

Berkreasi dengan **ANDROID STUDIO**

Rahasia Membangun Aplikasi Android dalam Pendidikan

Android Studio, tools yang sangat powerful untuk membangun aplikasi Android. Dalam buku ini akan dipelajari langkah demi langkah dalam membuat sebuah proyek, menambahkan elemen-elemen penting seperti tombol, gambar, dan teks, yang menjadikan aplikasi tampak profesional dan interaktif. Tak hanya itu, buku juga memandu kita untuk menguasai berbagai layout yang memungkinkan kita merancang tampilan aplikasi yang menarik dan responsif—semua dengan cara yang mudah, tanpa memerlukan pengalaman coding yang rumit. Memahami konsep Activity—komponen utama yang membuat aplikasi kita dapat bergerak dan berfungsi. Dengan demikian kita akan belajar cara menambahkan berbagai interaksi pengguna seperti tombol yang bisa ditekan, tampilan pesan yang muncul (Toast), serta elemen dropdown yang membuat aplikasi kita semakin dinamis dan menarik. Buku ini juga membahas penggunaan logika dan perhitungan dalam aplikasi, seperti menghitung luas, keliling, hingga merancang sistem yang lebih kompleks. Dengan latihan-latihan yang mengasah keterampilan, kita akan semakin percaya diri dalam mengembangkan aplikasi untuk memecahkan masalah.

Puncaknya, buku ini memperkenalkan kepada kita bagaimana menghubungkan aplikasi kita dengan database (*BigData*). Aplikasi yang tidak hanya menampilkan informasi, tetapi juga dapat menyimpan, mengambil, dan mengubah data secara real-time. Dengan membangun API untuk operasi CRUD (*Create, Read, Update, Delete*) dengan JSON dan menghubungkan aplikasi kita ke database MySQL, sehingga kita akan mampu menciptakan aplikasi yang memiliki fungsionalitas luar biasa.

Buku ini bukan hanya untuk programmer berpengalaman, akan tetapi juga sangat cocok untuk pemula yang ingin belajar membuat aplikasi Android dan awal, bahkan jika kita belum pernah menulis kode pemrograman sebelumnya hingga pada aplikasi yang dikembangkan dapat diupload ke Play Store. Setiap langkah dijelaskan secara rinci, dan latihan-latihan yang disediakan akan membantu kita mempraktikkan ilmu yang baru dipelajari.

wadegroup .id



Berkreasi dengan Android Studio
Rahasia Membangun Aplikasi Android dalam Pendidikan

Dr. Supratman Zakir, S.Kom, M.Pd., M.Kom
Efmi Maiyana, S.Kom., M.Kom
Sadar Martua Haholongan Sir

wadegroup .id

Berkreasi dengan **ANDROID STUDIO**

Rahasia Membangun Aplikasi Android
dalam Pendidikan



Dr. Supratman Zakir, S.Kom, M.Pd., M.Kom
Efmi Maiyana, S.Kom., M.Kom
Sadar Martua Haholongan Sir

Berkreasi dengan

Android Studio

Rahasia Membangun Aplikasi Android dalam Pendidikan

Dr. Supratman Zakir, S.Kom. M.Pd., M.Kom
Efmi Maiyana, S.Kom., M.Kom
Sadar Martua Haholongan Sir

Sanksi Pelanggaran Pasal 113
Undang-undang Nomor 28 Tahun 2014
Tentang Hak Cipta

1. Setiap orang yang dengan atau tanpa hak melakukan pelanggaran terhadap hak ekonomi yang sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan ancaman pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp. 100.000.000 (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 Rp. 500.000.000 (lima ratus juta rupiah).
3. Setiap orang 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 a, huruf b, huruf e, dan/atau huruf g untuk penggunaan Secara Komersial dipidana dengan pidana penjara paling lama (empat) tahun dan/atau pidana denda paling banyak Rp. 1.000.000.000 (satu miliar rupiah).
4. Setiap orang memenuhi unsur sebagaimana dimaksud ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/ atau pidana denda paling banyak Rp. 4.000.000.000 (empat miliar rupiah)

Berkreasi dengan

Android Studio

**Rahasia Membangun Aplikasi
Android dalam Pendidikan**

Berkreasi dengan Android Studio

Rahasia Membangun Aplikasi Android dalam Pendidikan

© Dr. Supratman Zakir, S.Kom. M.Pd., M.Kom
Efmi Maiyana, S.Kom., M.Kom
Sadar Martua Haholongan Sir

Editor : Team wadegroup.id
Layout : Dana Muntaha
Design Cover : Dana Muntaha

Diterbitkan oleh:



Jl. Pos Barat Km.1, Melikan, Ngimput,
Purwosari, Babadan, Ponorogo, Jawa Timur

Anggota IKAPI 182/JTI/2017

Cetakan Pertama, Oktober 2025

ISBN: 978-623-6243-50-3

Hak Cipta dilindungi undang-undang.

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa seizin tertulis dari Penerbit.

15x23cm



KATA PENGANTAR

Alhamdulillah, Puji Sukur kehadiran Allah SWT, berkat Rahmat-Nya buku ini sampai ke tangan pembaca. Selawat dan salam kita doakan kepada Allah semoga disampaikan shalawat kita kepada Nabi Muhammad SAW. Selamat datang dalam perjalanan mengembangkan keterampilan di dunia pengembangan aplikasi Android melalui buku ini, "Berkreasi dengan Android Studio: Rahasia Membangun Aplikasi Android dalam Pendidikan." Buku ini dirancang untuk memberikan wawasan yang mendalam dan praktis mengenai cara membangun aplikasi Android menggunakan Android Studio, alat yang menjadi standar bagi para pengembang aplikasi di seluruh dunia.

Pendidikan di era digital kini semakin mengutamakan keterampilan praktis yang langsung diterapkan dalam kehidupan sehari-hari. Dunia teknologi terus berkembang, dan salah satu keahlian yang sangat dibutuhkan adalah kemampuan untuk menciptakan aplikasi yang dapat mempermudah aktivitas manusia. Android, sebagai sistem operasi paling banyak digunakan di dunia, menawarkan peluang besar untuk para pengembang aplikasi, terutama dalam konteks pendidikan.

Buku ini hadir sebagai panduan yang menggabungkan aspek pembelajaran dengan penerapan langsung. Dalam setiap bab, pembaca akan diajak untuk tidak hanya memahami konsep-konsep dasar dalam pengembangan aplikasi Android, tetapi juga untuk mengaplikasikannya dalam proyek nyata. Kami percaya bahwa dengan pendekatan yang terstruktur dan praktis, pembaca akan mampu menguasai Android Studio dan

membangun aplikasi Android yang tidak hanya fungsional, tetapi juga inovatif.

Dengan langkah-langkah yang jelas dan mudah dipahami, buku ini memberikan panduan yang ideal bagi para pemula yang ingin memulai karier di dunia pengembangan aplikasi, baik sebagai bagian dari pendidikan formal maupun sebagai keterampilan tambahan dalam kehidupan profesional. Selain itu, buku ini juga memberikan wawasan bagi para pengembang yang ingin memperdalam pengetahuan mereka dalam menggunakan Android Studio untuk membangun aplikasi Android yang lebih kompleks.

Selamat membaca, dan semoga buku ini menjadi langkah awal pembaca dalam menciptakan aplikasi-aplikasi Android yang inspiratif dan bermanfaat!

Salam Literasi

Penulis

DAFTAR ISI

KATA PENGANTAR.....	5
DAFTAR ISI.....	7
BAB I PENGENALAN ANDROID.....	9
1. Pengertian Andoroid	9
2. Sejarah Singkat Android.....	10
3. Perkembangan Android	11
4. Keunggulan Android.....	19
BAB II ANDROID STUDIO.....	21
1. Menenal Android Studio.....	21
2. Spesifikasi Hardware.....	21
3. Membuat Projek Baru Di Android Studio	22
4. Cara Menjalankan Projek Menggunakan Emolator Android Studio	26
5. Menjalankan Projek Langsung Handphone	30
6. Attributes Dalam Android Studio Android Studio	32
7. Menambahkan Elemen Dalam Android Studio	41
8. Menambahkan Foto Dan Image Button Dalam Android Studio	41
9. Edit Text Dalam Android Studio	46
10. Scroll View.....	49
BAB III MEMBANGUN ACIVITY	57
1. Pengertian Activity	57
2. Menambah Activity	57
BAB IV LAYOUT ANDRIOD STUDIO.....	61
1. Layout Android Studio	61
2. Menambah Layout.....	61
3. Jenis - Jenis Layout	63

BAB V PENGGUNAAN BUTTON, TOAST, STRING DAN SPINNER.....	101
1. Penggunaan String Pada Button Menggunakan OnClick	101
2. Penggunaan Toast Pada Button Menggunakan setOnClickListener.....	111
3. Spinner Dan Contoh Penggunaannya.....	114
BAB VI VARIABEL DAN LOGIKA	123
1. Variabel Dan Deklarasinya	123
2. Operator Aritmatika Dan Logika	124
3. Latihan	124
BAB VII INTENT	179
1. Pengertian Intent.....	179
2. Contoh Penggunaan Implicit Intent	180
3. Contoh Penggunaan Explicit Intent	187
BAB VIII CRUD dengan Json dan Database MySQL	193
1. Database Dengan JSON.....	193
2. Instalasi XAMPP Sebagai Database.....	193
3. Create, Read, Update dan Delete Android Studio.....	200
BAB IX Build APK.....	245
1. Pengertian Dan Langkah Pembuatan.....	245
BAB X Unggah Aplikasi Ke Google Play Store.....	251
DAFTAR PUSTAKA.....	257
BIODATA PENULIS.....	259
INDEKS	265

BAB I

PENGENALAN ANDROID



ANDROID

1. Pengertian Andoroid

Berdasarkan keterangan dari Nazaruddin, Android adalah sistem operasi berbasis Linux yang dirancang untuk ponsel. Android menawarkan platform terbuka yang memungkinkan para pengembang menciptakan aplikasi mereka sendiri, yang dapat dioperasikan pada berbagai perangkat mobile. Sistem operasi ini sering digunakan pada smartphone dan tablet PC. Fungsinya serupa dengan sistem operasi

Symbian pada perangkat Nokia, iOS pada Apple, dan BlackBerry OS.

Menurut Hendra Nugraha Lengkong, Android adalah salah satu platform perangkat smartphone. Keunggulan utama Android adalah lisensinya yang bersifat terbuka (open source) dan gratis, sehingga dapat dikembangkan tanpa biaya royalti atau didistribusikan dalam berbagai bentuk. Hal ini memberikan kemudahan bagi para programmer dalam menciptakan aplikasi baru di dalam platform tersebut.

Jadi dapat disimpulkan Android merupakan *Operating System* (OS) berbasis Linux yang sifatnya open source. Dengan lisensi open source dan gratis, Android memudahkan pengembang menciptakan aplikasi baru tanpa batasan biaya, menjadikannya pilihan yang fleksibel dan inovatif.

2. Sejarah Singkat Android

Awalnya, Android dikembangkan oleh Android Inc. sebagai sistem operasi berbasis Linux dengan dukungan dari Google Finance hingga akhirnya diakuisisi oleh Google pada tahun 2005. Android resmi diluncurkan pada 5 November 2007 bersamaan dengan pendirian Open Handset Alliance, sebuah perusahaan telekomunikasi yang bertujuan untuk meningkatkan standar perangkat seluler. Google merilis kode Android di bawah lisensi Apache, yang memungkinkan pengguna



Gambar 1.1 Android

memanfaatkan dan mengembangkan perangkat lunak ini dengan standar terbuka.

Ada dua jenis distributor resmi Android: pertama, yang mendapat dukungan penuh dari Google (Google Mail Services), dan kedua adalah Open Handset Distribution (OHD) yang benar-benar bebas dari dukungan Google. Android didesain untuk perangkat mobile dengan layar sentuh seperti smartphone dan tablet, menyesuaikan spesifikasi dari kelas low-end hingga high-end, sehingga popularitasnya semakin meningkat.

Sebagai sistem operasi open source, Android memungkinkan siapa saja untuk mengembangkannya sesuai keinginan. Platform ini memiliki gudang aplikasi bernama Google Play Store, di mana pengguna dapat mengunduh aplikasi dan game secara bebas. Uniknya, setiap versi Android diberi nama makanan, dengan huruf depan yang menandakan peningkatan versi.

Penggunaan sistem operasi Android pada smartphone kini sangat populer di kalangan perusahaan pembuat ponsel. Keunggulannya sebagai software open source memungkinkan banyak aplikasi gratis diunduh oleh pengguna. Hal ini membuat smartphone dengan sistem operasi Android lebih terjangkau dibandingkan dengan perangkat yang menggunakan sistem operasi berbayar.

3. Perkembangan Android

Di dunia teknologi yang terus berputar, Android lahir sebagai sebuah inovasi yang menjanjikan. Pada tanggal 5 November 2007, versi Beta Android pertama kali diperkenal-

kan, menciptakan gelombang baru dalam pengalaman pengguna ponsel. Dengan semangat yang menggebu, versi komersialnya muncul pada 23 September 2008, menyuguhkan berbagai fitur menarik seperti Android Market, web browser, Gmail, dan Maps, yang membuka jendela dunia bagi penggunanya. Kemudian, pada 9 Maret 2009, Android melangkah lebih jauh dengan peluncuran versi 1.1. Dalam pembaruan ini, Android memperkenalkan sejumlah fitur baru, termasuk Google Mail Service (GMS), alarm clock, dan pencarian suara, yang semakin memperkaya pengalaman komunikasi dan interaksi pengguna. Setiap sentuhan baru terasa seperti melodi yang harmonis, membawa nuansa segar ke dalam dunia mobile.

Tepat pada 30 April 2009, versi 1.5 yang dikenal sebagai Cupcake muncul, membawa bersama keajaiban fitur baru. Kemampuan mengunggah video ke YouTube, integrasi home screen dengan widgets, serta fungsi salin dan tempel dalam browser memberikan kebebasan bagi para pengguna untuk mengekspresikan diri dan berkreasi. Cupcake menjadi simbol dari pertumbuhan dan inovasi, seolah mengundang para pengguna untuk menjelajahi kemungkinan tanpa batas.

Kemudian, pada 15 September 2009, versi 1.6 yang bernama Donut diperkenalkan, membawa fitur unggulan seperti integrasi kamera, video, dan galeri. Versi ini juga mendukung layar dengan resolusi WVGA, menawarkan kejelasan dan ketajaman visual yang sebelumnya tidak pernah terbayangkan. Dengan perbaikan pada Google Play, pengguna kini bisa menemukan berbagai aplikasi dengan lebih mudah, seolah mengubah smartphone mereka menjadi sebuah portal ke dunia digital.

Tak lama setelahnya, pada 9 Desember 2009, versi 2.0 - 2.1 atau Eclair lahir, menyuguhkan inovasi yang semakin canggih. Dengan dukungan untuk Google Maps Beta, pencarian SMS, dan Bluetooth 2.1 untuk transfer data yang lebih cepat, Eclair menjadi jembatan antara dunia fisik dan digital. Setiap pembaruan membawa harapan baru dan kemungkinan yang lebih besar.

Melalui setiap langkah dan perubahan, Android tidak hanya berkembang sebagai sebuah sistem operasi, tetapi juga sebagai teman setia yang mendampingi penggunanya dalam perjalanan digital. Dengan semangat inovasi dan kreativitas, Android terus menyemarakkan dunia mobile, menjadi simbol dari kemudahan, kebebasan, dan eksplorasi yang tak terbatas.

Di tengah perjalanan yang penuh inovasi, Android melanjutkan evolusinya dengan memperkenalkan versi 2.2 yang dikenal sebagai Froyo (Frozen Yoghurt) pada 20 Mei 2010. Versi ini bukan sekadar peningkatan biasa; ia datang dengan fitur-fitur canggih yang menggugah rasa ingin tahu para pengguna. Dukungan untuk Adobe Flash, kemampuan Hotspot Portable, serta perekaman video berkualitas HD menjadikannya sebagai salah satu tonggak penting dalam sejarah Android.

Kemudian, pada 6 Desember 2010, Android mengumumkan kelahiran Gingerbread, versi 2.3 hingga 2.3.7. Dalam pembaruan ini, antarmuka pengguna diperbaharui untuk lebih hemat energi, menambahkan dukungan untuk NFC, keyboard virtual, dan peningkatan fasilitas salin-tempel. Gingerbread adalah simbol dari kemajuan, mengajak pengguna untuk menjelajahi dunia digital dengan lebih efisien.

Masuki era baru pada Honeycomb, yaitu versi 3.0 hingga 3.2. Dirilis khusus untuk perangkat tablet, Android menghadirkan antarmuka yang lebih ramah pengguna, serta fitur multitasking yang memungkinkan pengguna untuk beralih antar aplikasi dengan mudah. Ini adalah langkah besar untuk menjadikan tablet sebagai alat produktivitas yang mumpuni.

19 Oktober 2011 menjadi momen bersejarah dengan peluncuran Android 4.0 (Ice Cream Sandwich). Fitur baru seperti Face Unlock membawa pengalaman yang futuristik, sementara perbaikan dalam input teks dan suara serta tombol virtual menggantikan tombol fisik, menjadikannya lebih responsif dan intuitif. Ice Cream Sandwich menggabungkan yang terbaik dari smartphone dan tablet dalam satu platform yang harmonis.

Pada 9 Juli 2012, Android 4.1 muncul sebagai Jelly Bean. Versi ini memperkenalkan fitur-fitur inovatif seperti Google Now, antarmuka pengguna yang lebih halus, widget di layar kunci, dan dukungan untuk Bluetooth Smart Ready. Jelly Bean membawa kecepatan dan responsivitas yang lebih baik, membuat pengalaman pengguna semakin memuaskan.

Setelah itu, pada 31 Oktober 2013, Android memperkenalkan KitKat (versi 4.4). Pembaruan kali ini menciptakan antarmuka yang lebih canggih, fitur perekaman layar, dukungan untuk cetak nirkabel, serta peningkatan dalam keamanan dan performa. KitKat bukan hanya menambah kemudahan, tetapi juga memberi rasa aman kepada penggunanya.

Dengan hadirnya Android 5.0 (Lollipop), yang dirilis setelah KitKat, sistem operasi ini mengalami transformasi signifikan. Material Design menjadi ciri khas antarmuka yang lebih berwarna dan responsif. Lollipop juga memperkenalkan

fitur Project Volta, yang berfungsi untuk menghemat daya baterai hingga 30%, menjadikan perangkat lebih tahan lama dalam penggunaan sehari-hari.

Tidak berhenti di situ, Android 6.0 (Marshmallow) memperkenalkan perintah suara yang lebih fungsional dan menjanjikan daya tahan baterai yang lebih baik berkat fitur Doze. Dengan dukungan untuk sistem pengisian daya menggunakan USB-C, proses pengisian menjadi lebih cepat, memberikan kenyamanan bagi pengguna dalam menjalani aktivitas digital mereka.

Melalui setiap pembaruan, Android telah membuktikan dirinya sebagai pionir dalam dunia teknologi, terus berinovasi dan beradaptasi dengan kebutuhan penggunanya. Setiap versi membawa keajaiban baru, memperkaya pengalaman, dan membuka jalan menuju masa depan yang lebih cerah dan terhubung. Sejak peluncurannya, Android terus bertransformasi dan menghadirkan inovasi yang memikat. Android 10, yang resmi dirilis pada September 2019, memperkenalkan berbagai fitur menarik. Salah satu perubahan paling dinanti adalah mode gelap sistem, yang tidak hanya menawarkan tampilan yang lebih elegan tetapi juga membantu mengurangi kelelahan mata. Dengan kontrol privasi yang diperbarui, pengguna kini dapat lebih leluasa mengatur informasi pribadi mereka. Ditambah dengan navigasi gesek yang lebih intuitif, Android 10 memberikan pengalaman pengguna yang lebih lancar dan menyenangkan. Selain itu, Project Mainline menjadi terobosan baru, memungkinkan pembaruan komponen sistem langsung melalui Google Play, menjadikan perangkat lebih aman dan terkini tanpa harus menunggu versi lengkap.

Mengikuti kesuksesan sebelumnya, Android 11 hadir pada September 2020 dengan sejumlah perbaikan signifikan. Versi ini mempersembahkan peningkatan pada notifikasi, membuatnya lebih mudah diakses dan dikelola. Kontrol media yang ditingkatkan memberikan pengalaman yang lebih baik dalam mengelola suara dan video. Fitur izin satu kali menjadi langkah maju dalam menjaga privasi pengguna, memberikan kendali penuh atas aplikasi yang mengakses data tertentu. Selain itu, Bubbles menghadirkan cara baru untuk berkomunikasi dengan percakapan yang terapung di atas aplikasi lain, sementara Device Controls memudahkan pengguna dalam mengelola perangkat IoT dari satu tempat, menciptakan pengalaman rumah pintar yang lebih terintegrasi.

Tidak berhenti di situ, Android 12 melangkah lebih jauh dengan peluncuran versi beta pada Mei 2021. Pembaruan desain yang signifikan dengan Material You memberikan pengguna kebebasan untuk menyesuaikan tema dan gaya antarmuka sesuai dengan preferensi mereka. Hal ini menciptakan pengalaman yang lebih personal dan menarik, menjadikan setiap perangkat unik mencerminkan karakter penggunanya.

Versi selanjutnya adalah Android 13 merupakan versi ke-20 dari sistem operasi Android. Sistem operasi ini pertama kali diumumkan pada 10 Februari 2022, dan peluncurannya terjadi sekitar empat bulan setelah versi stabil Android 12 dirilis. Android 13 berfokus pada peningkatan fitur Android 12L, serta menambahkan privasi, keamanan, dan pengoptimalan antarmuka pengguna.

Salah satu fitur baru di Android 13 adalah pemilih foto. Fitur ini memungkinkan pengguna untuk membatasi akses aplikasi hanya pada foto atau video tertentu, berbeda dengan

versi sebelumnya yang memberikan akses ke seluruh galeri. Selain itu, Android 13 juga mendukung kustomisasi ikon aplikasi agar selaras dengan warna tema atau wallpaper yang digunakan.

Android 14 resmi diluncurkan pada 4 Oktober 2023, menghadirkan berbagai pembaruan signifikan, termasuk peningkatan kualitas foto, tema yang lebih variatif, dan kemampuan membuat wallpaper menggunakan kecerdasan buatan (AI). Selain itu, Android 14 juga memperkuat privasi dalam hal kesehatan, keselamatan, serta perlindungan data pribadi.

Fitur Ultra HDR pada Android 14 memungkinkan pengguna untuk mengambil gambar HDR yang dapat dinikmati di layar HDR dengan warna lebih hidup, sorotan cahaya lebih tajam, dan detail bayangan yang lebih gelap. Pembaruan ini juga mencakup tema monokrom yang diterapkan pada widget, ikon, tombol, latar belakang, dan animasi, memberikan tampilan yang seragam dan modern.

Hal yang menarik dari Android 14 adalah kemampuan AI dalam menciptakan wallpaper. Pengguna dapat merancang wallpaper unik untuk mengekspresikan diri. Selain itu, sistem secara proaktif memberikan informasi tentang izin aplikasi, memungkinkan pengguna mengontrol akses yang diberikan kepada aplikasi. Fitur lainnya adalah kemampuan Android 14 menggunakan lampu kilat kamera sebagai notifikasi visual, menambah fungsionalitas yang menarik.

Dengan setiap versi baru, Android terus menunjukkan komitmennya untuk meningkatkan pengalaman pengguna, menghadirkan fitur-fitur inovatif yang memudahkan

kehidupan sehari-hari dan membawa teknologi lebih dekat kepada penggunanya.

Berikut tabel perkembangan android:

No	Nama Versi	Tahun Rilis
1.	Beta	2007
2.	Android 1.1	2008
3.	Cupcake	2009
4.	Donut	2009
5.	Eclair	2009
6.	Froyo	2010
7.	Gingerbread	2010
8.	Honeycomb	2011
9.	Ice Cream Sandwich	2011
10.	Jelly Bean	2012
11.	KitKat	2013
12.	Lollipop	2014
13.	Marshmallow	2015
14.	Nougat	2016
15.	Oreo	2017
16.	Pie	2018
17.	Android 10 (Android Q)	2019
18.	Android 11 (Red Velvet Cake)	2020
19.	Android 12 (Snow Cone)	2021
20.	Android 13 (Tiramisu)	2022
21.	Android 14 (Upside Down Cake)	2023

4. Keunggulan Android

Keunggulan android cukup banyak, diantaranya yaitu:

1) Open Source

Android merupakan sistem operasi yang bersifat open source, yang memungkinkan banyak pengembang untuk menciptakan aplikasi bagi platform ini.

2) Instant Run

Fitur Instant Run memungkinkan pengguna untuk menjalankan program yang telah dikembangkan dengan cepat melalui perintah run, mempercepat proses pengujian aplikasi.

3) Fungsi Intelligent Code Editor

Dengan Android Studio, Anda dapat menulis kode dengan lebih efisien, cepat, dan produktif. Fitur Intelligent Code Editor memudahkan analisis kode serta menyediakan saran kode melalui sistem auto-complete.

4) Emulator yang Kaya Fitur

Android Studio dilengkapi dengan emulator yang kaya fitur, memungkinkan pengujian aplikasi langsung pada berbagai perangkat Android seperti ponsel, tablet, smartwatch, dan Smart TV. Anda juga dapat mensimulasikan berbagai fitur perangkat keras, termasuk lokasi GPS, sensor gerak, kondisi baterai, dan latensi jaringan.

5) Sistem yang Fleksibel

Android Studio menawarkan fleksibilitas dengan otomatisasi versi, manajemen dependensi, dan konfigurasi yang dapat disesuaikan. Anda dapat mengonfigurasi proyek untuk menyertakan library lokal maupun yang di-host, serta menjelaskan varian versi yang mengandung kode berbeda.

6) Kemampuan untuk Membuat Aplikasi Lengkap.

Dengan Android Studio, Anda memiliki kemampuan untuk mengembangkan aplikasi yang komprehensif, memenuhi berbagai kebutuhan pengguna.

7) Dapat Digunakan di Semua Perangkat Android

Android Studio menyediakan platform untuk mengembangkan aplikasi Android yang dapat dijalankan di berbagai perangkat, termasuk tablet, Android Wear, Android TV, dan Android Auto. Struktur fungsional ini memungkinkan Anda membagi proyek menjadi unit-unit yang lebih teratur, sehingga mempermudah proses pengembangan aplikasi Android.

BAB II

ANDROID STUDIO

1. Mengenal Android Studio

Android Studio adalah perangkat lunak yang menyediakan *Integrated Development Environment (IDE)*, dirancang khusus untuk mengembangkan aplikasi berbasis Android dengan dukungan dari IntelliJ IDEA. Bahasa pemrograman utama yang digunakan adalah Java, sementara XML digunakan untuk mendesain antarmuka atau format presentasi.

Android Studio juga terintegrasi dengan Software Development Kit (SDK), yang memungkinkan distribusi aplikasi ke perangkat Android. Sebelum menginstal Android Studio, komputer Anda harus memiliki SDK yang berfungsi sebagai alat API (Application Programming Interface) untuk membangun aplikasi Android. Android sendiri terdiri dari sistem operasi, middleware, dan aplikasi.

2. Spesifikasi Hardware

Spesifikasi hardware minimum yang bisa digunakan untuk menjalankan Android Studio dikutip dari web resmi android yaitu:

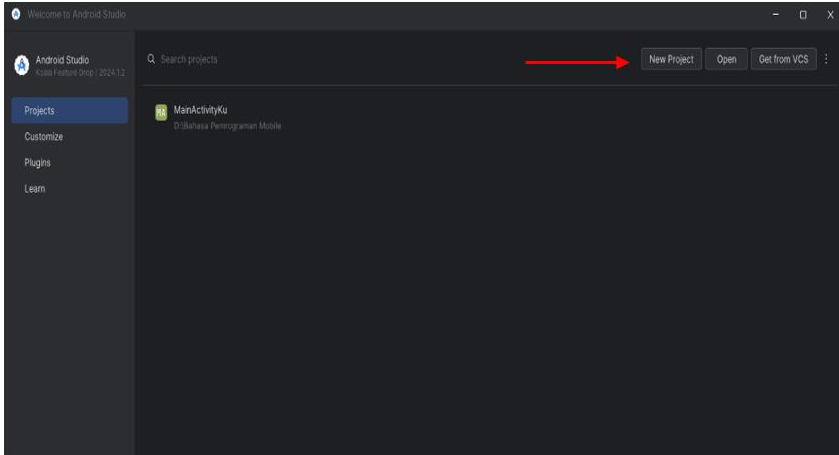
1. Untuk sistem operasi windows:
 - a) Microsoft® Windows® 7/8/10 (32/ 64-bit)
 - b) RAM minimum 4 GB, RAM yang disarankan 8 GB; tambah 1 GB untuk Emulator Android
 - c) Ruang disk minimum yang tersedia 2 GB, Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
 - d) Resolusi layar minimum 1280 x 800
 - e) Untuk emulator akselerasi: sistem operasi 64-bit dan prosesor Intel® dengan dukungan untuk Intel® VT-x, Intel® EM64T(Intel® 64), dan fungsionalitas Execute Disable (XD) Bit

2. Untuk sistem operasi Mac Os:
 - a) Mac® OS X® 10.10 (Yosemite) atau lebih tinggi, hingga 10.12 (macOS Sierra)
 - b) RAM minimum 3 GB, RAM yang disarankan 8 GB; tambah 1 GB untuk Emulator Android
 - c) Ruang disk minimum yang tersedia 2 GB, Disarankan 4 GB (500 MB untuk IDE + 1,5 GB untuk Android SDK dan gambar sistem emulator)
 - d) Resolusi layar minimum 1280 x 800.

3. Membuat Projek Baru Di Android Studio

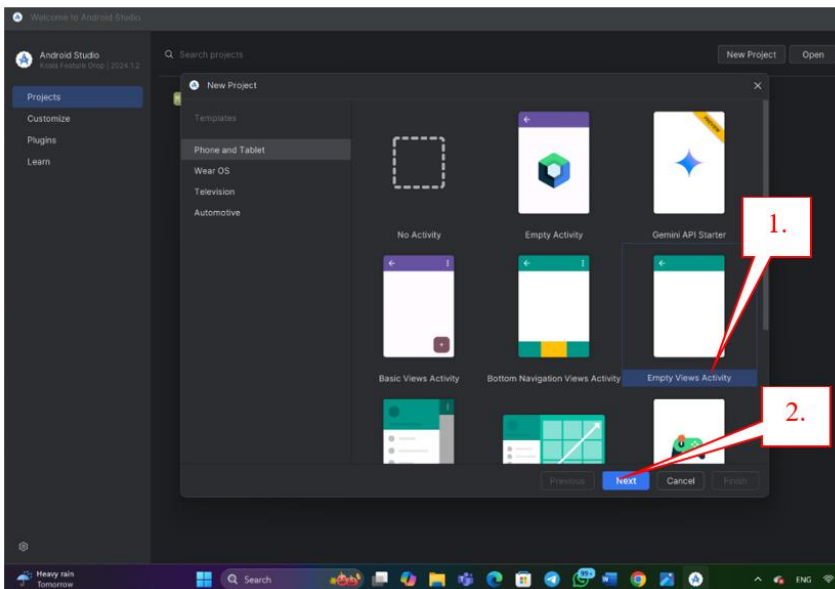
Dalam membuat projek di android studio ada beberapa langkah yang harus dilakukan diantaranya:

Langkah 1: Buka Android studio lalu klik new project



Gambar 2.1 Langkah 1 Pembuatan Projek Baru

Langkah 2: Kemudian klik empty views activity, kemudian pilih next.

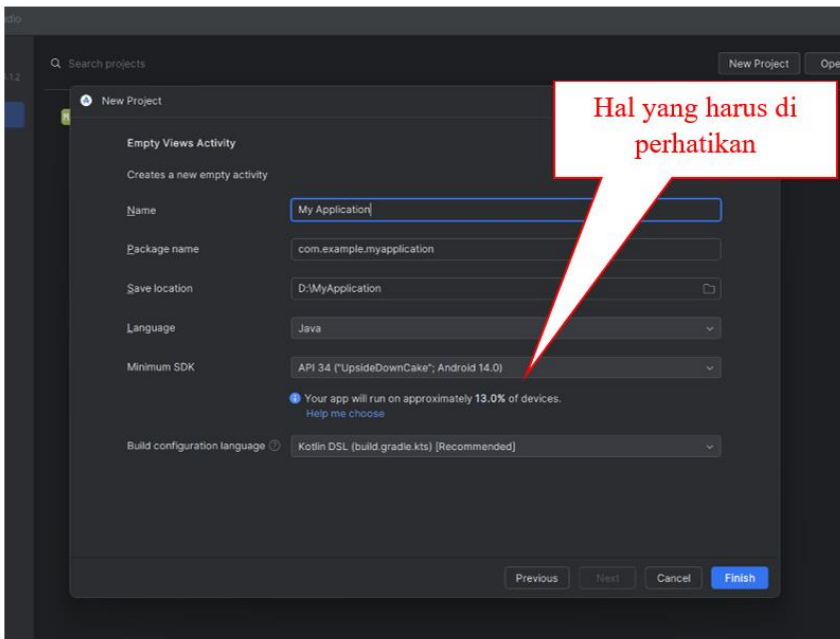


Gambar 2.2 Langkah 2 Pembuatan Projek Baru

Adapun hal yang harus diperhatikan yaitu pastikan kita menggunakan template phone and tablet.

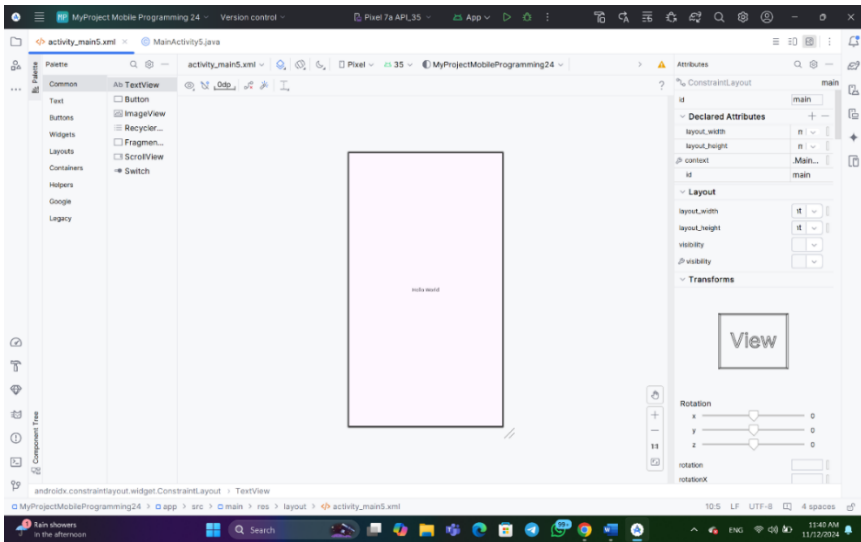
Langkah 3:

Dalam pembuatan projek yang ingin kita buat ada beberapa hal yang harus diperhatikan, yaitu dalam Minimum SDK. Penetapan itu sangatlah penting karena minimum SDK akan menentukan aplikasi atau projek yang kita buat dapat digunakan di android yang memiliki versi android yang telah kita tentukan di minimum SDK. Misalnya kita membuat projek dengan minimum SDK 33, maka android dengan versi dibawahnya tidak bisa menggunakan projek kita. Setelah memberikan identitas dari project kita lalu klik finish maka project telah siap dibuat.

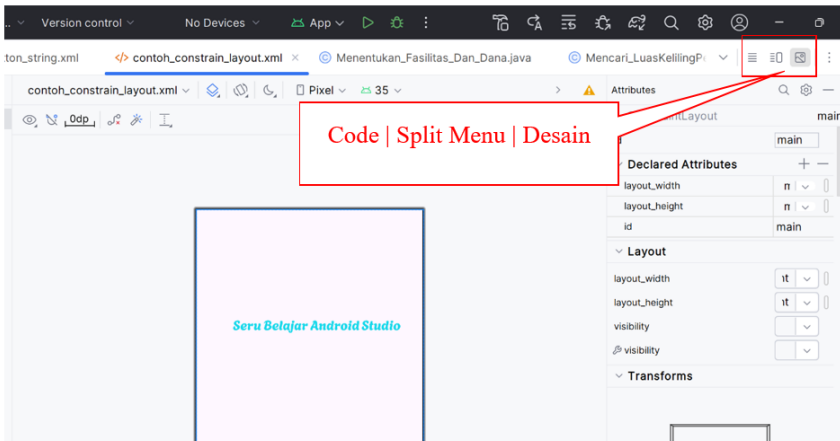


Gambar 2.3 Langkah 4 Pembuatan Projek Baru

Setelah itu akan muncul tampilan sebagai berikut:



Gambar 2.4 Lembar Proyek Baru



Gambar 2.5 Split Menu Pada Android Studio

Icon Code untuk melihat Koding Aplikasi, *Icon split* berfungsi untuk membagi layar menjadi 2 tampilan yaitu tampilan koding dan tampilan desain, Icon ini sangat berguna

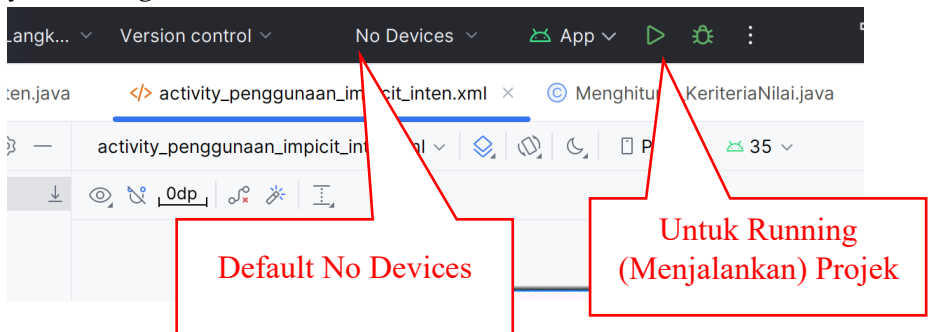
untuk langsung bisa melihat hasil sebuah Desain saat lagi membuat koding, sedangkan *Icon Desain* berguna untuk mendesain layout tampilan

4. Cara Menjalankan Proyek Menggunakan Emulator Android Studio

Melakukan running menggunakan emulator yang tersedia pada android studio dapat menggunakan beberapa langkah diantaranya:

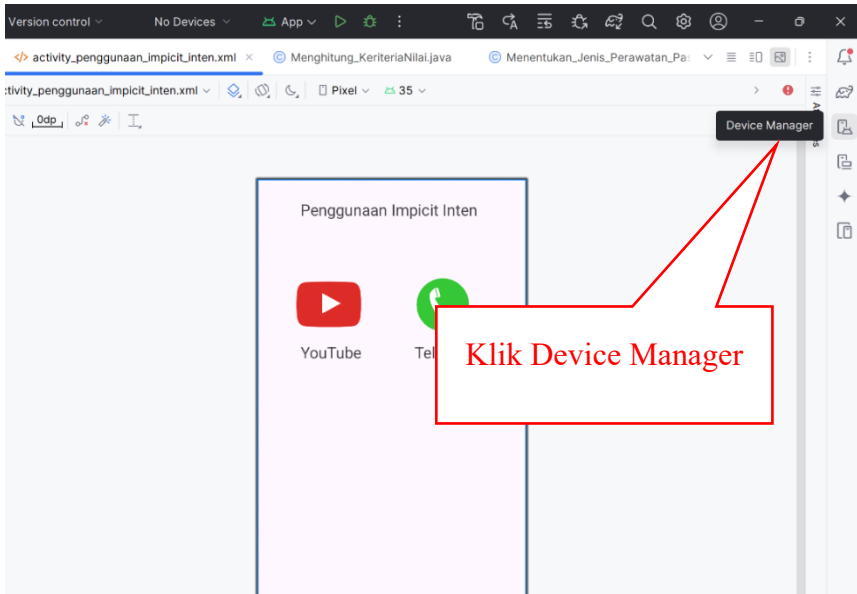
Langkah 1:

Untuk menjalankan proyek dalam emulator android studio yaitu dengan menekan tombol shit + f10.



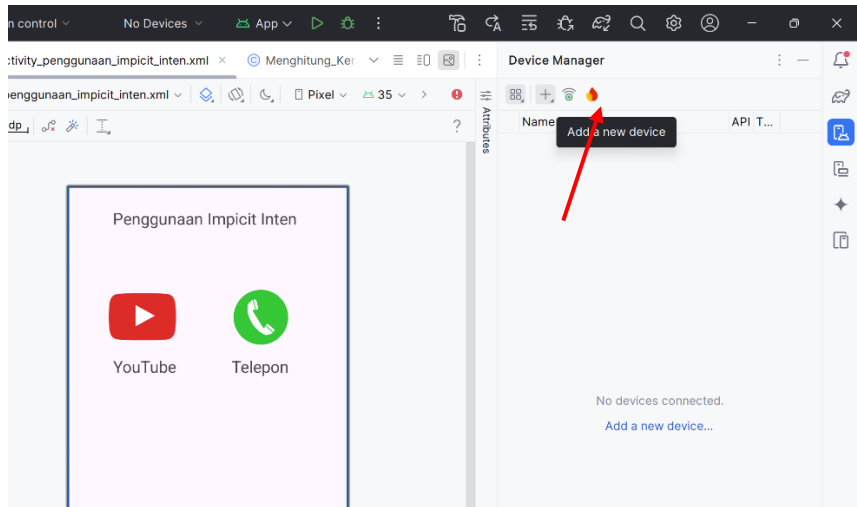
Gambar 2. 6 Langkah 1 Menjalankan Proyek Dengan Emulator

Secara default emulator dalam android studio harus di instal terlebih dahulu yaitu dengan cara:



Langkah 2:

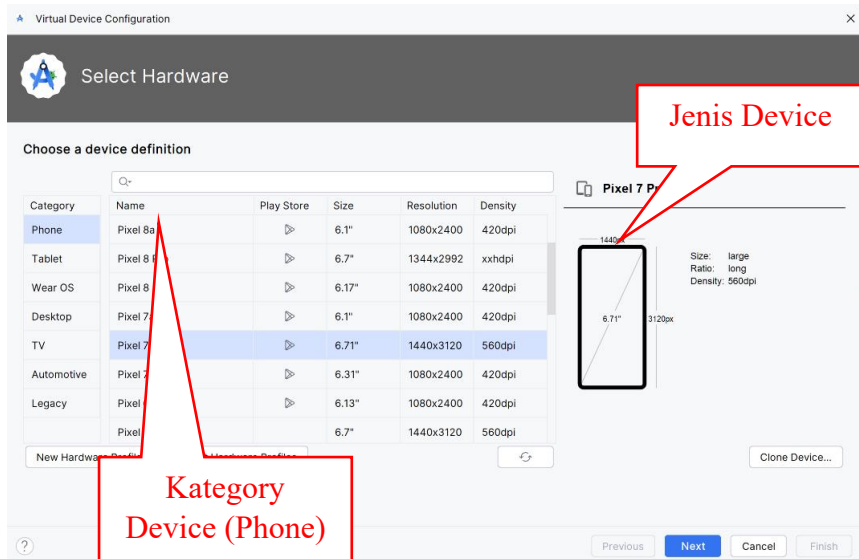
Klik tanda + untuk menambah virtual device.



Gambar 2. 7 Langkah 2 Menjalankan Proyek Dengan Emulator

Langkah 3:

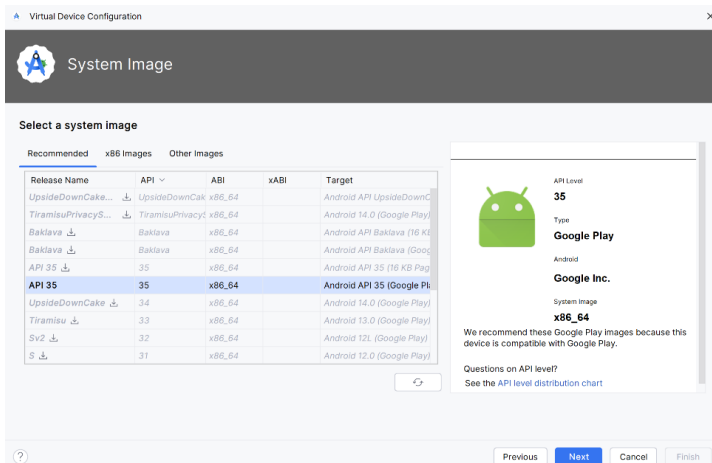
Pilih device yang diinginkan. Kemudian klik next.



Gambar 2. 8 Langkah 3 Menjalankan Proyek Dengan Emulator

Langkah 4:

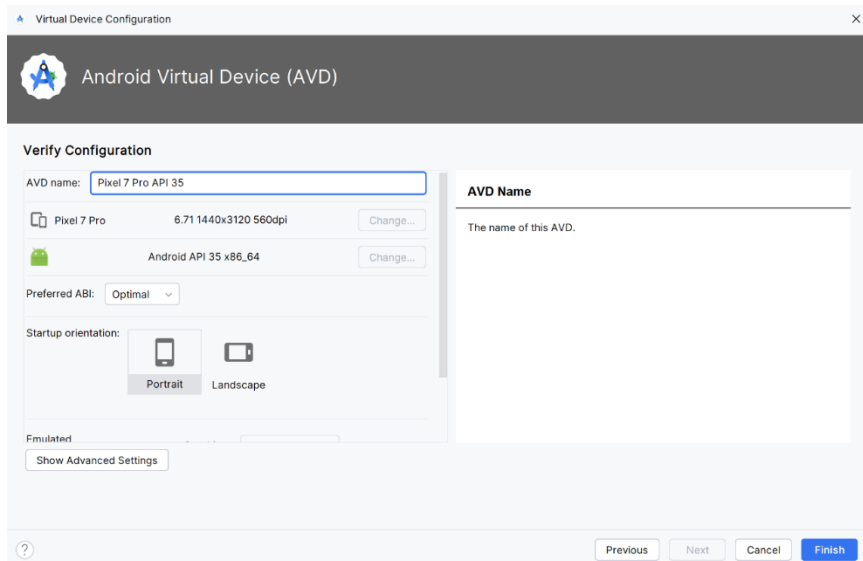
Pilih API 35. Lalu klik next.



Gambar 2. 9 Langkah 4 Menjalankan Proyek Dengan Emulator

Langkah 5:

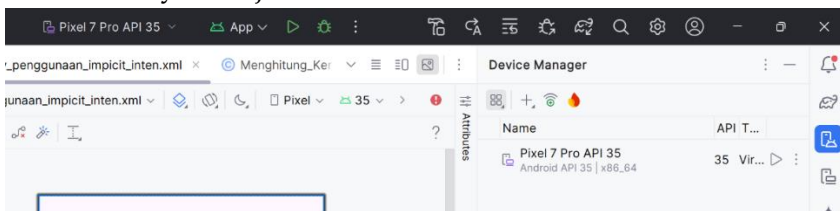
Pilih Orintasi dari virtual device. Kemudian klik next. Dan tunggu download emulator.



Gambar 2. 10 Langkah 5 Menjalankan Proyek Dengan Emulator

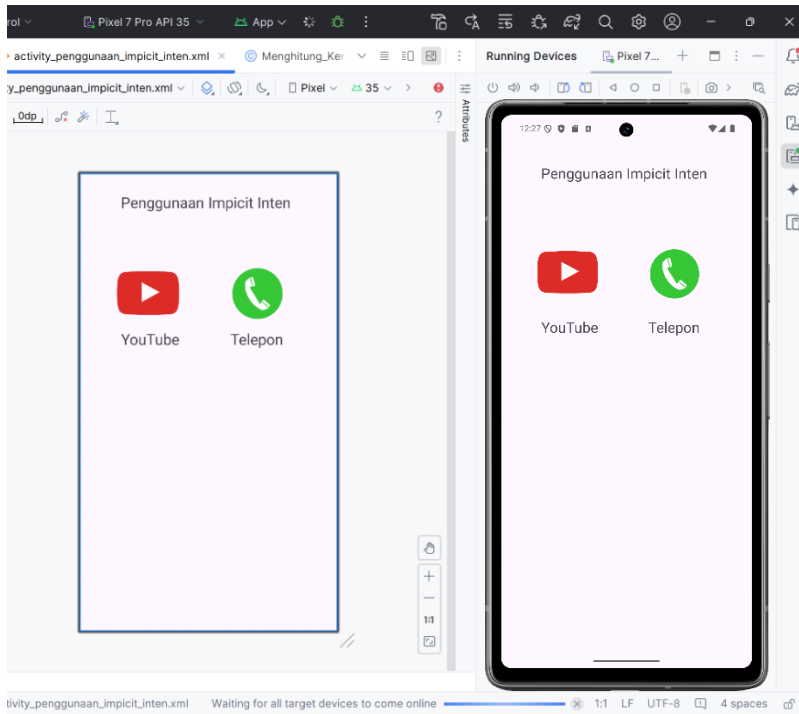
Langkah 6:

Maka emulator siap untuk digunakan dan akan tampil pada bilah atas layar kerja.



Gambar 2. 11 Langkah 6 Menjalankan Proyek Dengan Emulator

Berikut setelah di running:



Gambar 2. 12 Running Proyek Dengan Emulator

5. Menjalankan Proyek Langsung Handphone

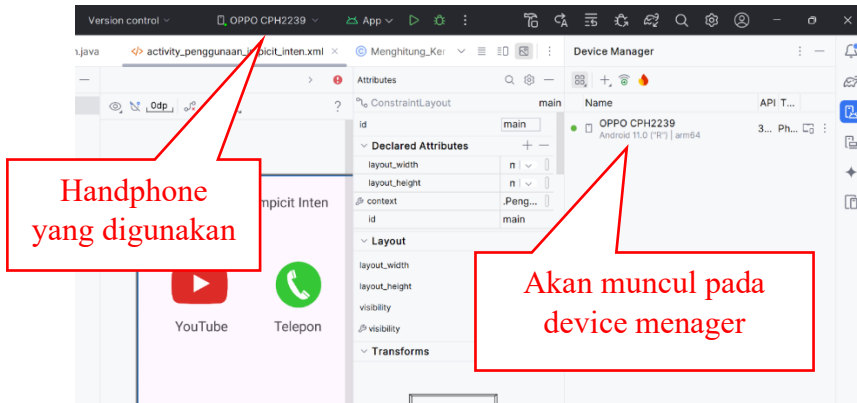
Menjalankan proyek menggunakan emulator dapat menyebabkan beberapa kendala seperti lama untuk melakukan running karena membutuhkan media penyimpanan yang atau RAM yang besar sehingga untuk mengatasi hal tersebut yaitu melakukan running langsung ke handphone sehingga lebih ringan untuk dijalankan. Adapun langkahnya sebagai berikut:

Langkah - Langkah:

Siapkan Hp android dan lakukan settingan sebagai berikut:

- 1) Aktifkan HP pilih Setting kemudian klik About Phone dan pilih versi kemudian ketuk Build Number sebanyak tujuh kali.
- 2) Aktifkan Pengaturan Pengembangan (Developer option) dan USB Debugging dengan cara: aktifkan HP pilih Setting kemudian pilih Developer Options. Kemudian USB Debugging Authorizations.
- 3) Hubungkan Hp dan komputer dengan kabel USB, cek apakah driver ADB sudah terinstal, caranya mengecek pada device manager dengan langkah-langkah sebagai berikut: Start - device manager - periksa driver ADB Jika masih error install terlebih dahulu driver ADB. Hal ini dilakukan jika android studio tidak dapat membaca perangkat kita.
- 4) Lalu running aplikasi di android studio, pilih nama hp kita yang tampil, maka akan ada pemberitahuan di hp, apakah di izinkan atau tidak, jika telah di izinkan, aplikasi akan tampil langsung pada layar hp kita.

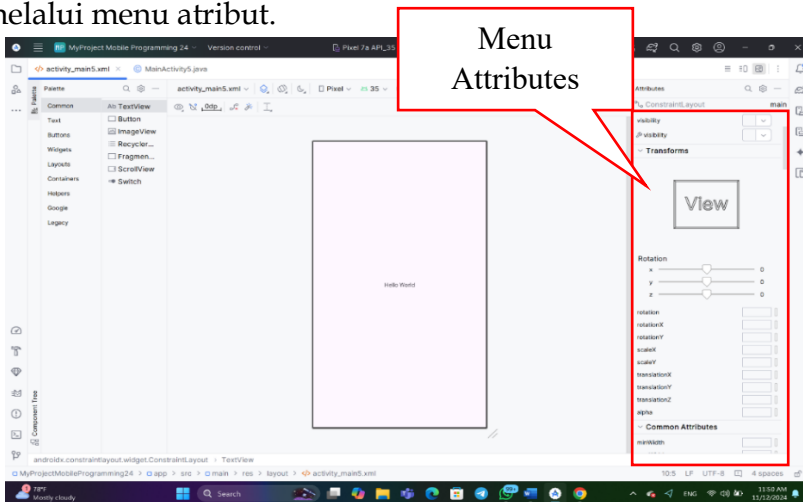
Berikut tampilan dengan menggunakan hp:



Gambar 2.13 Tampilan Penggunaan Handphone Untuk Menjalankan Projek

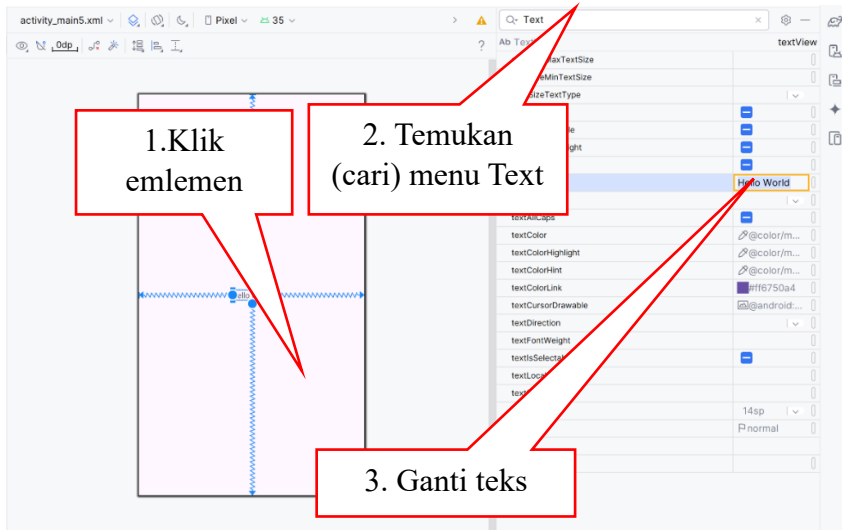
6. Attributes Dalam Android Studio Android Studio

Yaitu digunakan untuk melakukan editing terhadap elemen – elemen dalam android studio seperti mengganti teks, mengatur font, mengatur ukuran font, letak font dan lainnya melalui menu atribut.

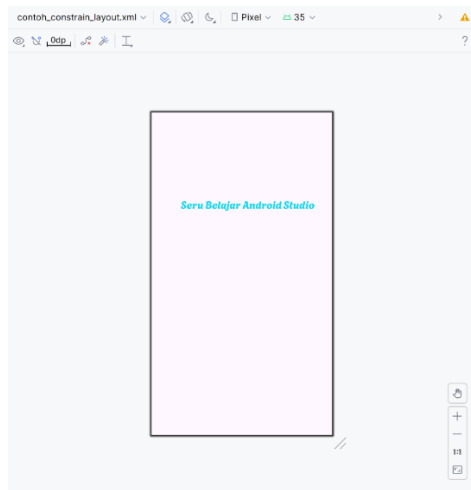


Gambar 2.14 Letak Menu Atribut Dalam Android Studio

- 1) Text, digunakan untuk mengganti kalimat pada elemen teks adapun contoh penerapannya: klik elemen yang ingin diganti kalimatnya kemudian cari menu text pada menu atribut.



Gambar 2.15 Mengganti Teks Pada Atribut

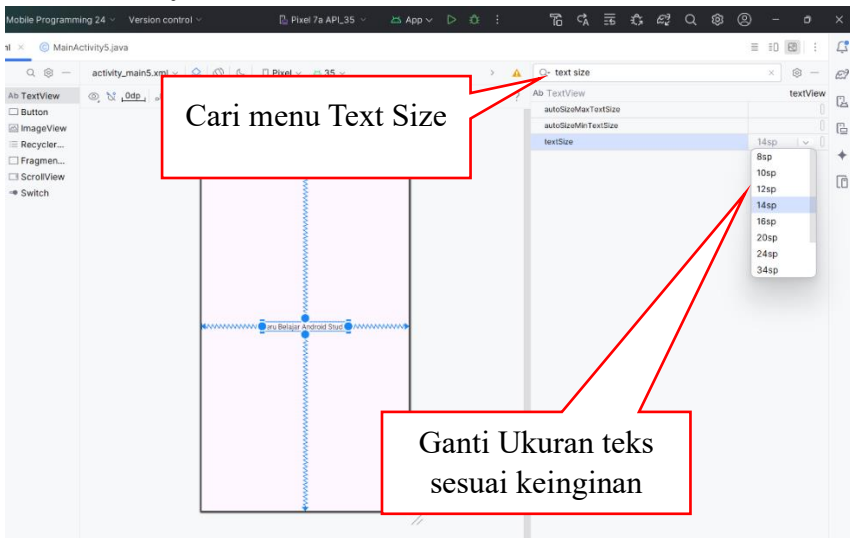


Gambar 2.16 Setelah Dilakukan Penggantian Teks Pada Atribut

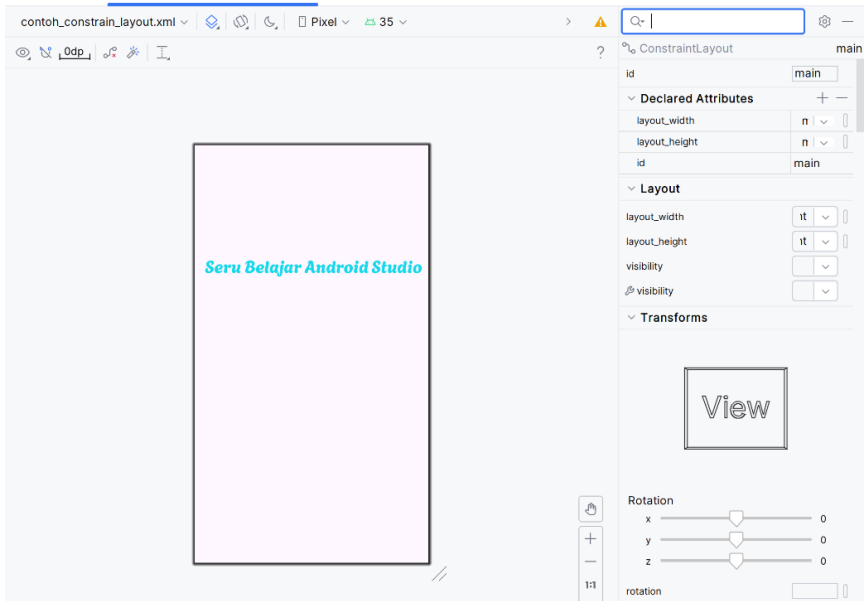
Berikut hasilnya pada koding .xml:

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Seru Belajar Android Studio" />
```

- 2) Text size, digunakan untuk mengubah ukuran teks. Caranya awalnya sama dengan yang sebelumnya hanya saja pada menu atribut menggunakan TextSize. Berikut contohnya:



Gambar 2.17 Mengganti Ukuran Teks Pada Atribut



Gambar 2.18 Setelah Dilakukan Penggantian Ukuran Teks

Berikut hasil koding .xml:

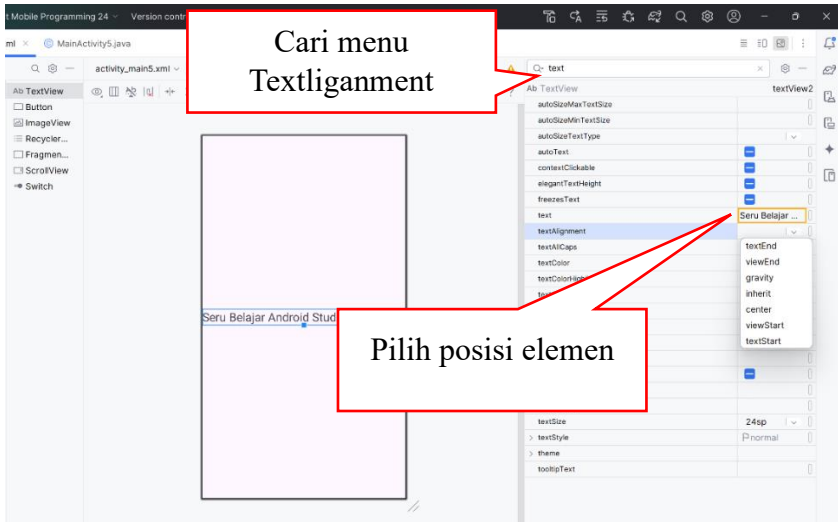
```

<TextView
    android:id="@+id/textView33"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:fontFamily="@font/agbalumo"
    android:text="Seru Belajar Android Studio"
    android:textColor="#14D9EB"
    android:textSize="24dp"
/>

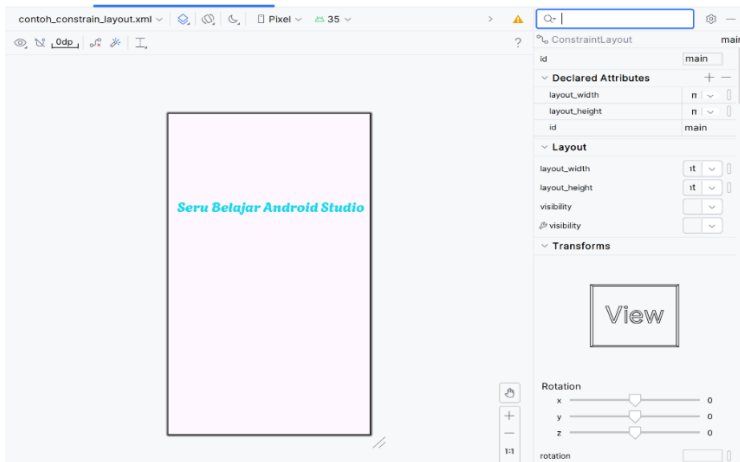
```

- 3) TextAligenment, digunakan untuk mengatur letak dari sebuah elemen apakah di kananl, kiri atau tengah. Dimana center untuk berposisi ditengah, Text End posisi elemen ke

kanan, dan start di kiri dan lainnya. Dan untuk warna cara yang dilakukan sama hanya saja pada menu atribut cari color text lalu pilih warna sesuai keinginan. Berikut contoh penggunaan TextAlignenment:



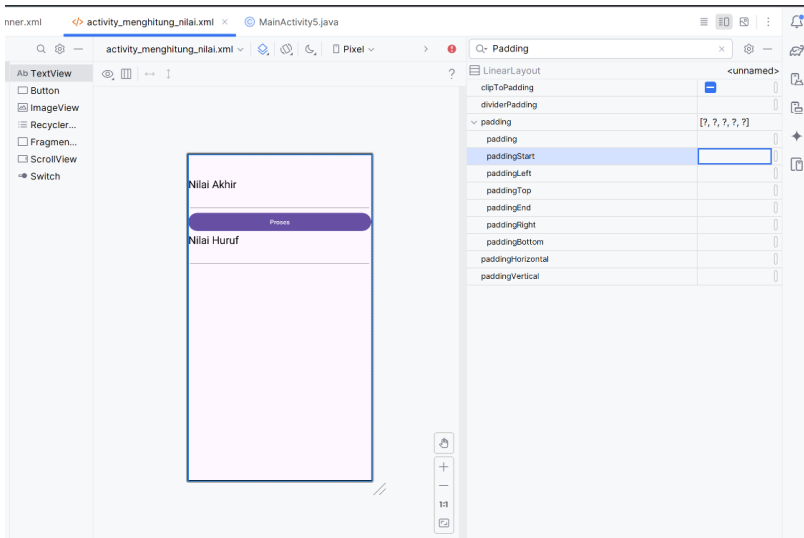
Gambar 2.19 Mengganti Letak Teks Pada Atribut



Gambar 2.20 Setelah Dilakukan Penggantian Letak Teks

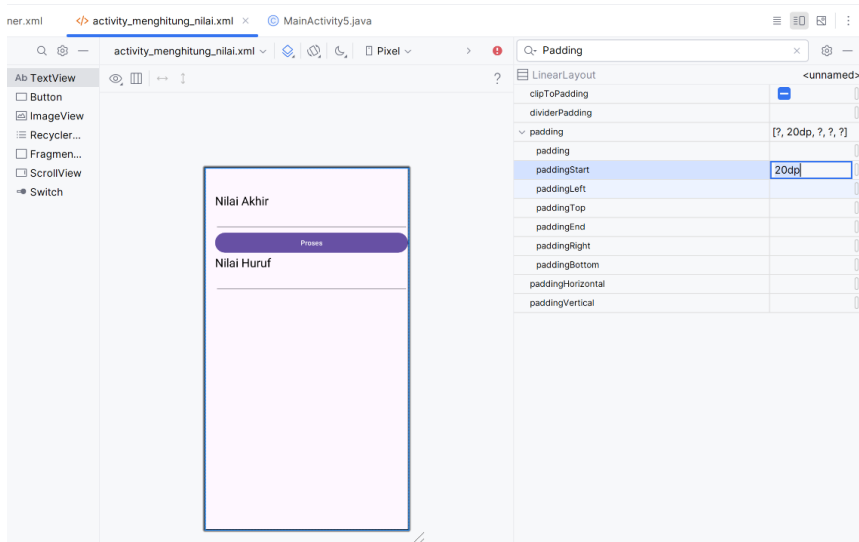
- 4) Padding Dan Margin, Padding adalah ruang kosong yang mengelilingi sebuah elemen, sementara margin adalah jarak antara elemen satu dengan elemen lainnya. Pengaturan padding dan margin umumnya diperlukan untuk menciptakan tampilan yang lebih presisi.

Berikut contoh penggunaan Padding: perhatikan gambar berikut dimana posisi elemen pada layout sangat dekat dengan sisi kiri layout, sehingga jika diberikan padding pada bagian kiri atau Start sebanyak 20dp akan mengalami perubahan.



Gambar 2.21 Layout Sebelum Menggunakan Padding

Pada gambar berikut akan digunakan padding dimana jika attribute yang digunakan Padding saja maknanya menerapkan semua padding pada semua sisi, sedangkan PaddingStart dan lainnya di sisi tertentu saja. Berikut perbedaan tampilan setelah menetapkan padding 20dp di bagian kanan atau PaddingStart.



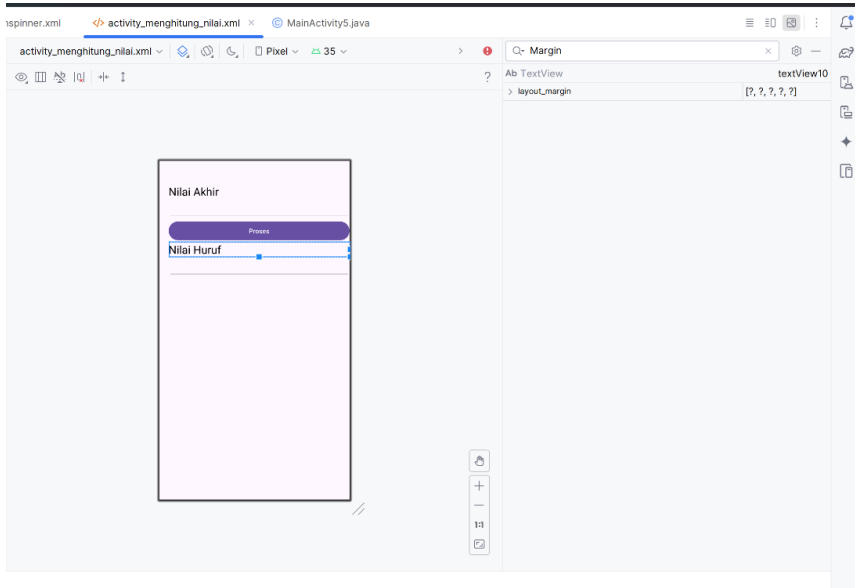
Gambar 2.22 Layout Setelah Menggunakan Padding

Berikut koding .xml:

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingStart="20dp">
</LinearLayout>
```

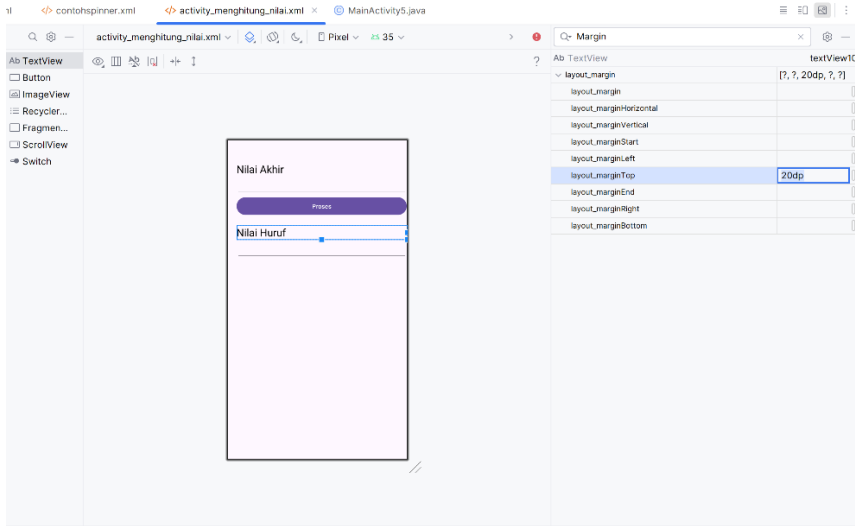
Berikut contoh penggunaan Margin: Perhatikan elemen pada gambar berikut, dimana nilai huruf memiliki jaeak yang sangat dekat dengan tombol proses, pada kasus ini akan menggunakan margin untuk memberi jarak antar elemen tersebut. Konsep margin dan padding memiliki konsep yang sama terhadap pemberian jarak jika ingin

memberi jarak elemen ke atas maka menggunakan `MarginTop` dan lain sebagainya.



Gambar 2.23 Elemen Sebelum Menggunakan Margin

Perhatikan Pada gambar 2.15 dimana jarak antara elemen Nilai Huruf dan tombol sangat dekat akan tetapi setelah menggunakan `MarginTop` pada elemen tersebut sehingga memberi jarak antara elemen tersebut.



Gambar 2.24 Elemen Setelah Menggunakan Margin

Berikut koding penggunaan Margin .xml:

```

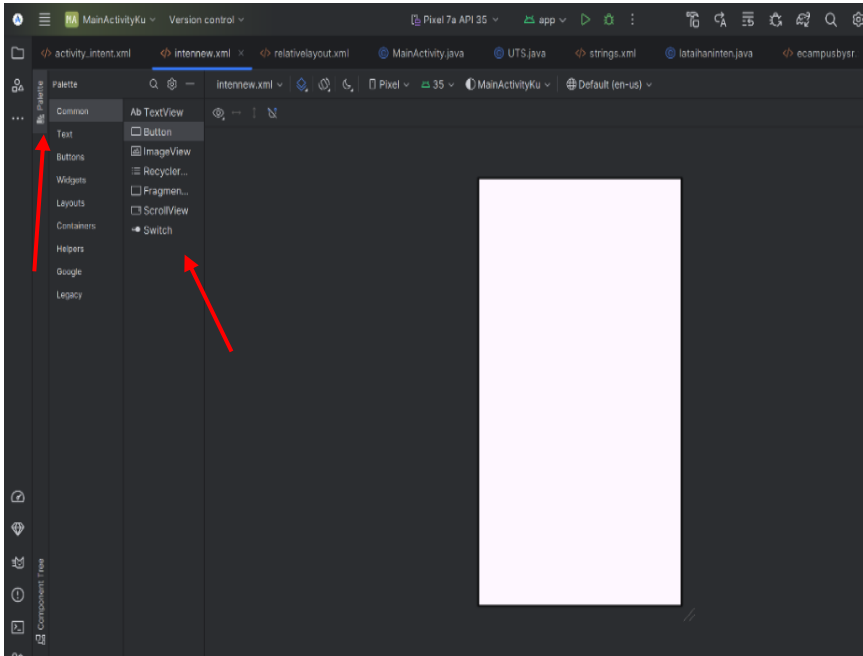
<TextView
    android:id="@+id/textView10"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="Nilai Huruf"
    android:textColor="@color/black"
    android:textSize="24sp" />

```

Perlu diketahui bahwa penggunaan Padding dan Margin tidak hanya digunakan di setiap elemen saja akan tetapi dapat digunakan disetiap elemen. Yaitu dengan menambahkan padding langsung ke Layout misalnya pada gambar 2.26 dan 2.27 dan juga pada koding .xml.

7. Menambahkan Elemen Dalam Android Studio

Menambahkan elemen di Android Studio bisa dilakukan dengan cara klik palatte kemudian memilih elemen yang diinginkan seperti text view, button, image view dan lainnya. Kemudian tarik ke halaman layout kita.



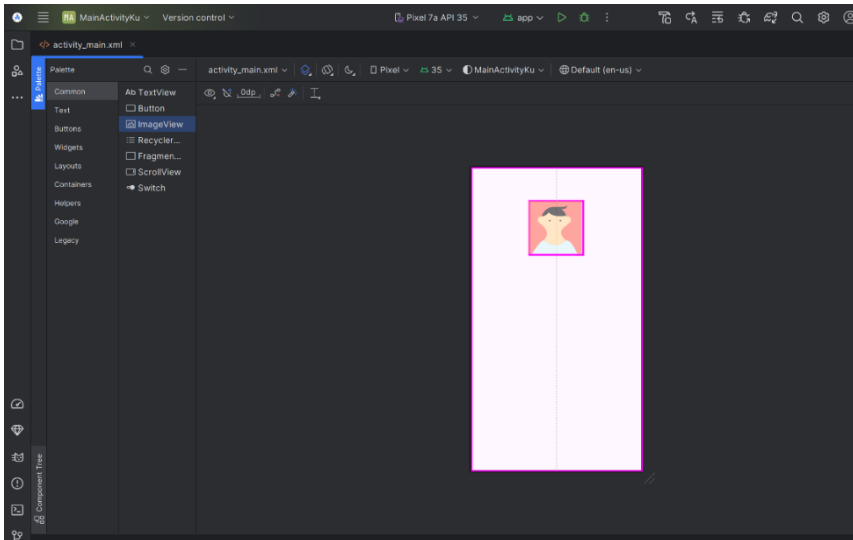
Gambar 2.25 Menambah Elemen Dalam Android Studio

8. Menambahkan Foto Dan Image Button Dalam Android Studio

Untuk menambahkan foto dalam android studio dapat melalui cara seperti berikut:

Langkah 1:

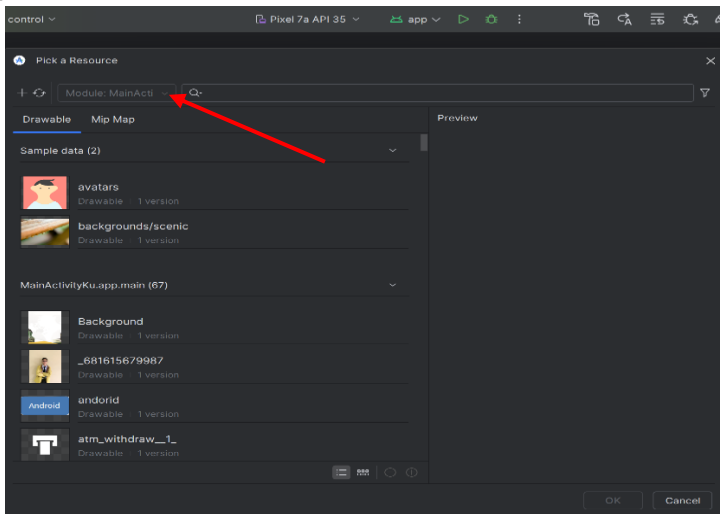
Klik Image view pada menu Palatte lalu tarik kedalam layout.



Gambar 2.26 Langkah 1 Menambahkan Foto Dalam Android Studio

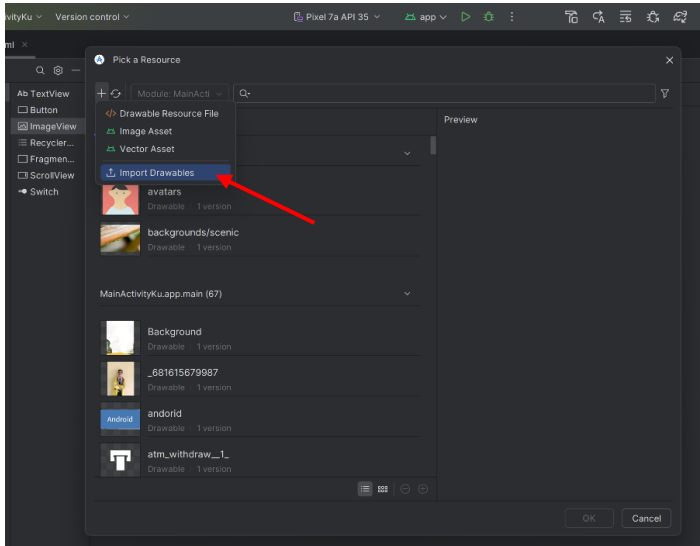
Langkah 2:

Kemudian pilih foto yang ingin ditambahkan jika belum diimport ke drawable maka klik simbol +.



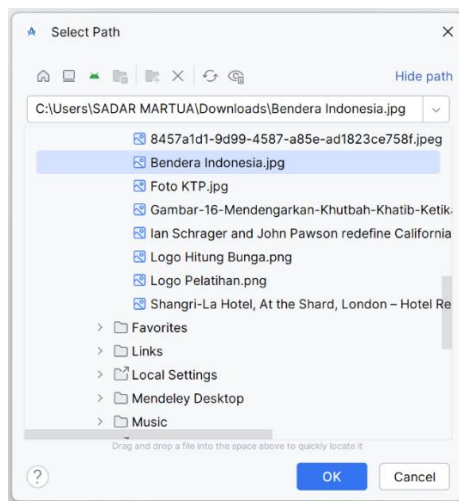
Gambar 2.27 Langkah 2 Menambahkan Foto Dalam Android Studio

Langkah 3: Kemudian klik Import Drawables.



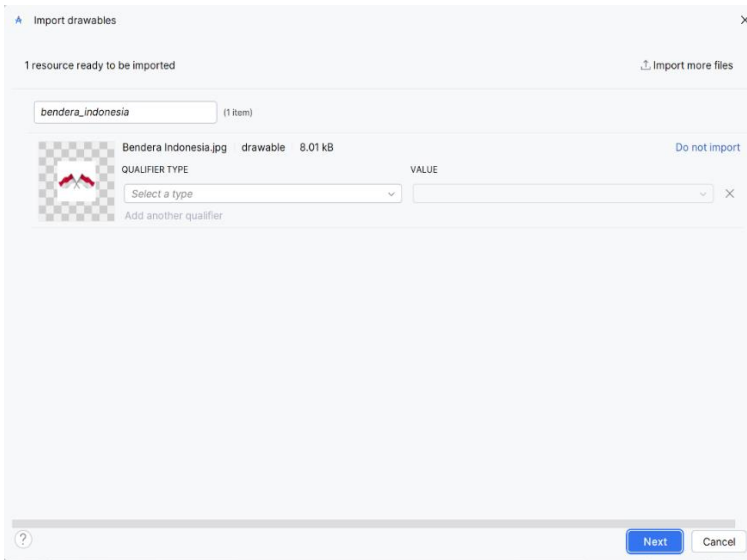
Gambar 2.28 Langkah 3 Menambahkan Foto Dalam Android Studio

Langkah 4: Lalu cari gambar yang ingin ditambahkan lalu klik ok.



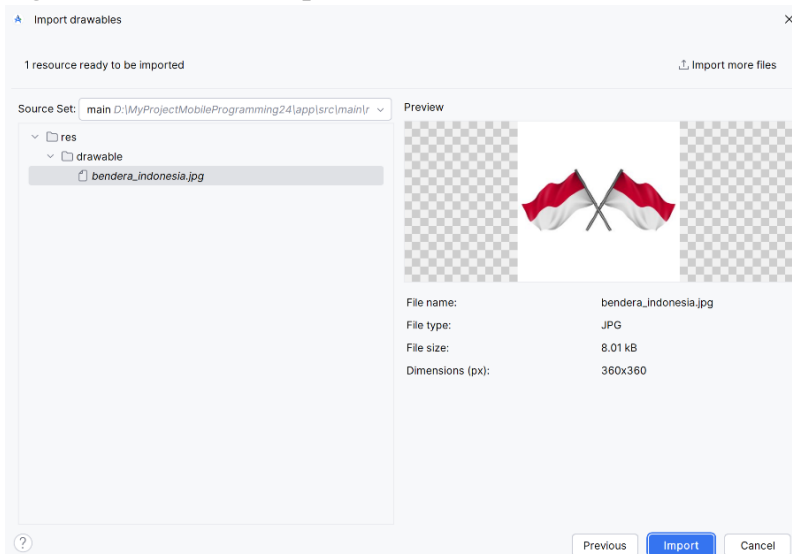
Gambar 2.29 Langkah 4 Menambahkan Foto Dalam Android Studio

Langkah 5: Lalu klik Next.



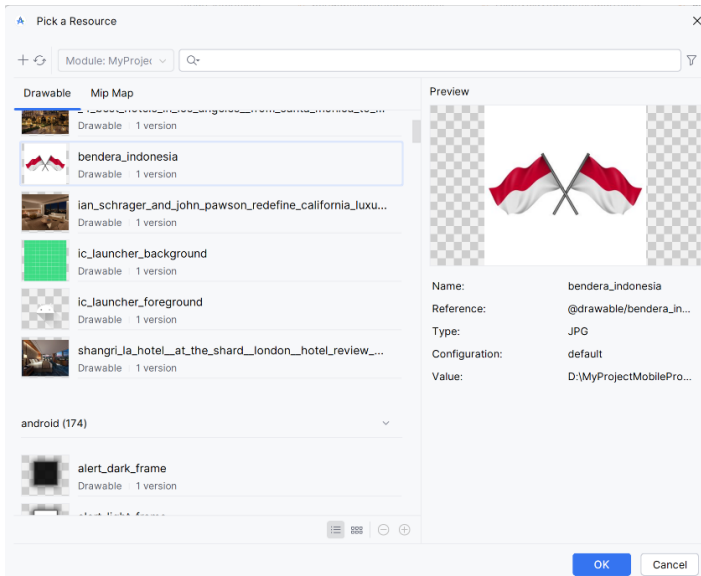
Gambar 2.30 Langkah 5 Menambahkan Foto Dalam Android Studio

Langkah 6: Lalu klik import



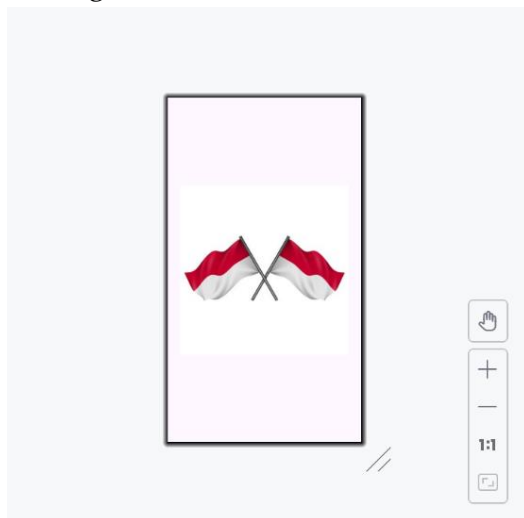
Gambar 2.31 Langkah 6 Menambahkan Foto Dalam Android Studio

Langkah 7: kemudian pilih gambar yang ingin ditambahkan. Kemudian klik ok.



Gambar 2.32 Langkah 7 Menambahkan Foto Dalam Android Studio

Langkah 8: Maka gambar berhasil di tambahkan.

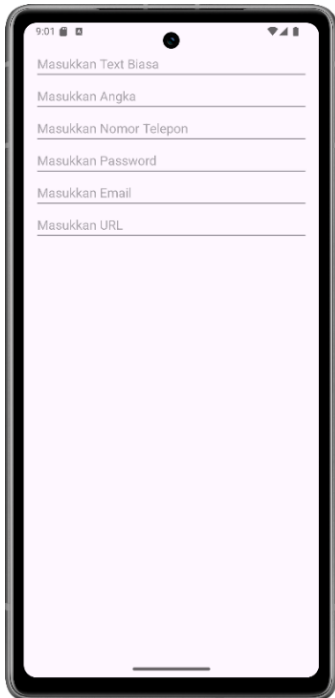


Gambar 2.33 Langkah 8 Menambahkan Foto Dalam Android Studio

Untuk membuat gambar menjadi sebuah button hanya saja pada langkah pertama pilih image button dan untuk langkah selanjutnya tetap sama.

9. Edit Text Dalam Android Studio

EditText sangat berguna sebagai interface atau tempat untuk menerima inputan berupa text atau kata-kata di aplikasi Android. Karena itu, jenis-jenis text yang bisa diterima oleh EditText pun bermacam-macam. Contohnya seperti password, nomor telepon, email address, dan sebagainya. Adapun contoh pengaplikasiannya:



Gambar 2.35 Hasil Running EditText Setelah Digunakan



Gambar 2.34 Hasil Running EditText Sebelum Digunakan

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
android:orientation="vertical"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="15dp"
android:layout_marginTop="20dp">
    <EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Masukkan Text Biasa"
android:inputType="text" /> //Untuk Kalimat Atau Teks

    <EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Masukkan Angka"
android:inputType="number" /> // Untuk Angka

    <EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Masukkan Nomor Telepon"
android:inputType="phone" /> // Untuk Nomor Telepon

```

```
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Masukkan Password"
android:inputType="textPassword" /> //Untuk Password
```

```
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Masukkan Email"
android:inputType="textEmailAddress" /> // Untuk
Alamat Email
```

```
<EditText
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:hint="Masukkan URL"
android:inputType="textUri" /> // Untuk Link URL
</LinearLayout>
```

Dalam EditText yang perlu menjadi perhatian adalah dalam inputType dimana inputType dapat disesuaikan dengan kebutuhan. Berikut jenis inputType:

- a) Teks biasa menggunakan inputType = "text";
- b) Angka menggunakan inputType = "number";
- c) Nomor telepon menggunakan inputType = "phone";
- d) Password menggunakan inputType = "password";
- e) Email menggunakan inputType "textEmailAddress";
- f) Url web menggunakan inputType = "textUri";

Dengan menggunakan InputType, selain berfungsi untuk menspesifikasikan data yang ingin kita inputkan, InputType juga akan mengubah bermacam hal, seperti layout keyboard, dan tampilan text. Untuk InputType number, maka layout keyboard Android kita akan berubah hanya menampilkan angka, begitu halnya dengan InputType email, layout keyboard kita akan menampilkan simbol '@' yang mudah diakses. Sedangkan untuk InputType password, maka apa yang kita ketikkan tidak akan tampil pada EditText, hanya ada titik-titik dan lain sebagainya.

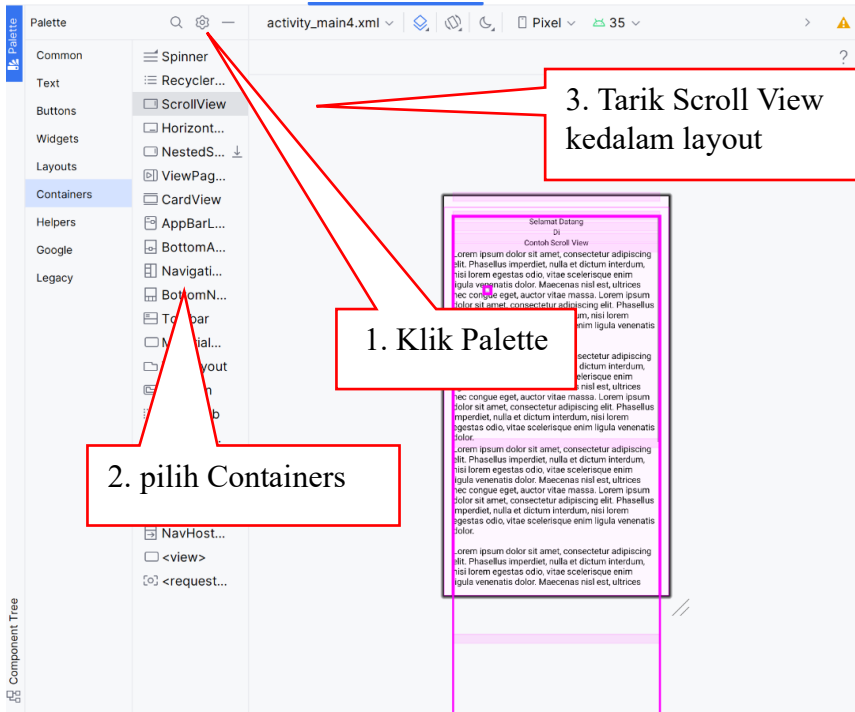
10. Scroll View

ScrollView memungkinkan pengembang untuk membuat tata letak antarmuka yang melebihi batas tampilan layar perangkat secara vertikal. Dengan menggunakan komponen ini, sebagian konten dapat diakses secara langsung, sementara sisanya akan terlihat saat pengguna menggulir layar ke bawah.

Dengan demikian, ScrollView memberikan ilusi ruang tampilan yang lebih luas dibandingkan dimensi layar sebenarnya, memungkinkan akses penuh ke konten dalam satu arah gulir.

Adapun contoh penggunaan dari Scroll View sebagai berikut:

Langkah 1:



Gambar 2.36 Langkah 1 Penggunaan Scroll View

Langkah 2:

Setelah itu kembali kemenu koding layout .xml yang telah diberikan scroll view.

```

2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
17     android:text="Selamat Datang" />
18
19 <TextView
20     android:id="@+id/textView8"
21     android:layout_width="match_parent"
22     android:layout_height="wrap_content"
23     android:text="Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum
24     android:textColor="#000000"
25     android:textSize="16sp" />
26
27 <TextView
28     android:id="@+id/textView2"
29     android:layout_width="match_parent"
30     android:layout_height="wrap_content"
31     android:layout_marginTop="28sp"
32     android:gravity="center"
33     android:text="Di Contoh Linear Layout"
34     android:textColor="@color/black"
35     android:textSize="30dp" />
36
37 <ScrollView
38     android:layout_width="match_parent"
39     android:layout_height="match_parent">
40
41     <LinearLayout
42         android:layout_width="match_parent"
43         android:layout_height="wrap_content"
44         android:orientation="vertical" />
45
46 </ScrollView>

```

Gambar 2.37 Penggunaan Scroll View Pada xml

Langkah 3:

Gantilah ConstraintLayout atau layout lainnya dengan ScrollView atau letakkan ScrollView sebagai elemen utama jika ingin semua konten dapat di-scroll. Ikuti langkah berikut:

```

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" />
</ScrollView>

```

Gambar 2.38 Tahap 2 Penggunaan Scroll View Pada xml

Berikut contoh penerapannya:

```
<ScrollView  
  
xmlns:android="http://schemas.android.com/apk/res/and  
roid"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    android:layout_marginTop="20dp"  
    android:padding="16dp">  
    <LinearLayout  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:orientation="vertical">  
  
        <TextView  
            android:id="@+id/textView4"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:gravity="center"  
            android:textColor="@color/black"  
            android:text="SelamatDatang"/>  
  
        <TextView  
            android:id="@+id/textView41"  
            android:layout_width="match_parent"  
            android:layout_height="wrap_content"  
            android:gravity="center"  
            android:textColor="@color/black"  
            android:text="Di"/>
```

```
<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Contoh Scroll View"
    android:textColor="@color/black" />
```

```
<TextView
    android:id="@+id/textView8"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
congue eget, auctor vitae massa. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. \n\nLorem ipsum dolor sit
amet, consectetur adipiscing elit. Phasellus imperdiet, nulla
et dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
congue eget, auctor vitae massa. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim          ligula          venenatis          dolor."
    android:textColor="#000000"
    android:textSize="16sp" />
```

```

<TextView
    android:id="@+id/textView7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
congue eget, auctor vitae massa. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. \n\nLorem ipsum dolor sit
amet, consectetur adipiscing elit. Phasellus imperdiet, nulla
et dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
congue eget, auctor vitae massa. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor."
    android:textColor="#000000"
    android:textSize="16sp" />
<TextView
    android:id="@+id/textView6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

    android:text="Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. Maecenas nisl est, ultrices nec

```

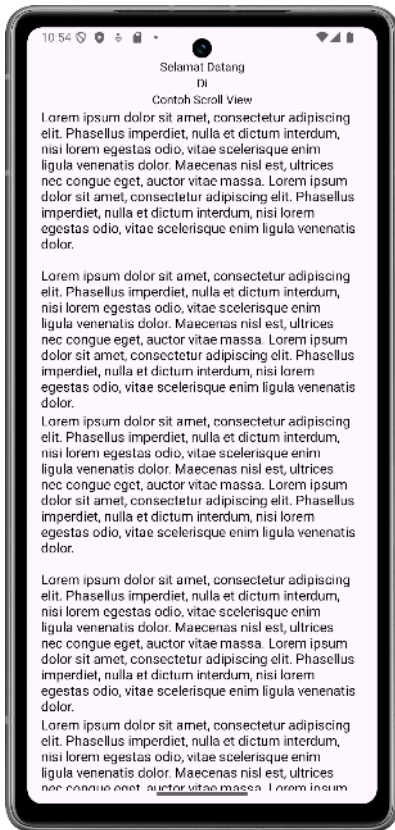
```

congue eget, auctor vitae massa. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. \n\nLorem ipsum dolor sit
amet, consectetur adipiscing elit. Phasellus imperdiet, nulla
et dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim ligula venenatis dolor. Maecenas nisl est, ultrices nec
congue eget, auctor vitae massa. Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus imperdiet, nulla et
dictum interdum, nisi lorem egestas odio, vitae scelerisque
enim          ligula          venenatis          dolor."
    android:textColor="#000000"
    android:textSize="16sp" />
<TextView
    android:id="@+id/textView9"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="Terimakasih Telah Mengakses Scroll
View"
    android:textColor="@color/black" />
</LinearLayout>
</ScrollView>

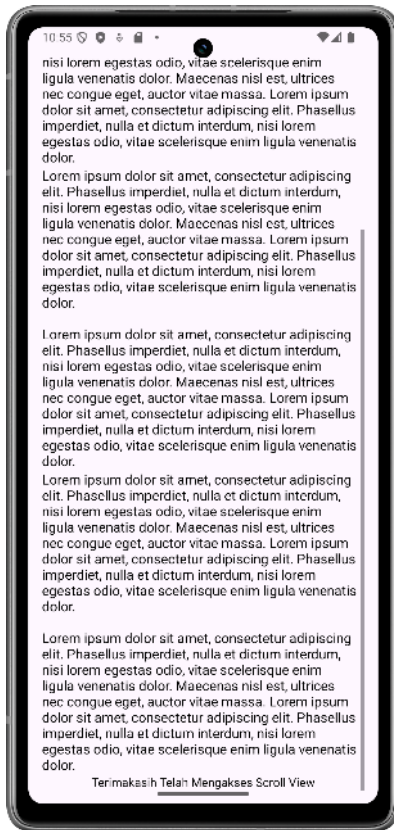
```

Dimana pada koding tersebut telah dilakukan perubahan fungsi utama yang mana menjadikan scroll view sebagai elemen utamanya.

Berikut adalah hasil dari penggunaan scroll view dari koding dan contoh di atas:



Gambar 2. 39 Hasil Running Penggunaan Scroll View



Gambar 2.40 Hasil Running Penggunaan scroll View Setelah Dilakukan Penggeseran

BAB III

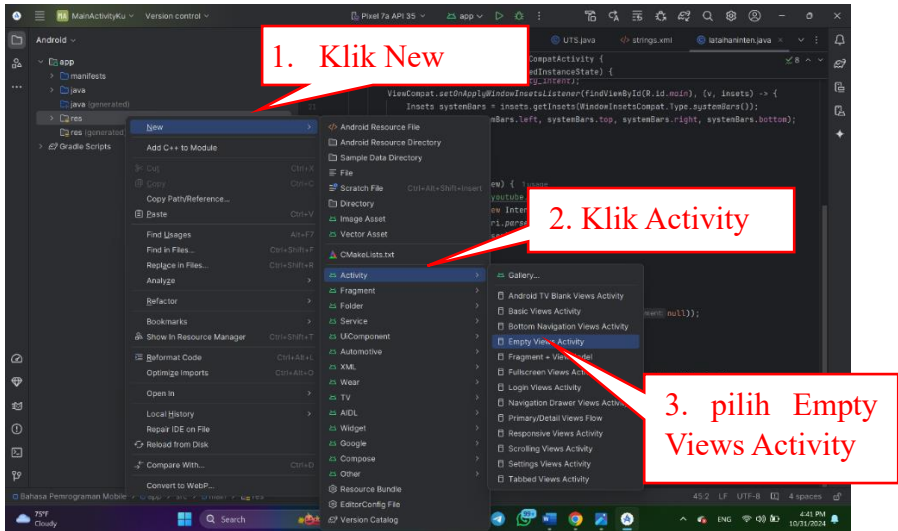
MEMBANGUN ACTIVITY

1. Pengertian Activity

Activity berguna untuk mendesain tampilan atau layout aplikasi yang digunakan sebagai komunikasi atau berintegrasi antara user dengan aplikasi seperti, memilih menu, mengambil foto, menampilkan peta, mengirim e-mail dan banyak lagi yang dapat dibangun pada activity. Satu aplikasi bisa memiliki lebih dari satu activity.

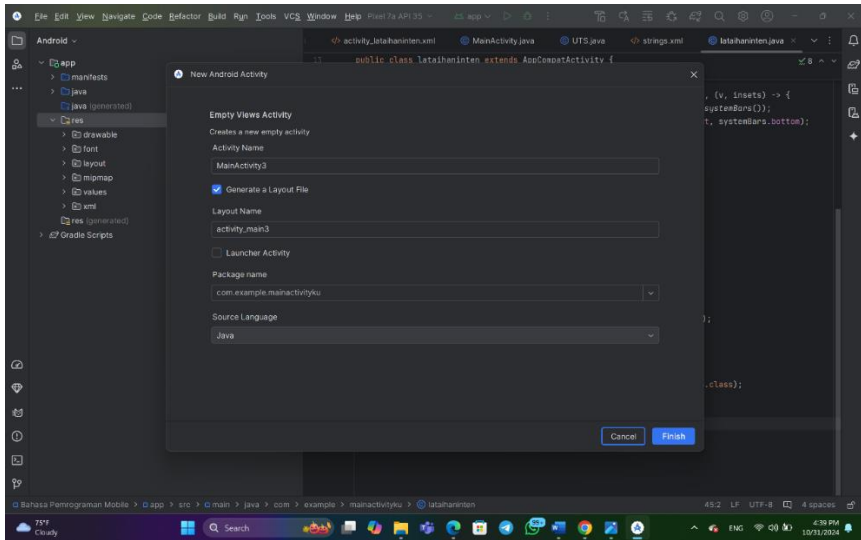
2. Menambah Activity

Pada menu APP pilih menu res lalu klik kanan kemudian ikuti langkah pada gambar berikut:



Gambar 3.1 Cara Menambah Activity

Setelah itu akan muncul tampilan berikut, kemudian berikan nama dan identitas lainnya. Dalam membuat penamaan dari activity dalam Android Studio pastikan nama activity yang kita buat tidak dikenali oleh Android Studio karna akan menyebabkan eror. Misalnya memberikan nama Intent atau IF dan lainnya maka akan menyebabkan eror oleh sebab itu berikan penamaan yang lebih spesifik misalnya LatihanIntent. Kemudian dalam penamaan pada activity usahakan menggunakan huruf kapital diawal nama project.



Gambar 3.2 Cara Memberikan Nama Activity

Setelah mengklik finish maka activity baru kita telah ditambahkan.

BAB IV

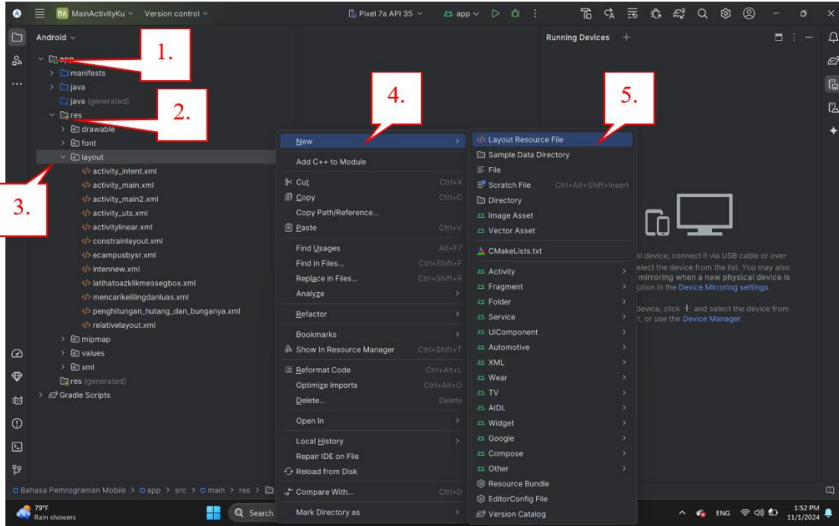
LAYOUT ANDRIOD STUDIO

1. Layout Android Studio

Layout di Android Studio adalah tampilan yang digunakan untuk mengatur posisi elemen seperti teks, gambar dan lainnya. Dalam mendesain layout activity pada android terdapat berbagai jenis layout yang ada yaitu: Linear Layout, Relative Layout dan Constarint Layout.

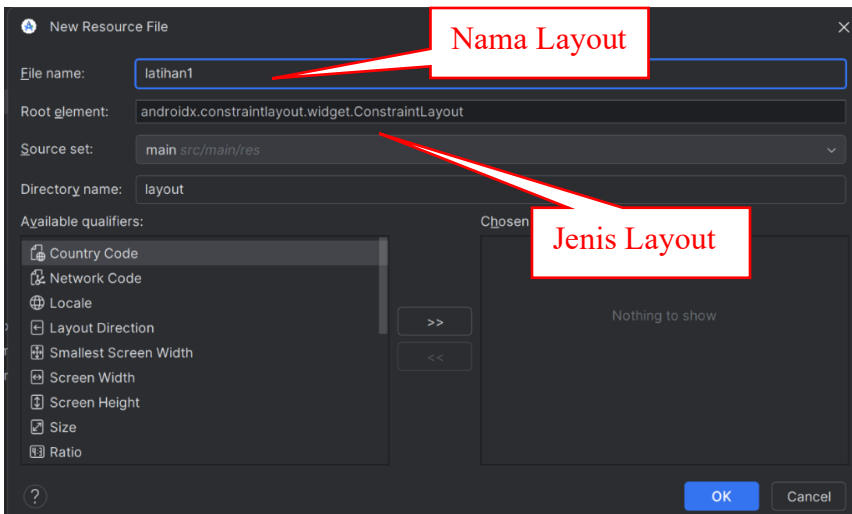
2. Menambah Layout

Untuk menambah layout dalam android studio dapat dilakukan dengan cara: Klik **App**, kemudian klik **res**, Pilih **Layout**, kemudian **new** kemudian **Layout Resource File**.



Gambar 4.1 Menambah Layout

Kemudian akan muncul tampilan berikut: kemudian klik ok jika sudah selesai mengisi identitas dari layout yang ditambahkan.



Gambar 4.2 Memberi Nama Dan Jenis Layout

3. Jenis - Jenis Layout

a. Linear Layout Dan Contoh Penggunaannya

Linear Layout adalah design tampilan pada aplikasi kita dengan tata letak secara vertical dan horizontal dimana tata letak aplikasi kita hanya bisa memasukan media secara mendatar dan menurun.

Berikut contoh penerapan dari linear layout:



Gambar 4.3 Contoh Penggunaan Linear Layout

Coding .xml

```
<?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"
android:orientation="vertical"
android:background="@color/white"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:layout_marginTop="30sp"
android:textColor="@color/black"
android:textSize="30dp"
android:text="Selamat Datang" />
    <TextView
android:id="@+id/textView2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20sp"
android:gravity="center"
android:text="Di Contoh Linear Layout"
android:textColor="@color/black"
android:textSize="30dp" />
    <ImageView
android:id="@+id/imageView"
```

```
android:layout_width="190dp"
android:layout_height="291dp"
android:layout_gravity="center"
android:layout_marginTop="5dp"
app:srcCompat="@drawable/bendera_indonesia" />
</LinearLayout>
```

b. Relative Layout Dan Contoh Penggunaannya

Merupakan Design tampilan pada aplikasi dengan tata letak secara bebas tanpa aturan sesuai keinginan, tidak seperti Linear Layout yang hanya terpaku pada salah satu tampilan vertical dan horizontal. RelativeLayout juga merupakan turunan dari ViewGroup, setiap komponen atau elemen UI (*User Interface*) di dalam layout tersebut akan tersusun secara Relatif terhadap elemen UI atau container lain.

Dalam pengaplikasiannya harus dilakukan penetapan posisi dari elemen yang kita muat. Untuk mengatur posisi elemen UI, misalnya Button, cukup mendragnya pada posisi yang diinginkan, tapi jika ingin menggunakan kode XML, ada beberapa atribut yang bisa kita gunakan pada elemen atau komponen UI didalam RelativeLayout untuk mengatur posisi dan juga jarak antar elemen, yaitu:

- a) `android:layout_centerInParent`: jika kita set menjadi "true", elemen UI akan berada di tengah relatif terhadap lebar dan tinggi Layout.
- b) `android:layout_centerHorizontal`: jika kita set menjadi "true", elemen UI akan berada di tengah relatif terhadap lebar Layout.

- c) `android:layout_centerVertical`: jika kita set menjadi "true", elemen UI akan berada di tengah relatif terhadap tinggi Layout.
- d) `android:layout_alignParentBottom`: jika kita set menjadi "true", elemen UI diletakan rata sisi bawah layout.
- e) `android:layout_alignParentTop`: jika kita set menjadi "true", elemen UI diletakan rata sisi atas layout.
- f) `android:layout_alignParentLeft`: jika kita set menjadi "true", elemen UI diletakan rata sisi kiri layout.
- g) `android:layout_alignParentRight`: jika kita set menjadi "true", elemen UI diletakan rata sisi kanan layout.

Ada beberapa atribut yang bisa dikombinasikan misalnya `android:layout_centerHorizontal` dengan `android:layout_alignParentBottom` jika kedua atribut tersebut di set menjadi "true", komponen UI akan berada di tengah bagian bawah layout.

Selain itu ada juga beberapa atribut yang tidak bisa di kombinasikan atau bisa disebut mutual exclusive, yaitu hanya bisa dipilih salah satu saja, misalnya `android:layout_centerVertical` dengan `android:layout_alignParentBottom`.

Adapun contoh pengaplikasiannya yaitu:



Gambar 4.4 Contoh Penerapan Relative Layout

Koding .xml:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="50dp"
    android:layout_marginBottom="50dp">
```

<Button

```
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerInParent="true"  
    android:text="Tengah" />
```

<Button

```
    android:id="@+id/button2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_alignParentLeft="true"  
    android:text="Kiri Atas" />
```

<Button

```
    android:id="@+id/button3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:text="T.Atas" />
```

<Button

```
    android:id="@+id/button4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_centerHorizontal="true"  
    android:layout_alignParentRight="true"  
    android:text="Kanan Atas" />
```

```
<Button
    android:id="@+id/button5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:text="Kiri.T" />

<Button
    android:id="@+id/button6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:layout_alignParentRight="true"
    android:text="Kanan.T" />

<Button
    android:id="@+id/button7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentLeft="true"
    android:text="Kiri Bawah" />

<Button
    android:id="@+id/button8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_alignParentBottom="true"
    android:text="T.Bawah" />
```

```
<Button
  android:id="@+id/button9"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignParentRight="true"
  android:layout_alignParentBottom="true"
  android:text="Kanan Bawah" />

</RelativeLayout>
```

Selain itu untuk mengatur suatu atribut agar dapat dijadikan menjadi acuan elemen lain juga dapat dilakukan. pengatur posisi relatif antara elemen UI dengan elemen UI lainnya, ada beberapa atribut yang bisa digunakan untuk mengatur posisi tersebut, yaitu:

- a) `android:layout_above`: jika di set menjadi "true", elemen UI akan berhimpit atau berada di atas elemen UI yang di picu/dituju.
- b) `android:layout_below`: jika di set menjadi "true", elemen UI akan berhimpit atau berada di bawah elemen UI yang di picu/dituju.
- c) `android:layout_toLeftOf`: jika di set menjadi "true", elemen UI akan berhimpit atau berada di sisi kiri elemen UI yang di picu/dituju.
- d) `android:layout_toRightOf`: jika di set menjadi "true", elemen UI akan berhimpit atau berada di sisi kanan elemen UI yang di picu/dituju.

Untuk contoh penerapannya, kita akan membuat 5 buah Button (masing-masing diberi ID), Button1 sebagai titik pusat (berada di tengah), lalu keempatnya lagi berada di sisi kanan, kiri, atas dan bawah Button yang di picu/ dituju.

Berikut contoh penerapannya:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="50dp">

    <Button
android:id="@+id/pusat"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Titik Pusat"
android:layout_centerInParent="true" />

    <Button
android:id="@+id/left"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Kiri"
android:layout_centerInParent="true"
android:layout_toLeftOf="@+id/pusat" />
```

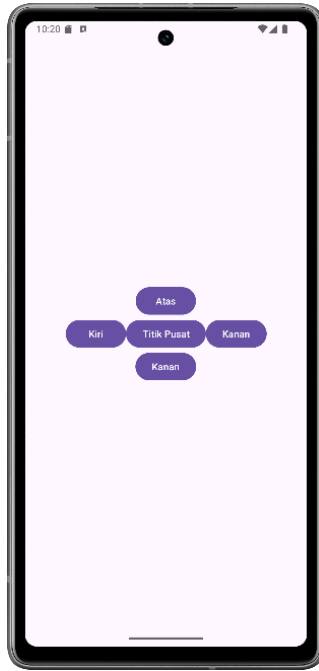
```
<Button
android:id="@+id/right"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_toRightOf="@id/pusat"
android:layout_centerInParent="true"
android:text="Kanan" />

    <Button
android:id="@+id/up"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Atas"
android:layout_above="@+id/pusat"
android:layout_centerHorizontal="true" />

    <Button
android:id="@+id/bottom"
android:layout_width="wrap_content"
/*android:layout_height="wrap_content"
android:layout_below="@id/pusat"
android:layout_centerInParent="true"
android:text="Kanan" />

</RelativeLayout>
```

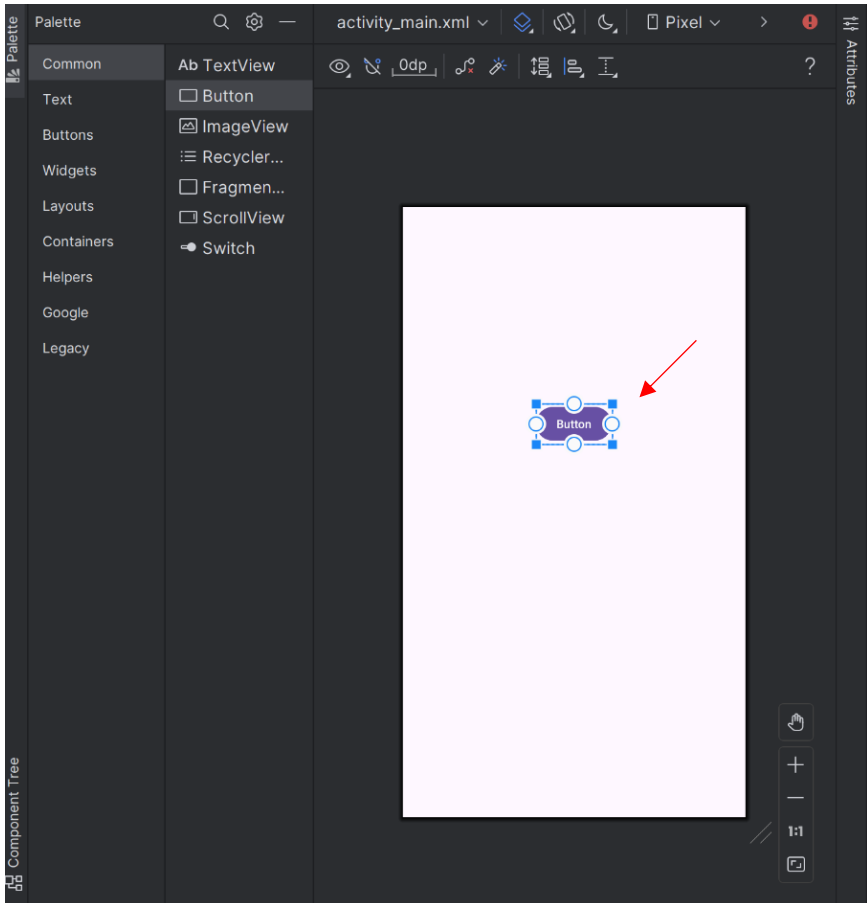
Berikut output yang dihasilkan dari koding .xml:



Gambar 4. 5 Hasil Running Relative Layout

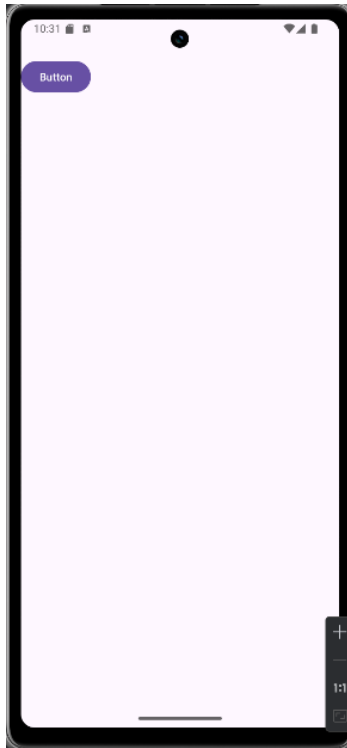
c. Constrain Layout Dan Contoh Penggunaanya

Constarint Layout memungkinkan kita membuat tata letak yang besar dan kompleks dengan tampilan datar. Ini hampir mirip dengan Relative Layout karena semua tampilan ditata berdasarkan hubungan antara satu objek dengan yang lain, tetapi lebih fleksibel daripada RelativeLayout dan lebih mudah digunakan dengan Editor Layout Android Studio. Dalam layout ini harus memperhatikan penarikan posisi dari elemen yang ingin ditetapkan. Seperti contoh berikut:



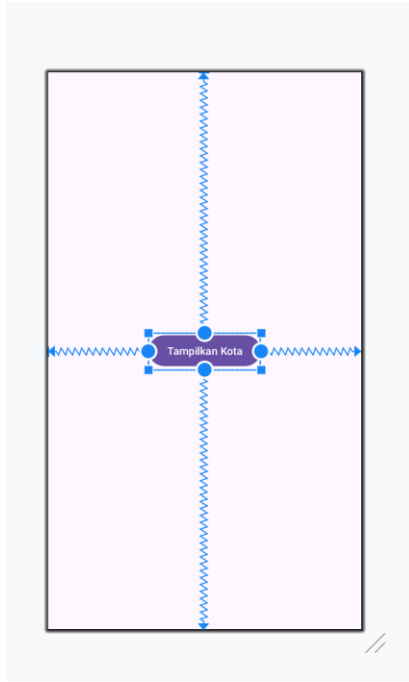
Gambar 4.6 Contoh Posisi Penarikan Constraint Layout

Pada tanda panah merah tersebut menunjukkan posisi dari constraint elemen dari sebuah elemen yang digunakan. Penarikan dari constraint elemen sangat diperlukan karena jika tidak dilakukan maka output dari elemen kita akan acak. Pada gambar 4.6 sebelum penarikan, elemen button berada ditengah setelah di running hasilnya sebagai berikut:



Gambar 4.7 Contoh Running Constraint Layout

Dimana elemen button yang berada ditengah setelah di running berada di kiri atas. Oleh sebab itu lakukan penarikan pada constrain elemen sebagai berikut:



Gambar 4.8 Contoh Penarikan Posisi Constrain Layout

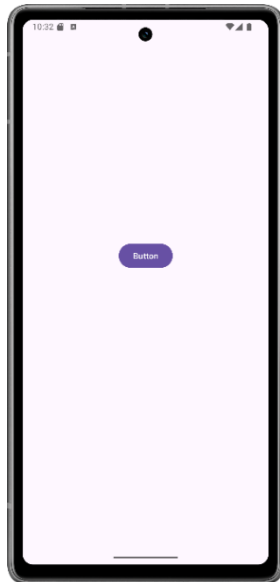
Lakukan penarikan pada ke empat sisi elemen constrain lalu posisi elemen akan sesuai dengan layout dan outputnya. Berikut contoh penerapannya:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    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=".MainActivity4">
```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Berikut hasil output setelah dilakukan penarikan ke empat sisi constrain elemen pada button:



Gambar 4.9 Hasil Running Setelah Dilakukan Penarikan Pada Constrain Layout

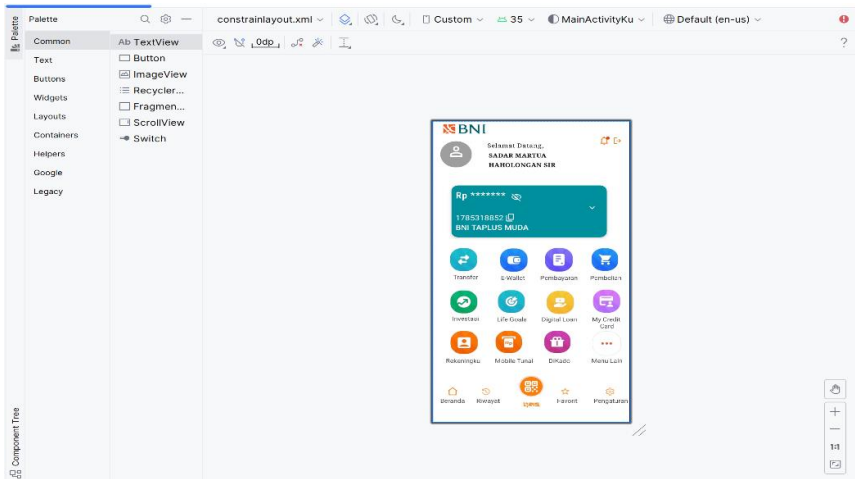
4. Contoh Penggunaan

a. Kasus 1

Membuat tampilan BNI Mobile menggunakan Constrain Layout, berikut penyelesaiannya

Langkah 1: siapkan elemen – elemen yang dibutuhkan.

Langkah 2: Aturlah posisi elemen yang disesuaikan dengan versi aslinya.



Gambar 4. 10 Layout Kasus 1

Berikut koding layout .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:layout_width="match_parent"
```

```

android:layout_height="match_parent"
android:background="@android:color/white"
android:fitsSystemWindows="true">
<ImageView
    android:id="@+id/logobankBNI"
    android:layout_width="121dp"
    android:layout_height="43dp"
    android:layout_marginStart="16dp"
    android:layout_marginEnd="16dp"
    android:scaleType="centerInside"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.002"

app:srcCompat="@drawable/bni_bank_negara_indonesia"/
>
<TextView
    android:id="@+id/textView7"
    android:layout_width="137dp"
    android:layout_height="23dp"
    android:fontFamily="serif"
    android:letterSpacing="0.04"
    android:text="Selamat Datang,"
    android:textColor="#000000"
    android:textSize="16sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"

```

```
app:layout_constraintHorizontal_bias="0.394"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.07" />
```

```
<TextView
```

```
    android:id="@+id/textView8"  
    android:layout_width="168dp"  
    android:layout_height="68dp"  
    android:fontFamily="serif"  
    android:letterSpacing="0.02"  
    android:lineSpacingExtra="8sp"  
    android:text="Sadar Martua Haholongan Sir"  
    android:textAllCaps="true"  
    android:textColor="#000000"  
    android:textSize="16sp"  
    android:textStyle="bold"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.444"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.11" />
```

```
<androidx.cardview.widget.CardView
```

```
    android:id="@+id/cardView"  
    android:layout_width="370dp"  
    android:layout_height="145dp"  
    android:layout_margin="30dp"  
    app:cardBackgroundColor="#008F9C"
```

```
app:cardCornerRadius="16dp"  
app:cardElevation="4dp"  
app:contentPadding="0dp"  
app:contentPaddingBottom="15dp"  
app:contentPaddingLeft="10dp"  
app:contentPaddingTop="10dp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.385"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.236"  
tools:ignore="DuplicateIds">
```

```
<TextView
```

```
    android:id="@+id/textView15"  
    android:layout_width="137dp"  
    android:layout_height="wrap_content"  
    android:fontFamily="sans-serif-condensed-medium"  
    android:letterSpacing="0.04"  
    android:text="Rp *****"  
    android:textColor="#FFFF"  
    android:textSize="25sp"  
    app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.394"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.07" />
```

```
<TextView
    android:id="@+id/textView16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom"
    android:fontFamily="@font/roboto_medium"
    android:text="BNI TAPLUS MUDA"
    android:textColor="#FFFEFE"
    android:textSize="20sp" />
```

```
<TextView
    android:id="@+id/textView17"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center | left"
    android:layout_marginTop="20dp"
    android:fontFamily="sans-serif-thin"

    android:text="1785318852"
    android:textColor="#FFFEFE"
    android:textSize="20sp"
    android:textStyle="bold" />
```

```
<ImageButton
    android:id="@+id/imageButton21"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_marginStart="105dp"
    android:layout_marginTop="55dp"
```

```
android:backgroundTint="@android:color/transparent"
    android:scaleType="fitCenter"

android:src="@drawable/content_copy_24dp_e8eaed_fill0_
wght400_grad0_opsz24" />

<ImageButton
    android:id="@+id/imageButton22"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_marginStart="120dp"

android:backgroundTint="@android:color/transparent"
    android:scaleType="fitStart"

android:src="@drawable/visibility_off_24dp_e8eaed_fill0_
wght400_grad0_opsz24" />

<ImageButton
    android:id="@+id/imageButton23"
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_marginStart="300dp"
    android:layout_marginTop="30dp"

android:backgroundTint="@android:color/transparent"
    android:scaleType="fitStart"

android:src="@drawable/keyboard_arrow_down_24dp_e8e
aed_fill0_wght400_grad0_opsz24" />
```

```
</androidx.cardview.widget.CardView>

<TextView
    android:id="@+id/textView11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="5dp"
    android:layout_marginTop="296dp"
    android:text="Transfer"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.11"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/button10" />

<TextView
    android:id="@+id/textView19"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="412dp"
    android:layout_marginRight="40dp"
    android:text="Investasi"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.109"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/button10" />
```

```
<TextView
```

```
    android:id="@+id/textView24"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="5dp"  
    android:layout_marginTop="660dp"  
    android:layout_marginRight="40dp"  
    android:text="Rekeningku"  
    android:textSize="16sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.079"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/textView25"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="15dp"  
    android:layout_marginTop="660dp"  
    android:layout_marginRight="25dp"  
    android:text="Mobile Tunai"  
    android:textSize="16sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.396"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView27"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="660dp"
    android:layout_marginRight="40dp"
    android:text="DiKado"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.727"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView26"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="660dp"
    android:text="Menu Lain"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.958"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
android:layout_marginLeft="11dp"  
android:layout_marginTop="428dp"  
android:layout_marginRight="10dp"  
android:text="E-Wallet"  
android:textSize="16sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.388"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textView14"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="428dp"  
android:gravity="center"  
android:text="Pembayaran"  
android:textSize="16sp"  
android:layout_marginRight="10dp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.694"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textView20"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginLeft="8dp"  
android:layout_marginTop="216dp"
```

```
    android:layout_marginEnd="16dp"
    android:text="Life Goals"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.402"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="@+id/imageButton4"
/>
```

```
<TextView
    android:id="@+id/textView21"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="544dp"
    android:layout_marginEnd="13dp"
    android:text="Digital Loan"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.699"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView22"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="216dp"
```

```
    android:layout_marginEnd="15dp"
    android:text="My Credit "
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.993"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="@+id/imageButton5"
/>
```

```
<TextView
    android:id="@+id/textView23"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="135dp"
    android:layout_marginEnd="25dp"
    android:text="Card"
    android:textSize="16sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.971"
    app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="@+id/textView18" />
```

```
<TextView
    android:id="@+id/textView18"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="8dp"
```

```
android:layout_marginTop="428dp"  
android:text="Pembelian"  
android:textSize="16sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.952"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textView130"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginLeft="15dp"  
android:layout_marginTop="772dp"  
android:text="Riwayat"  
android:textSize="16sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.225"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textView28"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="772dp"  
android:text="Beranda"  
android:textSize="16sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.045"
```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/textView29"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="772dp"  
    android:text="Pengaturan"  
    android:textSize="16sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.99"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/textView30"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="772dp"  
    android:layout_marginLeft="1dp"  
    android:text="Favorit"  
    android:textSize="16sp"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.705"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<ImageButton  
    android:id="@+id/imageButton17"
```

```
android:layout_width="60dp"
android:layout_height="90dp"
android:layout_marginTop="712dp"
android:backgroundTint="@android:color/transparent"
android:scaleType="fitCenter"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.237"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/riwayat" />
```

<Button

```
android:id="@+id/button10"
android:layout_width="75dp"
android:layout_height="73dp"
android:layout_marginTop="12dp"
android:backgroundTint="#5f5f5f"
android:foregroundGravity="center"
android:insetTop="0dp"
android:insetBottom="0dp"
```

```
app:icon="@drawable/person_24dp_f19e39_fill0_wght400_
grad0_opsz24"
```

```
app:iconGravity="top"
app:iconSize="45dp"
app:iconTintMode="src_in"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.047"
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/logobankBN  
I" />
```

```
<ImageButton  
    android:id="@+id/imageButton3"  
    android:layout_width="110dp"  
    android:layout_height="120dp"  
    android:layout_marginStart="20dp"  
    android:layout_marginTop="328dp"  
    android:backgroundTint="@android:color/transparent"  
    android:scaleType="fitCenter"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@drawable/transfer" />
```

```
<ImageButton  
    android:id="@+id/imageButton7"  
    android:layout_width="110dp"  
    android:layout_height="120dp"  
    android:layout_marginTop="444dp"  
    android:backgroundTint="@android:color/transparent"  
    android:scaleType="fitCenter"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.053"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@drawable/investasi" />
```

```
<ImageButton  
    android:id="@+id/imageButton11"
```

```
android:layout_width="105dp"
android:layout_height="120dp"
android:layout_marginTop="560dp"
android:backgroundTint="@android:color/transparent"
android:scaleType="fitCenter"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.065"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/rekeningku" />
```

<ImageButton

```
android:id="@+id/imageButton12"
android:layout_width="110dp"
android:layout_height="120dp"
android:layout_marginTop="556dp"
android:backgroundTint="@android:color/transparent"
android:scaleType="fitCenter"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.358"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/mobile_tunai" />
```

<ImageButton

```
android:id="@+id/imageButton2"
android:layout_width="110dp"
android:layout_height="120dp"
android:layout_marginTop="696dp"
android:backgroundTint="@android:color/transparent"
```

```
android:scaleType="fitCenter"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/qris" />
```

<ImageButton

```
android:id="@+id/imageButton15"  
android:layout_width="60dp"  
android:layout_height="90dp"  
android:layout_marginTop="712dp"  
android:backgroundTint="@android:color/transparent"  
android:scaleType="fitCenter"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.237"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/riwayat" />
```

<ImageButton

```
android:id="@+id/imageButton18"  
android:layout_width="60dp"  
android:layout_height="90dp"  
android:layout_marginTop="712dp"  
android:backgroundTint="@android:color/transparent"  
android:scaleType="fitCenter"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.697"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"
```

```
app:srcCompat="@drawable/bintang" />
```

```
<ImageButton
```

```
    android:id="@+id/imageButton19"  
    android:layout_width="60dp"  
    android:layout_height="90dp"  
    android:layout_marginTop="712dp"  
    android:backgroundTint="@android:color/transparent"  
    android:scaleType="fitCenter"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.959"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@drawable/pengaturan" />
```

```
<ImageButton
```

```
    android:id="@+id/imageButton16"  
    android:layout_width="60dp"  
    android:layout_height="90dp"  
    android:layout_marginTop="712dp"  
    android:backgroundTint="@android:color/transparent"  
    android:scaleType="fitCenter"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.046"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@drawable/home" />
```

```
<ImageButton
```

```
    android:id="@+id/imageButton13"
```

```
android:layout_width="110dp"
android:layout_height="120dp"
android:layout_marginTop="560dp"
android:backgroundTint="@android:color/transparent"
android:scaleType="fitCenter"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.674"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/kado" />
```

<ImageButton

```
android:id="@+id/imageButton14"
android:layout_width="105dp"
android:layout_height="120dp"
android:layout_marginTop="560dp"
android:backgroundTint="@android:color/transparent"
android:scaleType="fitCenter"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.993"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/menu_lainnya" />
```

<ImageButton

```
android:id="@+id/imageButton8"
android:layout_width="120dp"
android:layout_height="120dp"
android:layout_marginTop="444dp"
android:backgroundTint="@android:color/transparent"
```

```
android:scaleType="fitCenter"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.372"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/life_goals" />
```

<ImageButton

```
android:id="@+id/imageButton9"  
android:layout_width="110dp"  
android:layout_height="120dp"  
android:layout_marginTop="448dp"  
android:backgroundTint="@android:color/transparent"  
android:scaleType="fitCenter"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.697"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/digital_loan" />
```

<ImageButton

```
android:id="@+id/imageButton4"  
android:layout_width="110dp"  
android:layout_height="120dp"  
android:layout_marginTop="328dp"  
android:backgroundTint="@android:color/transparent"  
android:scaleType="fitCenter"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.392"  
app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toTopOf="parent"  
app:srcCompat="@drawable/ewallet" />
```

```
<ImageButton
```

```
    android:id="@+id/imageButton5"  
    android:layout_width="110dp"  
    android:layout_height="120dp"  
    android:layout_marginLeft="20dp"  
    android:layout_marginTop="328dp"  
    android:backgroundTint="@android:color/transparent"  
    android:scaleType="fitCenter"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.986"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@drawable/pembelian" />
```

```
<ImageButton
```

```
    android:id="@+id/imageButton10"  
    android:layout_width="105dp"  
    android:layout_height="110dp"  
    android:layout_marginLeft="20dp"  
    android:layout_marginTop="448dp"  
    android:backgroundTint="@android:color/transparent"  
    android:scaleType="fitCenter"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.993"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:srcCompat="@drawable/my_credit" />
```

```
<ImageButton
    android:id="@+id/imageButton6"
    android:layout_width="110dp"
    android:layout_height="120dp"
    android:layout_marginLeft="20dp"
    android:layout_marginTop="328dp"
    android:backgroundTint="@android:color/transparent"
    android:scaleType="fitCenter"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.671"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/bayar" />
```

```
<ImageView
    android:id="@+id/imageView2"
    android:layout_width="117dp"
    android:layout_height="109dp"
    android:layout_marginBottom="720dp"
    android:paddingStart="20dp"
    android:paddingBottom="30dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.992"
    app:layout_constraintStart_toStartOf="parent"
    app:srcCompat="@drawable/notifikasi" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

BAB V

PENGGUNAAN BUTTON, TOAST, STRING DAN SPINNER

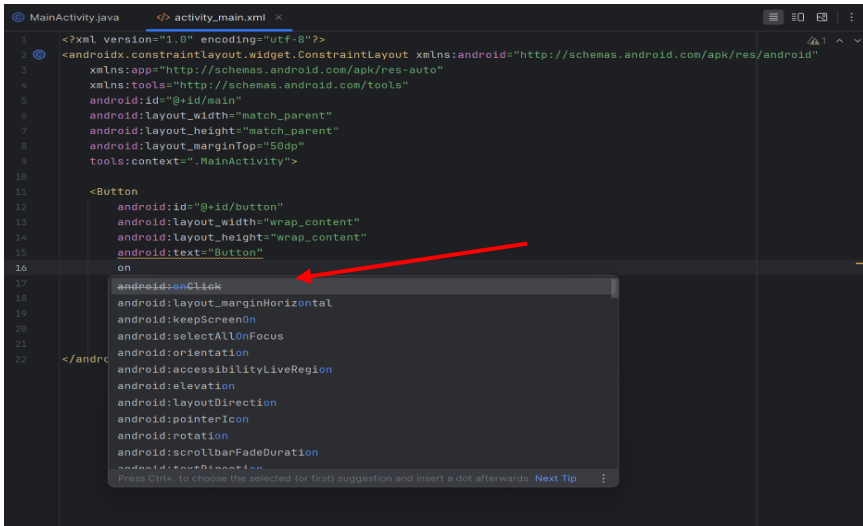
1. Penggunaan String Pada Button Menggunakan OnClick

Button adalah salah satu elemen antarmuka pengguna dalam aplikasi Android yang berupa tombol. Fitur ini sangat penting dan seringkali dianggap esensial dalam pengembangan aplikasi Android, karena banyak aktivitas dalam sistem operasi ini yang memerlukan interaksi melalui tombol untuk melaksanakan perintah-perintah tertentu. Contohnya termasuk tombol untuk login, logout, pencarian, dan berbagai menu lainnya.

Dalam mobile programming penggunaan toast berfungsi sebagai penampil pesan yang ingin ditampilkan ketika tombol tersebut diakses. Untuk itu dalam menggunakan toast tombol terlebih dahulu harus mengaktifkan fungsi OnClick. Berikut langkah dalam mengaktifkan fungsi onClik pada button dan penggunaan fungsi toast.

Langkah 1:

Pilih sebuah tombol yang ingin diaktifkan. Lalu ketik onClick.



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:id="@+id/main"
6   android:layout_width="match_parent"
7   android:layout_height="match_parent"
8   android:layout_marginTop="50dp"
9   tools:context=".MainActivity">
10
11   <Button
12     android:id="@+id/button"
13     android:layout_width="wrap_content"
14     android:layout_height="wrap_content"
15     android:text="Button"
16     on
17     android:onClick
18     android:layout_marginHorizontal
19     android:keepScreenOn
20     android:selectAllOnFocus
21     android:orientation
22     android:accessibilityLiveRegion
23     android:elevation
24     android:layoutDirection
25     android:pointerIcon
26     android:rotation
27     android:scrollbarFadeDuration
28   />
29 </androidx.constraintlayout.widget.ConstraintLayout>
```

Gambar 5.1 Langkah 1 Penggunaan Fungsi OnClick

Langkah 2:

Lalu akan muncul tampilan onClick yang masih kosong kemudian lakukan pengisian pada pengaktifan onClick, misalnya proses.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:
  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:layout_marginTop="50dp"
  tools:context=".MainActivity">

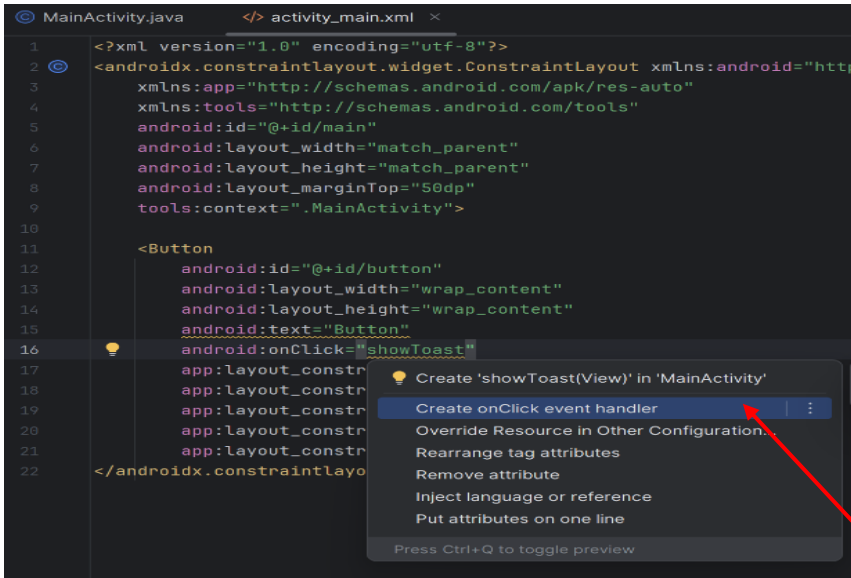
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:onClick=|||
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.395" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Gambar 5.2 Langkah 2 Penggunaan Fungsi onClick

Langkah 3:

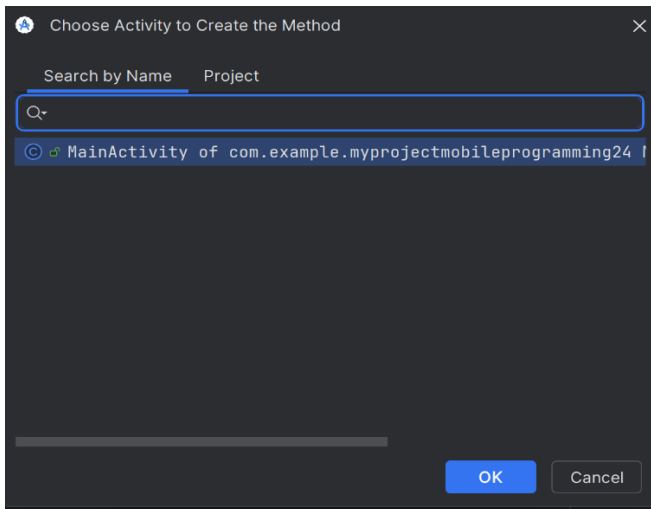
Setelah memberikan nama pada onClick, kemudian blok nama onClick tersebut kemudian tekan Alt + Enter secara bersamaan pada keyboard kemudian klik **Create onClick event handler**.



Gambar 5.3 Langkah 3 Penggunaan Fungsi OnClick

Langkah 4:

Kemudian pilih Activity yang akan digunakan untuk pemrosesan tombol tersebut, lalu klik ok.



Gambar 5.4 Langkah 4 Penggunaan Fungsi OnClick

Setelah itu akan diarahkan kepada menu ke menu activity.java untuk melakukan eksekusi selanjutnya. Artinya tombol tersebut sudah dapat untuk dieksekusi untuk menampilkan sebuah perintah seperti pesan dan lainnya.

Untuk penggunaan toast dilakukan pada coding Activity. Adapun cara mendeklarasikan toast sebagai berikut:

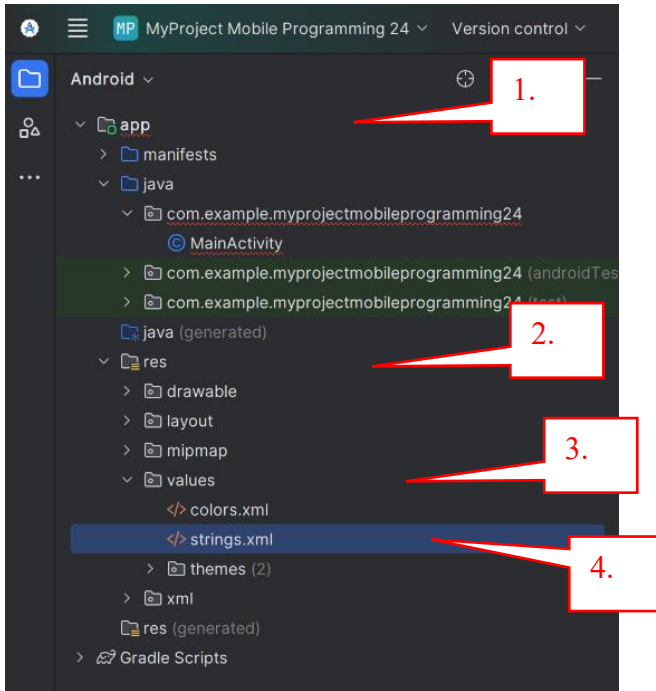


Gambar 5. 5 Pendeklarasian String Pada Fungsi Button

Untuk mengisi toast atau variable string.xml untuk menampilkan pesan lakukan langkah berikut:

Langkah 1:

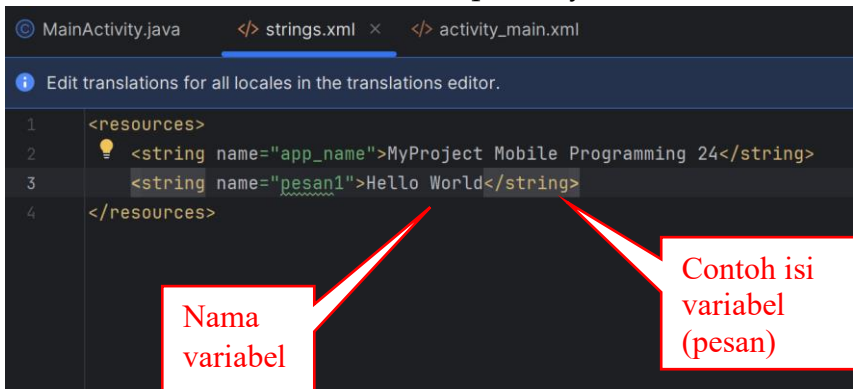
Klik **App**, kemudian **res**, lalu pilih **values** kemudian klik **strings.xml**.



Gambar 5.6 Langkah 1 Pendeklarasian String Pada Toast

Langkah 2:

Kemudian buat variabel toast dan pesannya.

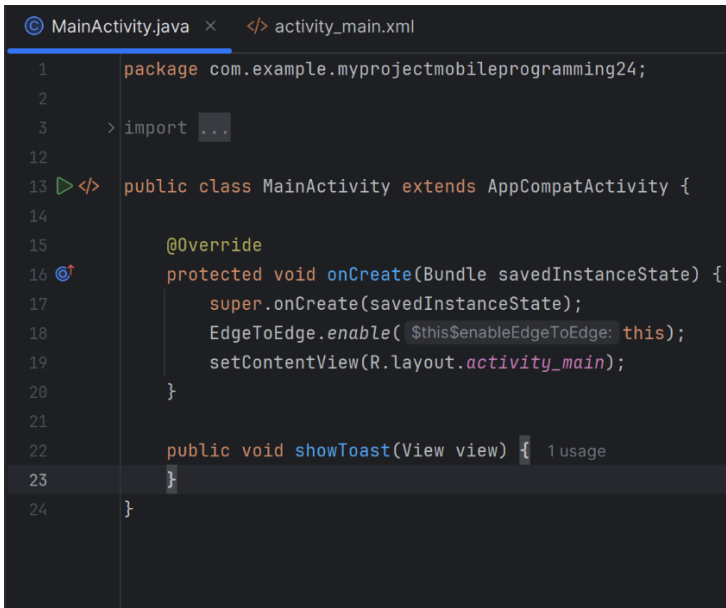


Gambar 5.7 Cara Mendeklarasikan Variabel Pada String

Berikut langkah penggunaan toast pada tombol yang telah diaktifkan:

Langkah 1:

Pada saat sebuah tombol diaktifkan maka langsung diarahkan sebuah Activity yang telah dipilih sebelumnya. Kemudian lakukan pengaktifan fungsi toast sesuai deklarasinya.

A screenshot of an IDE window showing the MainActivity.java file. The code is as follows:

```
1 package com.example.myprojectmobileprogramming24;
2
3 > import ...
12
13 <> public class MainActivity extends AppCompatActivity {
14
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         EdgeToEdge.enable($this$enableEdgeToEdge: this);
19         setContentView(R.layout.activity_main);
20     }
21
22     public void showToast(View view) { 1 usage
23     }
24 }
```

Gambar 5. 8 Langkah 1 Penggunaan Toast

Langkah 2:

Deklarasikan toast seperti langkah sebelumnya. Berikut contoh penggunaan toast pada langkah sebelumnya:

Coding layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
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:layout_marginTop="50dp"
tools:context=".MainActivity">

    <Button
android:id="@+id/button"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Button"
android:onClick="showToast"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.395" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Coding strings.xml

```
<resources>
    <string name="app_name">MyProject Mobile
Programming 24</string>
    <string name="pesan1">Seru Belajar Mobile
Programming</string>
</resources>
```

Coding activity.java

```
package com.example.myprojectmobileprogramming24;
import static android.widget.Toast.makeText;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
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);
    }

    public void showToast(View view) {
```

```
Toast toast1 = makeText(this, getString(R.string.pesan1),  
Toast.LENGTH_LONG);  
toast1.show();  
}  
}
```

Output yang dihasilkan ketika tombol Button di klik:



Gambar 5. 9 Contoh Penggunaan Toast Menggunakan Fitur OnClik

2. Penggunaan Toast Pada Button Menggunakan setOnClickListener

Selain menggunakan fitur onClick untuk mengaktifkan sebuah button dapat juga menggunakan fitur setOnClickListener. Adapun langkah untuk penerapannya:

Langkah 1:

Buatlah sebuah tombol dalam layout .xml

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Gambar 5.10 Langkah 1 Penggunaan setOnClickListener Pada Button

Langkah 2:

Berikanlah id button yang relevan untuk nama tombol tersebut misalnya bhitung.

```
<Button
    android:id="@+id/bhitung"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hitung"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Gambar 5. 11 Langkah 2 Penggunaan setOnClickListener Pada Button

Langkah 3:

Setelah itu lalu menuju Activity.Java untuk mendeklarasikan fitur setOnClickListener berikut langkahnya:

```
public class MainActivity4 extends AppCompatActivity {
    Button hitung;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });

        hitung = (Button)findViewById(R.id.bhitung);

        hitung.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(context: MainActivity4.this, text: "1 + 1 = 2", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

1. Deklarasikan variabel untuk mengakses button
2. Kaitkan variabel dengan elemen UI di layout XML
3. Gunakan variabel yang telah dideklarasikan untuk dijadikan tombol dengan fitur setOnClickListener
4. Pesan yang disampaikan

3. Menggunakan toas untuk menampilkan pesan

Gambar 5. 12 Pendekalasian setOnClickListener

Koding keseluruhan Activity.Java:

```
package com.example.myprojectmobileprogramming24;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

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 MainActivity4 extends AppCompatActivity {
    Button hitung;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main3);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(
            R.id.main), (v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
            return insets;
        });

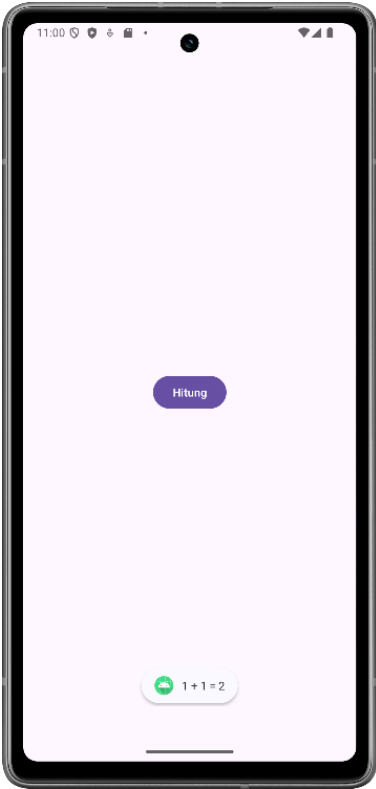
        hitung = (Button)findViewById(R.id.bhitung);

        hitung.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) {
                Toast.makeText(MainActivity4.this, "1 + 1 = 2",
                Toast.LENGTH_SHORT).show();
            }
        }
    }
}

```

```
});  
}  
}
```

Hasil Output:



Gambar 5. 13 Hasil Running Button Menggunakan setOnClickListener

3. Spinner Dan Contoh Penggunaannya

Spinner adalah elemen antarmuka pengguna yang berfungsi untuk memberikan opsi pemilihan dengan cara yang efisien dan terorganisir. Dalam aplikasi Android, spinner berfungsi seperti dropdown menu, di mana pengguna dapat

melihat dan memilih dari daftar opsi yang disediakan. Spinner sangat berguna ketika jumlah pilihan yang ada cukup banyak, sehingga tidak praktis untuk menampilkannya semua sekaligus di layar.

Penggunaan spinner membantu dalam menjaga tata letak antarmuka pengguna agar tetap bersih dan mudah dinavigasi. Selain itu, spinner juga meningkatkan pengalaman pengguna dengan memudahkan mereka dalam membuat pilihan, mengurangi waktu yang dibutuhkan untuk menelusuri berbagai opsi.

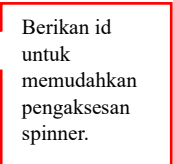
Spinner biasanya digunakan dalam berbagai konteks, seperti pemilihan item dari daftar, pengaturan preferensi pengguna, dan lainnya. Dengan implementasi yang tepat, spinner dapat menjadi alat yang efektif dalam pengembangan aplikasi yang intuitif dan responsif.

Berikut contoh pembuatan spinner:

Langkah 1:

Dalam layout xml tambahkan elemen spinner:

```
<Spinner
    android:id="@+id/namakota"
    android:layout_width="266dp"
    android:layout_height="35dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.62"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.216" />
```



Gambar 5. 14 Langkah 1 Penggunaan Spinner

Langkah 2:

Lakukan pendeklarasian array untuk menyimpan data-data dalam spinner pada string.xml. berikut contohnya:

```
<string-array name="namakota">
    <item>Bukittinggi</item>
    <item>Medan</item>
    <item>Padang</item>
    <item>Banda Aceh</item>
    <item>Bandung</item>
</string-array>
```

Gambar 5. 15 Pendeklarasian Array

Langkah 3:

Setelah itu lakukan pendeklarasian spinner pada Activity.Java:

```
package com.example.myprojectmobileprogramming24;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.Toast;

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 MainActivity4 extends AppCompatActivity {
```

```
Button tampilkankota;  
Spinner namakota;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    EdgeToEdge.enable(this);  
    setContentView(R.layout.activity_main3);
```

```
    ViewCompat.setOnApplyWindowInsetsListener(findViewById  
(R.id.main), (v, insets) -> {  
        Insets systemBars =  
insets.getInsets(WindowInsetsCompat.Type.systemBars());  
        v.setPadding(systemBars.left, systemBars.top,  
systemBars.right, systemBars.bottom);  
        return insets;  
    });
```

```
    namakota = findViewById(R.id.namakota);  
    tampilkankota = findViewById(R.id.btampil);
```

```
    ArrayAdapter<CharSequence> adapter =  
ArrayAdapter.createFromResource(this,  
    R.array.namakota,  
android.R.layout.simple_spinner_item);
```

```
    adapter.setDropDownViewResource(android.R.layout.simpl  
e_spinner_dropdown_item);  
    namakota.setAdapter(adapter);
```

```

        tampilkankota.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String kotapilihan=
namakota.getSelectedItem().toString();
        Toast.makeText(MainActivity4.this, "Kota yang
dipilih: " + kotapilihan, Toast.LENGTH_SHORT).show();
    }
});
}
}

```

Penjelasan coding:

- a) namakota dan tampilkankota diinisialisasi menggunakan `findViewById()`, yang menghubungkan variabel dalam kode dengan elemen UI di file layout `.xml`.
- b) namakota adalah elemen Spinner, yang berfungsi sebagai dropdown list atau daftar pilihan.
- c) tampilkankota adalah Button, yang akan digunakan untuk memicu aksi (menampilkan kota yang dipilih) ketika diklik.
- d) ArrayAdapter adalah objek yang menghubungkan data dengan tampilan Spinner.
- e) `mySpinner.setAdapter(adapter)`; menghubungkan adapter ke Spinner, sehingga Spinner akan menampilkan data namakota.

Dan untuk button menggunakan fungsi `setOnClickListener` Listener.

Koding keseluruhan dalam layout .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=".MainActivity4">

<Spinner
    android:id="@+id/namakota"
    android:layout_width="266dp"
    android:layout_height="35dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.62"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.216" />

<Button
    android:id="@+id/btampil"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

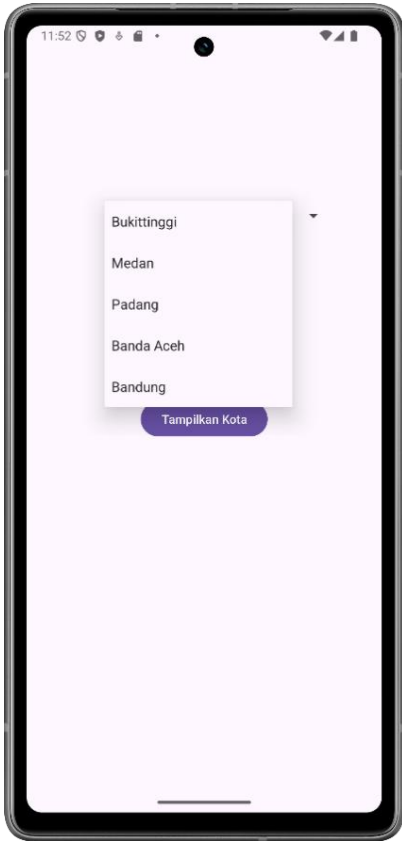
```
android:text="Tampilkan Kota"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

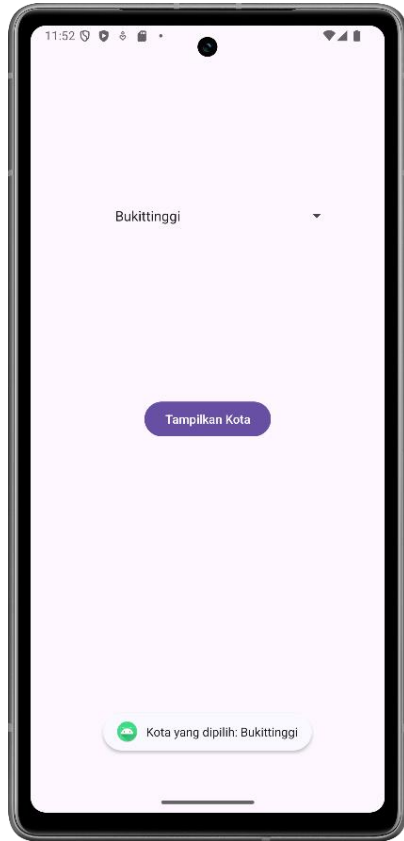
Koding dalam array pada string.xml:

```
<resources>  
  <string-array name="namakota">  
    <item>Bukittinggi</item>  
    <item>Medan</item>  
    <item>Padang</item>  
    <item>Banda Aceh</item>  
    <item>Bandung</item>  
  </string-array>  
</resources>
```

Hasil Output:



Gambar 5.16 Hasil Running Spinner



Gambar 5.17 Penggunaan Spinner

BAB VI

VARIABEL DAN LOGIKA

1. Variabel Dan Deklarasinya

Variabel adalah elemen dasar dalam pemrograman yang memungkinkan pengembang untuk menyimpan dan mengelola data. Setiap variabel memiliki nama yang unik dan dapat diisi dengan nilai yang berbeda-beda sepanjang eksekusi program. Tipe data dari variabel menentukan jenis nilai yang dapat disimpan, seperti angka bulat (integer), angka pecahan (float), karakter (char), atau teks (string).

Deklarasi variabel juga penting karena memberikan informasi kepada compiler atau interpreter tentang cara memperlakukan nilai yang tersimpan dalam variabel tersebut. Misalnya, jika sebuah variabel dideklarasikan sebagai tipe data integer, maka hanya nilai numerik bulat yang dapat disimpan dan operasi yang sesuai untuk tipe data tersebut dapat dilakukan.

Untuk mendeklarasikan sebuah variabel dalam `activity.java` yaitu menentukan tipe data dari variabel kemudian memberikan nama variabel tersebut. Contoh:

```

// Mendeklarasikan variabel bertipe int
int usia = 20;

// Mendeklarasikan variabel bertipe String
String univ = "UIN Sjech M. Djamil Djambek Bukittinggi";

// Mendeklarasikan variabel bertipe double
double tinggi = 1.85;

```

2. Operator Aritmatika Dan Logika

Operator aritmatika adalah tanda yang digunakan untuk melakukan berbagai operasi matematis dalam bahasa pemrograman, termasuk penjumlahan menggunakan simbol (+), pengurangan menggunakan simbol (-), perkalian menggunakan simbol (*), pembagian menggunakan simbol (/), dan operasi modulus menggunakan simbol (%). Operator logika dalam pemrograman dipakai untuk menghasilkan nilai boolean true atau false dari 2 kondisi atau lebih adapun operator logika dalam mobile programming seperti penggunaan IF, IF Else dan lainnya.

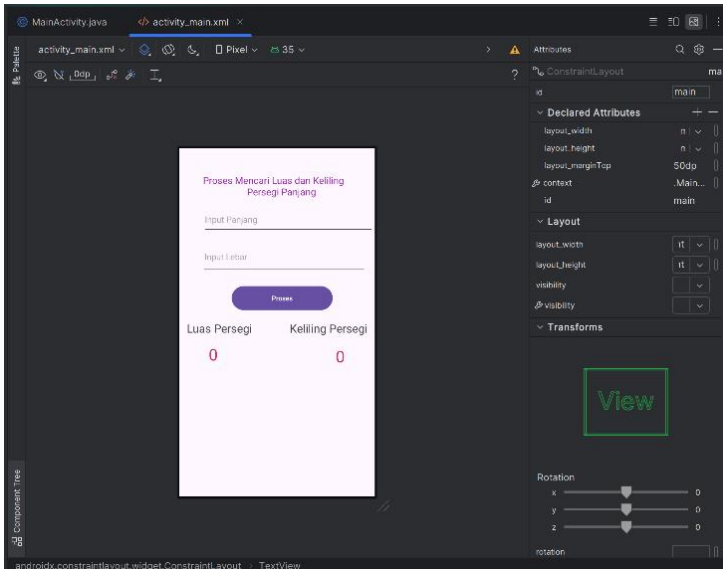
3. Latihan

a. Kasus 1 Mencari Luas Dan Keliling Persegi Panjang

Mencari luas dan keliling persegi panjang, dimana menggunakan form khusus dimana user dapat mengisi form tersebut untuk mencari luas dan keliling dari persegi panjang dan menggunakan button atau tombol yang di klik untuk melakukan pemrosesan. Berikut penyelesaiannya.

Langkah 1:

Desainlah pada layout sebagai berikut:



Gambar 6. 1 Desain Layout

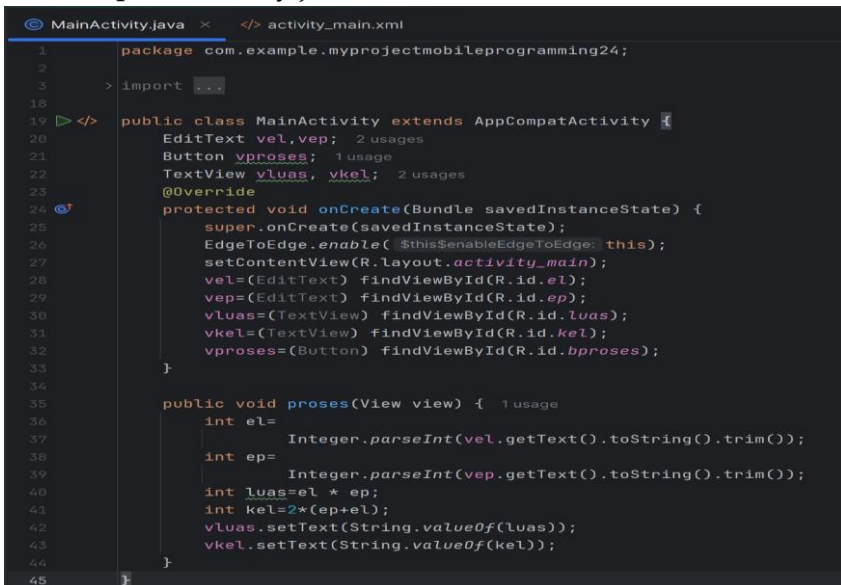
Pada setiap elemen penting seperti EditText diberikan sebuah Id ini berfungsi untuk dapat memanggil elemen tersebut. Contohnya:

```
<EditText
    android:id="@+id/e1"
    android:layout_width="343dp"
    android:layout_height="67dp"
    android:layout_marginTop="16dp"
    android:ems="10"
    android:hint="Input Lebar"
    android:inputType="number"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.72"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/ep" />
```

Gambar 6. 2 Deklarasi id

Langkah 2:

Setelah semua elemen yang diperlukan diberi id dan button juga di telah diaktifkan maka lakukan pendeklarasian variabel pada activity.java.



```
1 package com.example.myprojectmobileprogramming24;
2
3 > import ...
18
19 </> public class MainActivity extends AppCompatActivity {
20     EditText vel,vep; 2 usages
21     Button vproses; 1 usage
22     TextView vluas, vkel; 2 usages
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         EdgeToEdge.enable(this);
27         setContentView(R.layout.activity_main);
28         vel=(EditText) findViewById(R.id.el);
29         vep=(EditText) findViewById(R.id.ep);
30         vluas=(TextView) findViewById(R.id.luas);
31         vkel=(TextView) findViewById(R.id.kel);
32         vproses=(Button) findViewById(R.id.bproses);
33     }
34
35     public void proses(View view) { 1 usage
36         int el=
37             Integer.parseInt(vel.getText().toString().trim());
38         int ep=
39             Integer.parseInt(vep.getText().toString().trim());
40         int luas=el * ep;
41         int kel=2*(ep+el);
42         vluas.setText(String.valueOf(luas));
43         vkel.setText(String.valueOf(kel));
44     }
45 }
```

Gambar 6. 3 Pendeklarasian Variabael

Penjelasan koding:

- EditText vel, vep: Dua objek EditText yang digunakan untuk menerima input dari pengguna. vel mungkin digunakan untuk memasukkan nilai panjang, sedangkan vep untuk lebar.
- Button vproses: Sebuah tombol yang ketika ditekan, akan memicu perhitungan luas dan keliling.
- TextView vluas, vkel: Dua objek TextView yang digunakan untuk menampilkan hasil perhitungan luas dan keliling.
- findViewById(R.id.el) untuk Mengaitkan variabel Java dengan elemen UI dalam layout XML menggunakan ID

yang ditentukan. Misalnya, R.id.el adalah ID dari EditText untuk panjang.

- e) Integer.parseInt(...) untuk mengonversi teks yang dimasukkan di EditText menjadi integer setelah menghapus spasi di sekitar input dengan trim().
- f) int luas = el * ep; untuk menghitung luas dengan mengalikan panjang (el) dan lebar (ep).
- g) int kel = 2 * (ep + el); untuk menghitung keliling dengan rumus keliling persegi panjang.
- h) vluas.setText(String.valueOf(luas)); untuk mengupdate TextView untuk menampilkan hasil luas.
- i) vkel.setText(String.valueOf(kel)); untuk Mengupdate TextView untuk menampilkan hasil keliling.

Berikut koding keseluruhan:

Koding .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"
android:layout_marginTop="50dp"
tools:context=".MainActivity">

    <TextView
android:id="@+id/textView6"
```

```
android:layout_width="329dp"
android:layout_height="27dp"
android:layout_marginTop="4dp"
android:text="Proses Mencari Luas dan Keliling Persegi
Panjang"
android:textColor="#9730A8"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.609"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
android:id="@+id/textView3"
android:layout_width="329dp"
android:layout_height="27dp"
android:layout_marginTop="28dp"
android:text="Persegi Panjang"
android:textAlignment="center"
android:textColor="#9730A8"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.609"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
android:id="@+id/e1"
android:layout_width="343dp"
android:layout_height="67dp"
```

```

android:layout_marginTop="16dp"
android:ems="10"
android:hint="Input Lebar"
android:inputType="number"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.72"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ep" />

    <EditText
android:id="@+id/ep"
android:layout_width="358dp"
android:layout_height="58dp"
android:layout_marginTop="40dp"
android:ems="10"
android:hint="Input Panjang"
android:inputType="number"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.943"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/textView6"
/>

    <Button
android:id="@+id/bproses"
android:layout_width="212dp"
android:layout_height="58dp"
android:layout_marginTop="24dp"
android:onClick="proses"
android:text="Proses"

```

```
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.554"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/e1"  
tools:ignore="OnClick" />
```

```
<TextView  
android:id="@+id/textView8"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="312dp"  
android:text="Luas Persegi"  
android:textSize="24sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.058"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
android:id="@+id/text"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="312dp"  
android:text="Keliling Persegi"  
android:textSize="24sp"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.935"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
android:id="@+id/luas"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="360dp"
android:text="0"
android:textColor="#C53566"
android:textSize="34sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.158"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
android:id="@+id/kel"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="364dp"
android:text="0"
android:textColor="#C53566"
android:textSize="34sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.839"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Koding activity.java:

```
package com.example.myprojectmobileprogramming24;
import static android.widget.Toast.makeText;
```

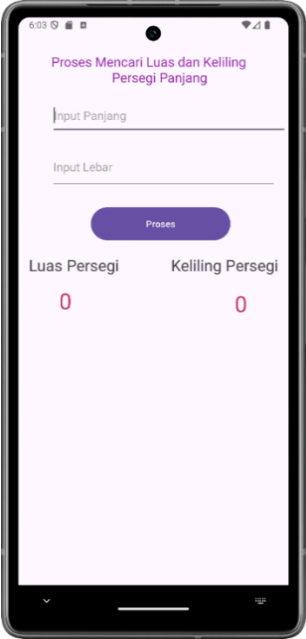
```
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

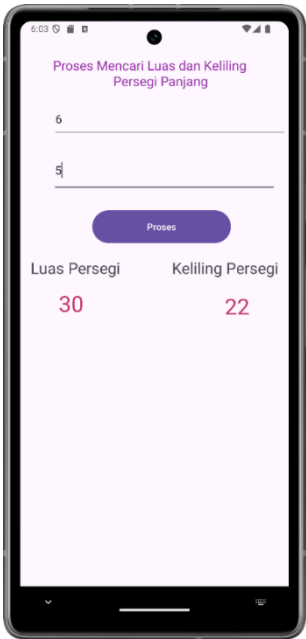
public class MainActivity extends AppCompatActivity {
    EditText vel,vep;
    Button vproses;
    TextView vluas, vkel;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        vel=(EditText) findViewById(R.id.el);
        vep=(EditText) findViewById(R.id.ep);
        vluas=(TextView) findViewById(R.id.luas);
        vkel=(TextView) findViewById(R.id.kel);
        vproses=(Button) findViewById(R.id.bproses);
    }
}
```

```
public void proses(View view) {
    int el=
        Integer.parseInt(vcl.getText().toString().trim());
    int ep=
        Integer.parseInt(vcp.getText().toString().trim());
    int luas=el * ep;
    int kel=2*(ep+el);
    vluas.setText(String.valueOf(luas));
    vkel.setText(String.valueOf(kel));
}
}
```

Hasil Output:



Gambar 6. 5 Hasil Running Latihan Kasus 1



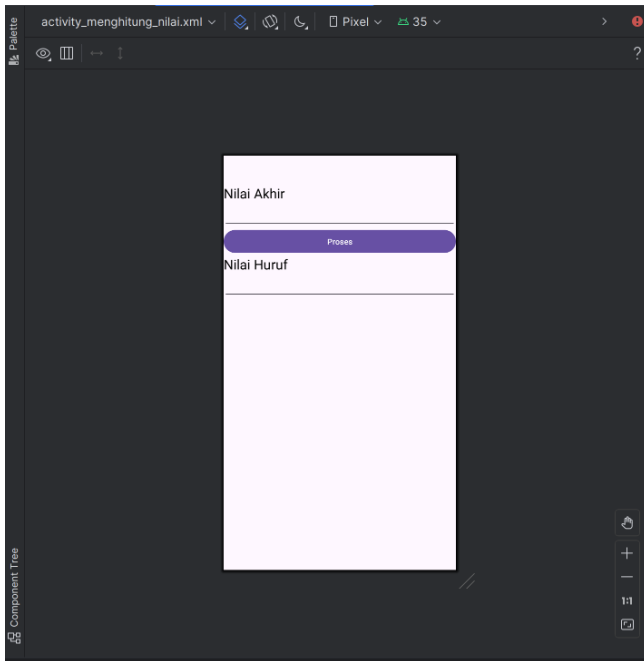
Gambar 6. 4 Hasil Running Setelah Di Proses

b. Kasus 2 Menghitung Kriteria Nilai

Menentukan penghitungan suatu nilai dimana jika nilai yang diinputkan diatas 85 maka akan mendapatkan nilai A, sedangkan nilai B untuk nilai lebih dari 75 dan selainnya mendapatkan nilai C. Berikut penyelesaiannya:

Langkah 1:

Lakukanlah desain komponen penting yang wajib ada sebagai berikut:




Gambar 6. 6 Desain Layout Latihan Logika IF

Pada setiap elemen penting seperti EditText diberikan sebuah Id ini berfungsi untuk dapat memanggil elemen tersebut. Contohnya:

```

<AutoCompleteTextView
    android:id="@+id/ena"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

```



Gambar 6. 7 Pendeklarasian id Pada Setiap Elemen

Langkah 2:

Setelah semua elemen yang diperlukan diberi id dan button juga di telah diaktifkan maka lakukan pendeklarasian variabel pada activity.java.

```

public class MenghitungNilai extends AppCompatActivity {
    EditText vna,vnh; 2 usages
    Button vbproses; 1 usage
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable( $this$enableEdgeToEdge: this);
        setContentView(R.layout.activity_menghitung_nilai);
        vna=(EditText) findViewById(R.id.ena);
        vnh=(EditText) findViewById(R.id.enh);
        vbproses=(Button)findViewById(R.id.bproses);
    }
    public void hitung(View view) { 1 usage
        int angka;
        String nh;
        String nangka= vna.getText().toString();
        angka=Integer.parseInt(nangka);
        if(angka>80)
        {
            nh="A";
        }
        else if(angka>70)
        {
            nh="B";
        }
        else
        {
            nh="C";
        }
        vnh.setText(nh);
    }
}

```

Gambar 6. 8 Contoh Penggunaan Logika If

Penjelasan koding:

- a) EditText vna dan vnh adalah variabel yang akan memegang referensi ke input teks (kolom isian angka), sementara Button vbproses adalah tombol yang akan digunakan untuk memproses perhitungan nilai.
- b) vna, vnh, dan vbproses dipetakan ke elemen UI dalam layout XML berdasarkan id. findViewById digunakan untuk mencari elemen berdasarkan id yang ada di file XML.
- c) `String angka = vna.getText().toString();` untuk mengambil nilai yang dimasukkan pengguna di kolom vna dan menyimpannya sebagai string angka.
- d) String nh merupakan variabel yang digunakan untuk menerima nilai yang diinputkan dan ini bersifat sementara.
- e) `angka = Integer.parseInt(angka);` untuk mengonversi string angka menjadi bilangan bulat (integer) dan menyimpannya dalam variabel angka.
- f) Struktur if-else digunakan untuk menentukan nilai huruf (nh) berdasarkan nilai angka jika nilai akhir yang diinputkan lebih besar dari 80, maka nh diberi nilai "A". Jika nilai akhir lebih besar dari 70, maka nh diberi nilai "B" selain itu, diberi nilai "C".
- g) `vnh.setText(nh);` untuk menampilkan nilai huruf (nh) pada kolom vnh setelah proses perhitungan selesai.

Berikut koding keseluruhan:

Koding .xml:

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

xmlns:android="http://schemas.android.com/apk/res/a
ndroid"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/textView7"
        android:layout_marginTop="50dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Nilai Akhir"
        android:textColor="@color/black"
        android:textSize="24sp" />

    <AutoCompleteTextView
        android:id="@+id/ena"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <Button
        android:id="@+id/bproses"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="hitung"
        android:text="Proses" />
```

```

<TextView
    android:id="@+id/textView10"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Nilai Huruf"
    android:textColor="@color/black"
    android:textSize="24sp" />
<AutoCompleteTextView
    android:id="@+id/enh"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>

```

Koding .java:

```

package com.example.myprojectmobileprogramming24;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

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 MenghitungNilai extends AppCompatActivity
{
    EditText vna,vnh;
    Button vbproses;

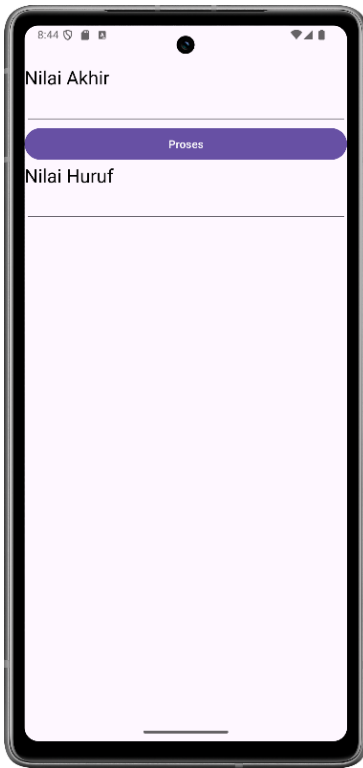
```

```

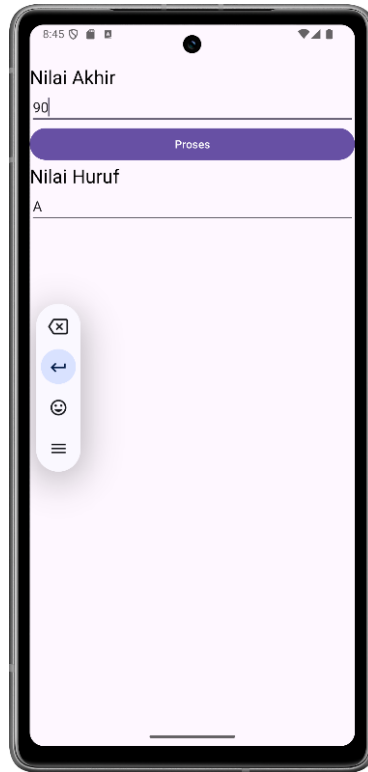
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_menghitung_nilai);
    vna=(EditText) findViewById(R.id.ena);
    vnh=(EditText) findViewById(R.id.enh);
    vbproses=(Button)findViewById(R.id.bproses);
}
public void hitung(View view) {
    int angka;
    String nh;
    String nangka= vna.getText().toString();
    angka=Integer.parseInt(nangka);
    if(angka>80)
    {
        nh="A";
    }
    else if(angka>70)
    {
        nh="B";
    }
    else
    {
        nh="C";
    }
    vnh.setText(nh);
}
}

```

Hasil output:



Gambar 6. 10 Hasil Running latihan Kasus 2



Gambar 6. 9 Hasil Running Setelah Diproses

c. Kasus 3 Menentukan Fasilitas Dan Dana

Yaitu untuk menentukan sebuah dana bantuan, fasilitas dan lainnya atau Tampilkan Pesan Pada Setiap Pemrosesan, sesuai dengan Kode Perwakilan Daerah yang diinputkan dengan ketentuan sebagai tabel berikut:

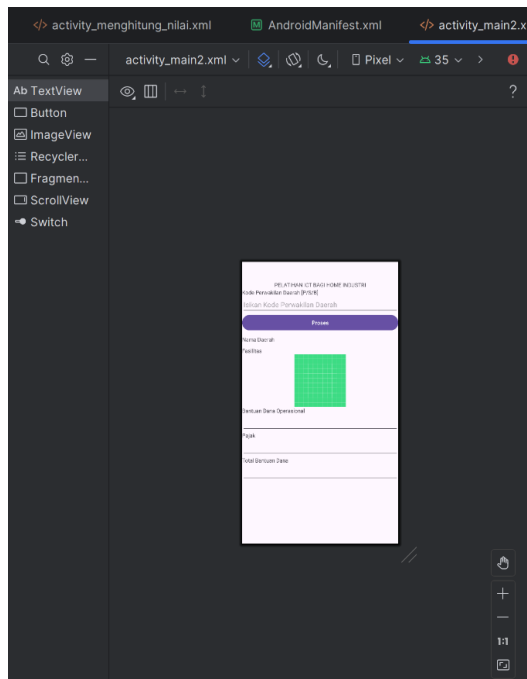
Kode	Daerah	Fasilitas	Dana Bantuan	Hotel
P	Padang	Penginapan + Buku Pelatihan	Rp. 4.000.000	Foto Hotel 1
B	Bukittinggi	Penginapan	Rp. 3.000.000	Foto Hotel 2
S	Solok	Penginapan + Buku Pelatihan	Rp. 2.000.000	Foto Hotel 3

- Pajak 10% dari Dana Bantuan Operasional

Penyelesaian:

Langkah 1:

Lakukanlah desain komponen penting yang wajib ada sebagai berikut:



Gambar 6. 11 Desain Layout Kasus 3

Langkah 2:

Setelah semua elemen yang diperlukan diberi id dan button juga di telah diaktifkan maka lakukan pendeklarasian variabel pada activity.java.

```
package com.example.myprojectmobileprogramming24;

import android.media.Image;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.text.NumberFormat;
import java.util.Locale;

public class MainActivity2 extends AppCompatActivity {
    EditText vedana, vepajak, vetotalbantuan,
    vkodepenginapan;
    TextView daerah, fasilitas;
    Button proses;
    ImageView imageView;
```

```

int vdana;
double vpajak, vttotal;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main2);
    vkodepenginapan=(EditText)
findViewById(R.id.kodepenginapan);
    vedana = (EditText) findViewById(R.id.dana);
    vepajak = (EditText) findViewById(R.id.pajak);
    vetotalbantuan = (EditText)
findViewById(R.id.totalbantuannya);
    daerah = (TextView) findViewById(R.id.daerah);
    fasilitas = (TextView) findViewById(R.id.fasilitas);
    proses = (Button) findViewById(R.id.bproses);
    imageView= (ImageView)
findViewById(R.id.gambarpenginapan);
}

public void proses1(View view) {
    String vvdaerah, vvfasilitas, vvkodepenginapan;
    Image img;

    vvkodepenginapan =
vkodepenginapan.getText().toString();
    if (vvkodepenginapan.equals("P"))
    {
        vvdaerah = "Padang";
    }
}

```

```
vvfasilitas = "Penginapan + Buku Pelatihan";  
vdana = 4000000;  
vpajak = 0.1 * vdana;  
vtotal = vdana - vpajak;  
imageView.setVisibility(view.VISIBLE);
```

```
imageView.setImageResource(R.drawable._4_best_hotels_in_  
los_angeles__from_santa_monica_to_downtown);
```

```
} else if (vvkodepenginapan.equals("B")) {  
    vvdaerah = "Bukittinggi";  
    vvfasilitas = "Penginapan";  
    vdana = 3000000;  
    vpajak = 0.1 * vdana;  
    vtotal = vdana - vpajak;  
    imageView.setVisibility(view.VISIBLE);
```

```
imageView.setImageResource(R.drawable.ian_schrager_and_  
john_pawson_redefine_california_luxury_at_west_hollywood_edit  
ion_hotel);
```

```
} else {  
    vvdaerah = "Solok";  
    vvfasilitas = "Penginapan + Buku Pelatihan";  
    vdana = 2000000;  
    vpajak = 0.1 * vdana;  
    vtotal = vdana - vpajak;  
    imageView.setVisibility(view.VISIBLE);
```

```
imageView.setImageResource(R.drawable.shangri_la_hotel__
```

```

    at_the_shard_london_hotel_review__cond__nast_traveler);
    }

    if (vvkodepenginapan.isEmpty()) {
        Toast.makeText(this, "Kode penginapan tidak boleh
kosong", Toast.LENGTH_SHORT).show();
        return;
    }

    Toast pesan = Toast.makeText(this,
getString(R.string.testoast) + vvdaerah,
Toast.LENGTH_LONG);
    pesan.show();

    NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(new Locale("id", "ID"));
    String formattedDana = formatRupiah.format(vvdana);
    String formattedPajak = formatRupiah.format(vpajak);
    String formattedTotal = formatRupiah.format(vtotal);

    daerah.setText(vvdaerah);
    fasilitas.setText(vvfasilitas);
    vedana.setText(formattedDana);
    vepajak.setText(formattedPajak);
    vetotalbantuan.setText(formattedTotal);
}
}

```

Penjelasan koding:

- a) EditText (vedana, vepajak, vetotalbantuan, vkodepenginapan) untuk komponen input teks yang memungkinkan pengguna memasukkan kode yang telah ditwntukan seperti P, B dan S untuk mengakses informasi.
- b) TextView (daerah, fasilitas) sebagai komponen teks untuk menampilkan hasil informasi seperti nama daerah dan fasilitas.
- c) Button (proses) sebagai tombol untuk memproses input dan menghitung hasil berdasarkan kode penginapan dan diinputkan oleh user.
- d) ImageView (imageView) digunakan untuk menampilkan gambar terkait dengan kode yang dipilih.
- e) int (vdana) merupakan variabel untuk menyimpan jumlah dana berdasarkan pilihan kode penginapan.
- f) double (vpajak, vtotal) merupakan variabel vpajak untuk menyimpan nilai pajak, dan vtotal menyimpan jumlah total dana setelah dikurangi pajak.
- g) findViewById() untuk menghubungkan variabel dengan komponen UI yang sesuai di dalam layout XML.
- h) Vvkodepenginapan untuk menyimpan kode penginapan yang dimasukkan oleh pengguna untuk menentukan kota, fasilitas, serta dana.
- i) NumberFormat.getCurrencyInstance(new Locale("id", "ID")) untuk mengatur format mata uang menjadi Rupiah (IDR).
- j) formatRupiah.format() untuk mengonversi vdana, vpajak, dan vtotal ke format mata uang Rupiah.

- k) `setText()` untuk menampilkan vvd daerah dan vvfasilitas di `TextView`, serta vdana, vpajak, dan vttotal yang sudah diformat sebagai Rupiah di `EditText`.
- l) `imageView.setVisibility(view.VISIBLE)` digunakan untuk menampilkan gambar kemudian `imageView.setImageResource(R.drawable.)` digunakan untuk menampilkan gambar dari direktori penyimpanan kita secara otomatis berdasarkan kode yang digunakan.

Koding secara keseluruhan:

Koding layout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
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"
android:textAlignment="center"
tools:context=".MainActivity2">

    <TextView
android:id="@+id/textView58"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:layout_marginTop="50dp"
```

```
android:text="PELATIHAN ICT BAGI HOME INDUSTRI"  
android:textAlignment="center" />
```

```
<TextView  
android:id="@+id/textView59"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:text="Kode Perwakilan Daerah [P/S/B]"  
android:textAlignment="textStart" />
```

```
<EditText  
android:id="@+id/kodepenginapan"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:ems="10"  
android:hint="Isikan Kode Perwakilan Daerah"  
android:inputType="text" />
```

```
<Button  
android:id="@+id/bproses"  
android:onClick="proses1"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:text="Proses" />
```

```
<TextView  
android:id="@+id/daerah"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dp"
```

```
android:text="Nama Daerah" />
```

```
<TextView  
android:id="@+id/fasilitas"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dp"  
android:text="Fasilitas" />
```

```
<ImageView  
android:id="@+id/gambarpenginapan"  
android:layout_width="match_parent"  
android:layout_height="136dp"  
app:srcCompat="@drawable/ic_launcher_background" />
```

```
<TextView  
android:id="@+id/textView61"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:text="Bantuan Dana Operasional"  
android:textAlignment="textStart" />
```

```
<EditText  
android:id="@+id/dana"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:ems="10"  
android:inputType="number" />
```

```
<TextView
```

```
android:id="@+id/textView62"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Pajak"
android:textAlignment="textStart" />

    <EditText
android:id="@+id/pajak"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10"
android:inputType="number" />

    <TextView
android:id="@+id/textView63"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Total Bantuan Dana"
android:textAlignment="textStart" />

    <EditText
android:id="@+id/totalbantuannya"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10"
android:inputType="number | numberDecimal | numberSigned" />
</LinearLayout>
```

Koding Activity.java:

```
package com.example.myprojectmobileprogramming24;

import android.media.Image;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.text.NumberFormat;
import java.util.Locale;

public class MainActivity2 extends AppCompatActivity {
    EditText vedana, vepajak, vetotalbantuan,
    vkodepenginapan;
    TextView daerah, fasilitas;
    Button proses;
    ImageView imageView;
    int vdana;
    double vpajak, vtotal;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_main2);
    vkodepenginapan=(EditText)
findViewById(R.id.kodepenginapan);
    vedana = (EditText) findViewById(R.id.dana);
    vepajak = (EditText) findViewById(R.id.pajak);
    vetotalbantuan = (EditText)
findViewById(R.id.totalbantuannya);
    daerah = (TextView) findViewById(R.id.daerah);
    fasilitas = (TextView) findViewById(R.id.fasilitas);
    proses = (Button) findViewById(R.id.bproses);
    imageView= (ImageView)
findViewById(R.id.gambarpenginapan);
}

public void proses1(View view) {
    String vvdaerah, vvfasilitas, vvkodepenginapan;
    Image img;

    vvkodepenginapan =
vkodepenginapan.getText().toString();
    if (vvkodepenginapan.equals("P"))
    {
        vvdaerah = "Padang";
        vvfasilitas = "Penginapan + Buku Pelatihan";
        vdana = 4000000;
        vpajak = 0.1 * vdana;
    }
}

```

```

        vtotal = vdana - vpajak;
        imageView.setVisibility(view.VISIBLE);

imageView.setImageResource(R.drawable._4_best_hotels_in_
los_angeles__from_santa_monica_to_downtown);

    } else if (vkodepenginapan.equals("B")) {
        vvdaerah = "Bukittinggi";
        vvfasilitas = "Penginapan";
        vdana = 3000000;
        vpajak = 0.1 * vdana;
        vtotal = vdana - vpajak;
        imageView.setVisibility(view.VISIBLE);

imageView.setImageResource(R.drawable.ian_schrager_and_
john_pawson_redefine_california_luxury_at_west_hollywood_edit
ion_hotel);

    } else {
        vvdaerah = "Solok";
        vvfasilitas = "Penginapan + Buku Pelatihan";
        vdana = 2000000;
        vpajak = 0.1 * vdana;
        vtotal = vdana - vpajak;
        imageView.setVisibility(view.VISIBLE);

imageView.setImageResource(R.drawable.shangri_la_hotel__
at_the_shard__london_hotel_review__cond__nast_traveler);
    }

```

```

    if (vvkodepenginapan.isEmpty()) {
        Toast.makeText(this, "Kode penginapan tidak boleh
kosong", Toast.LENGTH_SHORT).show();
        return;
    }

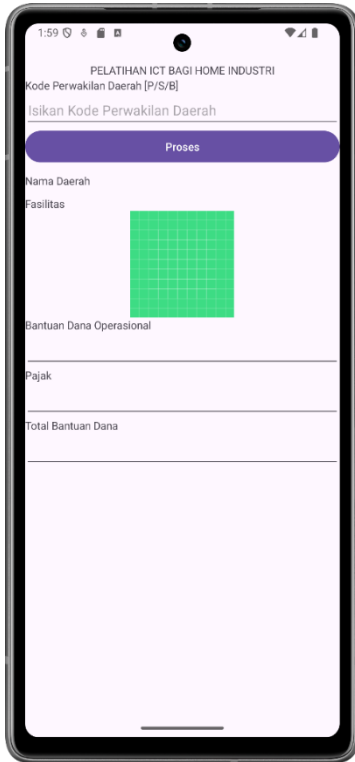
    Toast pesan = Toast.makeText(this,
getString(R.string.testoast) + vvdaerah,
Toast.LENGTH_LONG);
    pesan.show();

    NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(new Locale("id", "ID"));
    String formattedDana = formatRupiah.format(vdana);
    String formattedPajak = formatRupiah.format(vpajak);
    String formattedTotal = formatRupiah.format(vtotal);

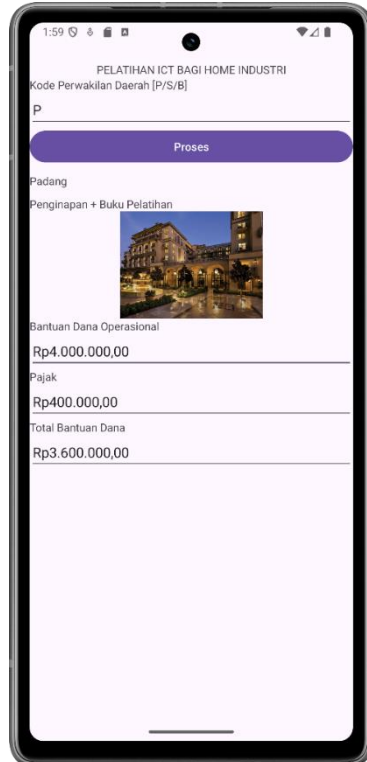
    daerah.setText(vvdaerah);
    fasilitas.setText(vvfasilitas);
    vedana.setText(formattedDana);
    vepajak.setText(formattedPajak);
    vetotalbantuan.setText(formattedTotal);
}
}

```

Hasil Output:



Gambar 6. 12 Hasil Running Latihan Kasus 3



Gambar 6. 13 hasil Running Setelah Di Proses

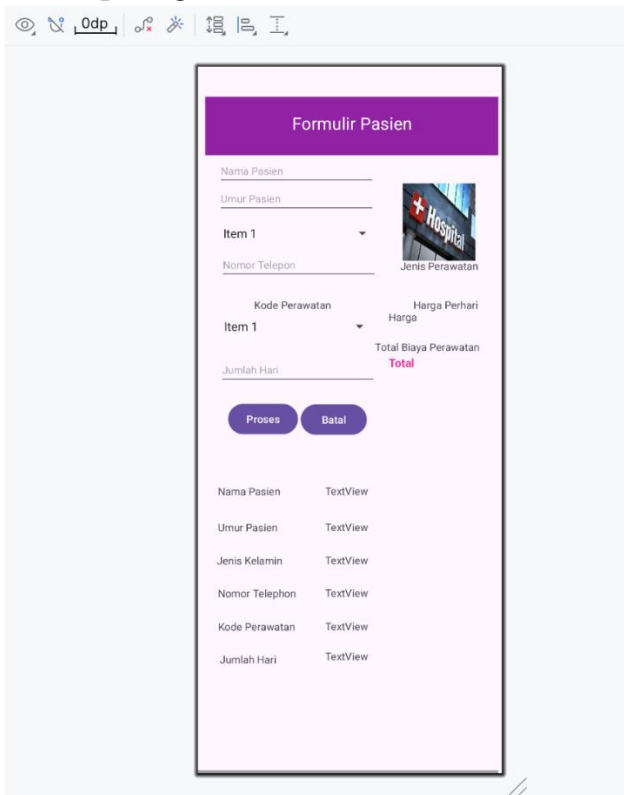
d. Kasus 4 Menentukan Jenis Perawatan Pasien Dalam Rumah Sakit

Yaitu untuk menentukan jenis perawatan, harga perawatan perhari dan total biaya yang dibayarkan berdasarkan lamanya perawatannya. Kemudian outputnya menghasilkan data – data pasien yang telah di inputkan setelah di tekan tombol proses dan tombol batal untuk membatalkan proses tersebut.

Berikut cara penyelesaiannya:

Langkah 1:

Menentukan desain yang wajib yaitu seperti edit text, button text view, image view dan lainnya lalu lakukan pendesainan kurang lebih seperti gambar berikut:



Gambar 6. 14 Desain Layout Kasus 3

Berikut koding .xml:

```
<?xml version="1.0" encoding="utf-8"?>
<HorizontalScrollView
xmlns:android="http://schemas.android.com/apk/res/and
roid"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:scrollbars="horizontal">
<ScrollView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="10dp"
android:scrollbars="vertical"
tools:layout_editor_absoluteX="-10dp"
tools:layout_editor_absoluteY="2dp">

<androidx.constraintlayout.widget.ConstraintLayout
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="wrap_content"
tools:context=".Mpasiem"
android:layout_marginTop="10dp"
tools:layout_editor_absoluteX="0dp"
tools:layout_editor_absoluteY="12dp">

<View
android:id="@+id/view"
```

```
        android:layout_width="390dp"
        android:layout_height="78dp"
        android:layout_marginTop="20dp"

    android:background="@android:color/system_accent1_800"
    android:backgroundTint="#9122A4"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/tjumlah"
    android:layout_width="208dp"
    android:layout_height="34dp"
    android:layout_marginTop="756dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.877"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/tkode"
    android:layout_width="208dp"
    android:layout_height="34dp"
    android:layout_marginTop="716dp"
    android:text="TextView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.877"
```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/tno"  
    android:layout_width="208dp"  
    android:layout_height="34dp"  
    android:layout_marginTop="672dp"  
    android:text="TextView"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.877"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/tjk"  
    android:layout_width="208dp"  
    android:layout_height="34dp"  
    android:layout_marginTop="628dp"  
    android:text="TextView"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.877"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/tumur"  
    android:layout_width="208dp"  
    android:layout_height="34dp"  
    android:layout_marginTop="584dp"
```

```
android:text="TextView"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.877"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/tnama"  
    android:layout_width="208dp"  
    android:layout_height="34dp"  
    android:layout_marginTop="536dp"  
    android:text="TextView"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.877"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView  
    android:id="@+id/textView8"  
    android:layout_width="142dp"  
    android:layout_height="32dp"  
    android:layout_marginTop="328dp"  
    android:text="Nomor Telephon"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.068"  
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/kode" />
```

```
<TextView
```

```
        android:id="@+id/textView4"
        android:layout_width="142dp"
        android:layout_height="32dp"
        android:layout_marginTop="192dp"
        android:text="Nama Pasien"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.068"
        app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/kode" />
```

```
<TextView
    android:id="@+id/textView9"
    android:layout_width="139dp"
    android:layout_height="32dp"
    android:layout_marginTop="372dp"
    android:text="Kode Perawatan"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.068"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/kode" />
```

```
<TextView
    android:id="@+id/tjenis"
    android:layout_width="157dp"
    android:layout_height="33dp"
    android:layout_marginTop="236dp"
    android:text="Jenis Perawatan"
    android:textAlignment="center"
```

```
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="1.0"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/harga"  
    android:layout_width="139dp"  
    android:layout_height="40dp"  
    android:layout_marginTop="304dp"  
    android:text="Harga"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.972"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
```

```
    android:id="@+id/textView7"  
    android:layout_width="142dp"  
    android:layout_height="32dp"  
    android:layout_marginTop="284dp"  
    android:text="Jenis Kelamin"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.061"  
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/kode" />
```

```
<TextView
```

```
    android:id="@+id/textView6"
```

```
        android:layout_width="142dp"
        android:layout_height="32dp"
        android:layout_marginTop="240dp"
        android:text="Umur Pasien"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.068"
        app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/kode" />
```

```
<TextView
```

```
    android:id="@+id/textView10"
    android:layout_width="139dp"
    android:layout_height="32dp"
    android:layout_marginTop="416dp"
    android:text="Jumlah Hari"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.078"
    app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@+id/kode" />
```

```
<EditText
```

```
    android:id="@+id/nama"
    android:layout_width="209dp"
    android:layout_height="36dp"
    android:ems="10"
    android:hint="Nama Pasien"
    android:inputType="text"
    android:textSize="14sp"
```

```
app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.093"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.137" />
```

```
<EditText
    android:id="@+id/umur"
    android:layout_width="209dp"
    android:layout_height="36dp"
    android:ems="10"
    android:hint="Umur Pasien"
    android:inputType="text"
    android:textSize="14sp"
```

```
app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.093"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.186" />
```

```
<Spinner
    android:id="@+id/jk"
    android:layout_width="217dp"
    android:layout_height="36dp"
    android:layout_marginTop="184dp"
```

```
android:entries="@array/jk"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.093"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
```

```
android:id="@+id/no"  
android:layout_width="209dp"  
android:layout_height="39dp"  
android:ems="10"  
android:hint="Nomor Telepon"  
android:inputType="text"  
android:textSize="14sp"
```

```
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.109"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.3" />
```

```
<Spinner
```

```
android:id="@+id/kode"  
android:layout_width="212dp"  
android:layout_height="36dp"  
android:layout_marginTop="308dp"  
android:entries="@array/kodep"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.096"
```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
<EditText
```

```
    android:id="@+id/jumlah"  
    android:layout_width="209dp"  
    android:layout_height="39dp"  
    android:ems="10"  
    android:hint="Jumlah Hari"  
    android:inputType="text"  
    android:textSize="14sp"
```

```
app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.104"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.486" />
```

```
<Button
```

```
    android:id="@+id/proses"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:onClick="Proses"  
    android:text="Proses"
```

```
app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.103"
```

```
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.572" />
```

```
<Button
```

```
    android:id="@+id/batal"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Batal"
```

```
app:layout_constraintBottom_toBottomOf="parent"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.421"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintVertical_bias="0.573" />
```

```
<TextView
```

```
    android:id="@+id/textView3"  
    android:layout_width="201dp"  
    android:layout_height="28dp"  
    android:layout_marginTop="40dp"  
    android:gravity="center"  
    android:text="Formulir Pasien"  
    android:textColor="@color/white"  
    android:textSize="60px"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.502"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toTopOf="parent" />
```

```
<ImageView
    android:id="@+id/foto"
    android:layout_width="97dp"
    android:layout_height="103dp"
    android:layout_marginTop="136dp"
    android:scaleType="fitXY"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.897"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"

    app:srcCompat="@drawable/hospital" />
```

```
<TextView
    android:id="@+id/total"
    android:layout_width="130dp"
    android:layout_height="59dp"
    android:layout_marginTop="364dp"
    android:text="Total"
    android:textColor="#EB2E96"
    android:textSize="40px"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.938"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

```
<TextView
    android:id="@+id/textView"
```

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="288dp"  
android:text="Kode Perawatan"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.226"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

```
android:id="@+id/textView12"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="288dp"  
android:text="Harga Perhari"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.913"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

<TextView

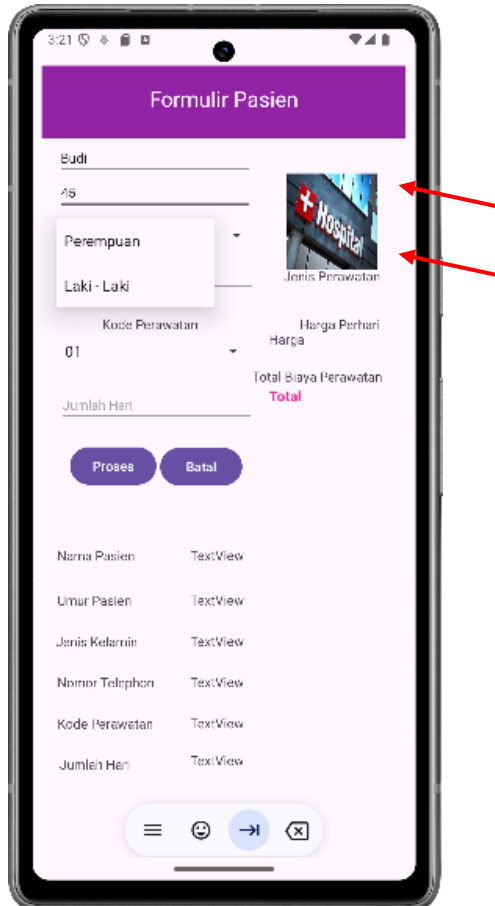
```
android:id="@+id/textView13"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="344dp"  
android:text="Total Biaya Perawatan"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.901"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent" />
```

```
</androidx.constraintlayout.widget.ConstraintLayout>  
</ScrollView>  
</HorizontalScrollView>
```

Pada koding tersebut sudah termasuk penggunaan scroll view, spinner dan elemen lainnya untuk memberikan tampilan yang menarik bagi user di bagian *user interface* UI.

Langkah 2:

Setelah itu tetapkanlah setiap id dari elemen yang di buat untuk memudahkan dalam pendeklarasiannya pada main activity.java. Setelah itu kita harus mendeklarikan array yang telah kita bangun pada spinner perhatikan gambar dimana spinner digunakan. Untuk penggunaannya sama seperti langkah pada BAB sebelumnya.



Gambar 6. 15 Contoh Penggunaan Spinner Pada Contoh Kasus

Berikut array dari spinner yang telah di deklarasikan di string.xml :

```
<string-array name="jk">
  <item>Perempuan</item>
  <item>Laki - Laki</item>
</string-array>
```

```
<string-array name="kodep">
  <item>01</item>
  <item>02</item>
  <item>03</item>
</string-array>
```

Penjelasan koding:

Array name = "jk" merupakan array yang digunakan pada spinner yaitu untuk memuat dua pilihan jenis kelamin pasien yaitu Perempuan dan Laki - Laki dan array "kodep" untuk menampung kode yang ada pada spinner yang mana kode 01, 02 dan 03 yang nantinya nilainya akan di gunakan untuk mengakses perintah logika pada main activity.java.

Langkah 3:

Setelah itu lalu pendeklarasian setiap elemen dan variabel di main activity.java kemudian menggunakan operator logika dan aritmatika pada pemrosesannya. Dan menggunakan format rupiah pada biaya untuk memudahkan *user* dalam mengakses aplikasinya.

Berikut koding pada activity.java :

```
package com.example.myprojectmobileprogramming24;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.Spinner;
```

```

import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;

import com.example.myprojectmobileprogramming24.R;

import java.text.NumberFormat;
import java.util.Locale;

public class PasienUtsa extends AppCompatActivity {
    EditText vnama, vumur, vno, vjumlah;
    Spinner vjk, vkode;
    TextView vtnama, vtumur, vtno, vtjumlah, vtjk, vtkode,
    vtharga, vttotal, vtjenis;
    Button vproses;
    ImageView img1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_pasien_utsa);

        vnama = (EditText) findViewById(R.id.nama);
        vumur = (EditText) findViewById(R.id.umur);
        vno = (EditText) findViewById(R.id.no);
        vjumlah = (EditText) findViewById(R.id.jumlah);
        vjk = (Spinner) findViewById(R.id.jk);
    }
}

```

```

vkode = (Spinner) findViewById(R.id.kode);
vtnama = (TextView) findViewById(R.id.tnama);
vtumur = (TextView) findViewById(R.id.tumur);
vtno = (TextView) findViewById(R.id.tno);
vtjenis = (TextView) findViewById(R.id.tjenis);
vtjumlah = (TextView) findViewById(R.id.tjumlah);
vtjk = (TextView) findViewById(R.id.tjk);
vtkode = (TextView) findViewById(R.id.tkode);
vtharga = (TextView) findViewById(R.id.harga);
vttotal = (TextView) findViewById(R.id.total);
img1 = (ImageView) findViewById(R.id.foto);
vproses = (Button) findViewById(R.id.proses);
}

public void Proses(View view) {
    String vkodekmr, jns;
    int hrg, jml, total;
    jml =
Integer.parseInt(vjumlah.getText().toString().trim());
    vtnama.setText(String.valueOf(vnama.getText()));
    vtumur.setText(String.valueOf(vumur.getText()));
    vtjumlah.setText(String.valueOf(vjumlah.getText()));

vtkode.setText(String.valueOf(vkode.getSelectedItem()));
vtno.setText(String.valueOf(vno.getText()));
vtjk.setText(String.valueOf(vjk.getSelectedItem()));
vkodekmr = vkode.getSelectedItem().toString();

    if (vkodekmr.equals("01")) {
        jns = "Rawat Inap";
    }
}

```

```

        hrg = 100000;
        img1.setImageResource(R.drawable.foto1);
    } else if (vkodekmr.equals("02")) {
        jns = "Rawat Jalan";
        hrg = 200000;
        img1.setImageResource(R.drawable.foto3);
    } else {
        jns = "Rawat Jalan";
        hrg = 300000;
        img1.setImageResource(R.drawable.kamarf2);
    }

    total = jml * hrg;
    vtjenis.setText(jns);
    vtharga.setText(rupiah(hrg));
    vttotal.setText(rupiah(total));
    Toast.makeText(getApplicationContext(),
        "Terima Kasih, Semoga lekas Sehat " +
vtnama.getText(),
        Toast.LENGTH_LONG).show();
    }

    private String rupiah(int number) {
        NumberFormat formatRupiah =
NumberFormat.getCurrencyInstance(new Locale("in", "ID"));
        return formatRupiah.format(number);
    }
}

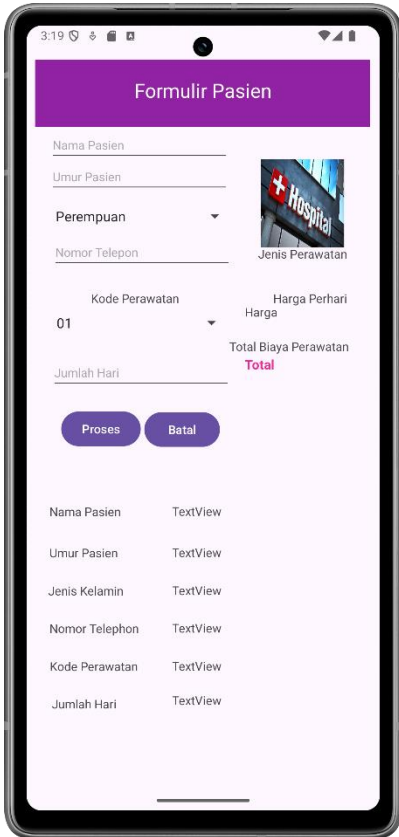
```

Penjelasan koding:

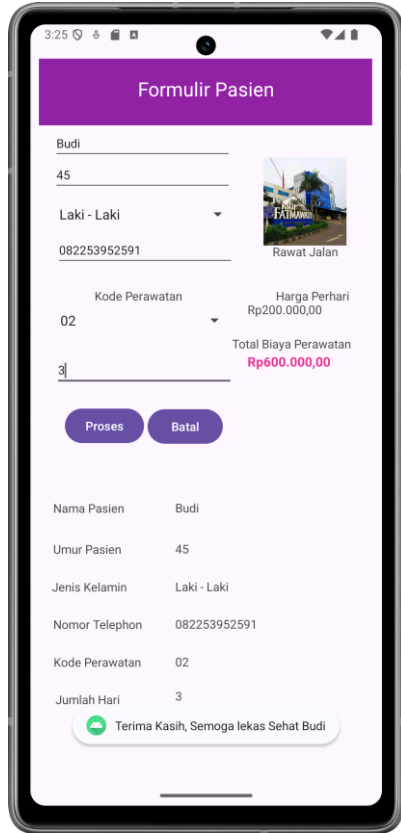
- a) EditText (vnama, vumur, vno, vjumlah) Komponen input teks yang memungkinkan pengguna untuk memasukkan informasi. Nilai yang dimasukkan pengguna pada komponen ini akan diproses lebih lanjut untuk menampilkan informasi pasien dan menghitung biaya.
- b) TextView (vtnama, vtumur, vtno, vtjumlah, vtjk, vtkode, vtharga, vttotal, vtjenis) Digunakan untuk menampilkan hasil input dari pengguna (seperti nama, umur, nomor pasien, jumlah hari) dan hasil perhitungan (jenis kamar, harga per hari, total biaya).
- c) Spinner (vjk, vkode) Komponen dropdown untuk memilih opsi. vjk digunakan untuk memilih jenis kelamin (misalnya, Laki-Laki atau Perempuan). vkode digunakan untuk memilih kode kamar perawatan (misalnya, "01" untuk Rawat Inap atau "02" untuk Rawat Jalan).
- d) Button (vproses) Tombol yang digunakan untuk memproses data yang diinputkan oleh pengguna.
- e) ImageView (img1) Digunakan untuk menampilkan gambar terkait dengan jenis kamar yang dipilih. Gambar ditentukan berdasarkan kode kamar yang dipilih melalui vkode.
- f) int (hrg, jml, total) dimana hrg: Variabel untuk menyimpan harga per hari berdasarkan jenis kamar. jml: Variabel untuk menyimpan jumlah hari perawatan yang diinputkan oleh pengguna melalui vjumlah. total: Variabel untuk menyimpan hasil perkalian antara hrg dan jml, yaitu total biaya perawatan.
- g) findViewById() Digunakan untuk menghubungkan variabel Java dengan elemen UI di file layout XML.

Misalnya, `vnama = (EditText) findViewById(R.id.nama);` menghubungkan elemen `EditText` bernama `nama` di layout XML dengan variabel `vnama` di kode Java.

- h) `Vkodekmr` Variabel lokal yang menyimpan kode kamar yang dipilih melalui `vkode`. Berdasarkan kode ini, jenis kamar (`jns`) dan harga per hari (`hrg`) ditentukan menggunakan pernyataan `if-else`.
- i) `NumberFormat.getCurrencyInstance(new Locale("in", "ID"))` Digunakan untuk memformat angka menjadi format mata uang Indonesia (Rupiah). `Locale ("in", "ID")` memastikan format menggunakan simbol `Rp` dan format angka yang sesuai dengan standar Indonesia.
- j) `rupiah(int number)` Metode ini memanfaatkan `NumberFormat` untuk mengubah nilai numerik seperti harga dan total biaya menjadi format mata uang Rupiah. Misalnya, `rupiah(100000)` akan menghasilkan `Rp100.000`.
- k) `setText()` Digunakan untuk menampilkan teks ke dalam komponen `TextView` di layar.
- l) `Toast` Digunakan untuk menampilkan pesan singkat kepada pengguna setelah proses selesai.



Gambar 6. 16 Hasil Running Latihan Kasus 4



Gambar 6. 17 Hasil Running Setelah Diproses

BAB VII

INTENT

1. Pengertian Intent

Inten merupakan objek yang menyediakan waktu proses untuk mengikat komponen komponen yang terpisah, biasanya untuk berpindah ke suatu activity. Intent berfungsi sebagai penghubung yang memungkinkan perpindahan halaman antara activity dalam sebuah aplikasi.

Dengan menggunakan intent, suatu activity dapat berpindah ke activity lain, baik dengan membawa data atau tanpa data. Bahkan, intent juga memungkinkan aplikasi untuk membuka aplikasi lain seperti Gmail, Google Maps, dan sebagainya.

Secara umum intent terdiri dari:

- a) Implicit Intent untuk memanggil fungsi yang sudah tersedia pada fungsi internal android seperti Kamera, Dial Number, Open Browser dan lainnya.
- b) Explicit Intent untuk memanggil activity yang berada dalam suatu project ataupun berbeda, seperti berpindah dari activity yang satu ke activity lainnya.

Pemanggilan Intent secara eksplisit dilakukan dengan menetapkan secara langsung nama Activity yang ingin dituju. Intent memiliki dua parameter utama: yang pertama adalah

Context, yang merepresentasikan tempat di mana activity tersebut sedang aktif, dan yang kedua adalah *destination*, yaitu activity tujuan yang akan dipanggil saat pengguna melakukan suatu tindakan, seperti menekan tombol.

Berikut contoh deklarasi dari Explicit intent:

```
Intent tentang = new Intent(mintent.this, latprofil2.class);  
startActivity(tentang);
```

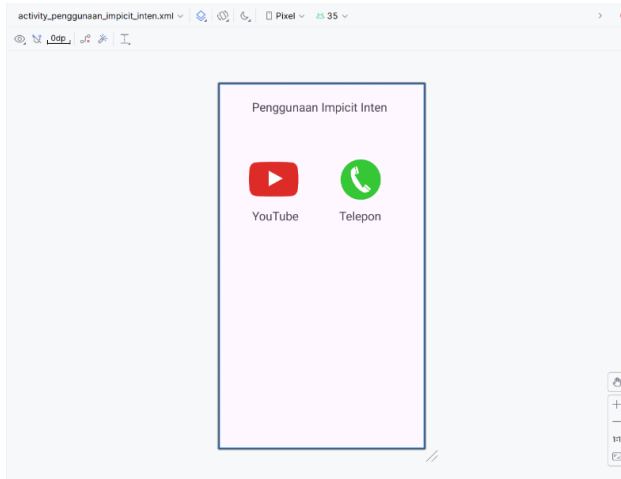
Penjelasan: *mintent.this*, merupakan activity saat ini dan *latprofil2.class* adalah activity tujuannya.

2. Contoh Penggunaan Impicit Intent

Pada contoh kali ini yaitu untuk mendial atau ingin menelfon sebuah nomor telepon menggunakan tombol kemudian tombol lainnya untuk membuka website YouTube.

Langkah 1:

Buatlah sebuah layout yang terdiri dari dua tombol untuk mendial nomor telepon dan membuka website YouTube yang masing - masing sudah dilakukan pengaktifan fitur `onClick`. Sebagai berikut:



Gambar 7. 1 Langkah 1 Penggunaan Impicit Inten

Langkah 2:

Setelah dilakukan pengaktifan fitur onClik pada tombol maka akan melakukan pendeklarasian dari Impicit intent yaitu sebagai berikut:

```
public void panggil(View view) { 1 usage
    String nomor = "082215678890" ;
    Intent panggil = new Intent(Intent.ACTION_DIAL);
    panggil.setData(Uri.fromParts( scheme: "tel", nomor, fragment: null));
    startActivity(panggil);
}
```

Gambar 7. 2 Contoh Deklarasi Impicit Inten

ACTION_DIAL merupakan fungsi yang telah disediakan oleh sistem, intent panggil untuk mengarahkan kepada pendialan telepon berdasarkan nomor yang terdapat pada string. Dan untuk wesite menggunakan ACTION_VIEW digunakan untuk mengakses sebuah wesite atau link url. startActivity untuk menjalankan perintah pada deklarasi intent.

Deklarasi untuk url:

```
public void buka(View view) { 1 usage
    String url = "https://youtube.com" ;
    Intent bukabrowser = new
        Intent(Intent.ACTION_VIEW);
    bukabrowser.setData(Uri.parse(url));
    startActivity(bukabrowser);
}
```

Gambar 7. 3 Deklarasi Impiclit Inten Untuk Url

Berikut koding lengkapnya:

Koding layout.xml:

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

    <ImageButton
        android:id="@+id/imageButton"
        android:layout_width="147dp"
        android:layout_height="121dp"
        android:backgroundTint="@android:color/transparent"
        android:drawingCacheQuality="high"
        android:scaleType="fitCenter"
```

```
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.136"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.209"  
android:onClick="YouTube"  
app:srcCompat="@drawable/logo_youtube" />
```

<ImageButton

```
android:id="@+id/imageButton20"  
android:layout_width="168dp"  
android:layout_height="100dp"  
android:backgroundTint="@android:color/transparent"  
android:drawingCacheQuality="high"  
android:scaleType="fitCenter"  
android:onClick="panggil"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.823"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.225"  
app:srcCompat="@drawable/logo_telepon" />
```

<TextView

```
android:id="@+id/textView5"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Penggunaan Implicit Inten"
```

```
android:textSize="24sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.47"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.043" />
```

<TextView

```
android:id="@+id/textView31"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="YouTube"  
android:textSize="24sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.208"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.356" />
```

<TextView

```
android:id="@+id/textView32"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="Telepon"  
android:textSize="24sp"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.738"
```

```
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.356" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Koding activity.java:

```
package
com.example.panduansuksesandroidstudiolangkah_langka
hmembangunaplikasidengancruddatabase;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;

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 Penggunaan_Implicit_Inten extends
AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);

        setContentView(R.layout.activity_penggunaan_implicit_inten);
```

```

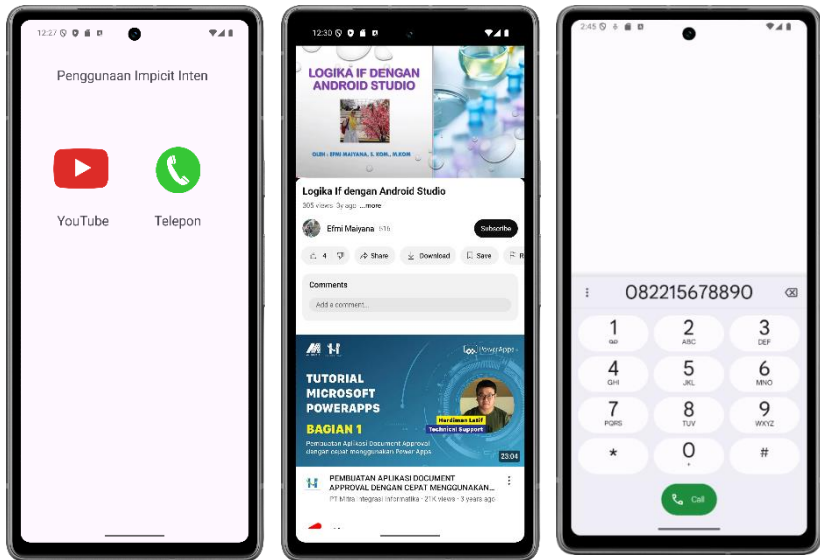
ViewCompat.setOnApplyWindowInsetsListener(findViewById
(R.id.main), (v, insets) -> {
    Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
    return insets;
});
}

public void panggil(View view) {
    String nomor = "082215678890" ;
    Intent panggil = new Intent(Intent.ACTION_DIAL);
    panggil.setData(Uri.fromParts("tel", nomor, null));
    startActivity(panggil);
}

public void YouTube(View view) {
    String url = "https://youtube.com" ;
    Intent bukabrowser = new
        Intent(Intent.ACTION_VIEW);
    bukabrowser.setData(Uri.parse(url));
    startActivity(bukabrowser);
}
}

```

Hasil Output:



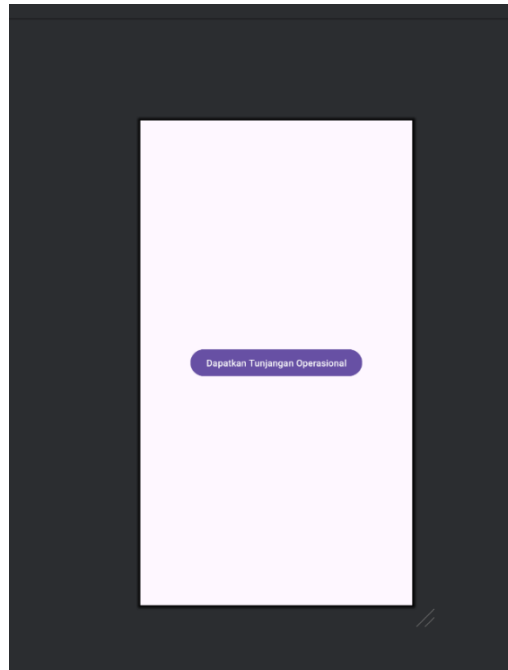
Gambar 7. 4 Hasil Running Penggunaan Impiclit Inten

3. Contoh Penggunaan Explicit Intent

Pada contoh kali ini yaitu untuk mengarahkan menu activity awal ke menu activity pada kasus 3 yaitu untuk menentukan fasilitas dan dana operasional bantuan berdasarkan kode.

Langkah 1:

Membuat sebuah halaman layout awal misalnya satu buah tombol dan fitur onClick nya sudah di aktifkan.



Gambar 7.5 Langkah 1 Penggunaan Explicit Inten

Langkah 2:

Setelah dilakukan pengaktifan fitur onClik pada tombol maka akan melakukan pendeklarasian dari Impicit intent yaitu sebagai berikut:

```
public void dapatkan(View view) { 1 usage
    Intent tentang = new Intent( packageContext: MainActivity3.this, MainActivity2.class);
    startActivity(tentang);
}
```

Activity saat ini

Activity yang dituju

Gambar 7.6 Contoh Deklarasi Explicit Inten

Koding keseluruhan:

Koding layout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="50dp">

    <Button
        android:id="@+id/pusat"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="dapatkan"
        android:text="Dapatkan Tunjangan Operasional"
        android:layout_centerInParent="true" />

</RelativeLayout>
```

Koding activity.java:

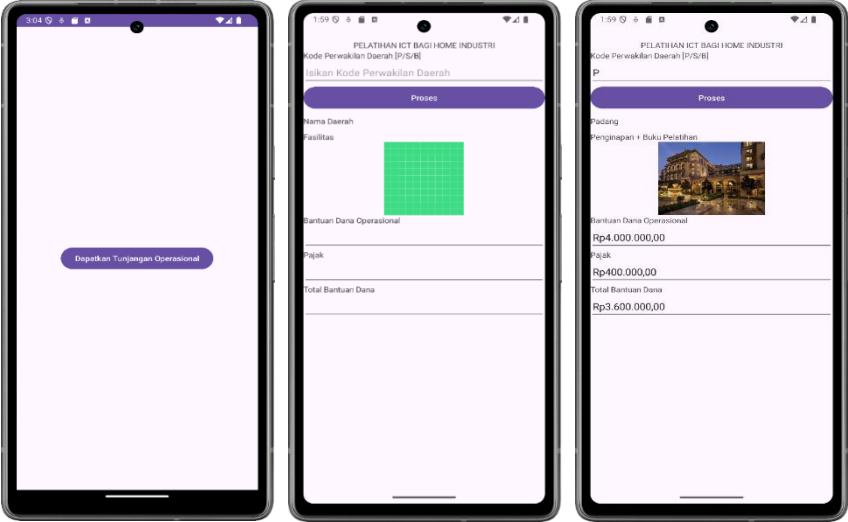
```
package com.example.myprojectmobileprogramming24;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.view.WindowCompat;
```

```
public class MainActivity3 extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        WindowCompat.setDecorFitsSystemWindows(getWindow(),  
false);  
        setContentView(R.layout.relativelayoutpenerapan1);  
    }  
  
    public void dapatkan(View view) {  
        Intent tentang = new Intent(MainActivity3.this,  
MainActivity2.class);  
        startActivity(tentang);  
    }  
}
```

Hasil output:



Gambar 7. 7 Hasil Running Explicit Intent

BAB VIII

CRUD dengan Json dan Database MySQL

1. Database Dengan JSON

Database atau Basis data adalah sekelompok data yang terhubung satu sama lain dan diorganisir berdasarkan skema atau struktur tertentu. Data tersebut disimpan di perangkat keras komputer dan dikelola menggunakan perangkat lunak. Melalui perangkat lunak ini data bisa dimanipulasi seperti diperbaharui, dicari, diolah dengan berbagai perhitungan atau dihapus sesuai dengan tujuan yang hendak dicapai.

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer.

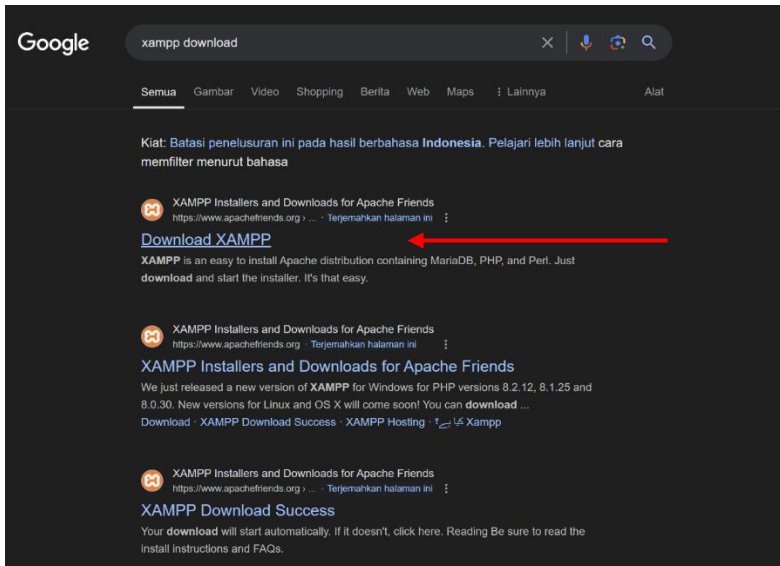
Jadi untuk bisa menghubungkan database dengan android salah satunya dapat menggunakan JSON. Dimana data yang dimanipulasi itu berasal dari database localhost baik menggunakan XAMPP atau Laragon sebagai server penyedia database baik menggunakan MySQL atau lainnya.

2. Instalasi XAMPP Sebagai Database

Adapun cara untuk melakukan instalasi XAMPP sebagai database sebagai berikut:

Langkah 1:

Cari XAMPP download pada browser, kemudian klik Download XAMPP



Gambar 7. 8 Instalasi XAMPP

Langkah 2:

Kemudian pilih XAMPP sesuai sistem operasi yang dimiliki. Dan pilih versi terbaru kemudian klik download.

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy. Installers created using InstallBuilder.

XAMPP for Windows 8.0.30, 8.1.25 & 8.2.12

Version	Checksum	Size
8.0.30 / PHP 8.0.30	What's Included? md5 sha1	Download (61.5k) 144 Mb
8.1.25 / PHP 8.1.25	What's Included? md5 sha1	Download (61.5k) 148 Mb
8.2.12 / PHP 8.2.12	What's Included? md5 sha1	Download (61.5k) 149 Mb

Requirements More Downloads >

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for those platforms [here](#).

Documentation/FAQs

There is no real manual or handbook for XAMPP. We write the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs

XAMPP for Browsers & 8.3.12

Version	Checksum	Size
8.3.12 / PHP 8.3.12	What's Included? md5 sha1	Run in the browser -

Requirements More Downloads >

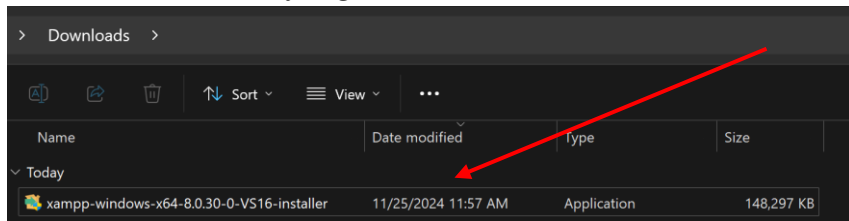
XAMPP for Linux 8.0.30, 8.1.25 & 8.2.12

Version	Checksum	Size
8.0.30 / PHP 8.0.30	What's Included? md5 sha1	Download (61.5k) 151 Mb
8.1.25 / PHP 8.1.25	What's Included? md5 sha1	Download (61.5k) 153 Mb

Gambar 7. 9 Langkah 2 Instalasi XAMPP

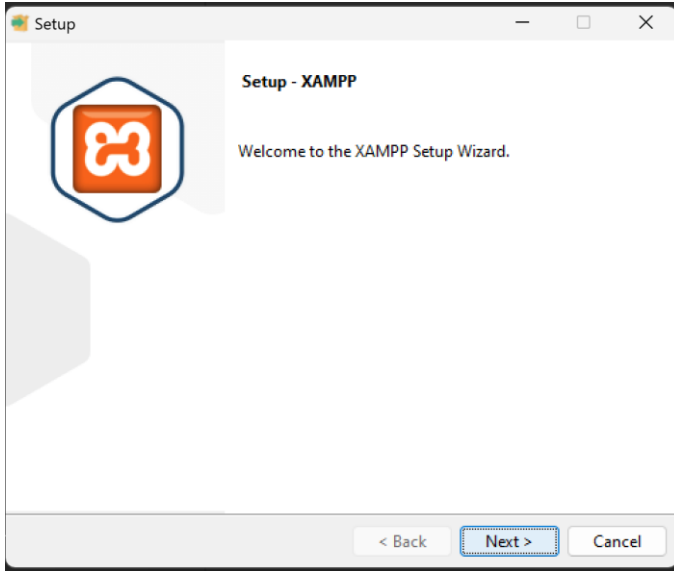
Langkah 3:

Kemudian buka file yang telah berhasil di download.



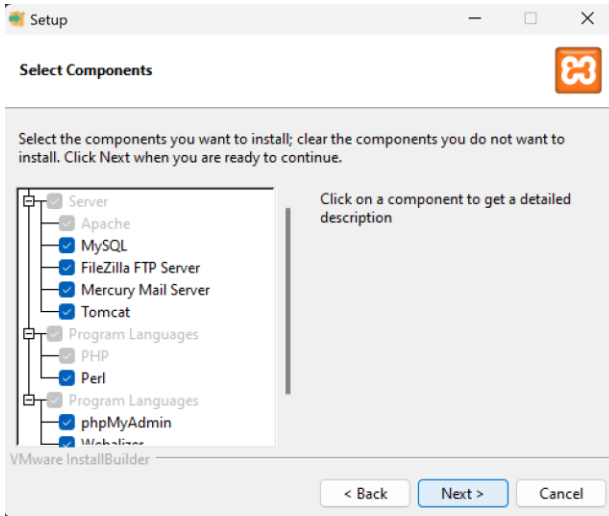
Gambar 7. 10 Langkah 3 Instalasi XAMPP

Langkah 4:
Klik next.



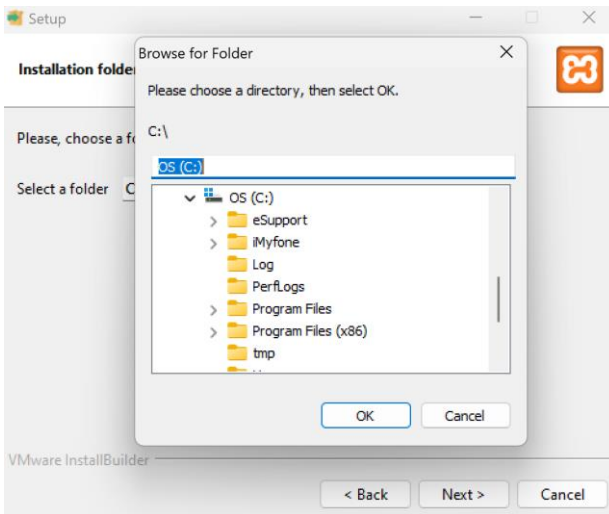
Gambar 7. 11 Langkah 4 Instalasi XAMPP

Kemudian centang pada bagian seperti gambar berikut kemudian klik next.



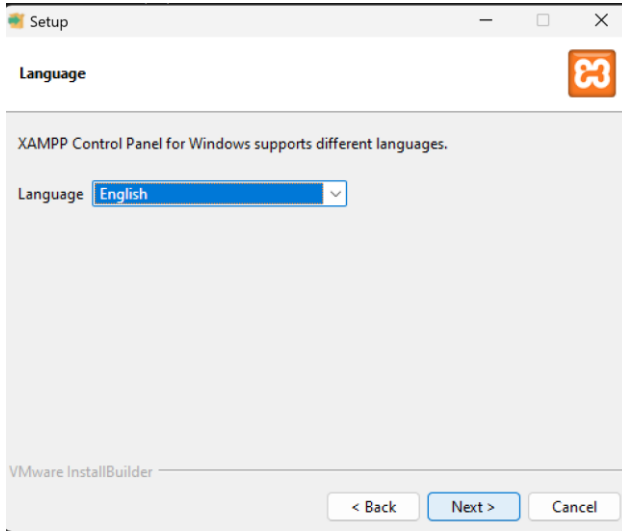
Gambar 7. 12 Langkah 4.1 Instalasi XAMPP

Kemudian simpan file XAMPP di folder yang di inginkan kemudian klik next.



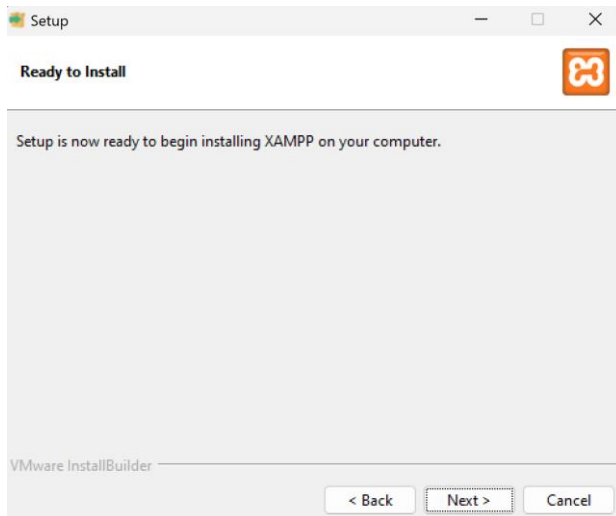
Gambar 7. 13 Langkah 4.2 Instalasi XAMPP

Kemudian klik pilih bahasa yang diinginkan kemudian klik next.



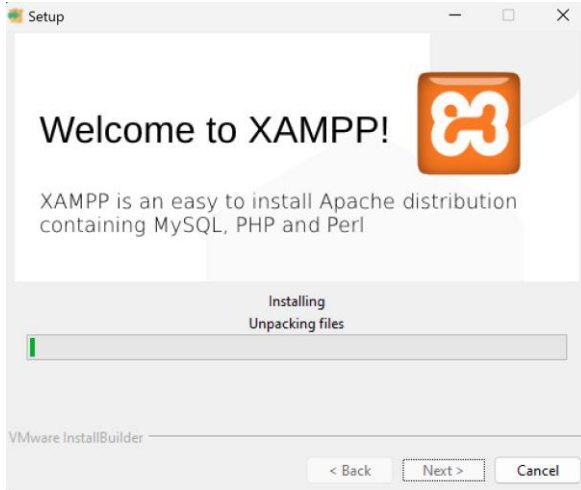
Gambar 7. 14 Langkah 4.3 Instalasi XAMPP

Kemudian klik next

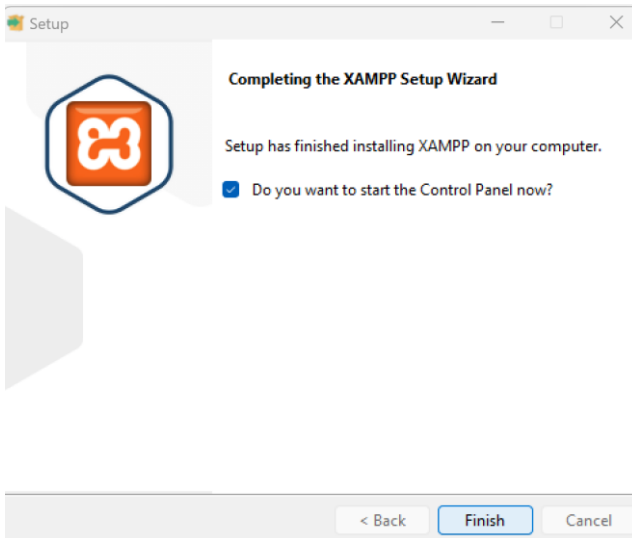


Gambar 7. 15 Langkah 4.4 Instalasi XAMPP

Kemudian tunggu proses instalasi. Kemudian klik finish setelah itu XAMPP berhasil untuk di instal dan siap digunakan.



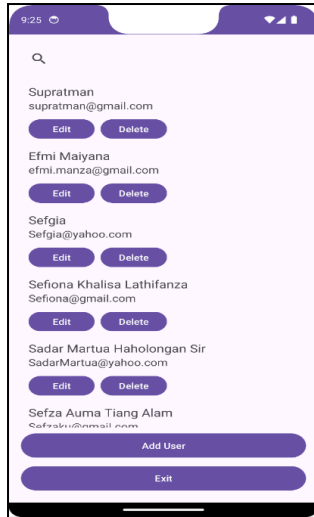
Gambar 7. 16 Langkah 4.5 Instalasi XAMPP



Gambar 7. 17 Langkah 4.6 Instalasi XAMPP

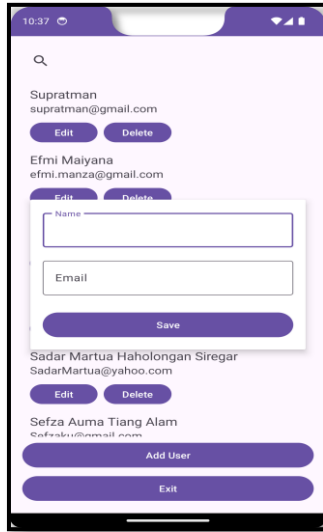
3. Create, Read, Update dan Delete Android Studio

Tampilan saat awal Aplikasi dijalankan adalah sebagai berikut :



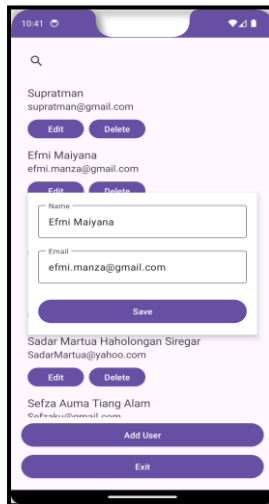
Gambar 7. 18 Tampilan Awal

Saat diklik tombol **Add User**, maka akan tampil gambar berikut :



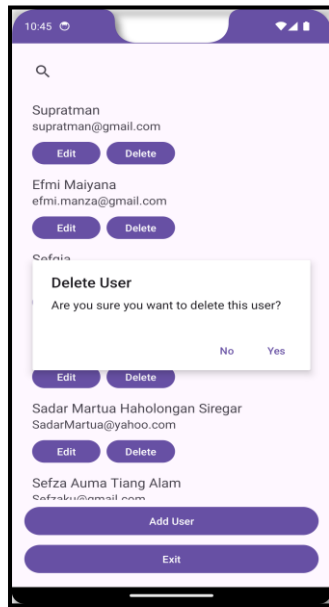
Gambar 7. 19 Tampilan Add User

Saat diklik Tombol Edit tampilan seperti gambar dibawah ini



Gambar 7. 20 Edit Data User

Saat diklik Tombol Hapus tampilan seperti gambar dibawah ini



Gambar 7. 21 Delete Data User

Aplikasi Android untuk CRUD (*Create, Read, Update, Delete*) menggunakan bahasa pemrograman Java dengan Android Studio dan database MySQL sebagai backend. Proyek ini juga akan menggunakan PHP sebagai API untuk berkomunikasi antara aplikasi Android dan database MySQL.

Langkah-Langkah dalam membangunnya :

A. Persiapan Database MySQL

1. Buat database di MySQL dengan nama **db_crud_android** dan tabel **users** dengan struktur berikut:

```
CREATE DATABASE db_crud_android;  
USE db_crud_android;
```

```
CREATE TABLE users (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  name VARCHAR(100),  
  email VARCHAR(100)  
);
```



Gambar 7. 22 Database CRUD

B. Membuat API PHP untuk CRUD

Buat folder dengan nama pada **C:/xampp/htdocs/android_crud** di server atau localhost, lalu tambahkan file berikut di dalamnya:

a. db_config.php

```
<?php  
$servername = "localhost";  
$username = "root";  
$password = "";  
$dbname = "db_crud_android";  
$conn = new mysqli($servername, $username, $password,  
$dbname);
```

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

b. create_user.php

```
<?php
include 'db_config.php';
$name = $_POST['name'];
$email = $_POST['email'];
$sql = "INSERT INTO users (name, email) VALUES
($name', '$email)";
if ($conn->query($sql) === TRUE) {
    echo json_encode(array("status" => "success", "message"
=> "User added successfully"));
} else {
    echo json_encode(array("status" => "error", "message" =>
"Failed to add user"));
}
$conn->close();
?>
```

c. read_user.php

```
<?php
include 'db_config.php';
$sql = "SELECT * FROM users";
$result = $conn->query($sql);
$users = array();
if ($result->num_rows > 0) {
    while($row = $result->fetch_assoc()) {
```

```
        array_push($users, $row);
    }
}
echo json_encode($users);
$conn->close();
?>
```

d. update_user.php

```
<?php
include 'db_config.php';
$id = $_POST['id'];
$name = $_POST['name'];
$email = $_POST['email'];
$sql = "UPDATE users SET name='$name', email='$email'
WHERE id=$id";
if ($conn->query($sql) === TRUE) {
    echo json_encode(array("status" => "success", "message"
=> "User updated successfully"));
} else {
    echo json_encode(array("status" => "error", "message" =>
"Failed to update user"));
}
$conn->close();
?>
```

e. delete_user.php

```
<?php
include 'db_config.php';
$id = $_POST['id'];
$sql = "DELETE FROM users WHERE id=$id";
```

```

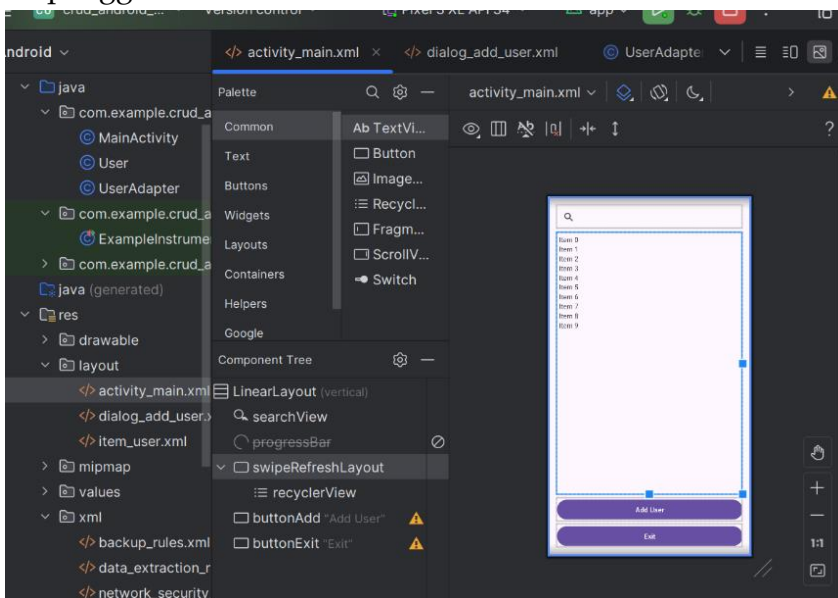
if ($conn->query($sql) === TRUE) {
    echo json_encode(array("status" => "success", "message"
=> "User deleted successfully"));
} else {
    echo json_encode(array("status" => "error", "message" =>
"Failed to delete user"));
}
$conn->close();
?>

```

C. Buka Android Studio dan buat project baru

Membuat Layout untuk Aplikasi

- a. Buat layout utama **activity_main.xml** untuk daftar pengguna



Gambar 7. 23 Contoh Layout Read (Tampil Data)

Berikut coding .xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:padding="16dp"
tools:context=".MainActivity">

<!-- SearchView untuk Pencarian Data Pengguna -->
<androidx.appcompat.widget.SearchView
    android:id="@+id/searchView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:queryHint="Search users"
    android:layout_marginBottom="8dp" />
<!-- ProgressBar untuk Indikator Loading -->
<ProgressBar
    android:id="@+id/progressBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:visibility="gone"
    android:layout_marginTop="2dp"
    android:layout_marginBottom="2dp" />
<!-- SwipeRefreshLayout untuk Pull-to-Refresh -->
```

```
<androidx.swiperefreshlayout.widget.SwipeRefreshLayout
    android:id="@+id/swipeRefreshLayout"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1">
```

```
<!-- RecyclerView untuk Menampilkan Daftar
Pengguna -->
```

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="4dp"
    android:clipToPadding="false" />
```

```
</androidx.swiperefreshlayout.widget.SwipeRefreshLayout>
```

```
<!-- Tombol untuk Menambahkan Pengguna Baru -->
```

```
<Button
    android:id="@+id/buttonAdd"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Add User"
    android:layout_marginTop="6dp" />
```

```
<Button
    android:id="@+id/buttonExit"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
android:text="Exit"
android:onClick="exit"
android:layout_marginTop="6dp" />
</LinearLayout>
```

Penjelasan:

1. **SearchView**: Untuk memungkinkan pengguna mencari data berdasarkan nama atau email.
2. **ProgressBar**: Ditampilkan saat data sedang dimuat dari server.
3. **SwipeRefreshLayout**: Membungkus RecyclerView untuk memungkinkan refresh dengan gesture swipe-down.
4. **RecyclerView**: Untuk menampilkan daftar pengguna.
5. **Button (buttonAdd)**: Untuk membuka dialog menambah pengguna baru.

Tambahkan dependency RecyclerView di file **build.gradle (Module: app)**. Tambahkan baris berikut di dalam blok dependencies:

```
implementation ('androidx.recyclerview:recyclerview:1.2.1')
```

Atau

```
implementation(libs.recyclerview)
```

Tambahkan Dependency SwipeRefreshLayout di build.gradle: SwipeRefreshLayout adalah bagian dari AndroidX, jadi Kita perlu memastikan bahwa dependency sudah ada di file build.gradle (Module: app). Tambahkan baris berikut di dalam blok dependencies:

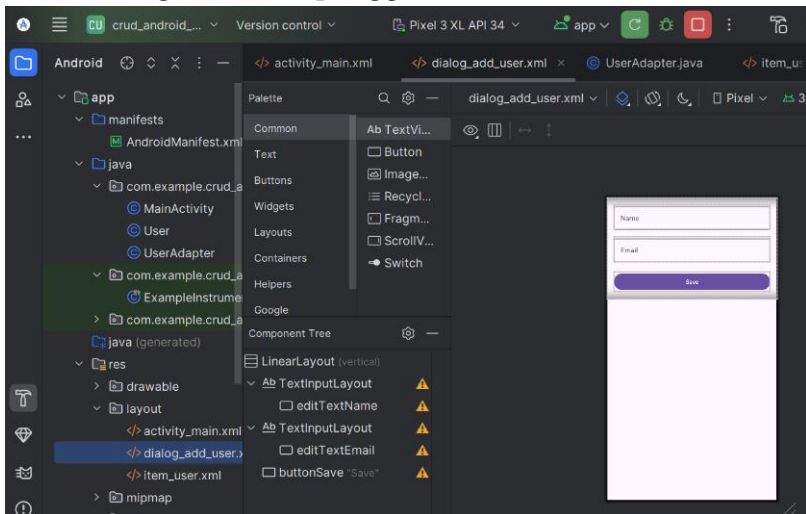
Implementation ('androidx.swiperefreshlayout:swiperefreshlayout:1.1.0')

Atau

implementation(libs.swiperefreshlayout)

Setelah menambahkan baris ini, lakukan **Sync Now** untuk sinkronisasi Gradle agar dependency terpasang.

- b. layout **dialog_add_user.xml** untuk dialog menambahkan atau mengubah data pengguna.



Gambar 7. 24 Contoh Layout Creat (Tambah)

Berikut coding .xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:orientation="vertical"
android:padding="16dp">

<!-- Input Field untuk Nama Pengguna -->
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Name">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPersonName" />

</com.google.android.material.textfield.TextInputLayout>

<!-- Input Field untuk Email Pengguna -->
<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Email"
    android:layout_marginTop="16dp">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress" />
```

```
</com.google.android.material.textfield.TextInputLayout>

<!-- Tombol Simpan untuk Menyimpan Data Pengguna -->
<Button
    android:id="@+id/buttonSave"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Save"
    android:layout_marginTop="24dp" />
</LinearLayout>
```

Menyempurnakan UI Menggunakan Material Design Components (Opsional)

Material Design menyediakan komponen UI yang modern dan lebih interaktif. Kita dapat menggunakan komponen seperti CardView, MaterialButton, atau TextInputLayout agar aplikasi lebih menarik secara visual dan mengikuti pedoman UI terkini.

Penjelasan Elemen Layout:

1. **TextInputLayout dan TextInputEditText:** Digunakan untuk menginput nama (editTextName) dan email (editTextEmail). Komponen TextInputLayout dari Material Design memberikan tampilan yang lebih baik dengan label yang melayang.
2. **Button (buttonSave):** Tombol untuk menyimpan data pengguna, baik saat menambah atau memperbarui data.

Catatan:

- Pastikan telah menambahkan dependency Material Components untuk menggunakan TextInputLayout dan TextInputEditText. Jika belum, tambahkan ini di build.gradle:

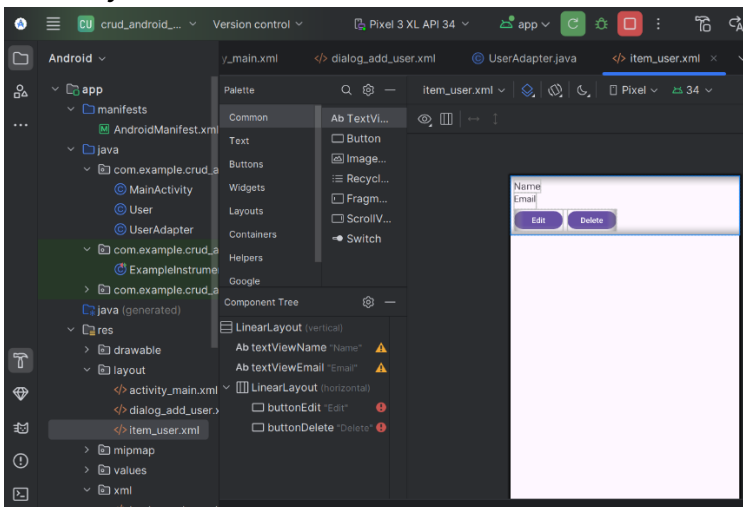
```
implementation  
('com.google.android.material:material:1.6.0')
```

Atau

```
implementation(libs.material.v160)
```

Dengan layout dialog_add_user.xml ini, Kita dapat membuka dialog untuk memasukkan atau memperbarui data pengguna pada aplikasi CRUD

c. Buat Layout item_user.xml



Gambar 7. 25 Contoh Layout Item User

Berikut coding .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="wrap_content"
    android:orientation="vertical"
    android:padding="6dp">

    <TextView
        android:id="@+id/textViewName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/textViewEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email"
        android:textSize="16sp"
        android:paddingBottom="8dp"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <Button
```

```
        android:id="@+id/buttonEdit"
        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:text="Edit"/>

<Button
    android:id="@+id/buttonDelete"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:text="Delete"
    android:layout_marginLeft="8dp"/>
</LinearLayout>
</LinearLayout>
```

D. Buat Class User dan Class Adapter

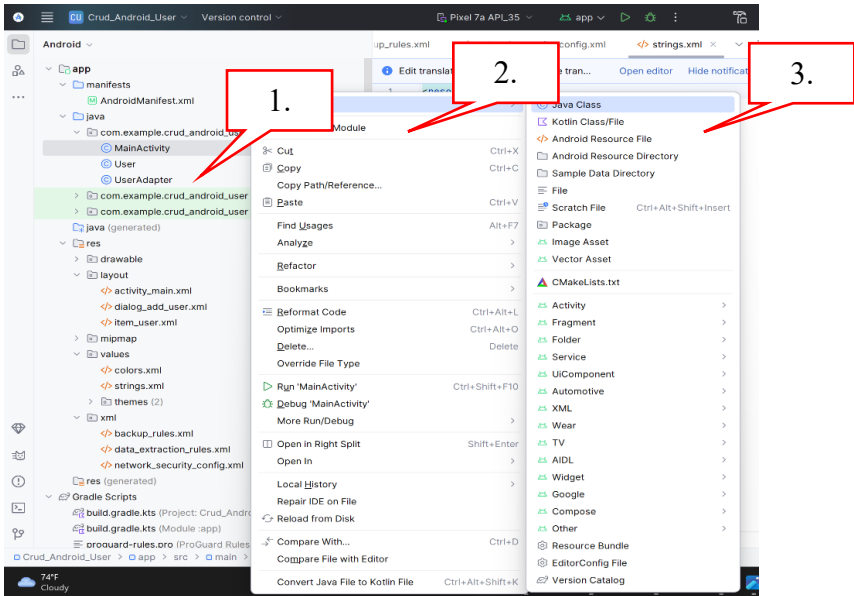
Adapun langkah - langkahnya sebagai berikut:

1. Buat package class **User**.
2. Buat package class **UserAdapter**.

Untuk membuatnya yaitu menggunakan class pada folder java, berikut caranya:

Langkah 1:

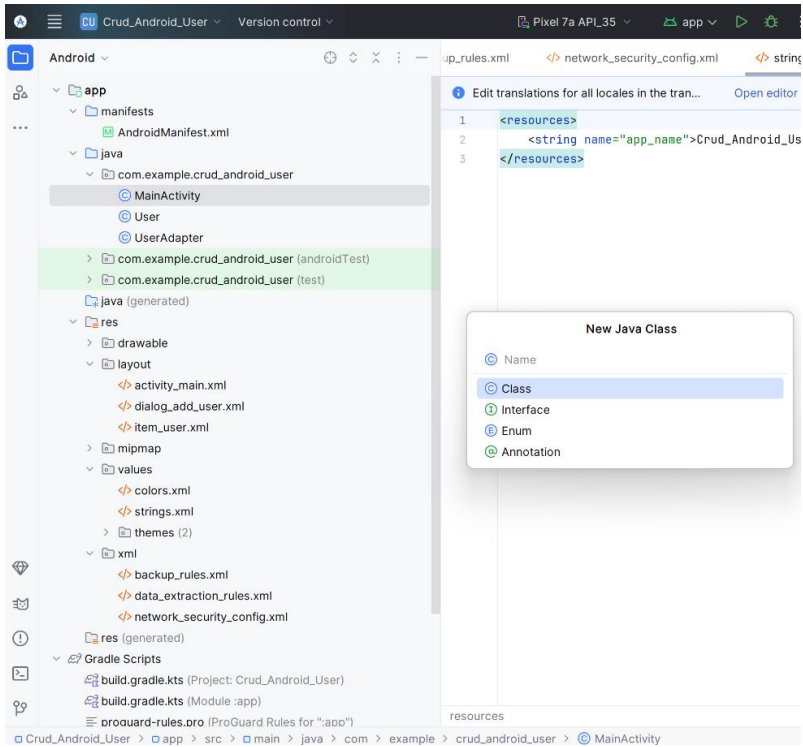
Pada folder java atau sejajar dengan MainActivity klik kanan lalu pilih new kemudian klik java class



Gambar 7. 26 Langkah 1 Pembuatan Class

Langkah 2:

Kemudian berikan nama dari class yang ingin dibuat.



Gambar 7. 27 Langkah 2 Pembuatan Class

Kemudian buatlah package model berikut:

a. Model User.java

```

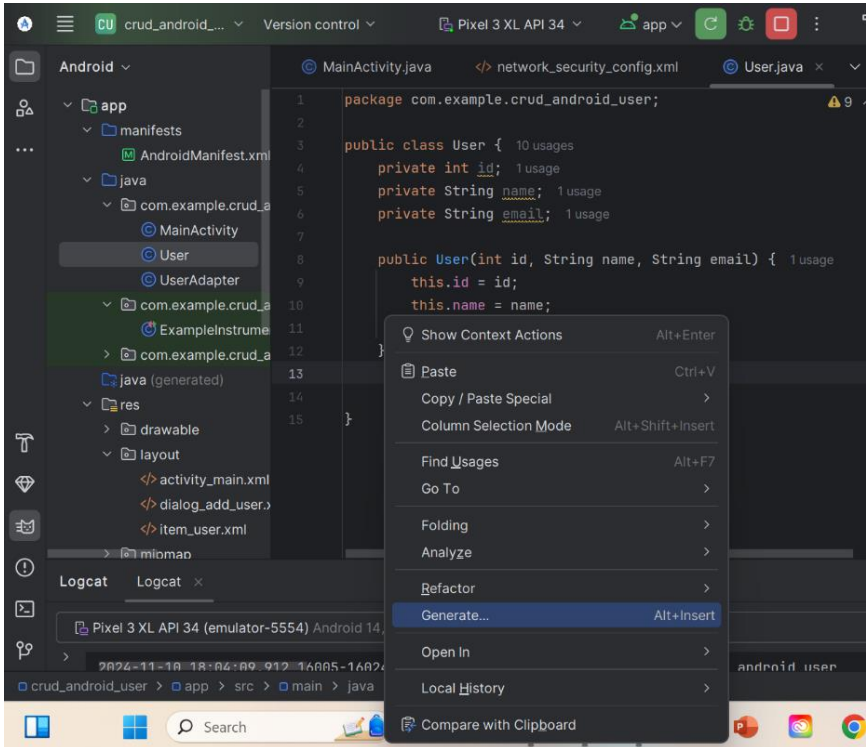
public class User {
    private int id;
    private String name;
    private String email;

    public User(int id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }
}

```

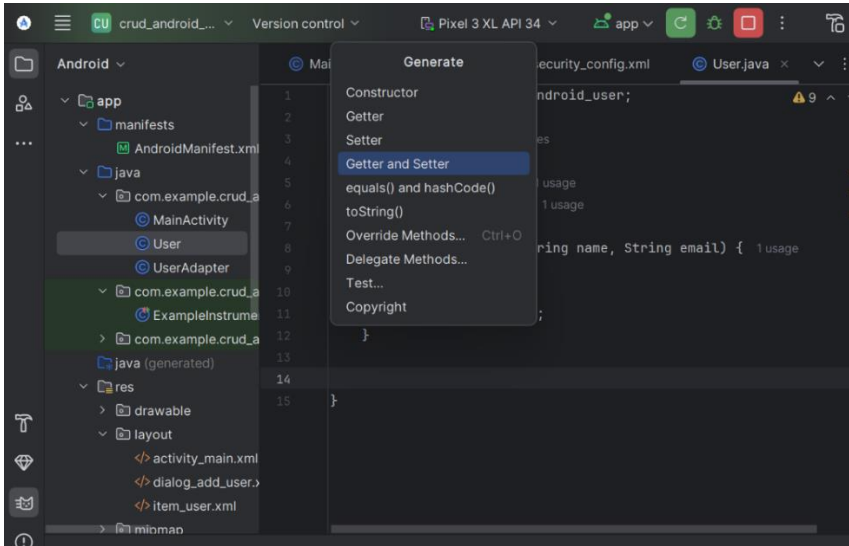
```
}  
  
//Getter dan Setter  
  
}
```

Pada //Getter dan Setter, lakukan klik kanan dekat kata-kata Getter dan Setter, maka tampil sebagai berikut:



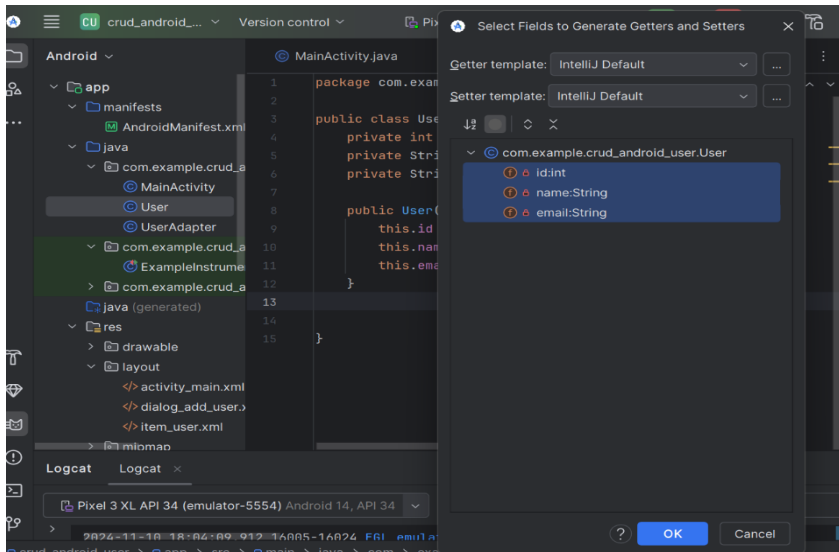
Gambar 7. 28 Langkah 1 Pembuatan Package Model User

Lalu Pilih Generate - Getter dan Setter



Gambar 7. 29 Langkah 2 Pembuatan Package Model User

Maka akan tampil sebagai berikut:



Gambar 7. 30 Langkah 3 Pembuatan Package Model User

Maka Koding pada **User.Java** sebagai berikut :

```
package com.example.crud_android_user;
public class User {
    private int id;
    private String name;
    private String email;

    public User(int id, String name, String email) {
        this.id = id;
        this.name = name;
        this.email = email;
    }

    public int getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

    public String getEmail() {
```

```

    return email;
}

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

```

b. Adapter UserAdapter.java

Buat adapter UserAdapter untuk RecyclerView guna menampilkan daftar data pengguna.

```

package com.example.crud_android_user;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import java.util.ArrayList;
import java.util.List;

public class UserAdapter extends
RecyclerView.Adapter<UserAdapter.UserViewHolder> {

    private List<User> userList;
    private UserActionListener userActionListener;

    // Constructor untuk UserAdapter

```

```

public UserAdapter(List<User> userList,
UserActionListener userActionListener) {
    this.userList = userList;
    this.userActionListener = userActionListener;
}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull
ViewGroup parent, int viewType) {
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.ite
m_user, parent, false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull
ViewHolder holder, int position) {
    User user = userList.get(position);
    holder.textViewName.setText(user.getName());
    holder.textViewEmail.setText(user.getEmail());

    // Event listener untuk button edit
    holder.buttonEdit.setOnClickListener(v -> {
        if (userActionListener != null) {
            userActionListener.showUserDialog(user);
        }
    });
}

```

```

// Event listener untuk button delete
holder.buttonDelete.setOnClickListener(v -> {
    if (userActionListener != null) {
        userActionListener.deleteUser(user.getId());
    }
});
}

@Override
public int getItemCount() {
    return userList.size();
}

// Update list pengguna dan perbarui tampilan adapter
public void updateList(List<User> newList) {
    userList = new ArrayList<>(newList);
    notifyDataSetChanged();
}

// ViewHolder untuk menampung referensi dari view
public static class UserViewHolder extends
RecyclerView.ViewHolder {
    TextView textViewName, textViewEmail;
    Button buttonEdit, buttonDelete;

    public UserViewHolder(@NonNull View itemView) {
        super(itemView);
        textViewName =
itemView.findViewById(R.id.textViewName);

```

```

        textViewEmail =
itemView.findViewById(R.id.textViewEmail);
        buttonEdit =
itemView.findViewById(R.id.buttonEdit);
        buttonDelete =
itemView.findViewById(R.id.buttonDelete);
    }
}

// Interface untuk menangani aksi pada item user
public interface UserActionListener {
    void showUserDialog(User user);
    void deleteUser(int userId);
}
}

```

Penjelasan Kode:

1. **Constructor UserAdapter:** Menerima daftar userList dan instance UserActionListener untuk menangani aksi pada item RecyclerView.
2. **onBindViewHolder:** Mengikat data pengguna (user) ke tampilan item, lalu menambahkan listener untuk tombol **Edit** dan **Delete**.
 - **Edit Button:** Memanggil showUserDialog dari MainActivity.
 - **Delete Button:** Memanggil deleteUser dari MainActivity.
3. **updateList:** Mengganti userList dengan daftar baru dan memperbarui tampilan adapter, yang digunakan saat ada pencarian atau perubahan data.

4. **UserViewHolder**: Menyimpan referensi TextView dan Button untuk setiap item di RecyclerView.
5. **Interface ActionListener**: Memiliki metode showUser Dialog dan deleteUser, yang akan diimplementasikan di MainActivity untuk memproses aksi dari adapter.

Koding build.gradle.kts(Module:App)

Detail Implementasi HTTP Request untuk CRUD Operations

Kita akan menggunakan OkHttp library untuk mengirimkan HTTP request. Tambahkan **dependency OkHttp** di build.gradle

Tambahkan dependency di build.gradle (Module: app)

```
implementation 'com.squareup.okhttp3:okhttp:4.9.1'
```

Atau

```
implementation(libs.okhttp)
```

Secara keseluruhan koding pada **build.gradle.kts (Module:App)**, kita bisa fokus kepada *dependencies* untuk penambahan **implementation**.

```
plugins {
    alias(libs.plugins.android.application)
}

android {
    namespace = "com.example.crud_android_user"
```

```
compileSdk = 34

defaultConfig {
    applicationId = "com.example.crud_android_user"
    minSdk = 29
    targetSdk = 34
    versionCode = 1
    versionName = "1.0"

    testInstrumentationRunner =
"androidx.test.runner.AndroidJUnitRunner"
}

buildTypes {
    release {
        isMinifyEnabled = false
        proguardFiles(
            getDefaultProguardFile("proguard-android-
optimize.txt"),
            "proguard-rules.pro"
        )
    }
}

compileOptions {
    sourceCompatibility = JavaVersion.VERSION_1_8
    targetCompatibility = JavaVersion.VERSION_1_8
}
}

dependencies {
```

```
implementation(libs.appcompat)
implementation(libs.material)
implementation(libs.activity)
implementation(libs.constraintlayout)
testImplementation(libs.junit)
androidTestImplementation(libs.ext.junit)
androidTestImplementation(libs.espresso.core)
implementation(libs.recyclerview)
implementation(libs.swiperefreshlayout)
implementation(libs.material.v160)
implementation(libs.okhttp)
}
```

c. Kode di MainActivity untuk CRUD

Isi MainActivity.java untuk mengatur fungsi-fungsi CRUD (Create, Read, Update, Delete).

```
package com.example.crud_android_user;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayoutManager;
```

```
import androidx.recyclerview.widget.RecyclerView;
import
androidx.swiperefreshlayout.widget.SwipeRefreshLayout;
import androidx.appcompat.widget.SearchView;
import com.google.android.material.snackbar.Snackbar;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import okhttp3.Call;
import okhttp3.Callback;
import okhttp3.OkHttpClient;
import okhttp3.Request;
import okhttp3.RequestBody;
import okhttp3.Response;
import okhttp3.FormBody;

public class MainActivity extends AppCompatActivity
implements UserAdapter.UserActionListener {

    private RecyclerView recyclerView;
    private Button buttonAdd;
    private ProgressBar progressBar;
    private SearchView searchView;
    private SwipeRefreshLayout swipeRefreshLayout;
    private UserAdapter adapter;
    private List<User> userList = new ArrayList<>();
```

```

private static final String BASE_URL =
"http://10.0.2.2/android_crud/";

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Inisialisasi View
    recyclerView = findViewById(R.id.recyclerView);
    buttonAdd = findViewById(R.id.buttonAdd);
    progressBar = findViewById(R.id.progressBar);
    searchView = findViewById(R.id.searchView);
    swipeRefreshLayout =
findViewById(R.id.swipeRefreshLayout);

    // Setup Adapter dan RecyclerView
    adapter = new UserAdapter(userList, this); // Pasang
'this' sebagai UserActionListener
    recyclerView.setLayoutManager(new
LinearLayoutManager(this));
    recyclerView.setAdapter(adapter);
    // Listener untuk Button Add
    buttonAdd.setOnClickListener(view ->
showUserDialog(null));

    // Setup Swipe to Refresh
    swipeRefreshLayout.setOnRefreshListener(() -> {
        loadUsers();
        swipeRefreshLayout.setRefreshing(false);
    });

```

```

});

// Listener untuk SearchView
searchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {
    @Override
    public boolean onQueryTextSubmit(String query) {
        filterUsers(query);
        return false;
    }

    @Override
    public boolean onQueryTextChange(String newText)
{
        filterUsers(newText);
        return false;
    }
});

// Memuat Data Pengguna dari Server
loadUsers();
}

private void loadUsers() {
    progressBar.setVisibility(View.VISIBLE);

    OkHttpClient client = new OkHttpClient();
    Request request = new Request.Builder()
        .url(BASE_URL + "read_user.php")
        .build();

```

```

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) {
        Log.e("MainActivity", "Error: " + e.getMessage());
        runOnUiThread() -> {
            progressBar.setVisibility(View.GONE);
            Toast.makeText(MainActivity.this, "Failed to
connect to server", Toast.LENGTH_SHORT).show();
        });
    }

    @Override
    public void onResponse(Call call, Response response)
throws IOException {
        if (response.isSuccessful()) {
            String json = response.body().string();
            try {
                JSONArray jsonArray = new JSONArray(json);
                userList.clear();
                for (int i = 0; i < jsonArray.length(); i++) {
                    JSONObject jsonObject =
jsonArray.getJSONObject(i);
                    int id = jsonObject.getInt("id");
                    String name = jsonObject.getString("name");
                    String email = jsonObject.getString("email");
                    userList.add(new User(id, name, email));
                }
                runOnUiThread() -> {
                    adapter.updateList(userList);
                }
            }
        }
    }
}

```

```

        progressBar.setVisibility(View.GONE);
    });
    } catch (JSONException e) {
        Log.e("MainActivity", "JSON Parsing Error: " +
e.getMessage());
        runOnUiThread() ->
progressBar.setVisibility(View.GONE));
    }
    } else {
        runOnUiThread() ->
progressBar.setVisibility(View.GONE));
    }
    }
    });
}

```

```

// Method untuk Menampilkan Dialog Tambah/Edit
User
@Override
public void showUserDialog(User user) {
    AlertDialog.Builder builder = new
AlertDialog.Builder(this);
    View view =
getLayoutInflater().inflate(R.layout.dialog_add_user, null);
    EditText editTextName =
view.findViewById(R.id.editTextName);
    EditText editTextEmail =
view.findViewById(R.id.editTextEmail);
    Button buttonSave =
view.findViewById(R.id.buttonSave);

```

```

if (user != null) {
    editTextName.setText(user.getName());
    editTextEmail.setText(user.getEmail());
}

buttonSave.setOnClickListener(v -> {
    String name =
editTextName.getText().toString().trim();
    String email =
editTextEmail.getText().toString().trim();

    if (name.isEmpty() || email.isEmpty()) {
        Toast.makeText(MainActivity.this, "Please fill all
fields", Toast.LENGTH_SHORT).show();
    } else {
        if (user == null) {
            createUser(name, email);
        } else {
            updateUser(user.getId(), name, email);
        }
    }
});

builder.setView(view);
builder.create().show();
}

private void createUser(String name, String email) {
    OkHttpClient client = new OkHttpClient();

```

```

RequestBody formBody = new FormBody.Builder()
    .add("name", name)
    .add("email", email)
    .build();

Request request = new Request.Builder()
    .url(BASE_URL + "create_user.php")
    .post(formBody)
    .build();

client.newCall(request).enqueue(new Callback() {
    @Override
    public void onFailure(Call call, IOException e) {
        Log.e("MainActivity", "Create User Error: " +
e.getMessage());
    }

    @Override
    public void onResponse(Call call, Response response)
throws IOException {
        if (response.isSuccessful()) {
            runOnUiThread() -> {
                loadUsers();
                Snackbar.make(recyclerView, "User added
successfully", Snackbar.LENGTH_SHORT).show();
            });
        }
    }
});
}

```

```

private void updateUser(int id, String name, String email)
{
    OkHttpClient client = new OkHttpClient();
    RequestBody formBody = new FormBody.Builder()
        .add("id", String.valueOf(id))
        .add("name", name)
        .add("email", email)
        .build();

    Request request = new Request.Builder()
        .url(BASE_URL + "update_user.php")
        .post(formBody)
        .build();

    client.newCall(request).enqueue(new Callback() {
        @Override
        public void onFailure(Call call, IOException e) {
            Log.e("MainActivity", "Update User Error: " +
e.getMessage());
        }

        @Override
        public void onResponse(Call call, Response response)
throws IOException {
            if (response.isSuccessful()) {
                runOnUiThread() -> {
                    loadUsers();
                    Snackbar.make(recyclerView, "User updated
successfully", Snackbar.LENGTH_SHORT).show();
                });
            }
        }
    });
}

```

```

    }
    });
}

// Method untuk Menghapus Pengguna
@Override
public void deleteUser(int userId) {
    new AlertDialog.Builder(this)
        .setTitle("Delete User")
        .setMessage("Are you sure you want to delete this
user?")
        .setPositiveButton("Yes", (dialog, which) -> {
            OkHttpClient client = new OkHttpClient();
            RequestBody formBody = new
FormBody.Builder()
                .add("id", String.valueOf(userId))
                .build();
            Request request = new Request.Builder()
                .url(BASE_URL + "delete_user.php")
                .post(formBody)
                .build();

            client.newCall(request).enqueue(new Callback() {
                @Override
                public void onFailure(Call call, IOException e)
{
                    Log.e("MainActivity", "Delete User Error: " +
e.getMessage());
                }
            });
        });
}
}

```

```

        @Override
        public void onResponse(Call call, Response
response) throws IOException {
            if (response.isSuccessful()) {
                runOnUiThread() -> {
                    loadUsers();
                    Snackbar.make(recyclerView, "User
deleted successfully", Snackbar.LENGTH_SHORT).show();
                });
            }
        });
    })
    .setNegativeButton("No", null)
    .show();
}

private void filterUsers(String query) {
    List<User> filteredList = new ArrayList<>();
    for (User user : userList) {
        if
(user.getName().toLowerCase().contains(query.toLowerCase()
||
user.getEmail().toLowerCase().contains(query.toLowerCase(
))) {
            filteredList.add(user);
        }
    }
    adapter.updateList(filteredList);
}

```

```

}

public void exit(View view) {
    new AlertDialog.Builder(this)
        .setTitle("Exit")
        .setMessage("Are you sure you want to exit?")
        .setPositiveButton("Yes", (dialog, which) -> {
            finishAffinity();
            System.exit(0);
        })
        .setNegativeButton("No", null)
        .show();
}
}
}

```

Teknik menentukan Base Url

1. Menggunakan 10.0.2.2 untuk Emulator Android:

- Jika menggunakan emulator Android, alamat 10.0.2.2 dapat digunakan untuk mengakses localhost pada mesin pengembangan.

BASE_URL = http://10.0.2.2/android_crud/;

2. Menggunakan IP Address Komputer:

- Temukan IP Address komputer, dengan cara:
 - Di Windows: Buka Command Prompt dan ketik **ip**, lalu temukan bagian IPv4 Address.

- Di macOS atau Linux: Buka Terminal dan ketik `ifconfig` atau `ip a`.
- Jika IP Address komputer adalah 192.168.1.223, URL-nya adalah
`BASE_URL = http://192.168.1.223/android_crud/;`

3. Jika Server Berjalan di Port Lain:

- Jika server berjalan di port selain 80 (misalnya, port 8080), tambahkan nomor port setelah IP.
- Misalnya, jika XAMPP berjalan di port 8080, URL-nya:
`URL url = new URL("http://192.168.1.223:8080/android_crud/");`

4. Menggunakan localhost di Peramban Komputer

Jika hanya menguji di peramban komputer, bisa menggunakan localhost, misalnya:

`http://localhost/android_crud/`

5. Menggunakan domain website

Jika kita onlinekan aplikasi atau database kita maka Url :

`URL url = new URL("http://namadomain/");`

Contoh

`URL url = new URL("http://www.efmimaiyana.com/");`

Penjelasan Koding Mainactivity.java

Aplikasi ini menggunakan HTTP requests (menggunakan `HttpURLConnection` atau `OkHttp`) untuk berkomunikasi dengan API PHP. Pastikan menyesuaikan URL API sesuai dengan lokasi server Kita.

Fungsi loadUsers akan mengirimkan GET request ke API `read_users.php` untuk mengambil data pengguna dari database MySQL dan mengupdate RecyclerView.

Fungsi createUser mengirim POST request ke API `create_user.php` dengan data name dan email

Fungsi updateUser mengirimkan POST request ke `update_user.php` dengan data id, name, dan email.

Fungsi deleteUser mengirim POST request ke `delete_user.php` dengan id pengguna yang ingin dihapus.

Fungsi showUserDialog untuk menampilkan Dialog form input nama dan email. Agar pengguna tahu ketika data sedang dimuat dari server, dapat menambahkan Progress Dialog atau ProgressBar untuk memberikan indikasi bahwa proses sedang berjalan. Tambahkan Toast untuk menampilkan pesan kesalahan jika terjadi kegagalan saat mengakses server atau API membuat class ApiClient untuk menangani semua permintaan HTTP agar kode lebih terorganisir. memanggil fungsi-fungsi ApiClient dari MainActivity untuk memisahkan logika jaringan. Contohnya, untuk memuat data pengguna: Tambahkan logika pencarian di dalam MainActivity yang akan memfilter daftar pengguna berdasarkan nama atau email yang dicari Pastikan bahwa setiap kali data dibuat, diubah, atau dihapus, RecyclerView diperbarui secara otomatis agar data terbaru selalu ditampilkan tanpa memuat ulang halaman. Tambahkan pemanggilan `loadUsers()` di akhir setiap fungsi `createUser`, `updateUser`, dan `deleteUser`

Menggunakan Snackbar untuk Feedback yang Lebih Interaktif untuk memberikan notifikasi pada pengguna (misalnya, saat pengguna berhasil menambahkan atau menghapus data), bisa menggunakan Snackbar untuk feedback

yang lebih elegan daripada Toast. Tambahkan Snackbar di fungsi `createUser`, `updateUser`, dan `deleteUser`.

Menyimpan Data Sementara dengan SharedPreferences (Opsional) Kita bisa menyimpan data sementara, seperti filter pencarian terakhir atau pengaturan tampilan, menggunakan SharedPreferences.

Catatan

- a. Gantilah `your-server-url` dengan URL server yang digunakan.
- b. Pastikan Kita memiliki kelas `UserAdapter` dan layout yang sesuai (`activity_main.xml`, `dialog_add_user.xml`, dll.).
- c. Snackbar digunakan sebagai feedback setelah penambahan, pengeditan, atau penghapusan data.

Jika terdapat masalah dalam menjalankan bisa melakukan beberapa langkah dibawah ini:

1. Error 'Cleartext HTTP traffic'

Error 'Cleartext HTTP traffic' terjadi karena Android 9 (Pie) dan versi lebih baru secara default memblokir permintaan HTTP yang tidak terenkripsi (HTTP tanpa HTTPS) untuk keamanan. Untuk mengatasi masalah ini, Anda bisa mengonfigurasi aplikasi Android agar mengizinkan lalu lintas 'Cleartext' (HTTP) dengan menambahkan konfigurasi khusus di 'AndroidManifest.xml'.

Langkah-langkah Mengatasi Cleartext HTTP Traffic

- a. Tambahkan Network Security Config di `res/xml`
 1. Buat folder baru bernama `xml` di dalam folder `res` jika belum ada.
 2. Di dalam folder `res/xml`, buat file baru bernama `network_security_config.xml`.

network_security_config.xml

```
xml
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain
includeSubdomains="true">YOUR_IP_ADDRESS_OR_DOMAIN</domain>
  </domain-config>
</network-security-config>
```

3. Gantilah `YOUR_IP_ADDRESS_OR_DOMAIN` dengan alamat IP lokal Anda atau domain server. Jika Anda ingin mengizinkan semua lalu lintas HTTP, gunakan `*` untuk `domain`, seperti ini:
Xml **<domain includeSubdomains="true">10.0.2.2</domain>**
- b. Ubah **AndroidManifest.xml** untuk Menggunakan Konfigurasi Ini
Di dalam file `AndroidManifest.xml`, tambahkan referensi ke `network_security_config.xml` di dalam tag ``:

Xml:

```
<application  
  
  android:networkSecurityConfig="@xml/network_security_  
  config"  
  ... >  
  ...  
  </application>
```

Contoh Lengkap **AndroidManifest.xml** xml:

```
<manifest  
  xmlns:android="http://schemas.android.com/apk/res/and  
  roid"  
  package="com.example.app">  
  <uses-permission  
  android:name="android.permission.INTERNET"/>  
  <application  
  
  android:networkSecurityConfig="@xml/network_security_  
  config"  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name"  
    android:roundIcon="@mipmap/ic_launcher_round"  
    android:supportsRtl="true"  
    android:theme="@style/AppTheme">  
    ...  
  </application>  
</manifest>
```

Setelah melakukan langkah-langkah ini, aplikasi akan mengizinkan lalu lintas HTTP (Cleartext) ke alamat yang Anda tentukan.

2. Bersihkan dan Rebuild Proyek

Kadang-kadang, Android Studio tidak mengenali komponen tertentu karena cache atau build yang belum bersih. Kita bisa membersihkan dan membangun ulang proyek dengan langkah-langkah berikut:

- Pilih **Build > Clean Project**.
- Setelah proses pembersihan selesai, pilih **Build > Rebuild Project**.

3. Invalidate Caches / Restart:

Buka File > Invalidate Caches / Restart dan pilih Invalidate and Restart.

BAB IX

Build APK

1. Pengertian Dan Langkah Pembuatan

Build APK merupakan langkah terakhir dalam mobile programming yaitu untuk menjadikan projek yang kita bangun menjadi sebuah aplikasi yang dapat di share dan di install oleh orang lain atau proses menghasilkan file apk (*Android Package*) yang dapat diinstal pada perangkat lainnya.

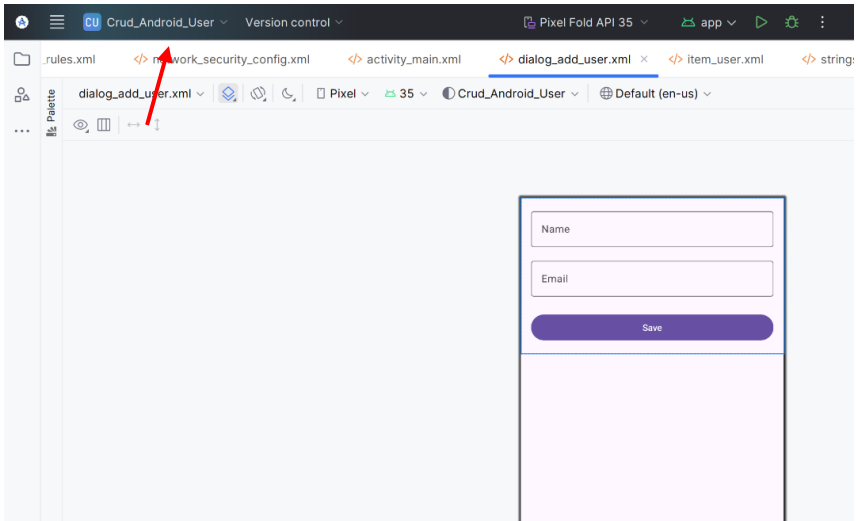
Adapun langkah pembuatannya:

Langkah 1:

Memastikan projek yang kita buat dalam android studio tidak memiliki eror, karena jika terdapat eror pada projek yang dibuat maka akan gagal dalam melakukan build apk.

Langkah 2:

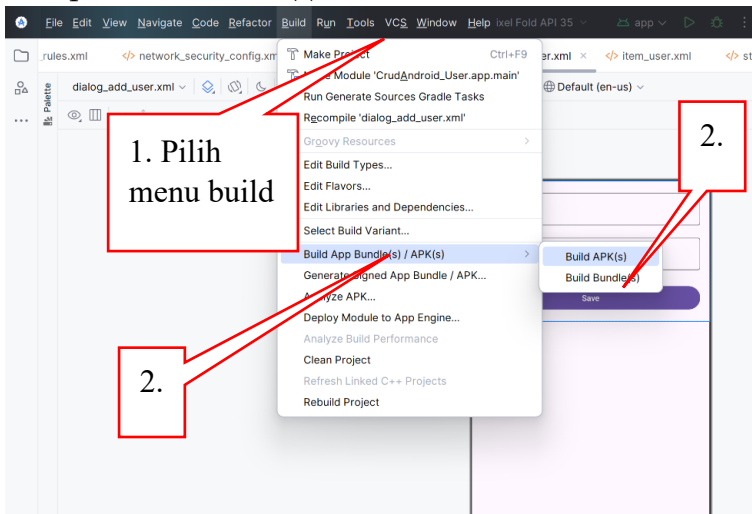
Setelah projek yang telah selesai dibuat dibuka maka klik garis tiga pada bagian atas kanan halaman kerja android studio.



Gambar 9. 1 Langkah Pembuatan Build APK

Langkah 3:

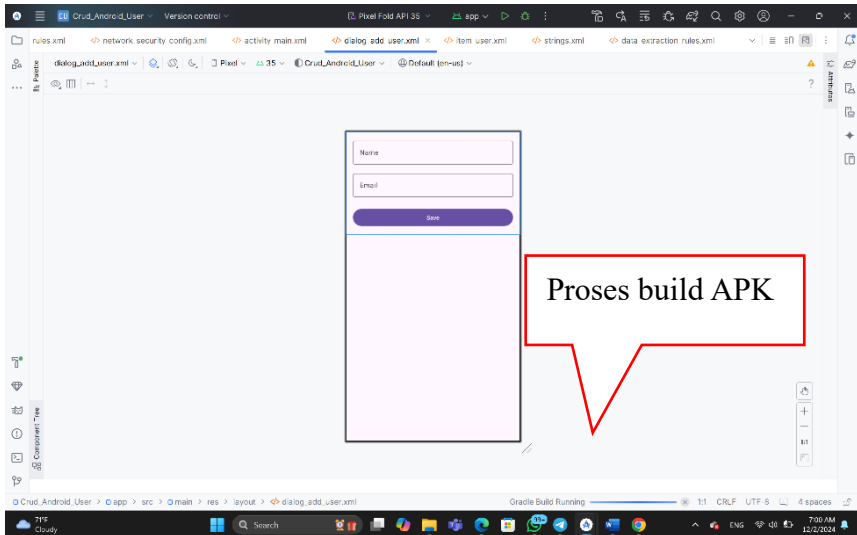
Kemudian pilih menu build lalu klik Build App Bundle(s) / APK(s) lalu pilih Build APK(s).



Gambar 9. 2 Langkah Pembuatan Build APK(s)

Langkah 4:

Tunggu proses pembuatan build APK(s)



Gambar 9. 3 Proses Pembuatan build Apk

Langkah 5:

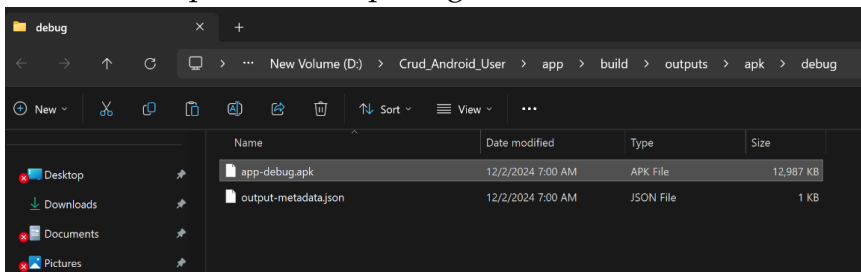
Berikut tampilan Build APK yang telah berhasil dan siap untuk di share. File build APK ini berada pada file proyek kita di bagian app kemudian pilih build kemudian outputs lalu apk kemudian debug akan tetapi jika ingin lebih cepat untuk menemukannya klik locate pada notifikasi build success. Maka akan langsung di arahkan ke tempat penyimpanan aplikasi yang telah berhasil kita buat.



Gambar 9. 4 Build APK Succesfully

Tampilan tersebut menunjukkan file apk yang telah berhasil dibuat dan siap untuk di share dan diinstal oleh perangkat lain langkah inilah yang menjadi langkah terakhir dalam pembuatan sebuah aplikasi.

Berikut tampilan pada file direktori atau file explorer yaitu app-debug.apk dan file dengan extensi .apk itulah yang akan diinstal pada sebuah perangkat mobile:



Gambar 9. 5 Lokasi Penyimpanan Build APK

Membuat file APK (*Android Package*) memiliki beberapa kegunaan penting, terutama Berikut adalah manfaat utamanya:

1. Distribusi Aplikasi

- **Google Play Store:** File APK digunakan untuk mengunggah aplikasi ke Google Play Store agar dapat diunduh oleh pengguna secara global.
- **Distribusi Langsung:** Anda dapat membagikan file APK melalui platform lain, seperti website, email, atau media sosial, untuk pengguna yang ingin menginstal aplikasi tanpa menggunakan Play Store.

2. Pengujian Aplikasi

- **Pengujian pada Perangkat Nyata:** File APK dapat diinstal langsung di perangkat Android untuk menguji aplikasi dalam kondisi dunia nyata. Ini penting untuk memastikan aplikasi berjalan dengan baik pada berbagai perangkat dan versi Android.
- **Beta Testing:** APK bisa digunakan untuk membagikan aplikasi kepada kelompok pengguna beta sebelum rilis resmi.

3. Dokumentasi dan Arsip

- File APK berfungsi sebagai arsip aplikasi yang dapat disimpan untuk kebutuhan masa depan, seperti pembaruan atau referensi.

4. Keperluan Bisnis atau Komersial

- APK memungkinkan perusahaan untuk mendistribusikan aplikasi internal secara langsung kepada karyawan atau mitra, tanpa mengunggahnya ke Play Store.
- Aplikasi khusus klien atau proyek freelance sering kali dibagikan dalam format APK.

5. Pengembangan Berkelanjutan

- Membuat APK memungkinkan pengembang untuk melihat hasil kerja mereka dan mengevaluasi apa saja yang perlu diperbaiki atau ditambahkan.

Aplikasi yang kita bangun siap untuk kita jalankan, dan kita gunakan secara bersama.

BAB X

Unggah Aplikasi Ke Google Play Store

Mengunggah aplikasi ke *Google Play Store* memiliki banyak kegunaan yang strategis, terutama jika ingin menjangkau audiens yang lebih luas atau memiliki tujuan bisnis. Berikut adalah beberapa manfaat utamanya:

1. Distribusi Global

- **Akses ke Jutaan Pengguna:** Google Play Store adalah platform distribusi aplikasi terbesar di dunia, dengan jutaan pengguna aktif. Mengunggah aplikasi ke sana memungkinkan Anda menjangkau pengguna di berbagai negara.

2. Meningkatkan Kredibilitas

- **Kepercayaan Pengguna:** Aplikasi yang tersedia di Play Store dianggap lebih aman dan terpercaya oleh pengguna, karena Google melakukan tinjauan sebelum aplikasi dipublikasikan.
- **Brand Image:** Memiliki aplikasi di Play Store dapat meningkatkan citra profesionalitas dan kredibilitas brand atau bisnis Anda.

3. Monetisasi Aplikasi

- Dapat menghasilkan uang melalui:
 - a. **Iklan Dalam Aplikasi:** Memanfaatkan platform seperti AdMob.
 - b. **Pembelian Dalam Aplikasi (In-App Purchases):** Untuk fitur premium atau konten eksklusif.
 - c. **Aplikasi Berbayar:** Menetapkan harga untuk mengunduh aplikasi.

4. Pembaruan Mudah dan Terpusat

- Google Play Store memungkinkan memberikan pembaruan aplikasi secara otomatis kepada semua pengguna, tanpa perlu membagikan file APK secara manual.

5. Analitik dan Insight

- Dengan Play Console, bisa mendapatkan data penting seperti jumlah unduhan, demografi pengguna, ulasan, dan feedback. Data ini membantu meningkatkan kualitas aplikasi.

6. Menjangkau Target Audiens yang Tepat

- Google Play Store menyediakan fitur segmentasi yang memungkinkan aplikasi Anda ditemukan oleh audiens yang sesuai dengan kategori, deskripsi, dan kata kunci aplikasi.

7. Integrasi dengan Ekosistem Google

- Aplikasi di Play Store dapat memanfaatkan integrasi dengan layanan Google lainnya seperti Google Pay,

Firestore, atau Google Maps untuk meningkatkan pengalaman pengguna.

8. Peluang Kolaborasi

- Aplikasi bisa menarik perhatian investor, mitra bisnis, atau pengguna yang mungkin tertarik untuk mendukung atau menggunakan produk lebih lanjut.

9. Kemudahan Pengelolaan

- Dapat mengatur distribusi aplikasi untuk negara tertentu, membatasi akses untuk target tertentu, atau bahkan menyediakan aplikasi hanya untuk penggunaan internal (melalui distribusi pribadi di Play Console).

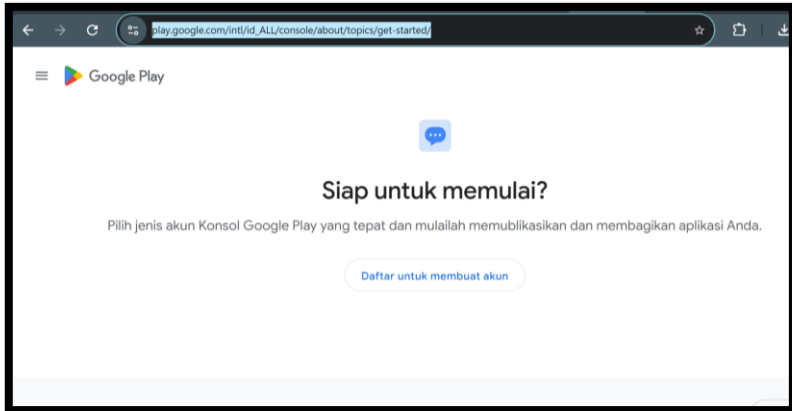
Jika memiliki aplikasi yang sudah siap diunggah, mengunggahnya ke Play Store adalah langkah strategis untuk memanfaatkan platform ini secara maksimal

Berikut adalah langkah-langkah untuk mengunggah aplikasi ke *Google Play Store*:

1. Siapkan Akun Google Play Console

https://play.google.com/intl/id_ALL/console/about/topics/get-started/

- **Buat Akun Developer:**
 - a. Kunjungi Google Play Console.



Gambar 10. 1 Buat Akun Developer

- b. Buat akun dan bayar biaya pendaftaran sebesar \$25 (biaya sekali bayar).
- c. Setelah verifikasi, Anda dapat mulai mengunggah aplikasi.

2. Siapkan Aplikasi Anda

- Pastikan aplikasi sudah selesai dan siap dirilis.
- APK/Bundle harus:
 - a. Ditandatangani dengan keystore untuk rilis.
 - b. Tidak memiliki error atau bug kritis.
- Pastikan aplikasi memenuhi pedoman Google Play (termasuk kebijakan privasi dan konten).

3. Buat Dokumen Pendukung

- **Deskripsi Aplikasi:** Buat deskripsi menarik dan informatif.
- **Ikon Aplikasi:** Berformat PNG, ukuran 512x512 piksel.

- **Screenshot:** Tangkapan layar dari aplikasi (minimal 2 untuk setiap jenis perangkat: ponsel, tablet, dll.).
- **Video Promosi (Opsional):** Anda dapat mengunggah video demonstrasi ke YouTube dan menyematkannya di Play Store.
- **Kebijakan Privasi:** Siapkan URL halaman kebijakan privasi jika aplikasi Anda mengumpulkan data pengguna.

4. Buat Proyek Baru di Google Play Console

1. **Login ke Play Console.**
2. Klik **All Apps > Create App.**
3. Isi informasi dasar:
 - a. Nama aplikasi.
 - b. Bahasa utama.
 - c. Kategori (misalnya, Pendidikan, Hiburan, dll.).
 - d. Pilih apakah aplikasi ini gratis atau berbayar.
4. Klik **Create.**

5. Unggah File Aplikasi (APK/AAB)

1. Masuk ke **Production > Releases > Create New Release.**
2. Pilih metode penandatanganan aplikasi (disarankan menggunakan **Google Play App Signing**).
3. Unggah file APK atau App Bundle (.aab).

6. Isi Informasi Aplikasi

- **Content Rating:** Isi kuesioner untuk menentukan peringkat konten aplikasi.
- **Target Audience and Content:** Tentukan usia target pengguna aplikasi.

- **Data Safety:** Jelaskan data apa saja yang dikumpulkan aplikasi Anda.
- **App Details:** Unggah ikon, deskripsi, dan screenshot.

7. Tentukan Harga dan Negara Distribusi

- Atur apakah aplikasi Anda gratis atau berbayar.
- Pilih negara tempat aplikasi Anda tersedia.

8. Periksa dan Publikasikan

- Periksa kembali semua informasi.
- Setelah yakin, klik **Publish** untuk memulai proses review.

9. Tunggu Proses Review

- Google akan meninjau aplikasi Anda, biasanya dalam waktu 1-7 hari.
- Jika disetujui, aplikasi Anda akan tersedia di Play Store.

DAFTAR PUSTAKA

- Dhita R. L., Faulina, S. T., & Wisnumurti. (2023). Rancang Bangun Aplikasi Layanan Pengaduan Pada Dinas Pendidikan Kabupaten Oku Berbasis Android Menggunakan Android Studio. *Jik, 14Faulina,(2)*, 25–35. <https://journal.unmaha.ac.id/index.php/jik/article/view/252/214>
- Hendriyani, Y., & Suryani, K. (2020). Pemrograman Android Teori & Aplikasi. In *Qiara Media* (Vol. 258, Nomor 1, hal. 258).
- Herdiana, Y. (2018). Aplikasi Rumus Matematika Sma Berbasis Mobile. *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, 3(1), 112–121.
- Nisa, A., Abduh, H., & Dani, A. A. H. (2022). Sistem Informasi Geografis Lokasi Dokter Praktek Berbasis Android Di Kota Palopo. *Komputa: Jurnal Ilmiah Komputer dan Informatika*, 11(2), 78–85. <https://doi.org/10.34010/komputa.v11i2.7358>
- Prabowo, I. A., Wijayanto, H., Yudanto, B. W., & Nugroho, S. (2020). Buku Ajar Pemrograman Mobile Berbasis Android. In *Angewandte Chemie International Edition*. https://eprints.sinus.ac.id/762/1/Buku_Ajar-Pemrograman_Android.pdf
- Sahrial, R., Fauzi, D. F., & Susilawati, E. (2022). Pemanfaatan Json Untuk Menampilkan Data Realtime Covid-19 Dengan Model View Presenter. *Jurnal Teknoinfo*, 16(1), 144. <https://doi.org/10.33365/jti.v16i1.780>

Saputra, W. A. (2020). *Pemrograman Berbasis Objek Pemrograman Mobile dengan Android Studio* (Banjar Mas). POLIBAN PRESS.

Stmik Indonesia, S. (2018). *Mobile Programming Stmik Stikom Indonesia 2018*. https://stiki-indonesia.ac.id/wp-content/uploads/2019/03/modul/MODUL_TI/Modul_Mobile_Programming_Android.pdf

<https://developer.android.com/>

https://play.google.com/intl/id_ALL/console/about/topics/get-started/

BIODATA PENULIS

Dr. Supratman Zakir, S. Kom., M. Pd., M. Kom, Lahir di sebuah desa kecil bernama Lolo Hilir Kecamatan Gunung Raya Kabupaten Kerinci Propinsi Jambi. Tahun 1990 melanjutkan pendidikan di Pondok Pesantren Modern Nurul Ikhlas Padang Panjang. Tahun 1995 memulai pendidikan Sarjana di Jurusan Teknik Komputer di Sekolah Tinggi Manajemen



Informatika & Komputer (STMIK) YPTK Padang dan selesai pada tahun 2000. Tahun 2005 menyelesaikan Pendidikan Magister Teknologi Pendidikan di Universitas Negeri Padang (UNP). Pada tahun 2008 menyelesaikan pendidikan Magister Ilmu Komputer di Universitas Putra Indonesia (UPI) Padang dan Pendidikan Doktor diselesaikan pada tahun 2018 di Universitas Negeri Padang (UNP)

Ketua Jurusan Pendidikan Teknik Informatika & Komputer (2007-2010), Ketua Pusat Teknologi Informasi dan Pangkalan Data IAIN Bukittinggi (2010-2014), Sekretaris Lembaga Penelitian dan Pengabdian Masyarakat (LP2M) IAIN Bukittinggi (2014-2019). Dan Wakil Dekan Bidang Kemahasiswaan dan Kerjasama Fakultas Tarbiyah dan Ilmu Keguruan UIN Bukittinggi (2019-2022). Wakil Dekan Bidang Akademik dan Kelembagaan Fakultas Tarbiyah dan Ilmu Keguruan UIN Bukittinggi (2024-Sekarang). Penulis juga merupakan Asesor Nasional Lembaga Akreditasi Mandiri Kependidikan (LAMDIK), sehingga Penulis aktif memberikan pendampingan Akreditasi Program Studi bidang Kependidikan.

Beberapa penelitian yang telah dilakukan di antaranya : Desain Sistem Informasi Administrasi UKM Berbasis Web (2012). Desain Web Base Learning pada Matakuliah Pendidikan Agama

Islam (2016), Pengembangan Model Pembelajaran Berbasis Web (Web Base Learning, 2018), Efektivitas Model Pembelajaran Berbasis Web (2018). Digitalisasi Motif Kain Batik Nasional dalam Menyerap Nilai-nilai Keberagaman sebagai upaya melestarikan nilai-nilai Kebudayaan (2020), University Digital Transformation Index in The Malay Family of Countries (Penelitian Kolaborasi Internasional, 2023).

Penulis juga produktif menghasilkan buku, di antaranya : Cara Asyik Belajar PHP dengan Framework Codeigniter 4 (2023); Sains, Teknologi dan Masyarakat di Era Digital (2022); Segenggam Mozaik untuk Masyarakat (2021); Menggagas Model Pembelajaran dari Rumah -Learning From Home- (2020); Kupas Tuntas Bahasa Pemrograman Berbasis Web : HTML, PHP, Java + MySQL (2019).

Doktor PTIK ini aktif menjadi narasumber baik di lingkungan perguruan tinggi maupun masyarakat umum, diantaranya : Narasumber Guru Cerdas dengan ICT se-Kabupaten Padang Pariaman (2013); E-Government kabupaten/kota se-Propinsi Sumatera Barat (2013); Konsultan ICT-Pura Kabupaten Tanah Datar, Sawah Lunto, Bukittinggi (2012-2014); Media Pembelajaran Berbasis ICT untuk Guru Kab. Agam (2013); Peningkatan Kompetensi Guru PAI Kab. Agam (2012); Seminar Nasional Etika Menggunakan IT (2018), Seminar Nasional Progress of Website Security System in the Future (2018); Bukittinggi International Conference in Engineering (BICIE, 2019); Bukittinggi International Conference on Education (BICED, 2020). Penulis dapat dihubungi di e-mail : supratman@uinbukittinggi.ac.id

Efmi Maiyana, S.Kom., M.Kom.

Lahir: Padang, 30 Mei 1978

ID Scopus : 57216151481

Email: efmi.manza@gmail.com |

efmi_maiyana@yahoo.com

Website: efmimaiyana.com |

<https://bit.ly/efmimaiyanaok>



Profil

Saya adalah seorang pendidik dan praktisi di bidang Teknologi Informasi yang tertarik dalam mengembangkan kemampuan bahasa pemrograman dan teknologi pendidikan. Dengan latar belakang akademik, saya menyelesaikan studi Magister Komputer di Universitas Pendidikan Indonesia YPTK Padang pada Tahun 2008 dengan predikat pujian dan kini tengah melanjutkan pendidikan S3 di Universitas Negeri Padang (UNP) pada program studi Pendidikan Teknologi dan Kejuruan (PTK). Saya memiliki pengalaman mengajar di berbagai institusi pendidikan tinggi, termasuk di AMIK Bukittinggi, UIN Bukittinggi, STES Manna Wa Salwa serta Tutor Pada Universitas Terbuka dalam matakuliah bidang Sistem Informasi. Jabatan Struktural saat ini sebagai Direktur di AMIK Bukittinggi

Keterlibatan Organisasi dan Profesional

- Sekretaris APTIKOM SUMBAR (Asosiasi Perguruan Tinggi Informatika dan Komputer, Sumatera Barat) 2018 – 2022
- Sekretaris APTIKOM SUMBAR (Periode 2022 – 2026)
- Trainer Microsoft di OCM VEA (Virtual Education Academy) 2021 - sekarang
- Tim Editor Jurnal Sains Informatika (LLDIKTI X) 2018 – Sekarang

- Managing Editor JISED (Journal of Information System and Education Development)
- Reviewer Teknoif 2020 - sekarang
- Reviewer Jurnal Pustaka Data 2021 – *Sekarang*

Sebagai pengajar, saya berfokus untuk memberikan pemahaman yang mendalam tentang pengembangan aplikasi, serta mengembangkan kurikulum yang relevan dengan perkembangan teknologi terkini. Saya percaya bahwa teknologi, terutama bahasa pemrograman, adalah kunci untuk menciptakan solusi inovatif dalam dunia pendidikan.

Sebagai seorang akademisi dan praktisi teknologi, berkomitmen untuk terus berinovasi dan berbagi pengetahuan, serta memberikan kontribusi dalam pengembangan pendidikan yang berbasis teknologi. Besar harapan saya untuk dapat berkolaborasi dengan berbagai pihak dalam mengembangkan dan menerapkan teknologi. Saya terbuka untuk kesempatan kerja sama yang dapat saling mendukung dan menciptakan dampak positif di bidang teknologi dan pendidikan



SADAR MARTUA HAHOLONGAN SIR

✉ sadarmartuahaholongansir@gmail.com

☎ 0822 - 5395 - 2592

📍 Ps. Latong, Kab. Padang Lawas, Prov. Sumatera Utara

Tentang Saya

Seorang individu yang gemar belajar hal-hal baru, memiliki minat besar terhadap bahasa pemrograman, teknologi, dan public speaking. Bersemangat untuk terus berkembang dengan menggali pengetahuan di bidang teknologi terkini dan memperluas kemampuan komunikasi. Kombinasi minat dalam teknologi dan public speaking menciptakan kemampuan unik untuk menyampaikan ide-ide kompleks dengan cara yang menarik dan mudah dipahami.

Soft Skill

- Desain
- Public Speaking
- Kepemimpinan
- Editing
- Multimedia

Pendidikan

UIN Sjech M. Djamil Djambek Bukittinggi 2022 - Sekarang
S1 Pendidikan Teknik Informatika Dan Komputer

Pencapaian

- Duta Coding Design Web 2024
- Medali Emas Pada Olimpiade KOSMIK Bidang Bahasa 2024
- Terbaik3 MSQ PKM DEMA F 2023
- Terbaik1 MSQ Tingkat Kabupaten Padang Lawas 2022

Indeks

A

Activity · 8, 63, 64, 65, 110, 111, 113, 118, 119, 123, 159, 187, 275
Android · 1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 29, 36, 37, 38, 40, 46, 47, 48, 49, 50, 51, 52, 55, 64, 67, 79, 107, 121, 208, 210, 214, 248, 251, 253, 255, 259, 267, 268, 275
Android Studio · 1, 4, 5, 6, 7, 8, 21, 22, 23, 24, 26, 29, 36, 37, 38, 40, 46, 47, 48, 49, 50, 51, 52, 64, 67, 79, 208, 210, 214, 253, 267, 275
APK · 9, 255, 256, 257, 258, 259, 260, 262, 264, 265, 275
Aritmatika · 9, 131, 275
Array · 123, 180, 275
Atribut · 37, 39, 41, 275
Attributes · 8, 36, 275

B

Base url · 275
Build APK · 256, 275
Button · 8, 9, 47, 71, 74, 77, 78, 83, 98, 107, 111, 115, 117, 118, 119, 121, 123, 125, 126, 134, 137, 139, 143, 145, 146, 150, 154, 156, 159, 174, 180, 184, 197, 217, 220, 221, 223, 230, 232, 233, 234, 236, 237, 238, 242, 275

C

Constraint Layout · 80, 275
CRUD · 9, 201, 210, 211, 222, 234, 236, 275

D

Database · 9, 201, 202, 211, 275
Deklarasi · 130, 133, 189, 190, 196, 275
Desain · 29, 132, 142, 149, 164, 269, 273, 275
Drawbel · 275

E

EditText · 52, 53, 54, 55, 132, 134, 136, 139, 142, 143, 146, 150, 154, 155, 156, 159, 171, 180, 184, 185, 236, 242, 275
ekonomi · 2
Elemen · 8, 44, 45, 46, 142, 221, 275
Emulator · 22, 25, 30, 31, 32, 33, 34, 248, 275
Error · 240, 241, 243, 244, 246, 251, 275
Explicit intent · 188, 276

F

File · 67, 253, 257, 259, 265, 276

G

Google · 10, 12, 13, 14, 16, 17, 187, 259, 261, 262, 263, 264, 265, 266, 276

I

Id · 132, 142, 276
Implicit intent · 276
InputType · 55, 276
Intent · 9, 64, 187, 188, 193, 195, 197, 276
IP Address · 248, 276

J

JSON · 9, 201, 241, 276

K

Koding · 29, 73, 119, 126, 127, 134, 139, 144, 146, 155, 159, 190, 193, 197, 229, 234, 249, 276

L

Layout · 5, 9, 42, 43, 46, 67, 68, 69, 70, 71, 72, 79, 84, 132, 142, 149, 164, 214, 215, 219, 221, 222, 276
Linear Layout · 67, 276
Logika · 9, 131, 142, 143, 276

M

Margin · 41, 43, 44, 45, 276
MySQL · 9, 201, 210, 211, 249, 270, 276

O

onClick · 108, 109, 115, 120, 125, 137, 145, 156, 174, 188, 191, 195, 197, 217, 276
Output · 117, 121, 128, 140, 163, 195, 276

P

Padding · 41, 42, 43, 45, 276
penyimpanan · 5
Platte · 276
Play Store · 10, 13, 259, 260, 261, 262, 263, 265, 266, 276
Projek · 8, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 276

R

Relative Layout · 67, 79, 276
Running · 34, 79, 121, 128, 141, 195, 277

S

Scroll View · 8, 55, 56, 57, 59, 61, 277
SDK · 24, 25, 27, 277
setOnClickListener · 9, 118, 119, 120, 121, 125, 231, 232, 238, 242, 277
Spinner · 9, 121, 122, 123, 125, 126, 128, 172, 179, 181, 184, 277
String · 9, 107, 111, 112, 113, 125, 131, 134, 140, 144, 147, 151, 153, 160, 182, 194, 226, 229, 230, 238, 239, 240, 241, 242, 243, 244, 245, 246, 277

T

Tabel · 277

Text View · 277

TextAligenment · 40, 277

Toast · 9, 112, 114, 116, 117, 119, 123,
139, 150, 159, 181, 185, 236, 240,
242, 249, 250, 277

V

Variabel · 9, 113, 130, 184, 185, 277

X

Xampp · 277