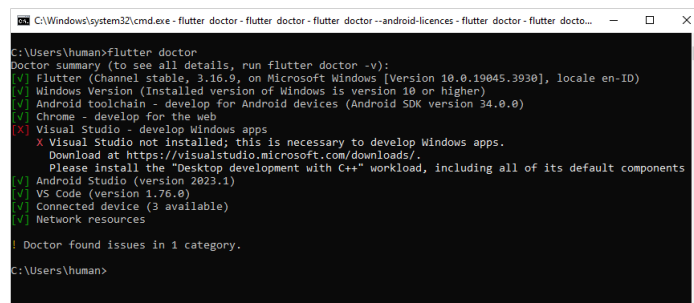
Gambar 37. Penyelesaian masalah lisensi Android

- b. Baca seluruh persyaratan yang ditampilkan, kemudian setuju dengan menekan tombol y pada keyboard (lihat Gambar 2.38).



Gambar 38. Menyetujui persyaratan dan ketentuan

- c. Setelah semua lisensi disetujui, verifikasi kembali menggunakan perintah *flutter doctor*. Hasil pada Gambar 2.39 menunjukkan bahwa seluruh persyaratan telah terpenuhi.

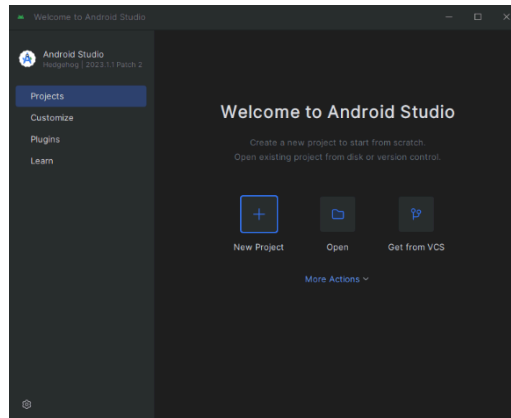


Gambar 39. Semua kelengkapan terpenuhi

Catatan: Instalasi Visual Studio tidak bersifat wajib karena fungsinya dapat digantikan oleh Android Studio. Visual Studio beserta pustaka C++ diperlukan apabila pengembangan aplikasi dilakukan untuk platform desktop berbasis Windows.

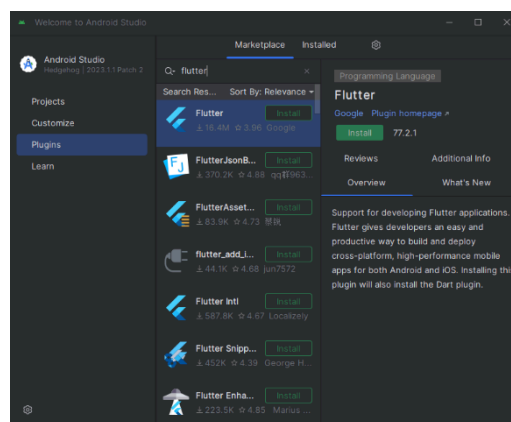
2.3. Instalasi Plugin Flutter di Android Studio

- a. Jalanlan Android Studio



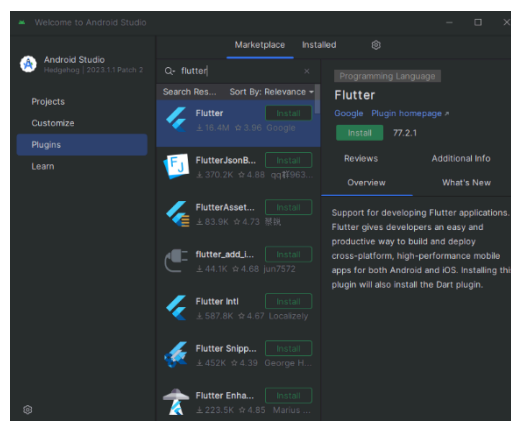
Gambar 40. Menu utama Android Studio

b. Pilih menu *plugin* di bagian kiri yang ditampilkan pada jendela utama.



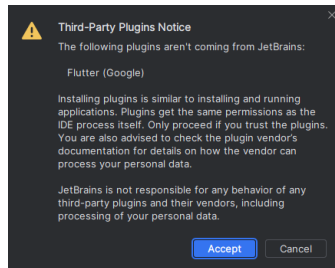
Gambar 41. Menu plugin

c. Lakukan pencarian dengan kata kunci “flutter”. Klik pada hasil pencarian plugin bernama “Flutter”



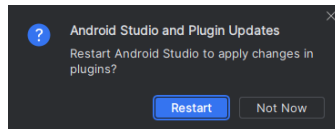
Gambar 42. Pencarian *plugin* Flutter

d. Kemudian klik tombol *install*. Tunggu sampai prosesnya sampai selesai. Pada jendela dialog yang ditampilkan klik “*Accept*”.



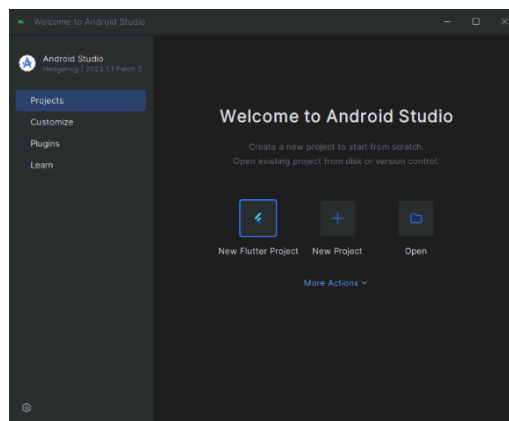
Gambar 43. Peringatan plugin dari pihak ketiga

e. Kemudian klik tombol “*restart*” pada dialog untuk *restart* Android Studio.



Gambar 44. Restart Android Studio

f. Jika *install plugin* Flutter berhasil maka di jendela utama Android Studio terdapat menu “*New Flutter Project*”



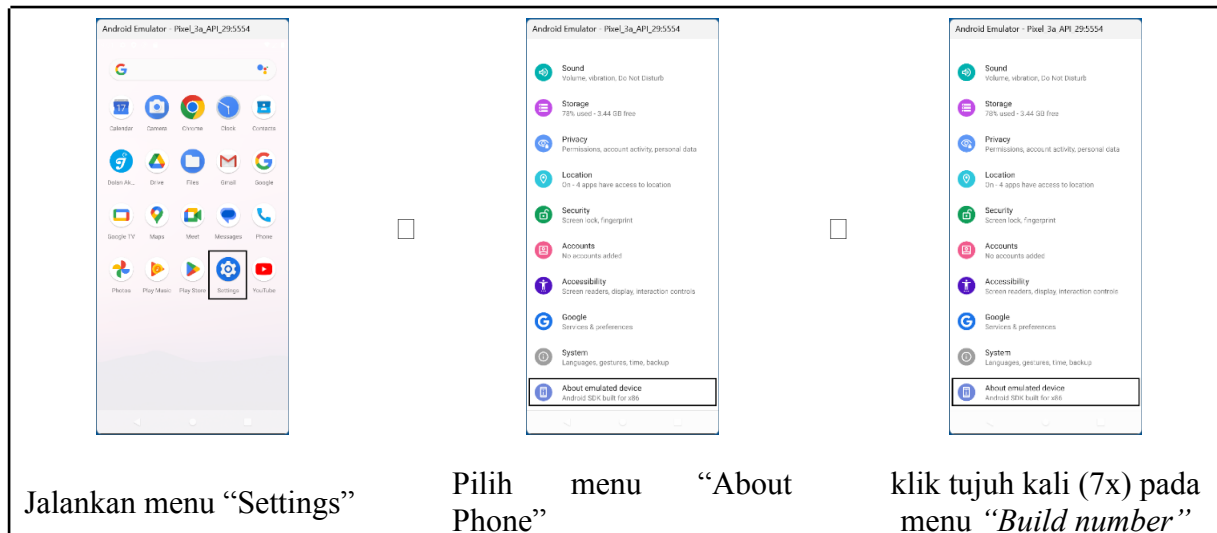
Gambar 45. Menu Flutter Project

2.4. Menyiapkan Perangkat Uji

Pada proses pengembangan aplikasi, perangkat uji diperlukan untuk memastikan program berjalan sesuai rancangan.

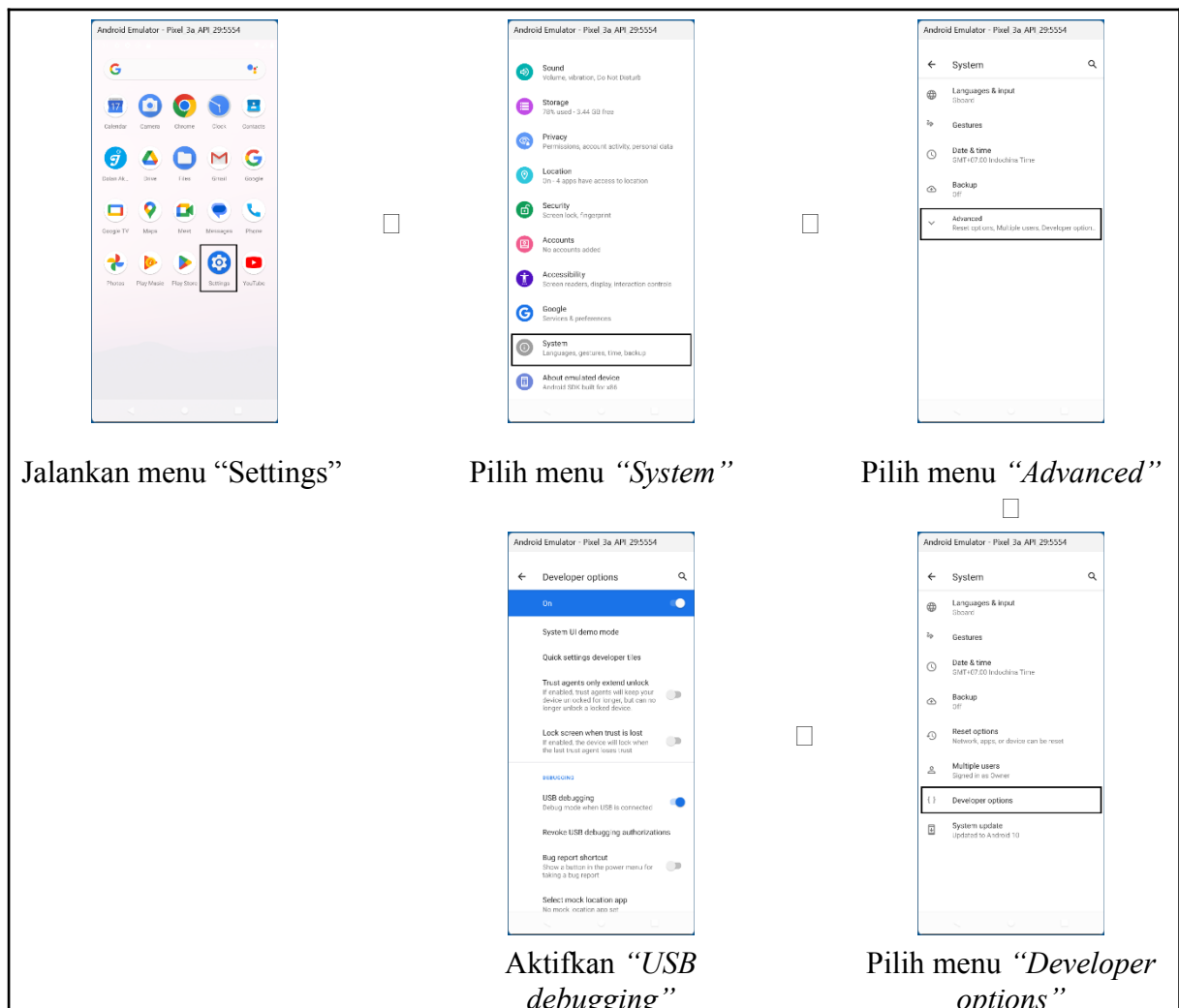
2.4.1. Menggunakan *Smartphone* Android

Smartphone perlu dikonfigurasi ke mode pengembang agar dapat digunakan untuk menjalankan program dari komputer. Proses aktivasi “*Developer mode*” ditunjukkan pada Gambar 46.



Gambar 46. Pengaktifan “Developer mode” pada *smartphone*

Setelah “Developer mode” diaktifkan, menu ‘USB debugging’ dapat dibuka dan diaktifkan sesuai tahapan pada Gambar 47.

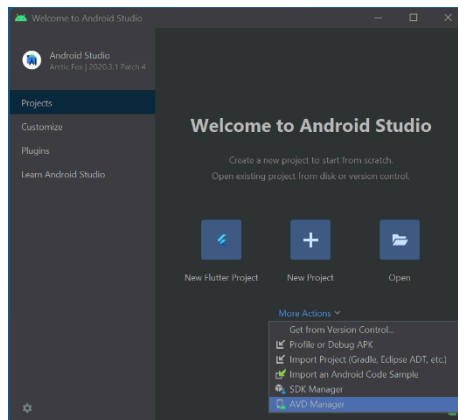


Gambar 47. Pengaktifan “USB debugging” pada *smartphone*

2.4.2. Menggunakan Emulator Android

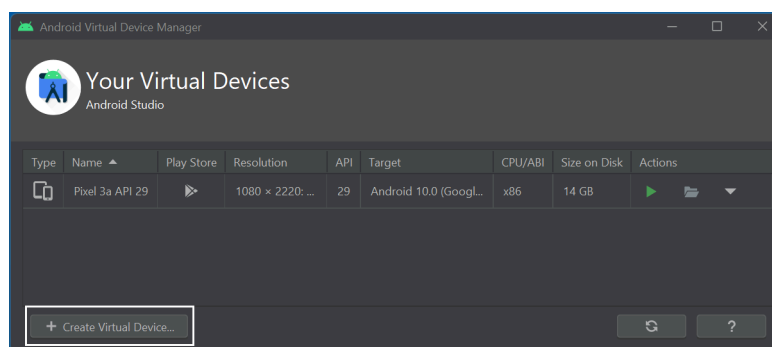
Emulator Android dapat dibuat melalui *AVD Manager* yang tersedia pada Android Studio dengan tahapan sebagai berikut:

d. Pada menu utama Android Studio, pilih *More Actions*, lalu klik *AVD Manager*.



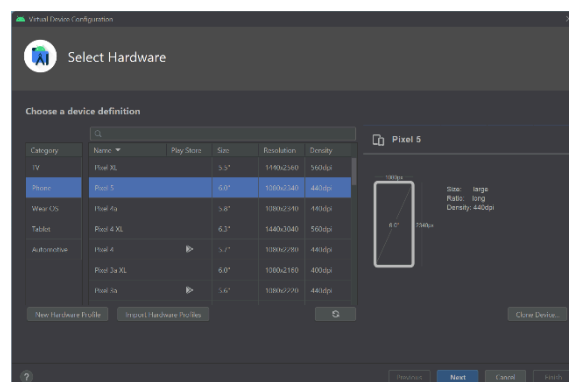
Gambar 48. Pilih menu AVD Manager

e. Klik tombol *Create Virtual Device* pada jendela *AVD Manager*.



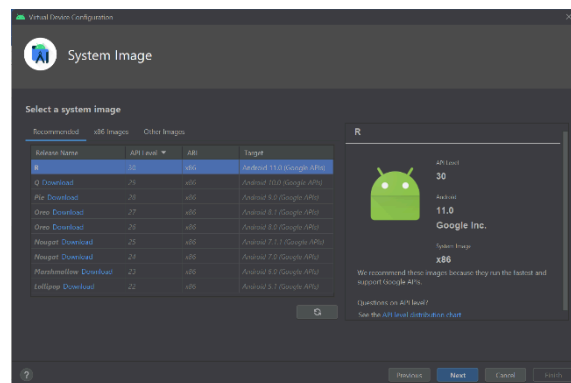
Gambar 49. Klik tombol “*Create Virtual Device*”

f. Di jendela *Select Hardware*, pilih kategori *Phone*, tentukan tipe perangkat yang diinginkan, misalnya Pixel 5, lalu klik tombol *Next* untuk melanjutkan.



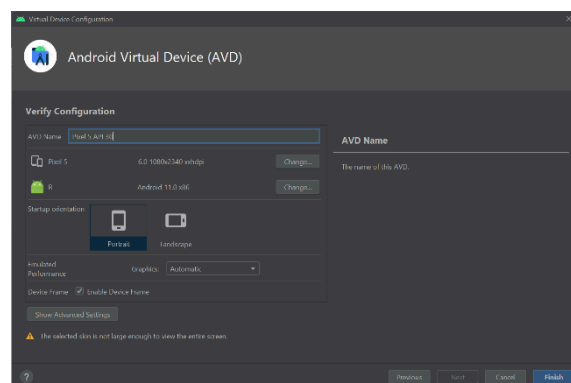
Gambar 50. Pilihan *device*

- g. Langkah selanjutnya adalah menentukan sistem operasi emulator. Pastikan tab *Recommended* terpilih, kemudian pilih Android 11.0 (R) pada kolom *Release Name* dan tekan *Next*.

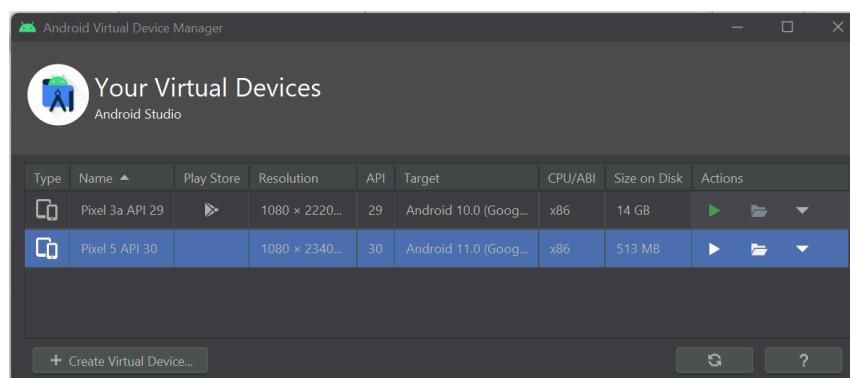


Gambar 51. Pilihan sistem operasi

- h. Pada jendela verifikasi konfigurasi, biarkan seluruh pengaturan tetap dalam kondisi *default*. Sebagai contoh, AVD Name otomatis terisi Pixel 5 API 30. Klik tombol *Finish* dan tunggu hingga proses selesai. Emulator baru akan muncul pada daftar di *AVD Manager*.

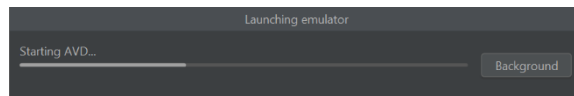


Gambar 52. Verifikasi konfigurasi

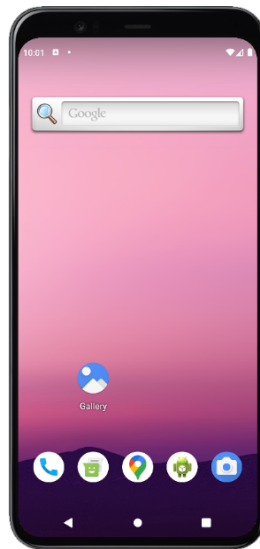


Gambar 53. Daftar device emulator

- i. Emulator dapat dijalankan melalui jendela *AVD Manager* dengan menekan ikon segitiga pada kolom Actions. Setelah jendela peluncuran tampil (Gambar 2.54), tunggu hingga emulator selesai dimuat.



Gambar 54. Peluncuran emulator

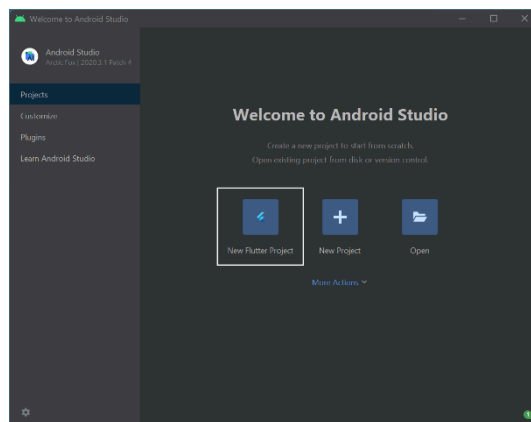


Gambar 55. Emulator android

2.5. Pengujian

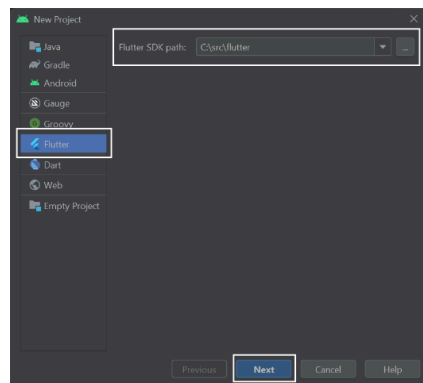
Pengujian lingkungan pengembangan dilakukan dengan memanfaatkan aplikasi sederhana hasil template bawaan Flutter saat inisialisasi proyek baru. Aplikasi ini akan dijalankan pada emulator Android serta perangkat *smartphone*.

- a. Dari jendela utama Android Studio, pilih menu *New Flutter Project*.



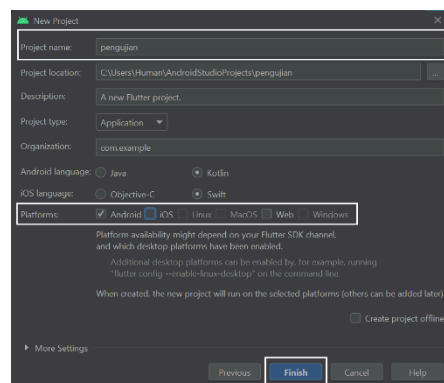
Gambar 56. *New Flutter Project*

- b. Pilih opsi Flutter, lalu tentukan lokasi folder “Flutter” sesuai dengan konfigurasi instalasi sebelumnya, misalnya di “C:\src\flutter”. Tekan *Next* untuk melanjutkan.



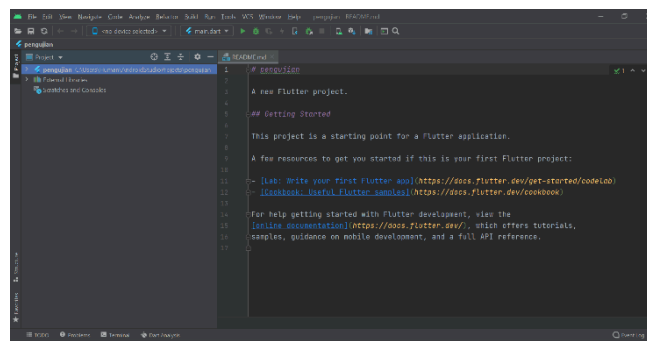
Gambar 57. Konfigurasi lokasi folder Flutter

- c. Isi kolom *Project name* dengan “pengujian”, biarkan *Project location* pada nilai *default*, dan aktifkan hanya *platform* Android. Klik *Finish*, kemudian tunggu hingga Android Studio selesai menyiapkan *template* proyek.

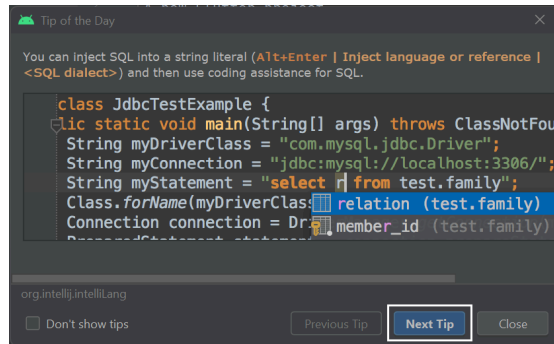


2.58. Pengisian nama proyek dan platform

- d. Setelah proses selesai, jendela kerja Android Studio akan ditampilkan. Jika muncul jendela *Tip of the Day*, klik *Close* untuk menutupnya.

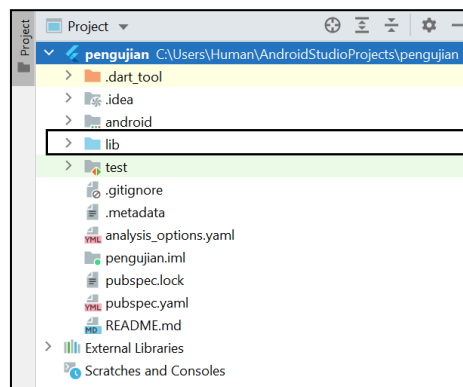


Gambar 59. Jendela kerja Android Studio



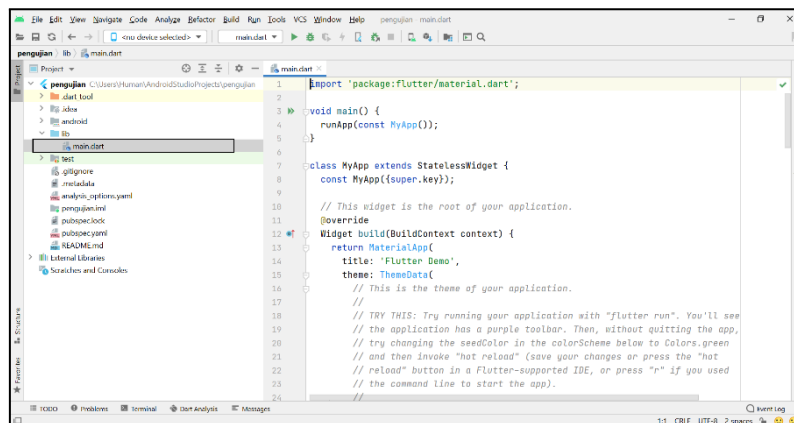
60. Jendela *Tip of the Day*

Tutup juga *file* Readme.md dengan klik ikon X. Selanjutnya, perluas struktur *folder* proyek “Pengujian” pada panel kiri.



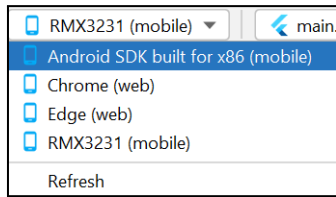
Gambar 2.61. Struktur file dan folder program

Untuk membuka kode program, klik *folder* “lib”, lalu klik ganda file “main.dart” sehingga kode tampil di *text editor*.



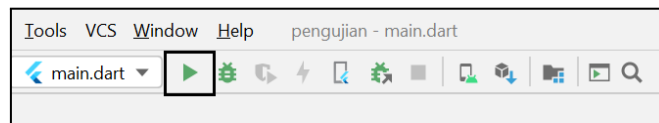
Gambar 62. Kode program aplikasi

- e. Jalankan program pada emulator Android. Pastikan emulator sudah aktif; jika belum, lakukan langkah pada Subbab 2.4 poin f. Pilih device emulator melalui menu dropdown Flutter Device Selection (Gambar 2.63).



Gambar 63. Pilihan *device* emulator

Kemudian klik ikon ► untuk menjalankan aplikasi (Gambar 2.64). Proses pertama kali memerlukan waktu beberapa menit. Setelah berhasil, tampilan aplikasi akan muncul pada emulator Android.



Gambar 2.64. Menjalankan program

Uji aplikasi dengan menekan tombol Tambah (+) sehingga angka di tengah layar bertambah. Hal ini menunjukkan aplikasi berhasil dijalankan di emulator.



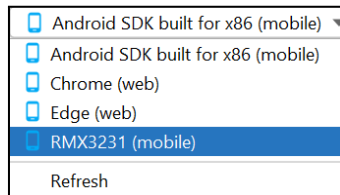
Gambar 65. Tampilan aplikasi di emulator Android

- f. Selanjutnya, jalankan aplikasi pada perangkat nyata (smartphone Android). Pastikan Developer mode dan USB debugging telah diaktifkan (lihat Gambar 2.46 dan 2.47). Hubungkan smartphone ke komputer menggunakan kabel USB. Jika muncul dialog izin USB debugging di layar perangkat (Gambar 2.66), pilih Allow.



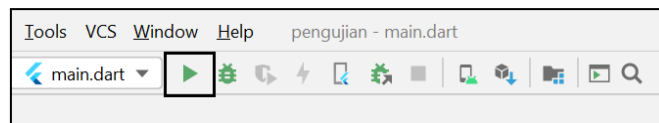
Gambar 66. Ijin akses USB debugging

- g. Pada menu *Flutter Device Selection*, pastikan perangkat *smartphone* yang terhubung terpilih dengan nama sesuai model *smartphone* (Gambar 2.67).

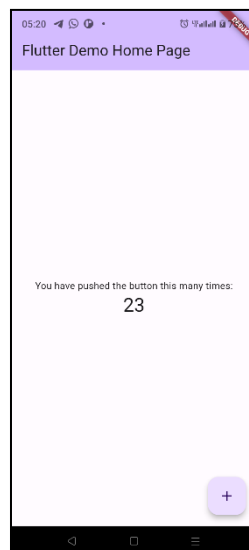


Gambar 2.67. Pilihan device *smartphone*

- h. Klik ikon ► untuk menjalankan program (Gambar 2.68). Tunggu beberapa saat hingga aplikasi berhasil ditampilkan pada layar *smartphone* Android (Gambar 2.69).



Gambar 68. Menjalankan program



Gambar 69. Tampilan aplikasi di *smartphone* Android

Dengan demikian, aplikasi berhasil dijalankan baik pada emulator maupun perangkat nyata. Hasil ini menunjukkan bahwa instalasi lingkungan pengembangan telah berhasil dan siap digunakan untuk pengembangan aplikasi selanjutnya.

3. Contoh Kode Program

Template bawaan Flutter berupa aplikasi Counter dapat dimodifikasi menjadi aplikasi sederhana Hello World. Lakukan perubahan kode program dengan langkah-langkah berikut:

- Buka file `main.dart` pada folder `lib` proyek.
- Hapus seluruh kode bawaan template Counter.

c. Ganti dengan kode berikut:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  @override
11  Widget build(BuildContext context) {
12    return MaterialApp(
13      home: Scaffold(
14        appBar: AppBar(
15          title: const Text("Praktikum Flutter"),
16        ),
17        body: const Center(
18          child: Text(
19            "Hello World",
20            style: TextStyle(fontSize: 24),
21          ),
22        ),
23      ),
24    );
25  }
26 }
```

4. Tugas Praktikum

Untuk memperkuat pemahaman mengenai proses instalasi dan konfigurasi lingkungan pengembangan, pada bagian ini dilakukan beberapa latihan sederhana menggunakan kode program Flutter. Latihan ini bertujuan untuk membiasakan penggunaan proyek Flutter, melakukan perubahan kode, serta menguji aplikasi baik di emulator maupun pada perangkat Android nyata. Selain itu, dilakukan pula pengamatan waktu eksekusi awal untuk mengetahui perbedaan performa antara emulator dan perangkat fisik.

a. Menjalankan Program

- Jalankan aplikasi menggunakan kode program pada Sub Bab 3.
- Pastikan aplikasi dapat berjalan tanpa error.

b. Modifikasi Kode Program

- Ubah teks pada AppBar menjadi “Program Pertamaku”.
- Ganti teks “Hello World” dengan nama masing-masing.
- Jalankan kembali program untuk melihat hasil perubahan.

c. Pengujian pada Perangkat

- Jalankan aplikasi pada emulator Android.
- Uji aplikasi pada smartphone nyata melalui USB debugging.
- Pastikan aplikasi dapat berjalan normal pada kedua perangkat.

d. Pengamatan Waktu Eksekusi

- Catat waktu yang dibutuhkan aplikasi untuk pertama kali dijalankan hingga tampil di layar.
- Bandingkan hasil waktu eksekusi antara emulator dan smartphone nyata.

7. Format Laporan Praktikum

Dari hasil praktikum ini, buatlah laporan dengan format sebagai berikut:

a. Cover

Memuat identitas laporan meliputi judul praktikum, nama, NIM, kelas, dan tanggal pelaksanaan.

b. Tujuan

Berisi uraian singkat mengenai tujuan praktikum sesuai dengan topik yang telah ditentukan dalam modul.

c. Langkah Kerja Singkat

Menyajikan ringkasan tahapan praktikum yang dilakukan, ditulis secara runtut dan jelas tanpa perlu detail teknis yang berulang dari modul.

d. Screenshot

Menampilkan bukti hasil praktikum, misalnya tampilan aplikasi pada emulator maupun perangkat nyata, lengkap dengan keterangan singkat.

e. Kesimpulan

Menguraikan ringkasan hasil praktikum, pemahaman yang diperoleh, serta pengalaman yang didapat dari proses pengujian dan modifikasi program.