

PEMROGRAMAN ANDROID DENGAN LADYBUG

Penulis

**Marwiyah Hasibuan; Ali Akbar Ritonga;
Irmayanti; Deci Irmayani**

Pemrograman Android dengan Ladybug



UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

Pemrograman Android dengan Ladybug

Marwiyah Hasibuan;

Ali Akbar Ritonga;

Irmayanti;

Deci Irmayani;



Pemrograman Android dengan Ladybug

Marwiyah Hasibuan; Ali Akbar Ritonga; Irmayanti; Deci Irmayani;

Desain Cover :
Musthafa Haris Munandar

Sumber :
<https://munandar.yayasanmmi.com/pemrograman-android>

Tata Letak :
Deci Irmayani

Proofreader :
Ali Akbar Ritonga

Ukuran :
Jml hal judul: 1, Jml hal isi naskah: 81, Uk: 15x23 cm

ISBN :
978-634-04-1799-9

Cetakan Pertama :
Juli 2025

Hak Cipta 2025, Pada Penulis

Isi diluar tanggung jawab percetakan

Copyright © 2025 by Yayasan MMI
All Right Reserved

Hak cipta dilindungi undang-undang
Dilarang keras menerjemahkan, memfotokopi, atau
memperbanyak sebagian atau seluruh isi buku ini
tanpa izin tertulis dari Penerbit.

PENERBIT YAYASAN MUNANDAR MEMBANGUN INDONESIA
Jl. Pasar Banjar Dusun XVI Desa Simpang Empat, Asahan, Sumatera Utara 21271
Telp/Wa : 082363080181
<https://munandar.yayasanmmi.com/>
E-mail: mmipublisher@yayasanmmi.com

PRAKATA

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya buku ini yang berjudul "Pemrograman Android dengan Ladybug" dapat diselesaikan dengan baik. Buku ini disusun sebagai panduan praktis sekaligus referensi akademik bagi mahasiswa, dosen, maupun praktisi yang ingin memahami proses pengembangan aplikasi Android dengan pendekatan visual menggunakan Ladybug.

Buku ini disusun secara sistematis, dimulai dari pengenalan antarmuka Ladybug, perancangan tampilan, logika aplikasi, hingga implementasi fitur-fitur umum dalam aplikasi Android. Setiap bab dilengkapi dengan contoh kasus dan latihan yang diharapkan dapat membantu pembaca memahami dan mempraktikkan materi secara langsung.

Penulis menyadari bahwa buku ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi penyempurnaan edisi selanjutnya. Semoga buku ini dapat memberikan manfaat nyata bagi para pembaca dalam memperluas wawasan dan keterampilan di bidang pemrograman Android.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan buku ini. Semoga karya sederhana ini menjadi kontribusi positif bagi dunia pendidikan dan pengembangan teknologi di Indonesia.

Hormat Kami,

Penulis

DAFTAR ISI

PRAKATA	vi
Daftar Isi.....	vii
BAB 1 Pengenalan Android	1
A. Pendahuluan	1
B. Sejarah Android.....	2
C. Versi Android	2
D. Fitur Utama Android	3
E. Arsitektur Android	4
F. Pasar Aplikasi Android	7
G. Tujuan Pengembangan Aplikasi Android.....	8
BAB 2 Android Studio Ladybug 2024 2024.2.1 Patch 3 .	10
A. Pendahuluan	10
B. Perangkat Pendukung Android Studio Ladybug.....	11
C. Instalasi JDK 23.....	11
D. Instalasi Android Studio Ladybug.....	12
E. Update SDK Component Installer	14
F. Update SDK Manager	15
G. Membuat Android Virtual Device.....	17
BAB 3 Menggunakan Android Studio Ladybug 2024.2.1	
Patch 3	22
A. Pendahuluan	22
B. Struktur Proyek Android.....	23
C. Membuat Aplikasi Android Pertama	26
D. Menjalankan Aplikasi dengan Android Virtual Device.....	30
E. Menjalankan Aplikasi di Perangkat Smartphone....	31

BAB 4 Layout.....	35
A. Pendahuluan	35
B. Jenis-Jenis Layout.....	35
BAB 5 Activity.....	44
A. Pendahuluan	44
B. Membuat Activitas(Aktivity) Baru.....	44
C. Intent.....	46
BAB 6 Database SQLite.....	47
A. Pendahuluan	47
B. Struktur SQLite.....	47
C. Membuat aplikasi CrudData dengan Database SQLite	50
BAB 7 Aplikasi Hand Speed.....	63
A. Pendahuluan	63
B. Membuat Aplikasi Game HandSpeed.....	63
Biografi Penulis	71
Sinopsis	73

BAB 1

Pengenalan Android

A. Pendahuluan

Setiap perangkat yang digunakan memiliki sistem operasi, mulai dari perangkat elektronik, komputer, smartphone, smartwatch, konsol game, kamera digital dan masih banyak lagi. Sistem operasi memiliki fungsi untuk menghubungkan antara hardware dengan software. Tanpa adanya sistem operasi ini, segala aktivitas yang Anda lakukan di layar komputer atau smartphone tidak dapat dilakukan.

Beragam sistem operasi yang dikembangkan oleh developer. Misalnya Windows, Linux, MacOS X, iOS, dan lainnya. Dari beragam sistem operasi tersebut Android menjadi salah satu sistem operasi yang dirancang secara khusus untuk perangkat seluler terutama layar sentuh seperti smartphone dan tablet.

Secara umum Android merupakan sistem operasi dan platform pemrograman yang dikembangkan oleh Google untuk ponsel cerdas dan perangkat seluler lainnya (seperti tablet). Android bisa berjalan di beberapa macam perangkat dari banyak produsen yang berbeda. Android menyertakan kit development perangkat lunak untuk penulisan kode asli dan perakitan modul perangkat lunak untuk membuat aplikasi bagi pengguna Android. Android juga menyediakan pasar untuk mendistribusikan aplikasi. Secara keseluruhan, Android menyatakan ekosistem untuk aplikasi seluler.

Pengertian Android menurut pmdapat (Anis Ramadhani, 2013) adalah sistem operasi untuk telepon seluler yang berbasis Linux. Selain itu Android juga merupakan perangkat lunak gratis dengan sumber terbuka, dalam artian Google memperbolehkan pengguna untuk mengembangkan sistem operasi tersebut. Android juga memiliki toko aplikasi yang bernama Google Play Store. Tentunya bagi siapa saja yang menggunakan smartphone dengan sistem Android, dapat bebas mendownload aplikasi atau game yang terdapat pada Google Play Store.

B. Sejarah Android

Sistem operasi Android pertama kali diluncurkan pada bulan September 2008, di mana Android dikembangkan oleh Open Handset Alliance yang disponsori secara komersial oleh Google. Saat itu dilakukan peluncuran Android pertama yaitu versi 1.0 pada tanggal 23 September 2008 yang dikenal dengan kode Alpha. sistem operasi ini terbilang cukup fungsional dengan menghadirkan beberapa aplikasi Google, seperti Gmail, Google Maps, kalender, dan YouTube.

Versi Android selanjutnya terus diperbaharui hingga muncul versi yang terkini yaitu Android versi 15 dirilis oleh Google pada 15 Oktober 2024. Android 15 memiliki nama kode Vanilla Ice Cream. Versi ini akan menghadirkan sejumlah pembaruan, baik dari segi pengalaman pengguna hingga sejumlah fitur lainnya. Android versi 16 adalah versi sistem operasi Android yang akan datang, rilis final Android versi 16 diperkirakan akan terjadi pada kuartal kedua tahun 2025, atau antara bulan April hingga Juni tahun 2025.

C. Versi Android

Android memiliki beberapa versi yang dirilis sejak pertama kali diluncurkan. Setiap versi Android diberi nama sesuai dengan urutan abjad, dimulai dari versi pertama dengan nama "Cupcake" dan diikuti dengan nama-nama makanan atau pencuci mulut. Berikut adalah beberapa versi Android yang lebih terkenal:

1. Android Cupcake (1.5) – Dirilis pada 2009
2. Android Donut (1.6) – Dirilis pada 2009
3. Android Eclair (2.0 - 2.1) – Dirilis pada 2009
4. Android FroYo (2.2) – Dirilis pada 2010
5. Android Gingerbread (2.3) – Dirilis pada 2010
6. Android Honeycomb (3.0 - 3.2) – Dirilis pada 2011
7. Android Ice Cream Sandwich (4.0) – Dirilis pada 2011
8. Android Jelly Bean (4.1 - 4.3) – Dirilis pada 2012
9. Android KitKat (4.4) – Dirilis pada 2013
10. Android Lollipop (5.0 - 5.1) – Dirilis pada 2014
11. Android Marshmallow (6.0) – Dirilis pada 2015
12. Android Nougat (7.0 - 7.1) – Dirilis pada 2016
13. Android Oreo (8.0 - 8.1) – Dirilis pada 2017
14. Android Pie (9.0) – Dirilis pada 2018
15. Android 10 (Q) – Dirilis pada 2019

16. Android 11 – Dirilis pada 2020
17. Android 12 – Dirilis pada 2021
18. Android 13 – Dirilis pada 2022
19. Android 14 – Dirilis pada 2023

D. Fitur Utama Android

Android terus berkembang dengan berbagai pembaruan sistem operasi yang lebih canggih dan fitur-fitur baru. Setiap versi Android diberi nama dengan urutan abjad, mulai dari Android Cupcake (1.5) hingga Android 14 yang terbaru. Setiap versi membawa pembaruan yang mencakup peningkatan kinerja, keamanan, tampilan antarmuka, dan berbagai fitur baru. Berikut ini fitur utama android:

1. Antarmuka Pengguna: Android menyediakan antarmuka yang dapat disesuaikan, memungkinkan pengguna untuk mengubah tampilan perangkat mereka dengan berbagai widget, wallpaper, tema, dan ikon.
2. Aplikasi: Android memungkinkan pengguna untuk mengunduh dan instal aplikasi dari Google Play Store. Selain itu, pengembang dapat membuat aplikasi menggunakan bahasa pemrograman Java, Kotlin, dan bahasa lainnya.
3. Multitasking: Android mendukung multitasking, yang memungkinkan pengguna untuk menjalankan beberapa aplikasi secara bersamaan.
4. Kustomisasi: Salah satu kekuatan utama Android adalah tingkat kustomisasi yang tinggi, yang memungkinkan pengguna untuk mengubah hampir setiap aspek perangkat mereka.
5. Google Services: Android memiliki integrasi yang mendalam dengan berbagai layanan Google, seperti Gmail, Google Maps, YouTube, dan Google Drive, memberikan pengalaman yang lebih terhubung dan efisien bagi penggunanya.
6. Keamanan: Android memiliki berbagai lapisan keamanan, seperti pembaruan sistem otomatis, enkripsi data, dan kontrol aplikasi yang lebih ketat. Selain itu, Play Protect juga memberikan perlindungan terhadap aplikasi berbahaya.

E. Arsitektur Android

Arsitektur Android merujuk pada cara sistem operasi Android dibangun dan disusun, termasuk pembagian tugas dan pengelolaan komponen-komponen di dalam aplikasi serta perangkat Android itu sendiri. Beberapa konsep dasar dalam arsitektur Android meliputi struktur aplikasi, lapisan sistem, dan pola desain yang digunakan untuk memastikan aplikasi Android berjalan dengan baik, efisien, dan dapat dipelihara. Berikut adalah beberapa elemen penting dalam arsitektur Android:

1. Arsitektur Aplikasi Android:

Aplikasi Android biasanya terdiri dari beberapa komponen utama yang berinteraksi satu sama lain:

- **Activities:** Komponen UI yang menampilkan layar dan memungkinkan interaksi dengan pengguna.
- **Services:** Komponen yang berjalan di latar belakang untuk melakukan tugas-tugas tanpa antarmuka pengguna.
- **Broadcast Receivers:** Komponen yang menangani pesan atau peristiwa dari sistem atau aplikasi lain.
- **Content Providers:** Komponen yang memungkinkan aplikasi untuk berbagi data dengan aplikasi lain.
- **Fragments:** Bagian dari UI dalam aplikasi yang dapat digunakan kembali dan lebih fleksibel, sering kali digunakan dalam aplikasi dengan layout yang kompleks.

2. Lapisan Sistem Android:

Android terdiri dari beberapa lapisan yang bekerja bersama untuk menyediakan pengalaman pengguna yang mulus:

- **Linux Kernel:** Menyediakan dasar bagi sistem operasi Android, termasuk pengelolaan perangkat keras seperti memori, prosesor, dan perangkat input/output.
- **Android Runtime (ART):** Tempat aplikasi dijalankan. Aplikasi Android dikompilasi ke dalam bytecode dan dijalankan di atas ART.

- **Libraries:** Android menyertakan berbagai pustaka yang menyediakan fungsionalitas untuk aplikasi, termasuk pustaka grafis, database, dan komunikasi jaringan.
- **Application Framework:** Berfungsi untuk menyediakan API dan layanan yang digunakan oleh aplikasi, seperti manajemen antarmuka pengguna, layanan lokasi, dan pengelolaan media.
- **Aplikasi:** Aplikasi Android itu sendiri, yang dibangun oleh pengembang dan berinteraksi dengan seluruh sistem.

3. Arsitektur Komponen Aplikasi:

Arsitektur aplikasi Android dapat diatur menggunakan pola desain tertentu untuk meningkatkan skalabilitas, keterbacaan, dan pemeliharaan aplikasi. Beberapa arsitektur yang sering digunakan adalah:

- **Model-View-ViewModel (MVVM):** Membagi aplikasi menjadi tiga komponen utama: Model (data), View (UI), dan ViewModel (logika presentasi). MVVM membuat kode lebih terpisah dan lebih mudah diuji.
- **Model-View-Presenter (MVP):** Serupa dengan MVVM, namun dengan pendekatan yang lebih berfokus pada logika presentasi yang dipisahkan ke Presenter.
- **Clean Architecture:** Sebuah pola yang mengatur aplikasi menjadi lapisan-lapisan yang terpisah dengan dependensi hanya mengarah dari lapisan luar ke lapisan dalam, untuk membuat aplikasi lebih modular dan mudah diuji.

4. Pengelolaan UI dan Data:

- **LiveData:** Digunakan dalam MVVM untuk mengelola data yang bisa diamati oleh komponen UI (seperti Activity atau Fragment).
- **Room Database:** Sebuah pustaka untuk mengelola data lokal menggunakan SQLite secara lebih efisien dan mudah diakses.
- **ViewModel:** Mengelola data dan logika bisnis untuk antarmuka pengguna, memungkinkan data bertahan lebih lama, meskipun terjadi perubahan konfigurasi (misalnya, putar

layar).

5. Pengelolaan Thread dan Asinkronisitas:

- AsyncTask: Sebelumnya digunakan untuk menjalankan tugas di latar belakang, namun kini telah digantikan dengan lebih modern seperti Executor, LiveData, atau Coroutines di Kotlin untuk menangani operasi asinkron.
- Kotlin Coroutines: Fitur Kotlin yang sangat berguna untuk menangani operasi asinkron, sangat populer di Android karena kesederhanaannya.

6. Dependency Injection (DI):

Pengelolaan dependensi dapat dilakukan dengan Dependency Injection menggunakan pustaka seperti Dagger atau Hilt (yang merupakan pustaka DI berbasis Dagger yang lebih mudah digunakan). Ini membantu dalam pengelolaan dan pemisahan dependensi antar komponen aplikasi.

7. Arsitektur Jetpack:

Jetpack adalah kumpulan pustaka dan alat yang dirancang oleh Google untuk mendukung pengembangan aplikasi Android dengan lebih efisien. Beberapa pustaka Jetpack yang penting adalah:

- LiveData dan ViewModel untuk manajemen UI dan data.
- Navigation Component untuk mengelola navigasi aplikasi secara deklaratif.
- WorkManager untuk menangani tugas latar belakang yang persisten.
- Paging untuk mengelola data dalam jumlah besar dengan cara yang efisien.

8. Pengujian:

Pengujian aplikasi Android sangat penting untuk menjaga kualitas dan stabilitas. Beberapa jenis pengujian yang umum dilakukan adalah:

- Unit Testing: Menguji logika aplikasi secara terisolasi menggunakan pustaka seperti JUnit.

- UI Testing: Menguji antarmuka pengguna menggunakan pustaka seperti Espresso.
- Integration Testing: Menguji integrasi antara berbagai komponen aplikasi.

Dengan memahami dan menerapkan arsitektur yang tepat dalam pengembangan aplikasi Android, pengembang dapat memastikan bahwa aplikasi tersebut dapat di-maintain, scalable, dan efisien dalam jangka panjang.

F. Pasar Aplikasi Android

Toko aplikasi Android adalah platform atau layanan tempat pengguna Android dapat mengunduh dan membeli aplikasi untuk perangkat mereka. Beberapa toko aplikasi Android yang paling populer adalah:

1. Google Play Store, Toko aplikasi resmi dari Google untuk perangkat Android. Play Store menyediakan berbagai aplikasi, game, musik, film, buku, dan lainnya. Ini adalah sumber utama bagi kebanyakan pengguna Android untuk mengunduh aplikasi.
2. Amazon Appstore, Toko aplikasi alternatif yang menawarkan berbagai aplikasi Android. Amazon Appstore juga sering memberikan penawaran khusus dan aplikasi berbayar secara gratis pada waktu- waktu tertentu.
3. Samsung Galaxy Store, Toko aplikasi yang khusus untuk perangkat Samsung. Selain aplikasi Android umum, Galaxy Store juga menyediakan aplikasi dan layanan eksklusif untuk perangkat Samsung.
4. Aptoide, Toko aplikasi yang memungkinkan penggunanya untuk mengunduh aplikasi secara bebas, bahkan aplikasi yang tidak tersedia di Play Store. Aptoide menyediakan aplikasi dalam berbagai kategori, termasuk aplikasi modifikasi.
5. Huawei AppGallery, Toko aplikasi milik Huawei yang digunakan pada perangkat Huawei dan Honor. AppGallery berkembang pesat, terutama setelah Huawei mengalami pembatasan dari Google.

6. SlideME, SlideME adalah platform alternatif untuk mengunduh aplikasi Android yang juga menyediakan aplikasi yang tidak ditemukan di Play Store.

Toko aplikasi ini memungkinkan pengguna untuk menelusuri, mengunduh, dan menginstal berbagai aplikasi Android yang bisa membantu mereka memaksimalkan penggunaan perangkat Android mereka.

G. Tujuan Pengembangan Aplikasi Android

Mengembangkan aplikasi Android membuka peluang karier dan inovasi di era digital yang terus berkembang. Aplikasi Android dikembangkan untuk berbagai alasan seperti berikut ini:

1. Pangsa pasar yang dominan.

Dengan lebih dari 2,5 miliar perangkat Android aktif di seluruh dunia, peluang untuk menciptakan aplikasi yang digunakan secara luas sangat besar. Banyak perusahaan mencari pengembang Android untuk memenuhi kebutuhan pengguna di berbagai sektor, mulai dari pendidikan, kesehatan, hingga hiburan.

2. Peluang karir yang menjanjikan

Industri teknologi saat ini menawarkan banyak posisi untuk pengembang aplikasi Android, baik di perusahaan besar maupun *startup*.

3. Ketersediaan sumber daya belajar yang lengkap.

Google dan komunitas pengembang global menyediakan berbagai sumber daya untuk belajar pengembangan Android. Mulai dari tutorial gratis, kursus *online*, hingga sertifikasi resmi, semuanya dapat membantu Anda memulai perjalanan sebagai *developer* Android.

4. Fleksibilitas dalam berkarya

Pengembangan aplikasi Android memungkinkan Anda untuk bekerja dari mana saja. Dengan keahlian ini, setiap pengembang dapat menjalankan proyek secara remote, bahkan membangun startup sendiri tanpa harus terikat pada lokasi tertentu.

5. Meningkatkan kemampuan problem-solving

Belajar pengembangan aplikasi Android bukan hanya soal

coding, tetapi juga tentang memecahkan masalah. Anda akan diajarkan cara merancang solusi untuk kebutuhan pengguna, mengembangkan fitur yang relevan, hingga mengatasi bug yang muncul dalam aplikasi. Buat

BAB 2

Adroid Studio Ladybug 2024

| 2024.2.1 Patch 3

A. Pendahuluan

Android Studio adalah IDE resmi untuk pengembangan Android, dan dilengkapi semua hal yang Anda perlukan untuk membangun aplikasi Android. Untuk pertama kalinya Android Studio diluncurkan oleh Google pada tanggal 16 Mei 2013 sebagai lingkungan pengembangan terintegrasi atau IDE (Integrated Development Environment) khusus pada Android. Didasarkan pada IntelliJ IDEA, Android Studio memiliki fitur tambahan dari IDE sebelumnya yaitu eclipse ADT. Versi stabil pertama dan resmi dirilis pada bulan Desember 2014. (Wanvy Arifha Saputra, 2020)

Android Studio menyediakan alat untuk mendesain, membangun, menjalankan, dan menguji aplikasi Android. Android Studio menyediakan emulator yang memungkinkan pengembang menguji aplikasi mereka di berbagai perangkat dan konfigurasi Android tanpa harus memiliki perangkat fisik. Versi stabil Android Studio terbaru adalah Android Studio Ladybug 2024.2.1 Patch 3 yang dirilis pada tanggal 31 Oktober 2024. Android Studio Ladybug 2024.2.1 Patch 3 adalah versi spesial atau pembaruan dari Android Studio yang dirancang untuk memberikan pengalaman pengembangan aplikasi Android yang lebih stabil dan cepat Android Studio Ladybug memiliki beberapa kelebihan, di antaranya memiliki fitur Editor Kode baru yang menggunakan Gemini untuk membantu pengguna menjadi lebih produktif, Android Studio Ladybug memiliki mode K2 terintegrasi untuk meningkatkan stabilitas Kotlin, Android Studio Ladybug memiliki fitur untuk meningkatkan efisiensi startup IDE, Android Studio Ladybug memiliki fitur untuk menyederhanakan pengelolaan ekspresi cron dan lingkungan pengembangan untuk aplikasi Android, bahasa

pemrograman utama yang digunakan adalah Java dan Kotlin.

B. Perangkat Pendukung Android Studio Ladybug | 2024.2.1 Patch 3

Agar Software Android Studio LadyBug dapat terpasang dengan sempurna dibutuhkan perangkat komputer dengan spesifikasi sebagai berikut:

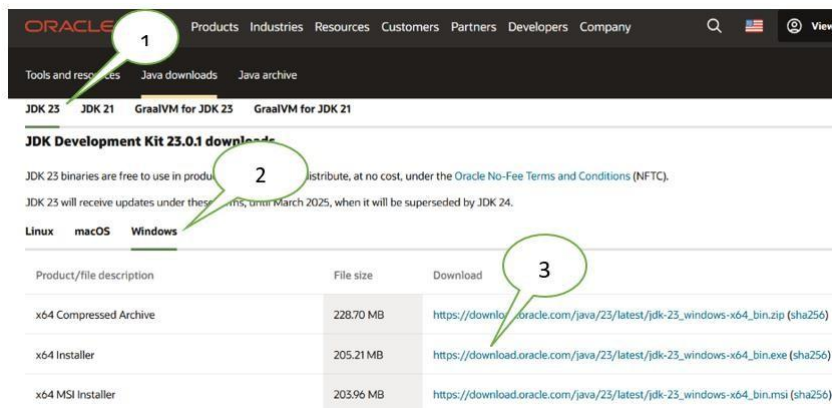
- a. Prosesor: Intel Core generasi ke-13 atau yang lebih baru
- b. RAM: 8 GB atau lebih
- c. Ruang disk minimum: 8 GB (IDE + Android SDK + Android Emulator)
- d. Resolusi layar minimum: 1280 x 800

C. Instalasi JDK 23

Fungsi dari SDK Component Installer adalah untuk mengelola dan memperbarui komponen-komponen yang diperlukan dalam pengembangan perangkat lunak, khususnya dalam konteks Software Development Kit (SDK). SDK adalah seperangkat alat yang digunakan oleh pengembang perangkat lunak untuk membangun aplikasi untuk platform tertentu, misalnya Android SDK untuk pengembangan aplikasi Android.

Untuk memperoleh JDK 23 buka link:

- <https://www.oracle.com/java/technologies/downloads/>



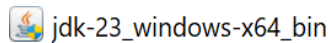
Gambar 1.1 Download JDK 23

- Pilih JDK versi 23 – pilih Windows – pilih link x64 Installer



Gambar 1.2 Proses Download JDK 23

- Setelah proses unduh selesai, selanjutnya klik 2 kali file JDK 23.



Gambar 1.3 File JDK 23

- Jika proses instalasi JDK 23 telah selesai klik tombol Close.

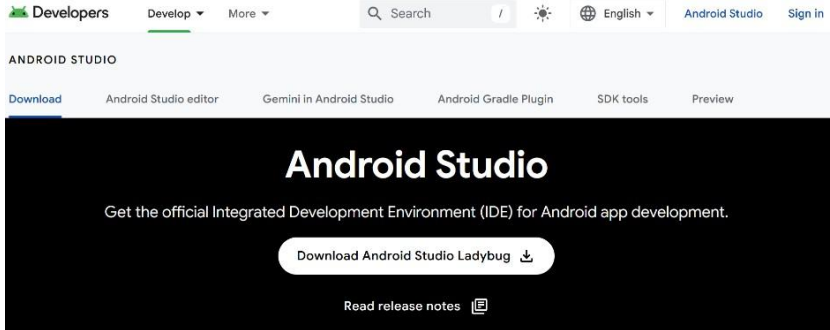


Gambar 1.4 Instal JDK 23

D. Instalasi Android Studio Ladybug | 2024.2.1 Patch 3

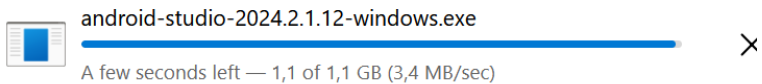
Sebelum melakukan instalasi Android Studio Ladybug | 2024.2.1 Patch 3:

- Pertama adalah mengunduh file Android Studio, buka link: <https://developer.android.com/studio>



Gambar 1.5 Link Android Studio Ladybug | 2024.2.1 Patch 3

- Untuk mengunduh Android Studio Ladybug | 2024.2.1 Patch 3, selanjutnya klik link Download Android Studio Ladybug



Show all downloads

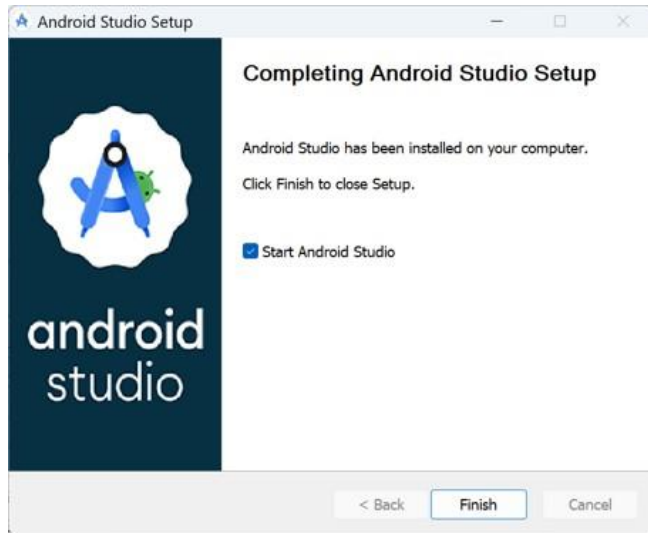
Gambar 1.5 Proses Download Android Studio Ladybug

- Setelah proses download Android Studio Ladybug | 2024.2.1 Patch 3, klik 2 kali pada file Android Studio Ladybug | 2024.2.1 Patch 3 yang telah diunduh

★ android-studio-2024.2.1.12-windows

Gambar 1.6 File Android Studio Ladybug | 2024.2.1 Patch 3

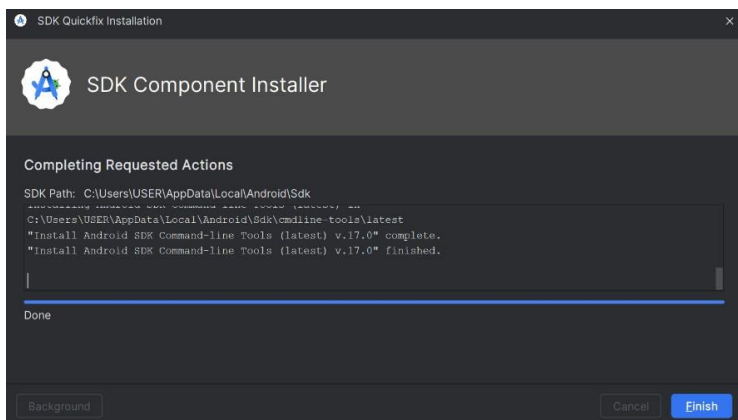
- Jika proses instalasi Android Studio Ladybug | 2024.2.1 Patch 3 telah selesai klik tombol Finish.



Gambar 1.7 Instal Android Studio Ladybug | 2024.2.1 Patch 3

E. Update SDK Component Installer

Fungsi dari SDK Component Installer adalah untuk mengelola dan memperbarui komponen-komponen yang diperlukan dalam pengembangan perangkat lunak, khususnya dalam konteks Software Development Kit (SDK). SDK adalah seperangkat alat yang digunakan oleh pengembang perangkat lunak untuk membangun aplikasi untuk platform tertentu, misalnya Android SDK untuk pengembangan aplikasi Android.



Gambar 1.8 Update SDK Component Installer

F. Update SDK Manager

Setelah Android Studio Ladybug dengan sempurna, selanjutnya adalah melakukan pengaturan Update SDK Manager. Tujuan Update SDK Manager adalah untuk memperbarui perangkat pengembangan perangkat lunak (SDK) yang digunakan dalam pembangunan aplikasi, khususnya untuk platform seperti Android. SDK Manager memungkinkan pengembang untuk mengelola dan memperbarui komponen SDK yang diperlukan untuk pengembangan aplikasi. Berikut ini langkah-langkah Update SDK Manager:

a. Update SDK Platform

Langkah-langkah pengaturan SDK Platform adalah:

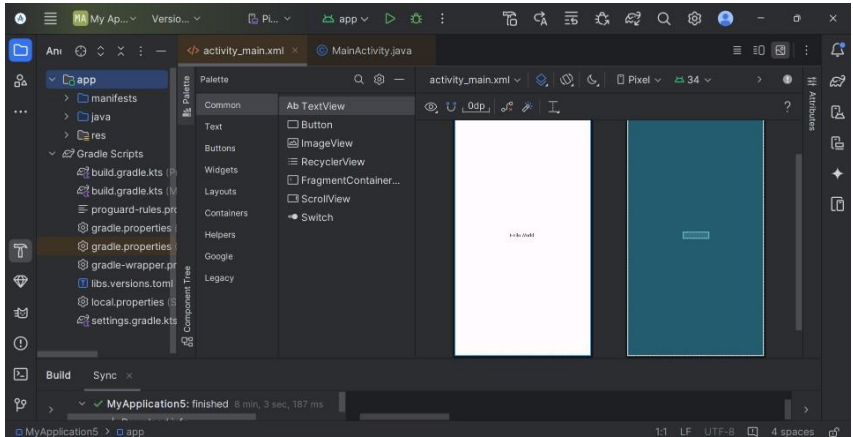
- Aktifkan Android Studio Ladybug | 2024.2.1 Patch 3



Gambar 1. 9 Android Studio Ladybug | 2024.2.1

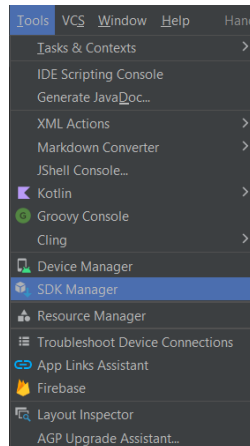
Patch 3 Maka akan tampil user interface dari Android Studio Ladybug |

2024.2.1 Patch 3



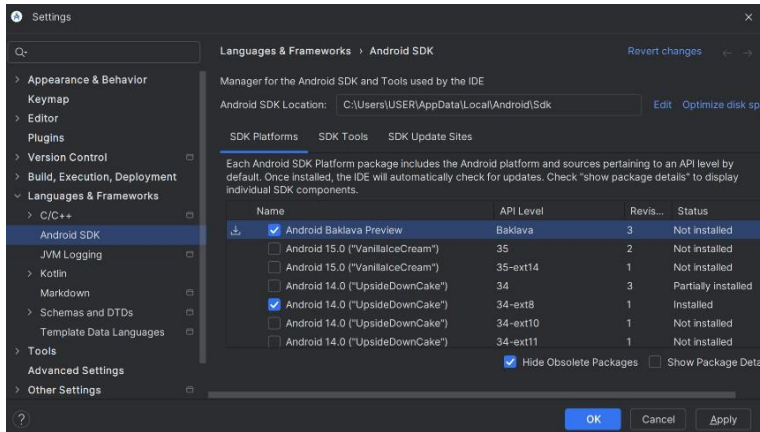
Gambar 1.10 Interface Android Studio Ladybug | 2024.2.1 Patch 3

- Pilih menu Tool – SDK Manager



Gambar 1.11 SDK Manager

- Pilih Android Android Bakara Preview – klik Apply - Ok



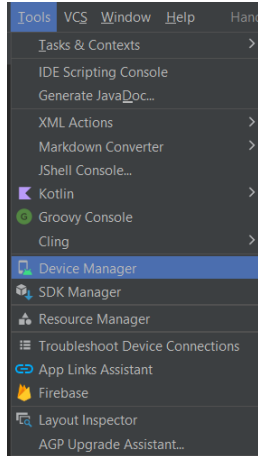
Gambar 1.12 Android Bakara Preview Update SDK Tools

- Centang semua item paket SDK Tool – klik Apply - Ok
- b. SDK Update Sites
 - Centang semua item paket SDK Update Site – klik Apply - Ok

G. Membuat Android Virtual Device

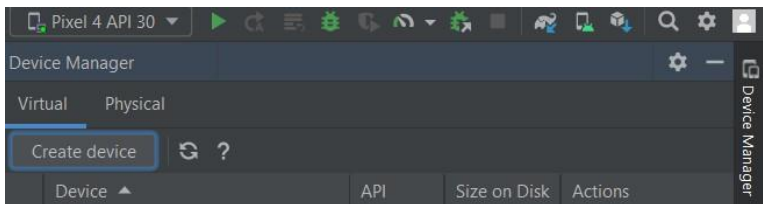
Setelah Update SDK Manager Selesai dengan baik selanjutnya adalah membuat Android Virtual Device (AVD) sebagai sarana pengujian untuk aplikasi Android yang dikembangkan. Langkah-langkahnya adalah:

- a. Klik menu Tools – Device Manager



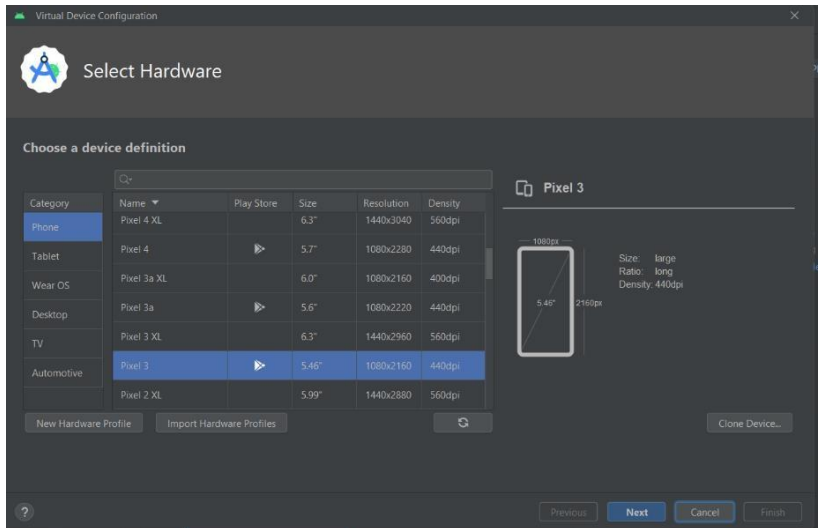
Gambar 1.14 Device Manager

- b. Klik Create Device



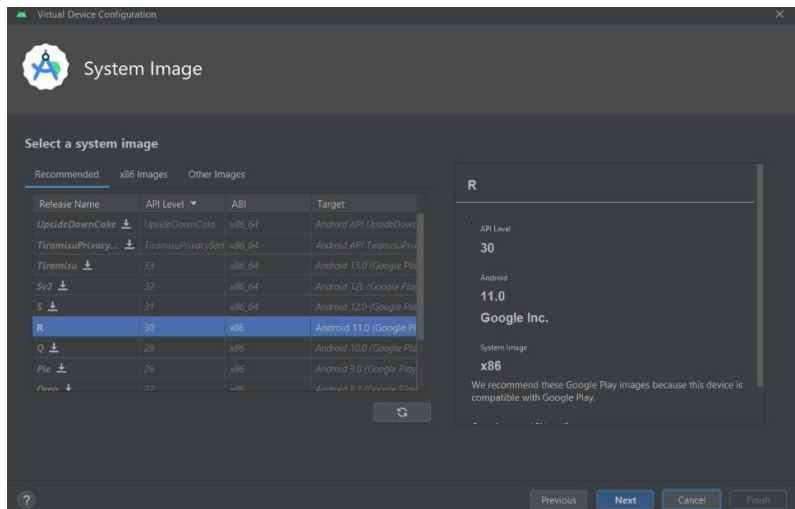
Gambar 1.15 Create Device

- c. Select Hardware – Choose a device definition – Category – Name = Pixel 3 – klik tombol Next



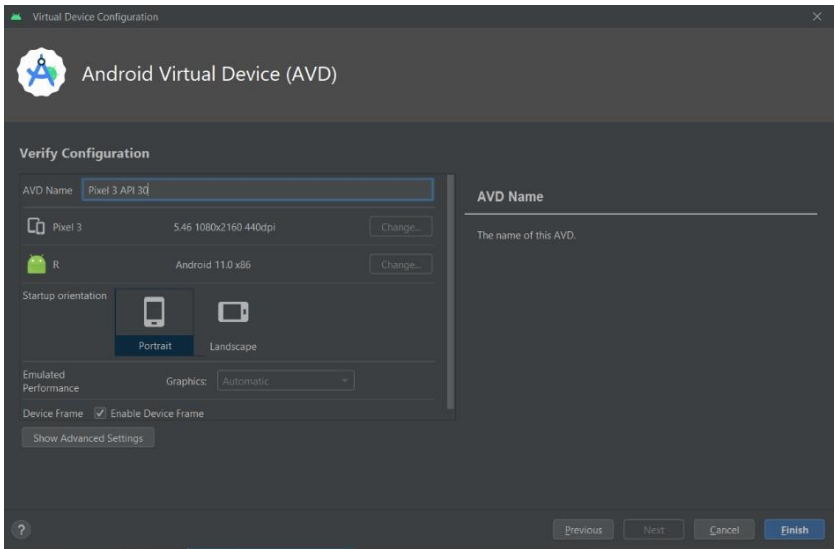
Gambar 1.16 Select Hardware

- d. System Image – Select a system image – API Level = 30
– klik tombol Next



Gambar 1.17 System Image

e. Android Virtual Device (AVD) – Klik tombol Finish

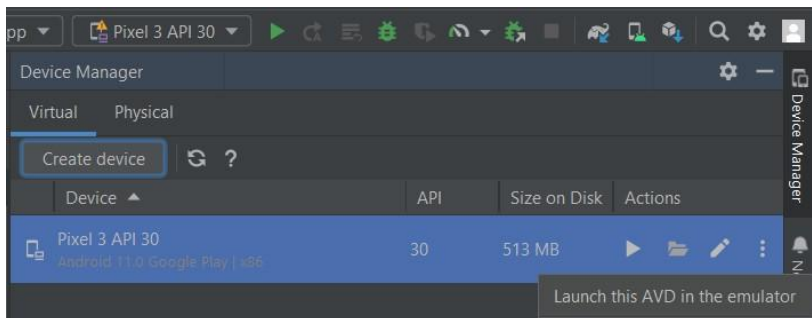


Gambar 1.18 Android Virtual Device (AVD)

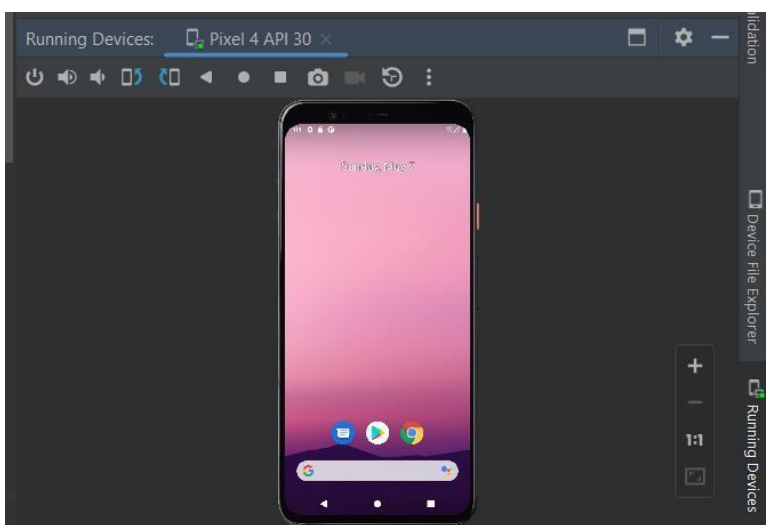
e. **Menguji Android Virtual Device**

Selanjutnya adalah menguji Android Virtual Device, langkahnya adalah:

- Klik menu Run(Shift + F10), atau langsung klik tombol run pada Device Pixel 3 API 30



Gambar 1.19 Launch Android Virtual Device (AVD)



Gambar 1.20 Running Devices Pixel 3 API 30

BAB 3

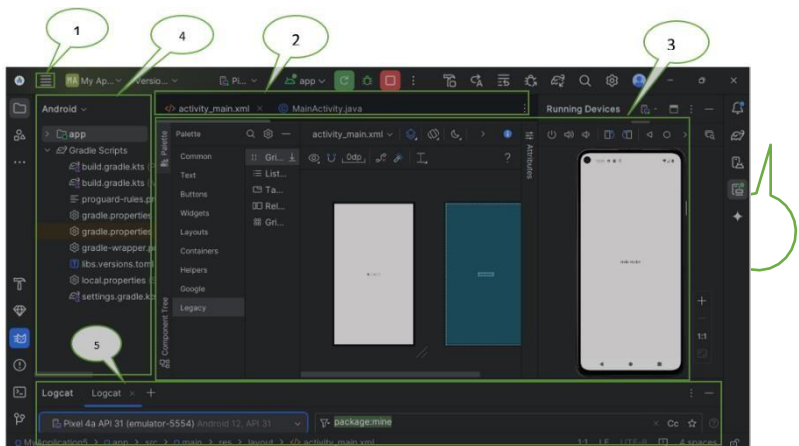
Menggunakan Adroid

Studio Ladybug | 2024.2.1

Patch 3

A. Pendahuluan

Android Studio Ladybug |2024.2.1 Patch 3 memiliki panel jendela yang menyediakan editor kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, Android Studio Ladybug |2024.2.1 Patch 3 juga memiliki alat untuk development, men-debug, menguji, dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih cepat dan mudah. Pengujian aplikasi dapat dimelalui berbagai macam emulator yang telah dikonfigurasi sebelumnya, dan membangun APK produksi untuk publikasi. Berikut ini adalah panel jendela Android Studio Ladybug |2024.2.1 Patch 3 :



Gambar 2.1 Jendela Panel

Berikut adalah penjelasan Panel jendela Android Studio Ladybug [2024.2.1 Patch 3 terdiri :

1. Bilah Menu. Bilah menu menjalankan beragam tindakan, termasuk menjalankan aplikasi Android dan meluncurkan alat Android.
2. Bilah Navigasi. Bilah navigasi memungkinkan navigasi melalui proyek dan membuka file untuk pengeditan. Bilah navigasi menyediakan tampilan struktur proyek yang lebih ringkas.
3. Panel Editor. Panel ini menampilkan materi file yang dipilih dalam proyek. Misalnya, setelah memilih layout (seperti yang ditampilkan dalam gambar), panel ini menampilkan editor layout dengan alat untuk mengedit layout. Setelah memilih file kode Java, panel ini menampilkan kode dengan alat untuk mengedit kode.
4. Bilah Status. Bilah status menampilkan status proyek dan Android Studio itu sendiri, serta peringatan atau pesan apa pun. Anda bisa mengamati kemajuan pembangunan di bilah status.
5. Panel Project. Panel proyek menampilkan file proyek dan hierarki proyek.
6. Gradle, Gradle berfungsi sebagai alat utama untuk menangani berbagai aspek pembangunan proyek, seperti mengompilasi kode, mengelola dependensi, dan menghasilkan file APK atau bundle yang dapat dipasang pada perangkat Android

B. Struktur Proyek Android

Struktur proyek Android merujuk pada susunan direktori dan file dalam sebuah proyek Android Studio. Proyek Android memiliki struktur yang terorganisir, di mana setiap file dan folder memiliki tujuan tertentu. Berikut adalah penjelasan tentang struktur proyek Android secara umum:

1. Folder app/
 - Ini adalah folder utama yang berisi semua sumber daya (resources) aplikasi Android Anda.
 - src/
Menyimpan kode sumber aplikasi, dibagi lagi menjadi dua folder utama:
 - main/
Menyimpan kode utama aplikasi.

- `libs/`
Menyimpan pustaka eksternal (libraries) yang tidak dikelola oleh Gradle.
- `test/`
Berisi file kode untuk pengujian unit (unit tests). Kode pengujian ini dapat dipisahkan untuk memudahkan proses pengujian aplikasi.
- `androidTest/`
Digunakan untuk menyimpan file pengujian yang melibatkan interaksi dengan Android Framework (UI tests, dll).

2. File Build dan Konfigurasi

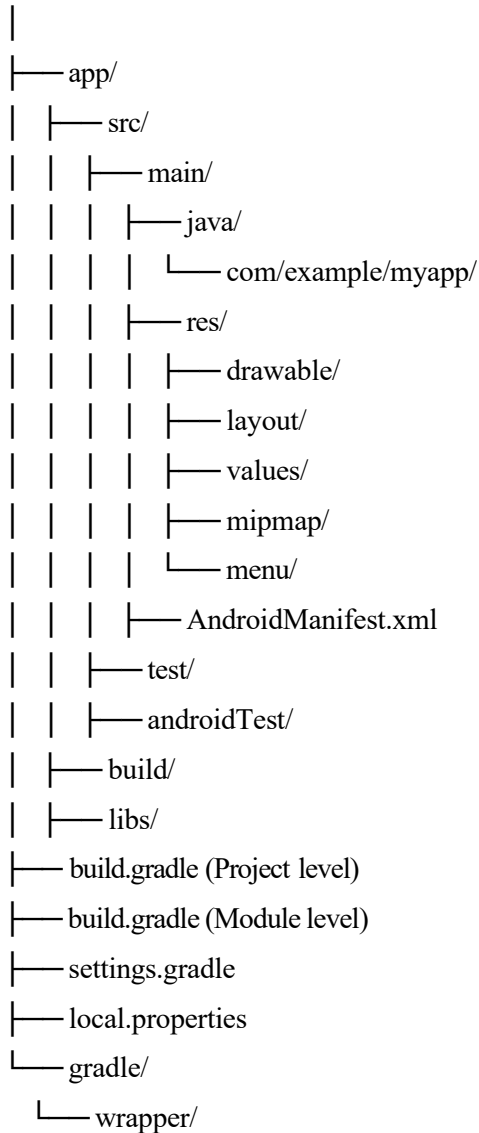
- `build.gradle` (Project level)
Menyimpan konfigurasi umum untuk seluruh proyek, seperti repository yang digunakan, plugin yang diperlukan, dan konfigurasi lainnya.
- `build.gradle` (Module level)
Digunakan untuk konfigurasi spesifik untuk setiap modul (misalnya, modul aplikasi `app/`), seperti dependensi, SDK versi, dan pengaturan kompilasi.
- `settings.gradle`
Digunakan untuk mengonfigurasi modul-modul yang akan disertakan dalam proyek. Biasanya digunakan saat proyek memiliki lebih dari satu modul.

3. File Lainnya

- `local.properties`
Menyimpan konfigurasi lokal yang bersifat spesifik untuk pengembangan di komputer tertentu, seperti lokasi SDK Android.
- `gradle.properties`
Menyimpan properti Gradle untuk pengaturan build yang lebih global.
- `gradle/wrapper/`
Folder yang berisi file konfigurasi untuk Gradle Wrapper yang memastikan bahwa proyek menggunakan versi Gradle yang sesuai.

Struktur Proyek Android

Umum: project/



Penjelasan Folder dan File Utama

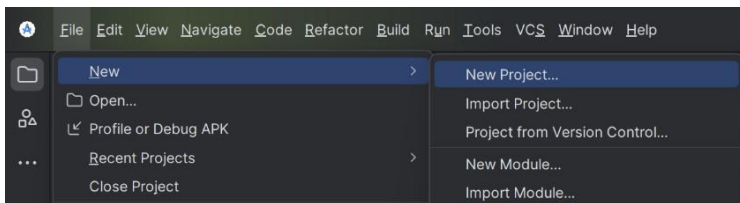
- `src/main/java/`: Menyimpan kode logika aplikasi seperti kelas-kelas Java/Kotlin yang menangani fungsionalitas aplikasi.

- `src/main/res/`: Menyimpan berbagai jenis resource yang digunakan dalam aplikasi.
- `src/main/AndroidManifest.xml`: File penting yang menyatakan konfigurasi komponen aplikasi Android seperti aktivitas dan izin yang dibutuhkan.
- `build.gradle`: File yang berisi pengaturan untuk proses build, dependensi yang digunakan, dan pengaturan plugin.
- Struktur proyek Android ini memberikan cara yang terorganisir untuk mengelola kode dan resource dalam aplikasi. Setiap komponen dikelompokkan dengan rapi untuk memudahkan pengembangan, pemeliharaan, dan distribusi aplikasi Android.

C. Membuat Aplikasi Android Pertama

Setelah Android Studio terinstal, buka aplikasi dan ikuti langkah-langkah berikut:

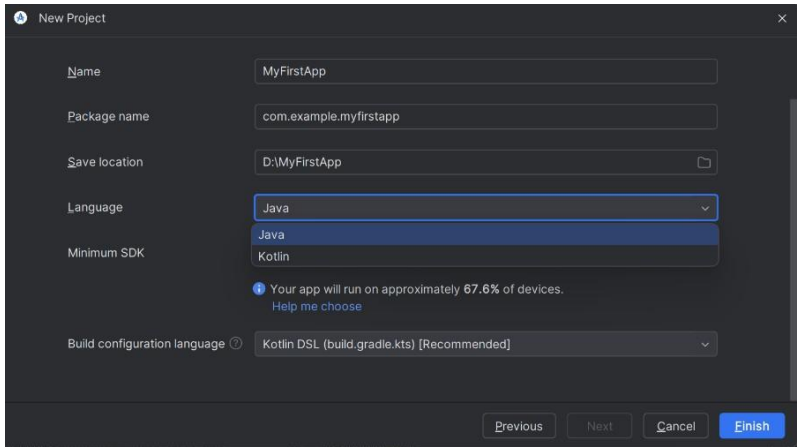
1. Buka Android Studio, jalankan Android Studio. Di layar utama, pilih opsi "Menu – New – New Project".



Gambar 2.2 New Project

2. Pilih Template And Tablet – Empty Views Activity. Tetapi Android Studio menyediakan beberapa template. Pilih salah satu template yang sesuai dengan aplikasi yang akan akan dibuat. Pilihan umum termasuk:
 - Empty Activity: Aktivitas kosong untuk memulai proyek dari awal.

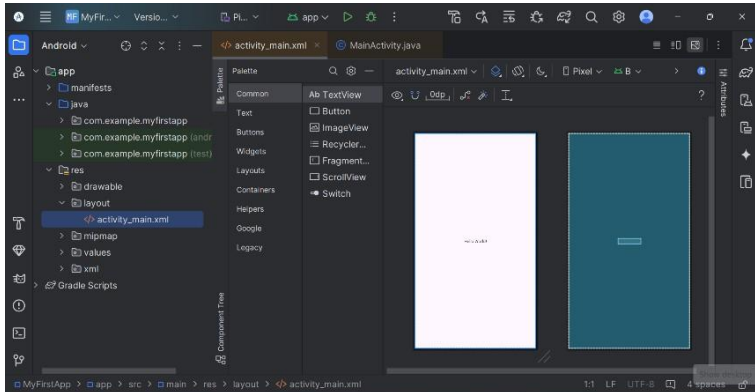
- Basic Views Activity: Menyediakan template dengan navigasi berbasis tab di bagian bawah.



Gambar 2.3 New Project

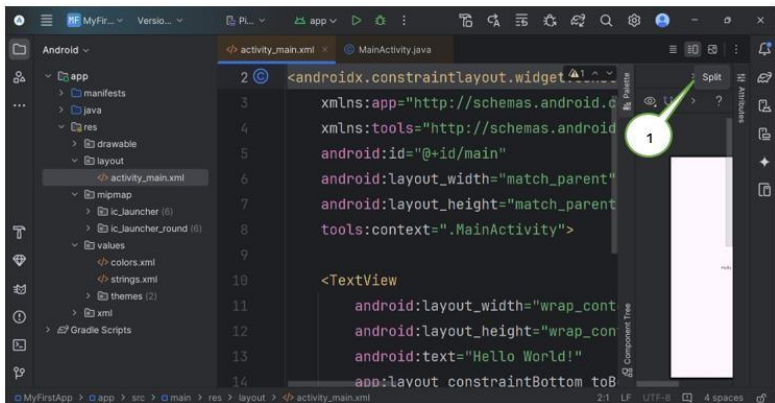
- Pilih Empty Views Activity untuk proyek dasar.
3. Menentukan Nama Project dan Lokasi, di jendela "Configure your project", isi:
 - Name: Nama aplikasi Anda (misalnya, MyFirstApp).
 - Save location: Lokasi tempat Anda ingin menyimpan proyek di komputer Anda.
 - Language: Pilih bahasa pemrograman Java diantara Java atau Kotlin).
 - Minimum SDK adalah API 30, Minimum SDK yang dipilih harus sesuai dengan versi Android yang mendukung dengan perangkat yang digunakan.

4. Klik Finish, setelah mengatur semua pengaturan, klik Finish untuk membuat project baru.



Gambar 2.4 User Interface MyFirsApp

5. Layout, untuk membuat Layout buka file `activity_main.xml` yang pada Java Modul (app)– Resources Root (res) - Package (layout). Untuk menampilkan kode editor layout klik split.



Gambar 2.5 Split

Kode main_activity.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```

xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="HelloWorld!"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>

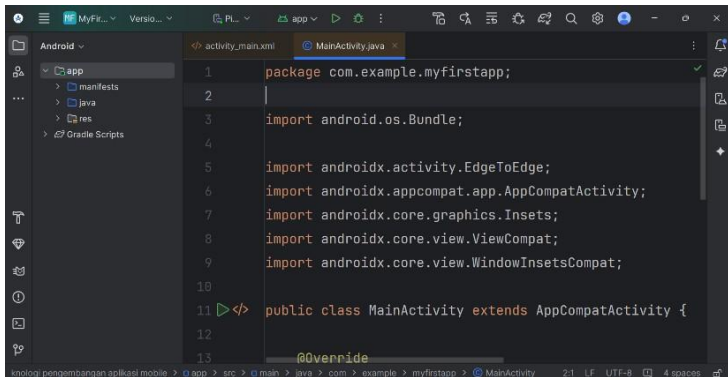
```

```

</androidx.constraintlayout.widget.ConstraintLayout>

```

6. Kode Java, untuk menulis kode java buka file MainActivity.java yang pada Java Modul (app)– Sources Root (java) - com.example.MyFirstApp MainActivity.java



Gambar 2.6 MainActivity.java

Kode MainActivity.java:

```

package
com.example.myfirstapp;
import android.os.Bundle;
import androidx.activity.EdgeToEdge;
import
androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;

```

```

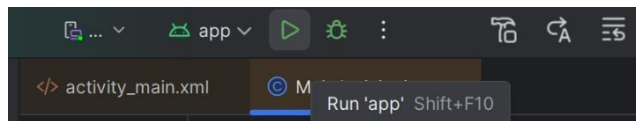
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends
    AppCompatActivity { @Override
    protected void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.m
ain), (v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        });
    }
}

```

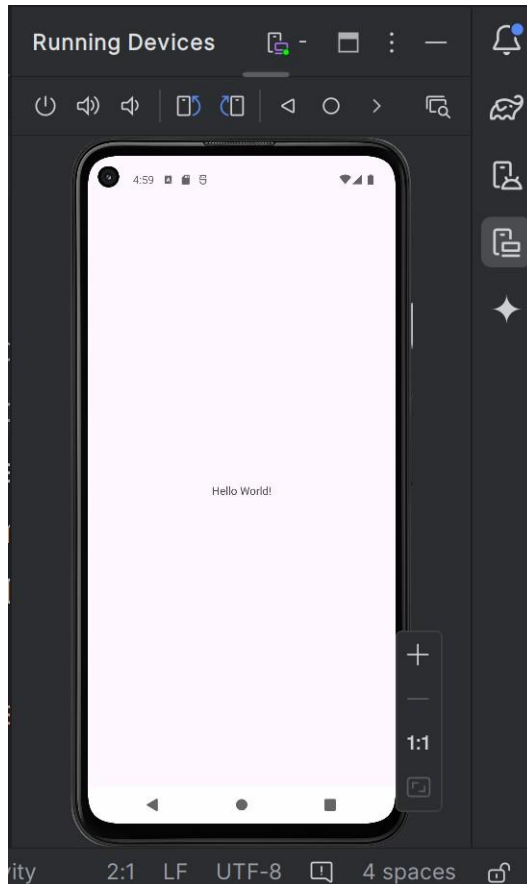
D. Menjalankan Aplikasi dengan Android Virtual Device

untuk menjalankan aplikasi dengan android virtual device klik Tool Run 'App' seperti gambar berikut:



Gambar 2.8 Run App

Hasil pengujian:



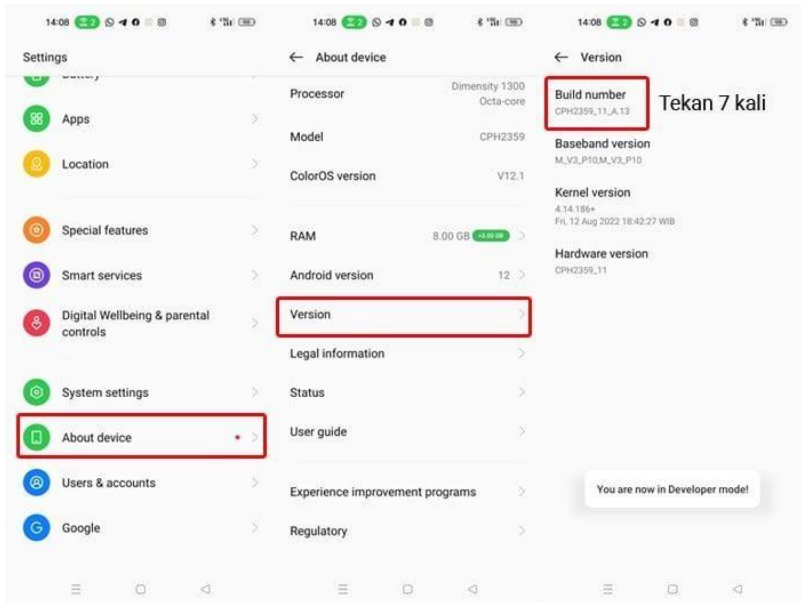
Gambar 2.9 Output

E. Menjalankan Aplikasi di Perangkat Smartphone

Agar Aplikasi Android dapat berjalan di perangkat smartphone harus melakukan pengaturan di perangkat smartphone. Langkah-langkah pengaturan pada perangkat smartphone adalah :

- Buka aplikasi Settings di Smartphone(sebagai contoh OPPO)
- Pilih menu About device
- Pilih opsi Version

- Lalu tekan 7 kali pada bagian Build number
- Nantinya akan muncul notifikasi You are now in Developer

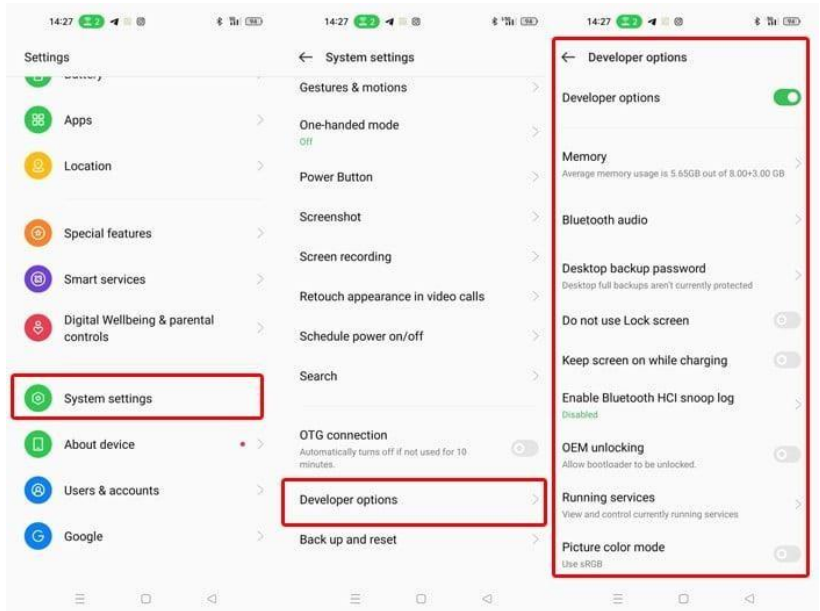


mode!

Gambar 2.10 Setting Perangkat Smartphone

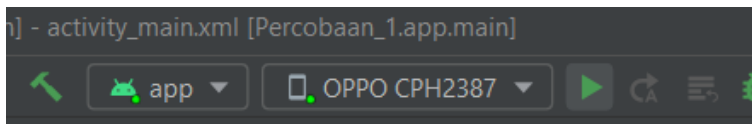
- Kembali ke halaman utama Settings
- Pilih menu System settings atau Additional settings tergantung versi dari ColorOS

- Masuk ke menu Developer options Kamu akan melihat menu yang biasanya tidak ditemukan.

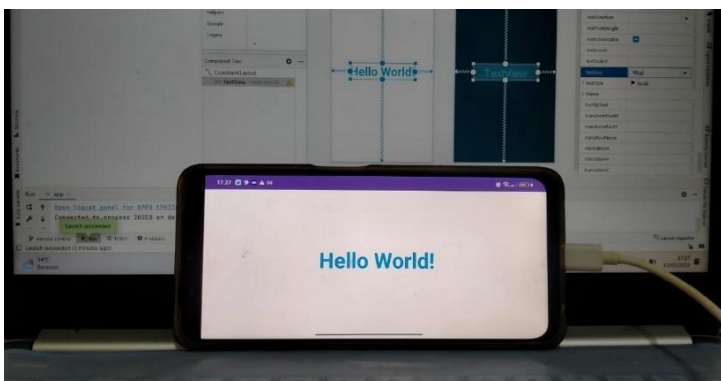


Gambar 1.41 Setting Developer Option

- Pada halaman Developer Options terdapat beragam fitur yang bisa kamu gunakan untuk mengembangkan aplikasi atau sekadar menguji kemampuan dari sistem yang dimiliki oleh handphone OPPO.
- Untuk menguji perangkat smartphone klik tombol run berwarna hijau.



Gambar 2.11 Pengujian Perangkat Smartphone



Gambar 2.13 Hasil Pengujian Perangkat Smartphone Oppo

BAB 4

Layout

A. Pendahuluan

Dalam pengembangan aplikasi Android, Layout mengacu pada tampilan dan pengaturan elemen-elemen UI (User Interface) di layar perangkat. Layout mengontrol bagaimana komponen seperti tombol, teks, gambar, dan elemen lainnya disusun di layar. Di Android Studio, layout biasanya dibuat menggunakan XML (Extensible Markup Language) yang memungkinkan pengembang untuk menentukan posisi dan gaya elemen-elemen tersebut.

Android menyediakan berbagai jenis layout yang dapat digunakan untuk menyusun tampilan aplikasi, jenis layout tersebut antara lain : LinearLayout, RelativeLayout, ConstraintLayout, GridLayout dan FrameLayout. Setiap jenis layout memiliki kelebihan dan kegunaan tertentu, tergantung pada kebutuhan aplikasi.

B. Jenis-Jenis Layout

Jenis – jenis layout yang tersedia di Android Studio Ladybug | 2024.2.1 Patch 3 adalah sebagai berikut:

1. LinearLayout

LinearLayout digunakan untuk menata elemen UI secara berurutan, baik secara vertikal (dari atas ke bawah) atau horizontal (dari kiri ke kanan). Pengembang dapat menggunakan atribut `android:orientation` untuk menentukan arah tata letak.

Contoh kode `activity_main.xml` untuk LinearLayout:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_
parent"
    android:layout_height="match_
parent"
    android:orientation="vertical"
    tools:context=".MainActivity"
>

    <TextView
        android:layout_width="wrap_co
        ntent"
        android:layout_height="wrap_co
        ntent" android:text="Halo,
        Selamat datang!" />

    <Button
        android:layout_width="wrap_con
        tent"
        android:layout_height="wrap_con
        tent" android:text="Klik Saya"
        />

</LinearLayout>

```

Hasil Pengujian Android:Orientation="vertical"



Gambar 4.1 LinearLayout, Orientation-vertical

Hasil Pengujian Android:Orientation="horizontal"



Gambar 4.2 LinearLayout, Orientation-horizontal

2. Relative Layout

RelativeLayout memungkinkan pemrogram untuk menyusun elemen-elemen UI relatif terhadap satu sama lain. Pemrogram bisa menentukan posisi elemen berdasarkan elemen lain (misalnya, mengatur elemen untuk tampil di bawah atau di sebelah kiri elemen lainnya).

Contoh kode activity_main.xml untuk RelativeLayout:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_
parent"
    android:layout_height="match_
parent"
    tools:context=".MainActivity"
>

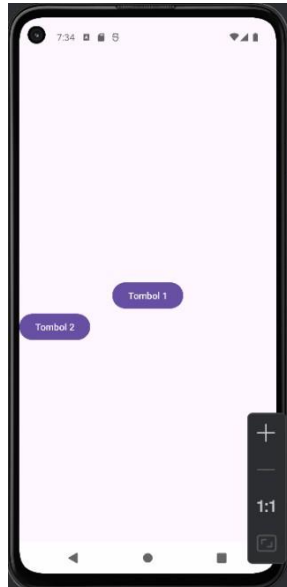
<Button
    android:id="@+id/b
utton1"
    android:layout_width="wrap_c
ontent"
    android:layout_height="wrap_c
ontent" android:text="Tombol
1"
    android:layout_centerInParent=
"true"/>

<Button
    android:id="@+id/b
utton2"
    android:layout_width="wrap_c
ontent"
    android:layout_height="wrap_c
ontent" android:text="Tombol
2"
    android:layout_below="@id/but
ton1"
```

```
android:layout_alignParentStart=  
"true"/>
```

```
</RelativeLayout>
```

Hasil Pengujian :



Gambar 4.3 Relative Layout

3. Constraint Layout

ConstraintLayout adalah layout yang lebih fleksibel dan kuat yang memungkinkan pemrogram untuk mendefinisikan hubungan antar elemen UI dengan membuat constraints atau batasan. Ini memungkinkan pemrogram untuk mendesain UI yang lebih kompleks dengan kontrol yang lebih besar terhadap posisi elemen.

Contoh kode file main_activity.xml untuk ConstraintLayout:

```
<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout  
xmlns:android="http://schemas.android.com/apk/res/a  
ndroid"
```

```

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools" android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Tombol"
    android:layout_marginTop="100dp"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Hasil Pengujian:



Gambar 4.4 Constraint Layout

4. Grid Layout

GridLayout memungkinkan pemrogram untuk menyusun elemen UI dalam bentuk grid (baris dan kolom). Ini sangat berguna jika Anda perlu menyusun elemen-elemen dalam bentuk tabel.

Contoh kode file `main_activity.xml` untuk GridLayout:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_
parent"
    android:layout_height="match_
parent"
    android:columnCount="2"
    android:rowCount="2"
    tools:context=".MainActivity">
    <Button
        android:layout_width="wrap_con
tent"
        android:layout_height="wrap_con
tent" android:text="Tombol 1"
    />

    <Button
        android:layout_width="wrap_con
tent"
        android:layout_height="wrap_con
tent" android:text="Tombol 2"
    />

    <Button
        android:layout_width="wrap_con
tent"
        android:layout_height="wrap_con
tent" android:text="Tombol 3"
    />

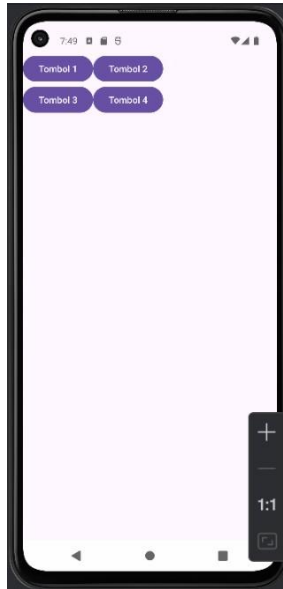
    <Button
        android:layout_width="wrap_con
tent"
```

```

        android:layout_height="wrap_content"
        android:text="Tombol 4"
    />
</GridLayout>

```

Hasil pengujian:



Gambar 4.5 GridLayout

5. Frame Layout

FrameLayout digunakan untuk menampilkan satu elemen UI di atas elemen lainnya. Elemen-elemen dalam FrameLayout biasanya akan diposisikan di sudut kiri atas, meskipun Anda bisa mengubahnya dengan pengaturan lain.

Contoh kode file `main_activity.xml` untuk FrameLayout:

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Tombol" />
```

</FrameLayout>

Hasil Pengujian:



Gambar 4.6 Framelayout

BAB 5

Activity

A. Pendahuluan

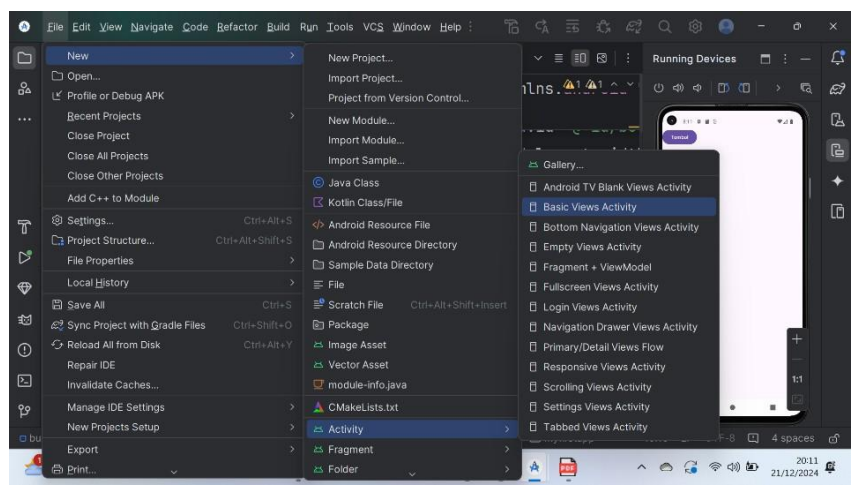
Umumnya, satu aktivitas(activity) di aplikasi ditetapkan sebagai aktivitas "utama", yang disajikan kepada pengguna saat membuka aplikasi untuk pertama kali. Kemudian setiap aktivitas bisa memulai aktivitas lainnya untuk melakukan tindakan yang berbeda. Aktivitas adalah subkelas dari kelas Activity atau salah satu dari subkelasnya. Jika membuat proyek baru di Android Studio, aktivitas tersebut secara default menjadi subkelas dari kelas AppCompatActivity. Kelas AppCompatActivity adalah subkelas Activity yang memungkinkan Anda menggunakan fitur aplikasi Android terbaru seperti bilah aksi dan desain material, sementara tetap memungkinkan aplikasi tersebut kompatibel dengan perangkat yang menjalankan Android versi lama.

Dalam Android Studio, Activity adalah komponen dasar dalam aplikasi Android yang bertanggung jawab untuk menampilkan antarmuka pengguna (UI) dan menangani interaksi pengguna. Setiap layar dalam aplikasi Android biasanya berhubungan dengan satu Activity. Activity dapat berupa layar login, layar utama, atau layar pengaturan, dan setiap Activity memiliki siklus hidup sendiri yang mengontrol bagaimana ia berinteraksi dengan sistem operasi Android dan pengguna.

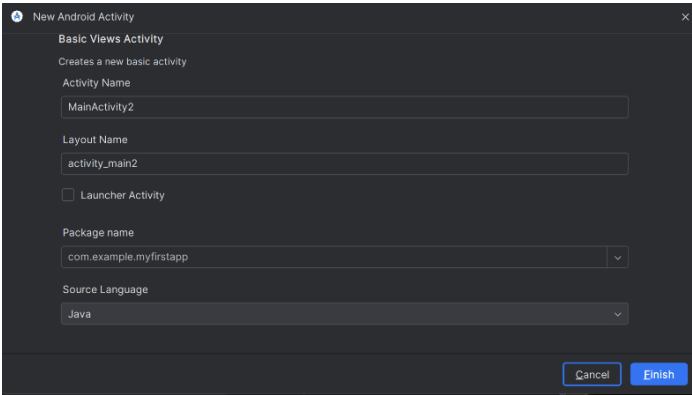
B. Membuat Aktivitas(Activity) Baru

Pemrogram dapat menambahkan aktivitas baru ke proyek di Android Studio dengan menu **File > New > Activity**. Pilih template aktivitas yang ingin pengembang gunakan, atau bisa langsung klik kanan di package yang berada dalam source root java > com.example.app_name > Activity > Basic Views Activity > Activity

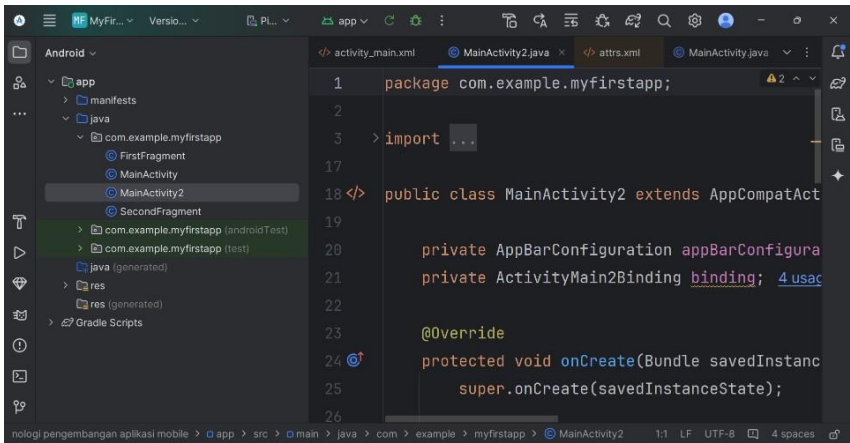
Name > Finish.



Gambar 5.1 Menambah aktivitas baru



Gambar 5.2 New Android Activity



Gambar 5.3 MainActivity

C. Intent

Dalam Android Studio, Intent adalah objek yang digunakan untuk memulai aktivitas baru, berkomunikasi antara komponen aplikasi atau mengirim data antar komponen aplikasi. Berikut contoh kodes Intent didalam android studio:

```
Intent intent = new Intent(MainActivity.this,
    MainActivity2.class); startActivity(intent);
```

Intent juga dapat digunakan untuk mengirim data dan menerima data. Berikut contoh kode Intent didalam android studio untuk mengirim data:

```
Intent intent = new Intent(MainActivity.this,
    MainActivity2.class); intent.putExtra("KEY_NAME",
    "Siti Nurbaya"); startActivity(intent);
```

Berikut contoh kode Intent didalam android studio untuk menerima data:

```
String name = getIntent().getStringExtra("KEY_NAME");
```

BAB 6

Database SQLite

A. Pendahuluan

SQLite adalah sebuah sistem manajemen basis data relasional (RDBMS) yang ringan, cepat, dan self-contained. Berbeda dengan kebanyakan sistem manajemen basis data yang memerlukan server terpisah untuk menjalankan database (seperti MySQL, PostgreSQL), SQLite menyimpan seluruh database dalam satu file tunggal di sistem berkas. Ini membuatnya ideal untuk aplikasi- aplikasi yang membutuhkan pengelolaan data tanpa konfigurasi yang rumit.(Wanvi Arifta Saputra, 2020)

SQLite adalah sistem manajemen basis data yang ringan dan bersifat lokal, yang sering digunakan dalam aplikasi Android untuk menyimpan data secara lokal. Android Studio menyediakan dukungan penuh untuk SQLite, memungkinkan pengembang untuk membuat, mengelola, dan melakukan query terhadap basis data SQLite dalam aplikasi Android yang dibangun.

B. Struktur SQLite

Berikut adalah langkah-langkah untuk membuat struktur SQLite dalam aplikasi Android Studio:

1. Membuat Kelas Adapter dan BaseAdapter

Kelas adapter digunakan untuk menghubungkan data dengan tampilan (view) dalam antarmuka pengguna, seperti ListView, RecyclerView, atau elemen tampilan berbasis daftar lainnya. Adapter bertindak sebagai perantara antara data dan tampilan, menampilkan data dalam bentuk yang sesuai dengan struktur tampilan.

Kelas BaseAdapter merupakan kelas dasar untuk membuat adapter yang

lebih kompleks. BaseAdapter memberi pemrogram fleksibilitas lebih besar untuk menampilkan data dalam berbagai cara.

Pemrogram perlu mengimplementasikan beberapa metode seperti getCount(), getItem(), getItemId(), dan getView() untuk mengontrol bagaimana data ditampilkan dalam tampilan.

Berikut contoh kode untuk kelas adapter :

```
public class Adapter extends
    BaseAdapter { private Activity
    activity;
    private LayoutInflater
    inflater; private
    List<Data> list;

    public Adapter(Activity activity, List<Data> list){
        this.activity = activity;
        this.list=list;
    }
```

2. Membuat kelas SQLiteOpenHelper

SQLiteOpenHelper adalah kelas yang digunakan untuk membuat dan mengelola database SQLite dalam Android. Kelas ini membantu dalam membuat, membuka, dan mengupgrade database.

Contoh kode file kelas SQLiteOpenHelper

```
public class Helper extends
    SQLiteOpenHelper { private static
    final int Database_Version = 1; static
    final String Database_name ="crud";
    }

    public Helper(Context context){
        super(context, Database_name, null, Database_Version);
    }
    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        final String SQL_CREATE_TABLE = "CREATE TABLE users (id
        INTEGER PRIMARY KEY autoincrement, name text NOT NULL, email text
        NOT NULL)";
        sqLiteDatabase.execSQL(SQL_CREATE_TABLE);
    }
```

```
}
```

3. Menggunakan database Sqlite

Setelah kelas SQLiteOpenHelper dibuat, pemrogram bisa menggunakannya untuk menambahkan, membaca, memperbarui, atau menghapus data dalam database SQLite.

- Menambah data, untuk menambahkan data ke dalam tabel, pemrogram bisa menggunakan metode insert() dari objek SQLiteDatabase.

Contoh kode metode tambah data dalam kelas SQLiteOpenHelper:

```
public void insert(String name, String email){
    SQLiteDatabase database
    =this.getWritableDatabase();
    String QUERY = "INSERT INTO users (name, email) VALUES
    ("'+name+"', '"+email+"')";
    database.execSQL(QUERY);
}
```

- Membaca data, untuk membaca data, pemrogram bisa menggunakan metode query() atau.rawQuery().

Contoh kode membaca data dalam kelas SQLiteOpenHelper:

```
SQLiteDatabase database =
this.getWritableDatabase(); Cursor cursor =
database.rawQuery(QUERY, null );
```

- Memperbaharui data, untuk memperbarui data pemrogram dapat menggunakan metode update(). Berikut contoh kode untuk metode update :

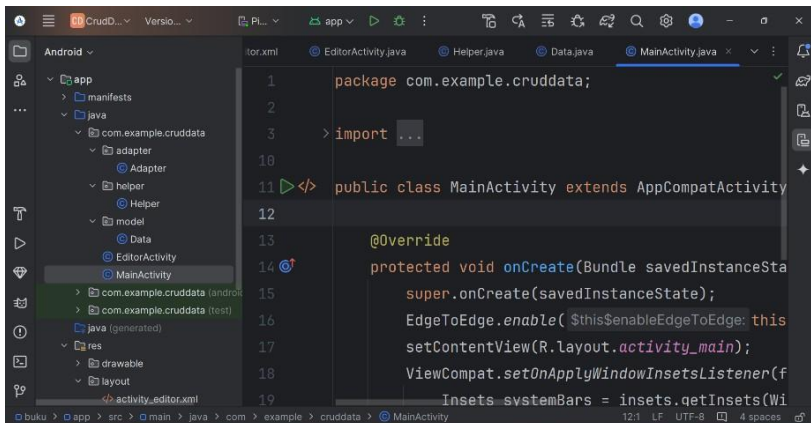
```
public void update(int id, String name, String
email){ SQLiteDatabase
database=this.getWritableDatabase();
String QUERY= "UPDATE users SET name = '"+name+"', email =
 '"+email+"' WHERE id = "+id;
database.execSQL(QUERY);
}
```

- Menghapus data, untuk menghapus data pemrogram bisa menggunakan metode delete(). Berikut contoh kode untuk metode delete(). Berikut contoh kode untuk metode update

```
public void delete(int id){
    SQLiteDatabase database =
    this.getWritableDatabase(); String QUERY =
    "DELETE FROM users WHERE id =" +id;
    database.execSQL(QUERY);
}
```

C. Membuat aplikasi CrudData dengan Database SQLite

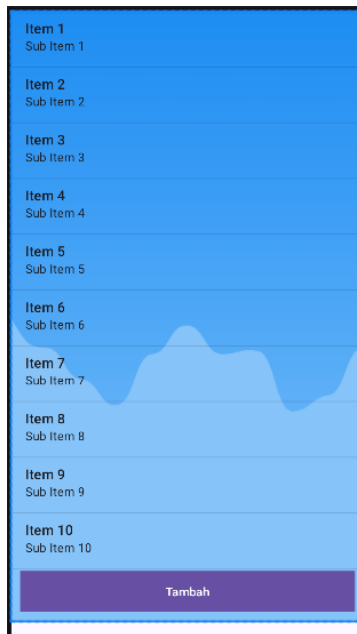
Berikut ini adalah struktur aplikasi CrudData:



Gambar 6.1 Struktur aplikasi CrudData

Berikut ini langkah-langkah membuat aplikasi CrudData di Android Studio Ladybug | 2024.2.1 Patch 3:

1. Buatlah proyek baru dengan nama : CrudData, pilih bahasa : java, tentukan minimum SDK : API 30.
2. Desainlah user interface dengan view atau tampilan sebagai berikut :



Gambar 6.2 Desain view

- Membuat background, sediakan gambar untuk background image android dan simpan di app > res > drawable > image.jpg
- Aktipkan file main_activity.xml (app > res > layout > main_activity.xml) kemudian klik split untuk mengaktifkan view code dan tuliskan kode sebagai berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:background="@drawable/bg"
    android:layout_width="match_parent"
```

```

android:layout_height="match_parent"
android:layout_marginBottom="20dp"
android:orientation="vertical"
tools:context=".MainActivity">
<ListView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="60dp"/>
<Button
    android:id="@+id/btn_add"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="10dp"
    android:layout_marginTop="10dp"
    android:layout_marginEnd="10dp"
    android:layout_marginBottom="10dp"
    android:background="@color/design_default_color_primary"
    android:text="Tambah"
    android:textColor="@color/white" />
</RelativeLayout>

```

- Menambah file xml baru dengan nama : activity_editor.xml, kemudian desain view sebagai berikut:

Gambar 6.3 Desain View activity_editor

Kode View activity_editor.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    android:orientation="vertical">
    <EditText
        android:id="@+id/edit_name"
        android:layout_width="match_
        parent"
        android:layout_height="wrap_conte
        nt"
        android:inputType="textPersonNam
        e" android:hint="Nama Lengkap"
        android:autofillHints=""
        android:layout_marginBottom="10d
        p"/>

    <EditText
        android:id="@+id/edit_email"
        android:layout_width="match_
        parent"
        android:layout_height="wrap_conte
        nt"
        android:inputType="textEmailAddre
        ss" android:hint="Email"
        android:autofillHints=""
        android:layout_marginBottom="10d
        p"/>

    <Button
        android:id="@+id/btn_save"
        android:layout_width="match_paren
        t"
        android:layout_height="wrap_conte
        nt" android:text="Simpan"
        android:background="@color/design_default_color_primary"
        android:textColor="@color/white"/>

</LinearLayout>
```

- Tambah file baru dengan nama list_user.xml, dengan desain view sebagai berikut:



Gambar 6.4 Desain View list_user

Kode View list_user.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10
sp"
    android:orientation="v
ertical">
    <TextView
        android:id="@+id/text_name
"
        android:layout_width="wrap_
content"
        android:layout_height="wrap_
content"
        android:text="Name"
        android:textColor="#000000
" android:textStyle="bold"/>

    <TextView
        android:id="@+id/text_emai
l"
        android:layout_width="wrap_
```

```

        content"
        android:layout_height="wrap_
        content"
        android:text="Email"
        android:textColor="#000000
        "
        android:textStyle="bold"/>

</LinearLayout>

```

3. Membuat paket helper, tambahkan kelas Helper di paket helper kemudian tuliskan kode Helper.java berikut ini:

```

package
com.example.cruddata.helper;
import android.content.Context;
import
android.database.Cursor;
import
android.database.sqlite.SQLiteDatabase;
import
android.database.sqlite.SQLiteOpenHelper;

import

androidx.annotation.Nullable;

import java.util.ArrayList;
import java.util.HashMap;

public class Helper extends
    SQLiteOpenHelper { private static final
        int Database_Version = 1; static final
        String Database_name ="crud";

        public Helper(@Nullable Context context, @Nullable String
name, @Nullable SQLiteDatabase.CursorFactory factory, int
version) {
            super(context, Database_name, null, Database_Version);
        }

        @Override
        public void onCreate(SQLiteDatabase sqLiteDatabase) {
            final String SQL_CREATE_TABLE = "CREATE TABLE
users (id INTEGER PRIMARY KEY autoincrement, name text
NOT NULL, email text NOT NULL)";
            sqLiteDatabase.execSQL(SQL_CREATE_TABLE);

```

```

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i,
        int i1)
    {
        sqLiteDatabase.execSQL("DROP TABLE IF EXISTS
            users"); onCreate(sqLiteDatabase);
    }

    public ArrayList<HashMap<String, String>> getAll(){

        ArrayList<HashMap<String, String>> list = new
        ArrayList<>(); String QUERY = "SELECT * FROM users ";
        SQLiteDatabase database =
        this.getWritableDatabase(); Cursor cursor =
        database.rawQuery(QUERY, null );
        if
        (cursor.moveTo
        First()){ do{
            HashMap<String, String> map = new
            HashMap<>(); map.put("id",
            cursor.getString(0));
            map.put("name",
            cursor.getString(1));
            map.put("email",
            cursor.getString(2));
            list.add(map);
        } while (cursor.moveToNext());
        }
        cursor.close();
        return
        list;
    }

    public void insert(String name, String email){
        SQLiteDatabase database
        =this.getWritableDatabase();
        String QUERY = "INSERT INTO users (name, email)
VALUES (" + name + ", " + email + ")";
        database.execSQL(QUERY);
    }

    public void update(int id, String name, String
        email){ SQLiteDatabase

```

```

        database=this.getWritableDatabase();
        String QUERY= "UPDATE users SET name = '"+name+"',
        email
        = '"+email+"' WHERE id = "+id;
        database.execSQL(QUERY);
    }

    public void delete(int id){
        SQLiteDatabase database =
        this.getWritableDatabase(); String QUERY =
        "DELETE FROM users WHERE id =" +id;
        database.execSQL(QUERY);
    }
}

```

4. Membuat Paket adapter, tambahkan kelas Adapter pada paket Adapter

kemudian tuliskan kode Adapter.java berikut ini:

```

package
com.example.cruddata.adapter;
import
com.example.cruddata.model.Data;
import java.util.List;
import
android.content.Context;
import
android.view.LayoutInflater;
import android.view.View;
import
android.view.ViewGroup;
import
android.widget.BaseAdapter;
import
android.widget.TextView;
import
com.example.cruddata.R;
import
com.example.cruddata.model.Data;
import java.util.List;

public class Adapter extends
    BaseAdapter { private Activity
    activity;

```

```

private LayoutInflater
inflater; private
List<Data> list;

public Adapter(Activity activity, List<Data> list) {
    this.activity = activity;
    this.list=list;
}

@Override
public int
    getCount() {
        return
            list.size();
    }

@Override
public Object
    getItem(int i) { return
        list.get(i);
    }

@Override
public long
    getItemId(int i) {
        return i;
    }

@Override

public View getView(int i, View view, ViewGroup viewGroup)
    { if(inflater == null){
        inflater = (LayoutInflater)
activity.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    }
    if(view == null && inflater !=null){
        view = inflater.inflate(R.layout.list_user, null);
    }
    if (view !=null) {
        TextView          name          =
view.findViewById(R.id.text_name);  TextView
email   = view.findViewById(R.id.text_email);

```

```

        Data data = list.get(i);
        name.setText(data.getNa
        ma());
        email.setText(data.getE
        mail());
    }
    return view;
}
}

```

5. Membuat Paket model, tambahkan kelas Data pada paket model kemudian tuliskan kode Data.java berikut ini:

```

package
com.example.cruddata.model;
public class Data {
    private String id, name, email;
    public Data(String id, String name, String email) {
        this.id=id;
        this.name=na
        me;
        this.email=e
        mail;
    }

    public String
        getId() {
            return id;
        }

    public void setId(String id)
        { this.id = id;
        }

    public String
        getNama() {
            return name;
        }

    public void setNama(String
        nama) { this.name = nama;
    }
}

```

```

        public String
            getEmail() {
                return email;
            }

        public void setEmail(String
            email) { this.email = email;
        }
    }
}

```

6. Tambah Activity dengan nama = EditorActivity.java, kemudian tuliskan kode EditorActivity.java berikut ini:

```

package com.example.cruddata;

import

androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import
android.util.Log;
import
android.view.View;
import
android.widget.Button;
import
android.widget.EditText;
import
android.widget.Toast;

import

com.example.crud_12rpl1_the_best.helper.Helper;

public class EditorActivity extends

AppCompatActivity {
    private EditText editName, editEmail;
    private Button btn_Save;
    private Helper db = new Helper

```

```

        (this); private String id, name, email;
    @Override
    protected void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_editor);
        editName = findViewById(R.id.edit_name);
        editEmail = findViewById(R.id.edit_email);
        btn_Save = findViewById(R.id.btn_save);

        id = getIntent().getStringExtra("id");
        name =
        getIntent().getStringExtra("name");
        email =
        getIntent().getStringExtra("email");

        if (id == null || id.equals("")) {
            setTitle("Tambah Users");
        } else {
            setTitle("Edit Users");
            editName.setText(name);
            editEmail.setText(email);
        }
        btn_Save.setOnClickListener(new
            View.OnClickListener() { @Override
            public void
                onClick(View v) {
                    try{
                        if(id == null ||
                            id.equals("")){ save();
                        } else {
                            Edit();
                        }
                    } catch (Exception
                        e){ Log log =
                            null;
                            log.e("Saving", e.getMessage());
                        }
                    }
            });
    }

```

```

    }
    private void save(){
        if(String.valueOf(editName.getText()).equals("") ||
        String.valueOf(editEmail.getText()).equals("")){
            Toast.makeText(getApplicationContext(), "Silahkan isi semua
            data",
            Toast.LENGTH_SHORT).show();
        }else{
            db.insert(editName.getText().toString(),
            editEmail.getText().toString());
            finish();
        }
    }

    private void Edit(){
        if(String.valueOf(editName.getText()).equals("") ||
        String.valueOf(editEmail.getText()).equals("")){
            Toast.makeText(getApplicationContext(), "Silahkan isi semua
            data",
            Toast.LENGTH_SHORT).show();
        }else{
            db.update(Integer.parseInt(id),
            editName.getText().toString(), editEmail.getText().toString());
            finish();
        }
    }
}

```

7. Uji coba

BAB 7

Aplikasi Hand Speed

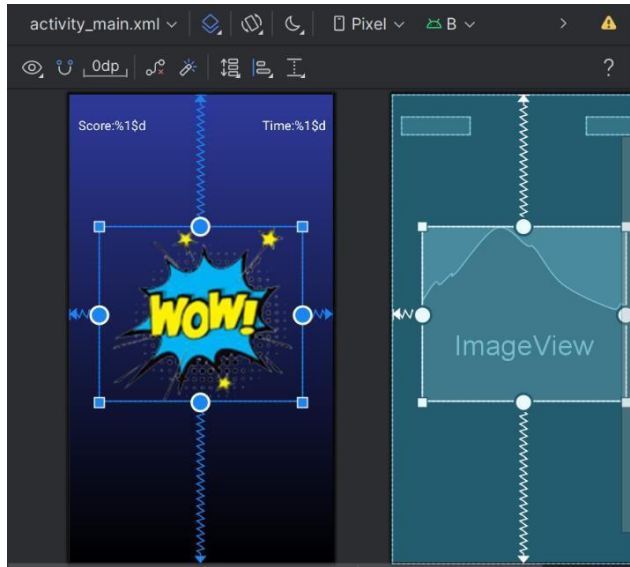
A. Pendahuluan

Aplikasi game hand speed adalah jenis game yang dirancang untuk menguji kecepatan tangan dan ketepatan pemain dalam melakukan tindakan tertentu. Game ini sering melibatkan tugas seperti menekan tombol dengan cepat, menghindari objek yang muncul tiba-tiba, atau menyelesaikan tantangan berdasarkan reaksi tangan yang cepat. Tujuannya adalah untuk melatih keterampilan motorik halus, meningkatkan waktu reaksi, serta melatih koordinasi antara mata dan tangan.

B. Membuat Aplikasi Game HandSpeed

Berikut ini adalah langkah-langkah membuat aplikasi game HandSpeed:

- Buatlah proyek baru dengan nama : HandSpeed, pilih bahasa : Kotlin, tentukan minimum SDK : API 30.
- Desainlah user interface dengan view atau tampilan sebagai berikut :



Gambar 9.1 Desain Interface

- Tulis Kode file main_activity.xml didalam Java Modul App
> res > layout > main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
xmlns:tools="http://schemas.android.com/tools"
```

```
xmlns:app="http://schemas.android.com/apk/res-auto" android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:background="@drawable/newbg"
```

```
tools:context=".MainActivity">
```

```
<TextView
```

```
android:id="@+id/game_score_text"
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```

android:text="Score:%1$d"
android:textColor="#fff"
android:textSize="20sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.045"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.049"
/>

```

```

<TextView
    android:id="@+id/time_left_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Time:%1$d"
    android:textColor="#fff"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.942"
    app:layout_constraintStart_toEndOf="@+id/game_score_text"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.049" />

```

```

<ImageView
    android:id="@+id/tap_me_image"
    android:layout_width="317dp"
    android:layout_height="273dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"

```

```

        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.451"
        app:srcCompat="@drawable/wawpng" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

- Tambah file baru dengan nama : btnbg.xml di Java Modul app > res > drawable > btnbg.xml dan tuliskan kode berikut ini:

```

<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android">
    <solid
        android:color="#ffffff"
    />
    <corners
        android:radius="200dp"
    />
</shape>

```

- Tambah file baru dengan nama : newbg.xml di Java Modul app > res > drawable > newbg.xml dan tuliskan kode berikut ini:

```

<?xml version="1.0" encoding="utf-8"?>
<shape
xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:startColor="#2e3a9b"
        android:endColor="#000"
        android:angle="270"
    />
</shape>

```

- Buka file MainActivity.kt di Java Modul app> kotlin+java > com.example > handspeed > MainActivity.kt, kemudian tuliskan kode Kotlin sebagai berikut:

```
package com.example.handspeed

import

android.annotation.SuppressLint

import android.content.IntentSender
import
androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import
android.os.CountDownTimer
import
android.widget.ImageView
import
android.widget.TextView
import android.widget.Toast

class MainActivity : AppCompatActivity() {
    private lateinit var gameScoreText :
    TextView private lateinit var timeLeftText
    : TextView private lateinit var
    tapMeImage : ImageView private var
    score=0
    private var gameStarted = false
    private lateinit var countDownTimer: CountDownTimer
    private var initialCountDown: Long = 60000
    private var countDownInterval: Long = 1000
    private var timeleft = 60

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        gameScoreText=findViewById(R.id.game_score_text)
        timeLeftText=findViewById(R.id.time_left_text)
        tapMeImage=findViewById(R.id.tap_me_image)
```

```

        tapMeImage.setOnClickListener{ incrementScore()}

        resetGame()
    }

    private fun
        incrementScore(){
            score++
            val newScore = getString(R.string.your_score, score)
            gameScoreText.text = newScore

            if(!gameStarted){
                startGame()
            }
        }

    }

    private fun
        resetGame(){
            score=0
            val initialScore = getString(R.string.your_score, score)
            gameScoreText.text = initialScore

            val initialTimeLeft = getString(R.string.time_left, 60)
            timeLeftText.text = initialTimeLeft

            countdownTimer=object :
                CountDownTimer(initialCountDown, countDownInterval){
                    override fun onTick(millisUntilFinished: Long) {
                        timeleft=millisUntilFinished.toInt()/1000
                        val timeleftString=getString(R.string.time_left,
timeleft)
                        timeLeftText.text=timeleftString
                    }

                    override fun
                        onFinish() {
                            endGame()
                        }
                }
        }
    gameStarted=false

```

```

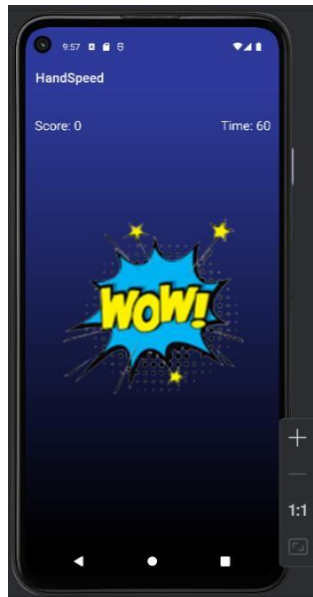
    }

    private fun startGame(){
        countdownTimer.start()
        gameStarted=true
    }

    private fun endGame(){
        Toast.makeText(this,
            getString(R.string.game_over_message,
                score), Toast.LENGTH_LONG).show()
        resetGame()
    }
}

```

- Untuk melihat tampilan dan mencoba game HandSpeed, jalankan aplikasi dengan menekan tombol Run 'App'. Jika aplikasi berhasil maka hasil pengujian dengan android virtual device akan tampil seperti gambar dibawah ini:



Gambar 9.1 Hasil pengujian dengan AVD
 Cara bermain Game HandSpeed adalah cara menekan gambar WOW secepat mungkin dimana waktu maksimal yang di sediakan

adalah 60 detik. Ketika gambar WOW ditekan pertama kali maka waktu akan menghitung mundur, dan ketika waktu yang disediakan habis maka akan muncul toas atau informasi score jumlah tekanan yang berikan pada tombol gambar WOW yang diperoleh dalam waktu 60 detik.



Gambar 9.2 Score

BIOGRAFI PENULIS



MARWIYAH HASIBUAN

Penulis lahir pada tanggal 29 September 1993 di Negeri Lama ,Kabupaten Labuhanbatu. Penulis menempuh Pendidikan di SD Negeri 112161 Mailil (2000-2006), MTS Teladan Guppi (2006-2009), Man Rantauprapat (2009-2012) dan STMIK Budidarma Medan Jenjang Diploma III (2012-2015).

Selama menjadi mahasiswa di STMIK Budidarma, penulis aktif di organisasi *Budidarma Networking Club* (BNC)

sebagai anggota. Kemudian Penulis melanjutkan studinya di Fakultas Sains dan Teknologi di Universitas Labuhanbatu – Rantau Prapat. Bidang minat yang diambil penulis adalah Sistem Informasi Selain masih aktif menjadi mahasiswa di Universitas Labuhanbatu, penulis juga saat ini bekerja Sebagai Aparatur Sipil Negara di Instansi Pemerintah Kabupaten Labuhanbatu Utara di Unit Dinas Kependudukan dan Pencatatan Sipil sebagai Pengelola Sistem dan Jaringan.



ALI AKBAR RITONGA

Lahir di Dusun Panduan Desa Silumajang Kec. NA.IX-X pada tanggal 24 Januari 1993. Menempuh pendidikan Sekolah Dasar di Sekolah Dasar Negeri 116901 Montong pada tahun 2006. Dan Melanjutkan Pendidikan Di MTS AT- Taufiqurrahman untuk tingkat Sekolah Menengah Pertama dan melanjutkan pendidikan di SMKN Negeri 4 Padang dengan konsentrasi Multimedia Komputer dan menyelesaikannya ada tahun 2011 Penulis melanjutkan pendidikan tingkat

tinggi ke Institut Teknologi Padang dan mengambil Jurusan Teknik Informatika. Dengan memperoleh Serjana Teknik. Selanjutnya penulis melanjutkan pendidikan program pasca serjana di Universitas Putra Indonesia YPTK Padang Sumatera Barat dan memperoleh gelar Magister Komputer pada tahun 2018. Penulis sekarang berprofesi

sebagai Dosen Tetap di Universitas Labuhanbatu, Penulis dapat dihubungi melalui email : aliakbarritonga@gmail.com.



IRMAYANTI

Lahir di Simonis 24 Agustus 1984. Setelah menamatkan SD Negeri 112324 Aek Kota Batu pada tahun 1996, Lalu melanjutkan sekolah Madrasah Tsanawiyah Daar Al-Ulum Asahan Kisaran lulus pada tahun 1999, SMU Al-Azhar Medan lulus pada tahun 2002, S1 Matematika UNIMED lulus pada tahun 2007, dan melanjutkan S2 Pendidikan Matematika lulus pada tahun 2013. Menikah dengan Beni Yunanta Hasibuan pada tahun 2009, mempunyai dua orang anak satu putri (Ratu Nabila Yunanta

Hasibuan) dan satu putra (Rauf Athaya Alby Yunanta Hasibuan). Memulai karir dari guru honor sampai diperguruan tinggi STKIP Labuhanbatu sebagai Dosen tetap. Melalui karya ini muda-mudahan dapat menjadi amal keluarga kami dan terkhusus kedua orang tua (Alm. H. Zulkarnaen Ritonga dan Hj. Normah Dalimunthe) semoga selalu mendapat rahmat dari Allah SWT. Lahir dari keluarga sederhana dengan didorong motivasi tinggi, Alhamdulillah dapat menerbitkan sebuah buku dengan pembahasan “Kalkulus, Statsistika, Teknik Riset Operasional, Analisis Variabel Kompleks, dll”. Besar harapan kiranya buku ini dapat berguna bagi mahasiswa dan masyarakat umum.



DECI IRMAYANI

Lahir di Rantauprapat pada tanggal 27 Mei 1986. Telah menyelesaikan Pendidikan Sarjana (S1) di STMIK Potensi Utama Medan dan melanjutkan Pendidikan Magister (S2) di UPI YPTK Padang. Saat ini bekerja sebagai Dosen di Universitas Labuhan Batu, Fakultas Sains dan Teknologi, dengan spesialisasi di bidang komputer. berkomitmen untuk memberikan

kontribusi dalam bidang pendidikan dan teknologi, serta membagikan pengetahuan kepada generasi mendatang.

SINOPSIS

Buku Pemrograman Android dengan Ladybug merupakan panduan praktis untuk mempelajari pembuatan aplikasi Android menggunakan pendekatan visual yang mudah dan intuitif. Ditujukan bagi pemula maupun pelajar yang belum memiliki pengalaman pemrograman, buku ini menghadirkan metode pembelajaran yang lebih ramah pengguna melalui platform Ladybug—sebuah alat bantu pengembangan aplikasi Android berbasis blok visual (visual programming).

Isi buku disusun secara bertahap dan sistematis, dimulai dari pengenalan antarmuka Ladybug, pembuatan desain antarmuka pengguna (UI), hingga penyusunan logika aplikasi tanpa perlu menulis kode secara manual. Pembaca akan diajak mengembangkan aplikasi-aplikasi sederhana hingga menengah, sambil memahami konsep dasar pemrograman, struktur aplikasi Android, serta prinsip kerja komponen seperti Activity, Intent, dan Layout.

Melalui pendekatan “drag and drop” serta visualisasi logika yang mudah dipahami, Ladybug menjadi media pembelajaran ideal untuk mengenalkan dunia pengembangan aplikasi mobile tanpa intimidasi kompleksitas teknis. Buku ini juga dilengkapi dengan studi kasus, latihan proyek, dan tips pengembangan untuk mempermudah pemahaman serta penerapan langsung.



PENERBIT
YAYASAN MMI

ISBN 978-634-04-1799-9



9

786340

417999