

MODUL PRAKTIKUM PEMROGRAMAN MOBILE



**LABORATORIUM REKAYASA PERANGKAT LUNAK
PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN
ALAM
UNIVERSITAS HASANUDDIN
2018**

IDENTITAS MATAKULIAH

Nama Mata Kuliah : Pemrograman Mobile

Kode Mata Kuliah :

SKS : Teori 2 SKS

Praktikum 1 SKS

Pengampu Mata Kuliah : A. Muh. Amil Siddik, S. Si., M. Si.

CPL-MATAKULIAH (CPL-MK)/ Sasaran Belajar

Capaian pembelajaran mata kuliah Pemrograman Mobile mendukung pencapaian visi dan misi prodi. Adapun capaian lulusan prodi yang dibebankan pada mata kuliah ini adalah sebagai berikut.

1. Berkontribusi dalam peningkatan mutu kehidupan bermasyarakat, berbangsa, bernegara dan kemajuan peradaban berdasarkan Pancasila (ST3).
2. Menunjukkan sikap bertanggungjawab atas pekerjaan di bidang keahliannya secara mandiri (ST9).
3. Mampu menerapkan pemikiran logis, kritis, sistematis, dan inovatif dalam konteks pengembangan atau implementasi ilmu pengetahuan dan teknologi yang memperhatikan dan menerapkan nilai humaniora yang sesuaidengan ilmu komputer. (KU1)
4. Mampu menunjukkan kinerja mandiri, bermutu, dan terukur. (KU2)
5. Mampu mengkaji implikasi pengembangan atau implementasi ilmu pengetahuan dan teknologi yang memperhatikan dan menerapkan nilai humaniora sesuai dengan keahliannya berdasarkan kaidah, tata cara dan etika ilmiah dalam rangka menghasilkan solusi, gagasan, desain atau kritik seni. (KU3)

6. Mampu mengambil keputusan secara tepat dalam konteks penyelesaian masalah di bidang keahliannya, berdasarkan hasil analisis informasi dan data. (KU5)
7. Menguasai konsep teoritis ilmu komputer dalam kaitannya pada pemecahan masalah melalui perancangan dan pengembangan algoritma untuk berbagai keperluan seperti *Network Security, Data Compression Multimedia Technologies, Mobile Computing Intelligent Systems, Information Management, Algorithms and Complexity, Human-Computer Interaction, Graphics and Visual Computing* (PI1).

Guna terpenuhinya capaian lulusan prodi tersebut maka capaian mata kuliah Pemrograman Mobile adalah mahasiswa mampu menggunakan Software Development Kit (SDK) Android Studio untuk mengembangkan aplikasi yang memanfaatkan UI dan sistem operasi Android, serta mampu mengembangkan aplikasi menggunakan SDK dengan menggunakan bahasa pemrograman Java dan file XML untuk menjelaskan sumber daya data dengan menunjukkan sikap jujur, bertanggungjawab, teliti, kritis, kreatif dan disertai kepercayaan diri yang tinggi untuk meningkatkan mutu kehidupan berbangsa dan bernegara.

Pengalaman Belajar

Selama proses perkuliahan ini, mahasiswa mengikuti kegiatan antara lain, Ceramah, Diskusi, Praktikum dan Simulasi.

Kata Pengantar

Puji syukur ke hadirat Tuhan Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Modul Praktikum Pemrograman Mobile dapat diselesaikan.

Modul praktikum di buat dalam rangka membantu proses pembelajaran khususnya untuk matakuliah yang mengharuskan adanya praktikum. Modul praktikum ini diharapkan dapat membantu mahasiswa dalam mempersiapkan dan melaksanakan praktikum dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan tujuan pelaksanaan praktikum dan semua kegiatan yang harus dilakukan oleh mahasiswa serta teori singkat untuk memperdalam pemahaman mahasiswa mengenai materi yang dibahas.

Penyusun menyakini bahwa dalam pembuatan Modul Praktikum Pemrograman Mobile ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran agar dapat menyempurnakan modul praktikum ini dimasa yang akan datang.

Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Makassar, September 2018

Penyusun

Daftar Isi

IDENTITAS MATAKULIAH.....	i
Kata Pengantar	iii
Daftar Isi	iv
BAB I Pengenalan Android.....	1
BAB II Layout	16
BAB III Membuat User Interface.....	28
BAB IV Aktivitas dan Intents	37
BAB V Siklus Hidup Aktivitas	46
BAB VI Dialog.....	51
BAB VII Multimedia.....	59
BAB VIII Build Apk.....	71

Sasaran Pembelajaran

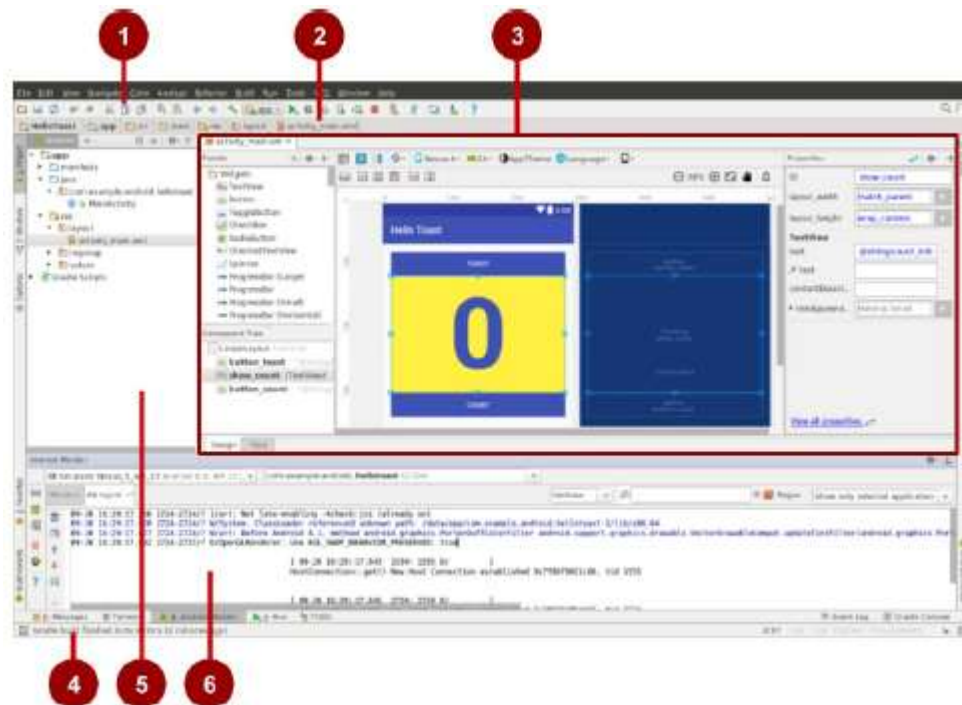
Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu menjelaskan tentang sistem operasi Android
2. Mahasiswa mampu membuat proyek Android dari template aplikasi dasar

❖ DASAR TEORI

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform yang bersifat open source bagi para pengembang untuk menciptakan sebuah aplikasi. Awalnya, Google Inc. mengakuisisi Android Inc. yang mengembangkan software untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, yaitu konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android.

Panel jendela Android Studio



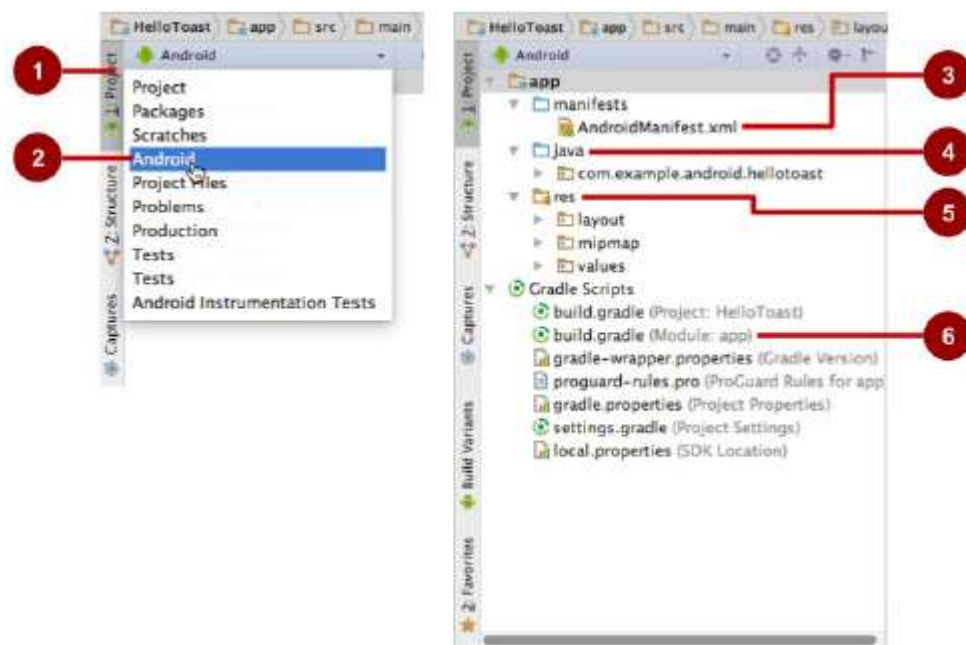
Gambar 1.1 Panel jendela Android Studio

1. **Bilah Alat.** Bilah alat menjalankan beragam tindakan, termasuk menjalankan aplikasi Android dan meluncurkan alat Android.
2. **Bilah Navigasi.** Bilah navigasi memungkinkan navigasi melalui proyek dan membuka file untuk pengeditan. Bilah navigasi menyediakan tampilan struktur proyek yang lebih ringkas.
3. **Panel Editor.** Panel ini menampilkan materi file yang dipilih dalam proyek. Misalnya, setelah memilih layout (seperti yang ditampilkan dalam gambar), panel ini menampilkan editor layout dengan alat untuk mengedit layout. Setelah memilih file kode Java, panel ini menampilkan kode dengan alat untuk mengedit kode.
4. **Bilah Status.** Bilah status menampilkan status proyek dan Android Studio itu sendiri, serta peringatan atau pesan apa pun. Anda bisa mengamati kemajuan pembangunan di bilah status.
5. **Panel Project.** Panel proyek menampilkan file proyek dan hierarki proyek.

- 6. Panel Monitor.** Panel monitor menawarkan akses ke daftar TODO untuk mengelola tugas, Android Monitor untuk memantau eksekusi aplikasi (ditampilkan dalam gambar), logcat untuk menampilkan pesan log, dan aplikasi Terminal untuk melakukan aktivitas Terminal.

Menjelajahi proyek

Setiap proyek di Android Studio berisi file AndroidManifest.xml, file kode sumber komponen, dan file sumber daya terkait. Secara default, Android Studio mengatur file proyek Anda berdasarkan jenis file, dan menampilkannya dalam proyek



Gambar 1.2 Project Android Studio

1. Tab **Project** . Klik untuk menampilkan tampilan proyek.
2. Pilihan **Android** di menu tarik-turun proyek.
3. File **AndroidManifest.xml**. Digunakan untuk menetapkan informasi tentang aplikasi bagi lingkungan waktu proses Android. Template yang Anda pilih membuat file ini.
4. Folder **java**. Folder ini menyertakan aktivitas, pengujian, dan komponen lain di kode sumber Java. Setiap aktivitas, layanan, dan komponen lain

didefinisikan sebagai kelas Java, biasanya dalam file miliknya. Nama aktivitas pertama (layar) yang pengguna lihat, yang juga melakukan inisialisasi sumber daya seluruh aplikasi, biasanya adalah MainActivity.

5. Folder **res**. Folder ini menyimpan sumber daya, seperti layout XML, string UI, dan gambar. Aktivitas biasanya dikaitkan dengan file sumber daya XML yang menetapkan layout tampilannya. File ini biasanya diberi nama setelah aktivitas atau fungsinya.

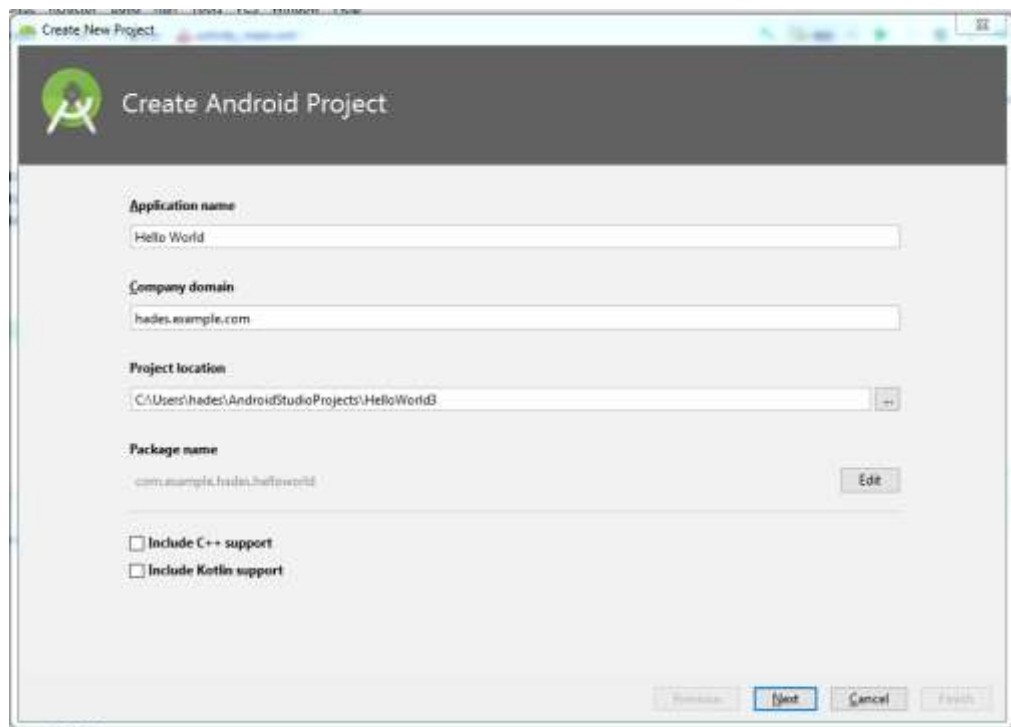
File ****build.gradle (Module: App.)***. *File ini menetapkan konfigurasi pembangunan modul. Template yang Anda pilih membuat file ini, yang mendefinisikan konfigurasi pembangunan, termasuk atribut **minSdkVersion** yang mendeklarasikan versi minimum untuk aplikasi, dan atribut **targetSdkVersion** yang mendeklarasikan versi tertinggi (terbaru) untuk aplikasi yang dioptimalkan. File ini juga menyertakan daftar dependensi*, yaitu pustaka yang diperlukan oleh kode — seperti pustaka AppCompatActivity untuk mendukung beragam versi Android.*

❖ LATIHAN PRAKTIKUM

Latihan 1 : Membuat program “Hello World”

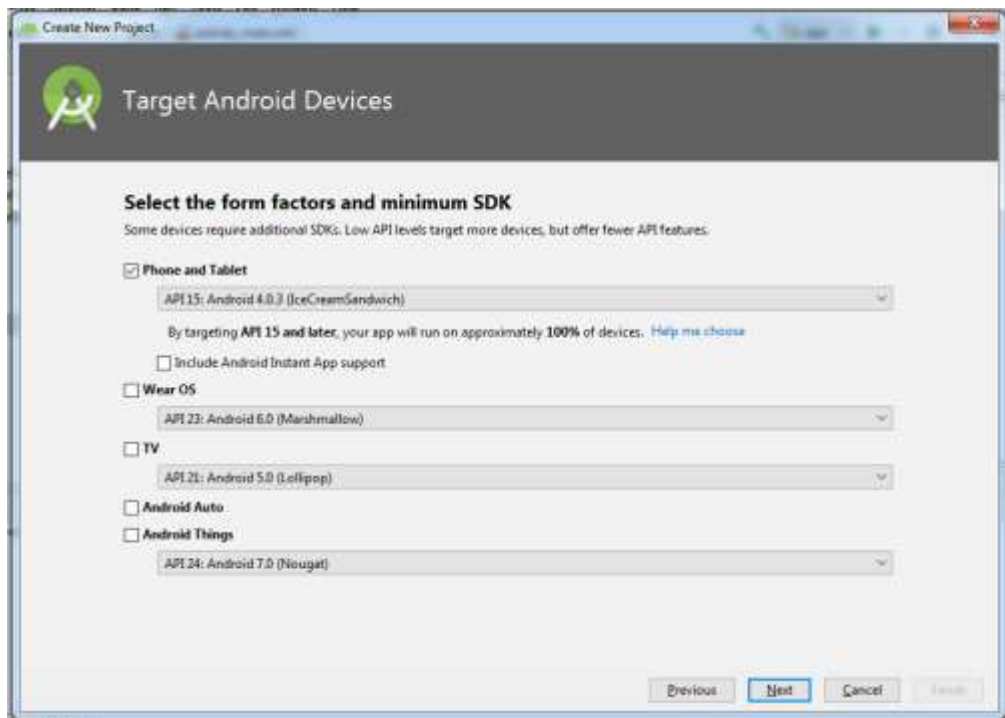
1. Buka Android Studio jika belum terbuka.
2. Di jendela utama **Welcome to Android Studio**, klik "Start a new Android Studio project".
3. Di jendela **New Project**, berikan aplikasi Anda **Application Name**, seperti "Hello World".
4. Verifikasi lokasi Proyek, atau pilih direktori yang berbeda untuk menyimpan proyek.
5. Pilih **Company Domain** yang unik.
 - 5.1. Aplikasi yang dipublikasikan di Google Play Store harus memiliki nama paket yang unik. Karena domain unik, mengawali nama aplikasi dengan nama atau nama domain perusahaan Anda akan menghasilkan nama paket yang unik.

- 5.2. Jika tidak berencana mempublikasikan aplikasi, Anda bisa menerima domain contoh default. Perhatikan bahwa mengubah nama paket aplikasi di kemudian hari berarti melakukan pekerjaan ekstra.
6. Verifikasi bahwa **Project Location** default adalah tempat Anda menyimpan aplikasi Hello World dan proyek Android Studio lainnya, atau ubah lokasi ke direktori yang diinginkan. Klik Next.



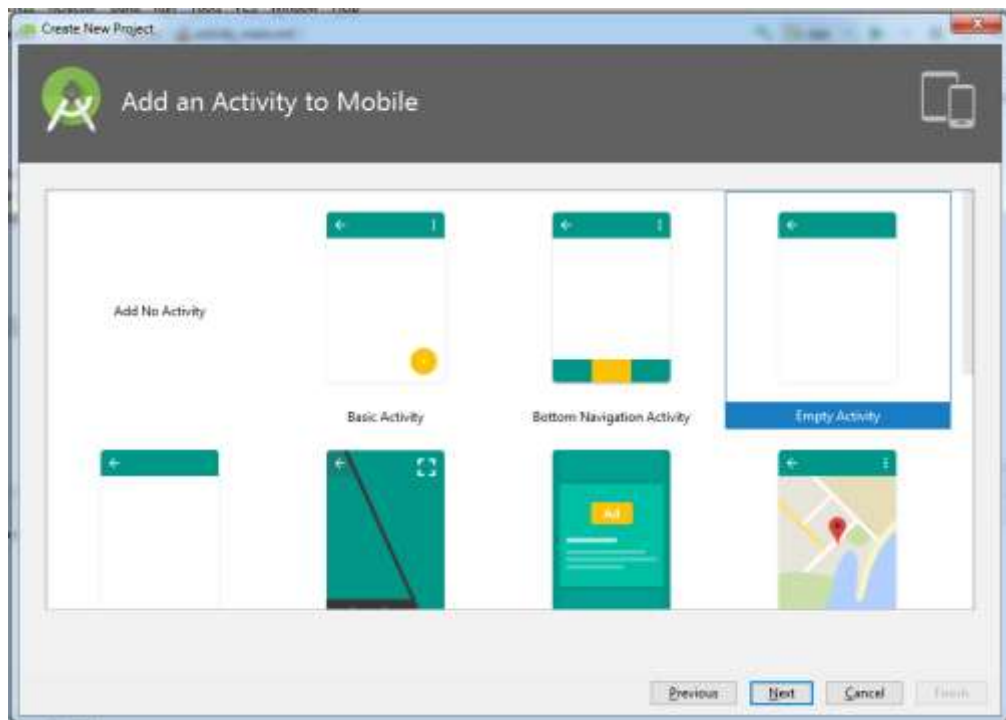
Gambar 1.3 Tampilan jendela **Create Android Project**

7. Di layar **Target Android Devices**, "Phone and Tablet" harus dipilih. Dan Anda harus memastikan bahwa API 15: Android 4.0.3 IceCreamSandwich disetel sebagai **Minimum SDK**. (Perbaiki jika perlu.)
- 7.1. Saat buku ini ditulis, memilih tingkat API ini menjadikan aplikasi "Hello World" kompatibel dengan 97% perangkat Android yang aktif di Google Play store.
- 7.2. Ini adalah setelan yang digunakan oleh contoh di buku ini.



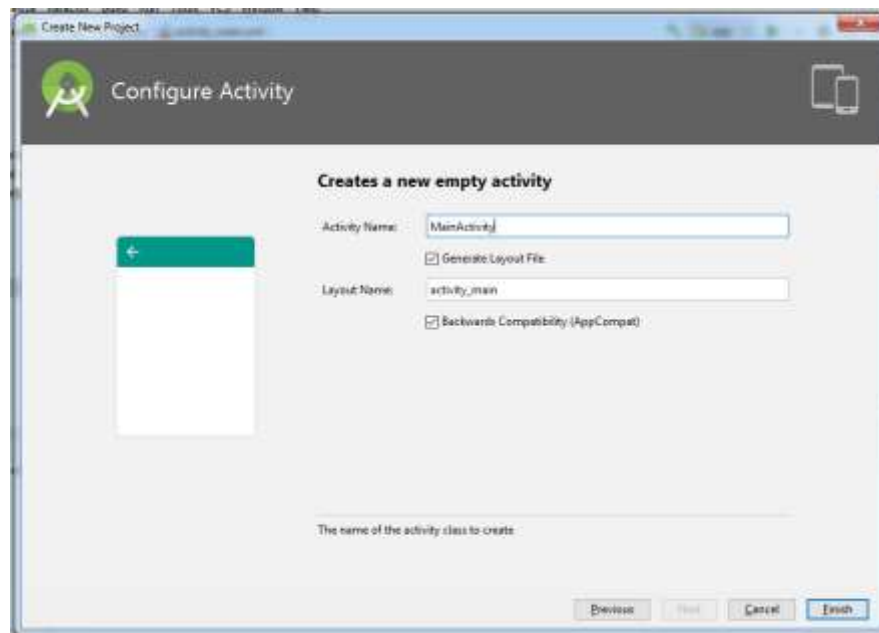
Gambar 1.4 Tampilan jendela **Target Android Devices**

8. Klik **Next**.
9. Jika proyek Anda memerlukan komponen tambahan untuk SDK target terpilih, Android Studio akan memasangnya secara otomatis. Klik **Next**.
10. Jendela **Add an Activity to Mobile**. Setiap aplikasi membutuhkan paling tidak satu aktivitas. Aktivitas mewakili satu layar dengan satu antarmuka pengguna dan Android Studio menyediakan template untuk membantu Anda memulai. Untuk proyek Hello World, pilih template termudah (saat ini ditulis, template proyek "Empty Activity" adalah template paling sederhana) yang tersedia.

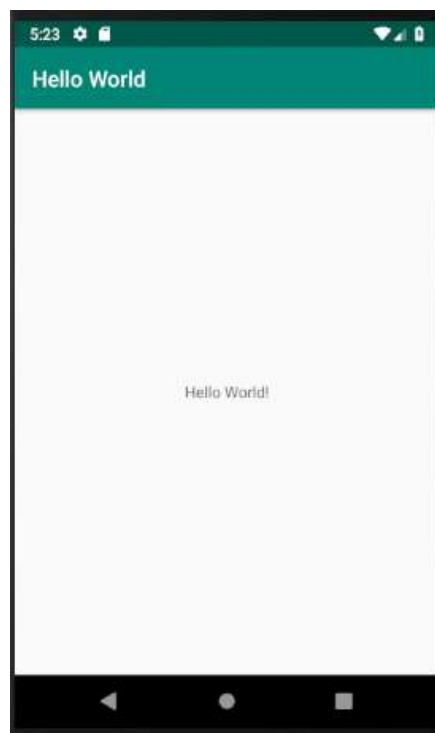


Gambar 1.5 Tampilan jendela **Add an Activity to Mobile**

11. Activity name pada jendela **Configure Activity** isi dengan MainActivity. Ini bukan persyaratan.
12. Pastikan kotak **Generate Layout file** dicentang (jika terlihat).
13. Pastikan kotak **Backwards Compatibility (App Compat)** dicentang.
14. Biarkan **Layout Name** sebagai activity_main. Layout biasanya diberi nama sesuai dengan aktivitasnya. Terima defaultnya dan klik **Finish**.



Gambar 1.6 Tampilan jendela **Configure Activity**



Gambar 1.7 Tampilan aplikasi **Hello World**

Latihan 2 : Jelajahi struktur proyek dan layout

Langkah-langkah ini mengasumsikan bahwa proyek Hello World Anda dimulai seperti yang ditampilkan pada diagram di atas.

1. Luaskan folder **manifests**.

Folder ini berisi **AndroidManifest.xml**. File ini menjelaskan semua komponen aplikasi Android Anda dan dibaca oleh sistem waktu proses Android saat program dijalankan.

2. Luaskan folder **java**. Semua file bahasa Java dikelola di folder ini. Folder **java** berisi tiga subfolder:

- **com.example.hello.helloworld (atau nama domain yang telah Anda tetapkan):** Semua file untuk paket ada di folder yang bernama sama dengan paket tersebut. Untuk aplikasi Hello World, ada satu paket dan paket berisi MainActivity.java (ekstensi file dapat dihilangkan di tampilan Proyek).
- **com.example.hello.helloworld(androidTest):** Folder ini untuk pengujian berinstrumen Anda, dan memulai dengan file pengujian kerangka.
- **com.example.hello.helloworld(test):** Folder ini untuk pengujian unit dan memulai dengan file pengujian unit kerangka yang secara otomatis dibuat.

3. Luaskan folder **res**. Folder ini berisi semua sumber daya untuk aplikasi Anda, termasuk gambar, file layout, string, ikon, dan penataan gaya. Folder ini berisi subfolder berikut:

- **drawable.** Simpan semua gambar aplikasi Anda di folder ini.
- **layout.** Setiap aktivitas memiliki paling tidak satu file layout yang menjelaskan UI di XML. Untuk Hello World, folder ini berisi activity_main.xml.
- **mipmap.** Simpan ikon peluncur Anda di folder ini. Ada subfolder untuk setiap kepadatan layar yang didukung. Android menggunakan kepadatan layar, yaitu, jumlah piksel per inci untuk menentukan resolusi gambar

yang diperlukan. Android mengelompokkan kepadatan layar yang sebenarnya ke dalam kepadatan umum seperti medium (mdpi), tinggi (hdpi), atau ekstra-ekstra-ekstra-tinggi (xxxhdpi). Folder `ic_launcher.png` berisi ikon peluncur default untuk semua kepadatan yang didukung oleh aplikasi Anda.

- **values.** Sebagai ganti melakukan hardcode nilai-nilai seperti string, dimensi, dan warna di file XML dan Java, praktik terbaiknya adalah mendefinisikannya menurut file nilai. Ini membuatnya lebih mudah diubah dan konsisten di seluruh aplikasi Anda.
4. Luaskan subfolder **values** dalam folder `res`. Folder ini berisi subfolder berikut:
- **colors.xml.** Tunjukkan warna default untuk tema pilihan Anda, dan Anda bisa menambahkan warna sendiri atau mengubahnya berdasarkan persyaratan aplikasi Anda.
 - **dimens.xml.** Simpan ukuran tampilan dan objek untuk resolusi yang berbeda.
 - **strings.xml.** Buat sumber daya untuk semua string Anda. Ini memudahkan penerjemahan string ke bahasa lain.
 - **styles.xml.** Semua gaya untuk aplikasi dan tema Anda ada di sini. Gaya membantu memberikan aplikasi Anda tampilan yang konsisten untuk semua elemen UI.

Latihan 3: Membuat perangkat virtual

1. Di Android Studio, pilih Tools > Android > AVD Manager atau klik ikon



AVD Manager di bilah alat.


2. Klik **+Create Virtual Device...** (Jika Anda telah membuat perangkat virtual sebelumnya, jendela akan menunjukkan semua perangkat yang ada dan tombolnya ada di bagian bawah.)

Layar Select Hardware muncul menunjukkan daftar perangkat keras yang telah dikonfigurasi sebelumnya. Untuk setiap perangkat, tabel menunjukkan ukuran tampilan (Size), resolusi layar dalam piksel (Resolution), dan kepadatan piksel (Density).

Untuk perangkat Nexus 5, kepadatan pikselnya adalah xxhdpi, yang berarti aplikasi Anda menggunakan ikon peluncur di folder xxhdpi dari folder mipmap. Aplikasi Anda juga akan menggunakan layout dan drawable dari folder yang didefinisikan untuk kepadatan itu juga.

3. Pilih perangkat keras Nexus 5 dan klik **Next**.
4. Pada layar **System Image**, dari tab **Recommended**, pilih versi sistem Android yang akan dijalankan pada perangkat virtual. Anda bisa memilih gambar sistem terbaru.
Ada lebih banyak versi yang tersedia dari yang ditunjukkan di tab **Recommended**. Lihat tab **x86 Images** dan **Other Images** untuk melihatnya.
5. Jika tautan **Download** terlihat di samping versi gambar sistem, berarti versi tersebut belum terpasang dan Anda perlu mengunduhnya. Jika perlu, klik tautan untuk mulai mengunduh dan klik **Finish** setelah selesai.
6. Pada layar **System Image**, pilih gambar sistem dan klik **Next**.
7. Verifikasi konfigurasi Anda dan klik **Finish**. (Jika jendela AVD Manager **Your Android Devices** tetap terbuka, Anda bisa melanjutkan dan menutupnya.)

Latihan 4: Menjalankan aplikasi di emulator

1. Di Android Studio, pilih **Run > Run app** atau klik ikon **Run**  di bilah alat.
2. Di jendela **Select Deployment Target**, di bawah **Available Emulators**, pilih **Nexus 5 API 23** dan klik **OK**.

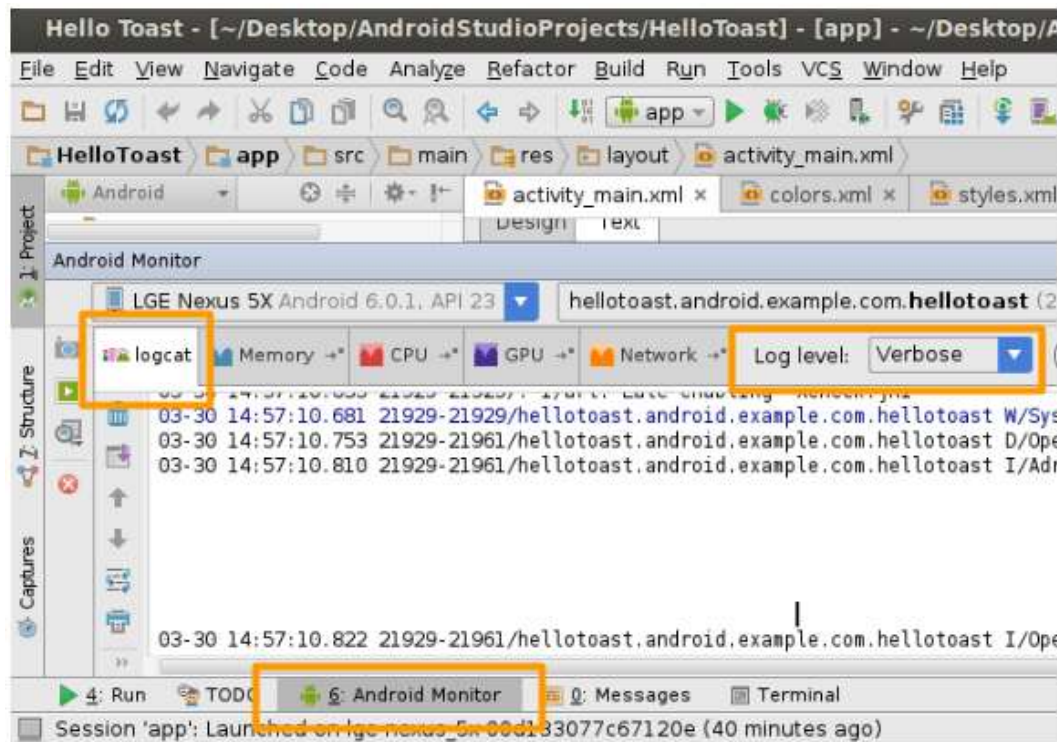
Latihan 5 : Menambahkan pernyataan log ke aplikasi Anda

1. Klik tombol **Android Monitor** di bawah Android Studio untuk membuka Android Monitor.

Secara default, ini membuka tab **logcat**, yang menampilkan informasi tentang aplikasi Anda saat sedang dijalankan. Jika Anda menambahkan pernyataan log ke aplikasi, pernyataan dicetak di sini juga.

Anda juga bisa memantau Memori, CPU, GPU, dan kinerja jaringan aplikasi dari tab lain Android Monitor. Ini dapat membantu debug dan kinerja menyempurnakan kode.

2. Level log default adalah **Verbose**. Di menu tarik-turun, ubah level log ke **Debug**.



Gambar 1.8 Android Monitor

Pernyataan log yang Anda tambahkan ke kode aplikasi mencetak pesan yang ditentukan oleh Anda di tab logcat Android Monitor. Misalnya:

```
Log.d("MainActivity", "Hello World");
```

Bagian dari pesan tersebut adalah:

- Log – [Kelas Log](#). API untuk mengirim pesan log.
- d – Tingkat Log. Digunakan untuk memfilter tampilan pesan di logcat. "d" untuk debug. Tingkat log lainnya adalah "e" untuk error, "w" untuk warning, dan "i" untuk info.
- "MainActivity" – Argumen pertama adalah tag yang dapat digunakan untuk memfilter pesan di logcat. Ini biasanya adalah nama aktivitas asal pesan. Tetapi, Anda dapat menjadikannya apa saja yang berguna untuk men-debug.
- "Hello world" – Argumen kedua adalah pesan yang sebenarnya.

Latihan 6: Jelajahi file AndroidManifest.xml

1. Buka aplikasi Hello World di Android studio, dan di folder **manifests**, buka **AndroidManifest.xml**.
2. Baca file dan pertimbangkan apa yang ditunjukkan oleh setiap kode. Kode di bawah ini dianotasi untuk memberi Anda petunjuk.

Kode:

```
<!-- XML version and character encoding -->
<?xml version="1.0" encoding="utf-8"?>
<!-- Required starting tag for the manifest -->
<manifest
  <!-- Defines the android namespace. Do not change. -->
  xmlns:android="http://schemas.android.com/apk/res/android"
  <!-- Unique package name of your app. Do not change once app is
    published. -->
    package="com.example.hello.helloworld">
    <!-- Required application tag -->
    <application
      <!-- Allow the application to be backed up and restored. ->
      android:allowBackup="true"
      <!-- Icon for the application as a whole,
        and default icon for application components. ->
        android:icon="@mipmap/ic_launcher"
      <!-- User-readable for the application as a whole,
        and default icon for application components. Notice that
        Android
        Studio first shows the actual label "Hello World".
        Click on it, and you will see that the code actually refers
        to a string
        resource. Ctrl-click @string/app_name to see where the
        resource is
        specified. This will be covered in a later practical . ->
        android:label="@string/app_name"
      <!-- Whether the app is willing to support right-to-left
        layouts.->
        android:supportsRtl="true"
      <!-- Default theme for styling all activities. ->
      android:theme="@style/AppTheme">
      <!-- Declares an activity. One is required.
        All activities must be declared,
        otherwise the system cannot see and run them. ->
        <activity
          <!-- Name of the class that implements the activity;
            subclass of Activity. ->
            android:name=".MainActivity">
          <!-- Specifies the intents that this activity can respond
            to.->
```

```

        <intent-filter>
            <!-- The action and category together determine what
                happens when the activity is launched. ->
            <!-- Start activity as the main entry point.
                Does not receive data. ->
            <action android:name="android.intent.action.MAIN" />
            <!-- Start this activity as a top-level activity in
                the launcher . ->
            <category
android:name="android.intent.category.LAUNCHER" />
        <!-- Closing tags ->
        </intent-filter>
    </activity>
</application>
</manifest>

```

❖ TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Android?
2. Jelaskan sejarah perkembangan Android!

❖ TUGAS PRAKTIKUM

1. Buat proyek baru Android Studio
2. Ganti sapaan “Hello world!!!” dengan “Hello *nama Anda*”. Contoh: Hello Ilmu Komputer!!!
3. Ganti latar belakang aplikasi dengan gambar sesuai keinginan Anda.

Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu menggunakan editor layout Android Studio.
2. Mahasiswa mampu membuat beberapa varian layout pada Android Studio.

❖ DASAR TEORI

Android Studio menyediakan banyak jenis layout, seperti **Constraint Layout**, **Linear Layout**, **Table Layout**, **Guide Line**, **Frame Layout**. Layout default Android Studio saat buku ini dibuat adalah **Connstraint Layout**, dimana layout ini serupa dengan **Relative Layout** tapi lebih fleksibel. **Relative Layout** sendiri dapat ditemukan di bagian **Legacy**.



Gambar 2.1 Layout Android Studio

Linear Layout

Adalah jenis layout yang mengatur child element (contoh : TextView, Button, CheckBox, dll) ke dalam suatu baris horizontal ataupun vertikal. Untuk mengatur direction dari Linear Layout ini sendiri bisa mengedit pada atribut `android:orientation`.

Untuk lebih jelasnya sebagai contoh:

Misal kita mempunyai suatu text view dengan nama TextView1 dan TextView2. Ketika kita menuliskan kode XML nya kita tulis berurutan TextView1 ditulis pertama dan TextView2 setelahnya. Jika kita menggunakan `android:orientation="vertical"` maka TextView1 akan berada di atas karena ditulis pertama dan TextView2 berada di bawah TextView1 karena ditulis kedua.

Sedangkan jika kita menggunakan `android:orientation="horizontal"` maka TextView 1 akan berada di sebelah kiri dan TextView2 akan berada di sebelah kanan.

Layout sendiri memiliki suatu atribut "weight" yang mempengaruhi size atau besar dari element child yang ada di dalamnya terhadap layar dari suatu device.

Relative Layout

Relative Layout memiliki karakteristik menempatkan view secara relatif. Posisi dari setiap view bergantung kepada view yang lain. Mudahnya adalah, programmer memiliki kebebasan untuk menempatkan view yang diinginkan. Penempatan satu view dapat berupa di sisi kanan, kiri, atas, ataupun bawah dari view lain. Jika tidak di tetapkan di sisi mana view akan berada, maka dapat terjadi penumpukan antara satu view dengan view yang lain.

Table Layout

Table Layout memiliki karakteristik yang mirip dengan pembuatan layout di HTML. Jika anda familiar dengan pemrograman web berjenis html, `<table>` dan `<TableLayout />` adalah 2 hal yang hampir sama. TableLayout mengatur tampilan berdasarkan kolom dan baris. Dan jangan lupa untuk selalu menambahkan tag `<TableRow />` jika ingin membuat baris baru dalam layout.

Frame Layout

Grup tampilan anak bertumpuk. FrameLayout didesain untuk memblokir area di layar guna menampilkan satu tampilan. Tampilan anak digambar bertumpuk, dengan anak yang baru saja ditambahkan di atas. Ukuran FrameLayout adalah ukuran tampilan anak terbesarnya.

Guide Line

Guide line adalah garis yang digunakana biasanya pada **Constraint Layout** untuk membantu dalam menyusun item-item di layout

Grid View

GridView merupakan serangkaian item-item atau daftar yang tersusun secara vertikal dan horizontal yang dapat ditampilkan secara keseluruhan seperti bentuk kotak. Adapun contoh nya seperti Kalender yang anda pakai. Nah didalam GridView ini diisi menggunakan Array ataupun Query Database, sama layaknya dengan ListView. GridView ini banyak dipakai saat kita membuat sebuah project dengan daftar seperti kotak-kotak. Pasti anda tahu bagaimana bentuknya. Pada Tutorial kali ini kita akan membuat sebuah masih mengenai daftar bulan dalam bentuk GridView.



Gambar 2.2 Guide Line, garis putus-putus melintang (horizontal) dan tegak (vertical)

❖ LATIHAN PRAKTIKUM

Latihan 1: Linear Layout

3. Buat project baru dengan nama Layout
4. Masukkan kode berikut ke activity_main.xml

Kode activity_main.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.hades.layout.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent">
```



```

        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:id="@+id/text1"/>
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="click"
    android:id="@+id/button1"/>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:weightSum="2">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Hello World!"
        android:id="@+id/text2"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="click"
        android:id="@+id/button2"/>
</LinearLayout>
</LinearLayout>

```

Hasil yang diperoleh:



Gambar 2.3 Linear Layout

Catatan :

Coba hilangkan sintaks

`android:layout_weight="1"`

pada button 2, perhatikan perbedaan tampilan dengan sintaks sebelumnya, khususnya pada tampilan landscape.

Latihan 2: Relative Layout

Kode activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.hades.layout.MainActivity">

    <TextView
        android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="0dp"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="0dp"
        android:layout_marginTop="41dp"
        android:text="click" />

</RelativeLayout>
```

Hasilnya :



Gambar 2.4 Relative Layout

Latihan 3: Table Layout

Kode activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout 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:orientation="vertical"
    tools:context="com.example.hades.layout.MainActivity">

    <TableRow>
        <TextView
            android:id="@+id/text2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:gravity="center"
            android:background="#dcdcdc"
            android:text="Hello World!"/>
    </TableRow>

    <TableRow>
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:weightSum="2">

        <TextView
            android:id="@+id/text1"
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:background="#0000ff"
            android:layout_alignParentStart="true"
            android:layout_alignParentTop="true"
            android:layout_marginStart="0dp"
            android:layout_marginTop="0dp"
            android:text="Hello World!"
            android:layout_weight="1"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintLeft_toLeftOf="parent"
            app:layout_constraintRight_toRightOf="parent"
            app:layout_constraintTop_toTopOf="parent" />

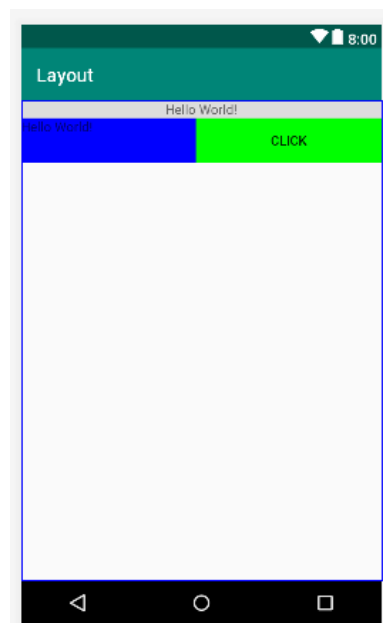
        <Button
            android:id="@+id/button1"
            android:layout_weight="1"
            android:background="#00ff00"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:text="click" />

    </TableRow>

</TableLayout>

```

Hasilnya:



Gambar 2.5 Table Layout

Latihan 4: Frame Layout

Kode activity_main.xml

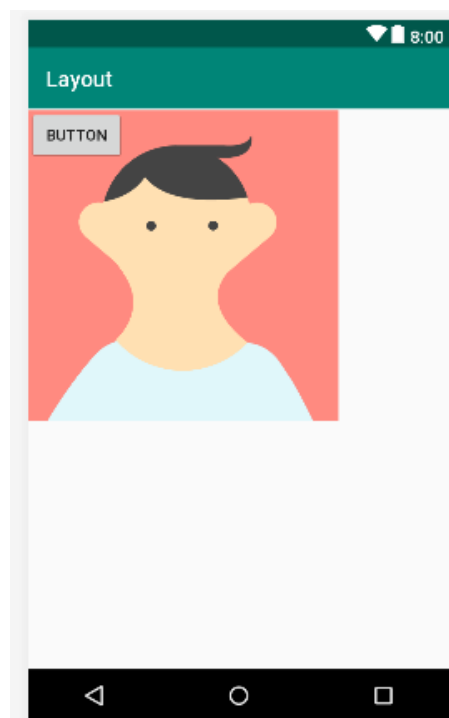
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.hades.layout.MainActivity">

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

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="283dp"
        android:layout_height="286dp"
        tools:srcCompat="@tools:sample/avatars[0]" />

</FrameLayout>
```

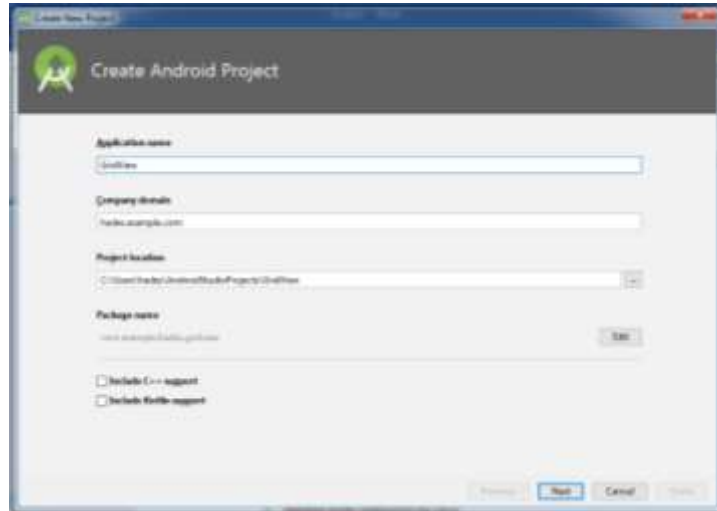
Hasilnya



Gambar 2.6 Frame Layout

Latihan 5: Grid View

1. Buat project baru dengan parameter



Gambar 2.7 Create Android Project

2. Berikut code untuk activity_main.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.hades.helloworld.MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Bulan"
        android:textSize="20sp"
        android:id="@+id/text1"
        android:gravity="center"/>

    <GridView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_below="@id/text1"
        android:stretchMode="columnWidth"
        android:id="@+id/grid1">

    </GridView>

</RelativeLayout>
```

3. Berikut kode MainActivity.java

```
package com.example.hades.gridview;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.GridView;

public class MainActivity extends AppCompatActivity {

    private String[] bulan = {"Januari", "Februari", "Maret", "April",
        "Mei", "Juni",
        "Juli", "Agustus", "September", "Oktober", "November", "Desember"};
    private GridView grid1;
    private ArrayAdapter<String> adapter;

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

        grid1 = (GridView) findViewById(R.id.grid1);

        adapter = new ArrayAdapter<String>(MainActivity.this,
            android.R.layout.simple_list_item_1, bulan);

        grid1.setAdapter(adapter);
    }
}
```



Gambar 2.8 Grid View

4. Tambahkan

```
android:numColumns="auto_fit"
```

pada GridView di activity_main.xml, maka hasilnya



Gambar 2.9 Grid View auto fit

❖ TUGAS PENDAHULUAN

Jelaskan apa yang dimaksud dengan layout?

❖ TUGAS PRAKTIKUM

Buat proyek baru dengan 5 tampilan. Buat agar satu tampilan menggunakan setengah atas layar, dan tampilan 4 lainnya berbagi setengah bawah layar. Hanya gunakan LinearLayout, gravitasi, dan berat untuk melakukannya.

Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu menggunakan antarmuka pengguna interaktif dalam Layout Editor, dalam XML, dan secara terprogram.

❖ DASAR TEORI

Antarmuka pengguna yang di pada layar perangkat seluler Android terdiri dari hierarki "tampilan". Tampilan adalah blok pembangun antarmuka pengguna dasar. Anda menentukan tampilan dalam file layout XML. Misalnya, tampilan dapat berupa komponen yang:

- menampilkan teks (kelas TextView)
- mengizinkan Anda mengedit teks (kelas EditText)
- mewakili tombol yang dapat diklik (kelas Button) dan komponen interaktif lainnya
- berisi teks yang dapat digulir (ScrollView) dan item yang dapat digulir (RecyclerView)
- menampilkan gambar (ImageView)
- menampung tampilan lain dan memosisikannya (LinearLayout).
- memunculkan menu dan komponen interaktif lainnya.

Anda bisa menjelajahi hierarki tampilan aplikasi dalam panel Component Tree Layout Editor.

Kode Java yang menampilkan dan menjalankan antarmuka pengguna berada dalam kelas yang memperluas Aktivitas dan berisi metode untuk mengembangkan tampilan, yaitu, mengambil layout XML tampilan dan menampilkannya di layar. Misalnya, MainActivity pada aplikasi Hello World mengembangkan tampilan teks dan mencetak Hello World. Pada aplikasi yang lebih kompleks, aktivitas dapat

mengimplementasikan klik dan handler kejadian lainnya, meminta data dari database atau internet, atau menggambar konten grafis.

Android memudahkan proses pemisahan elemen UI dan data dari satu sama lain, dan menggunakan aktivitas tersebut untuk menyatukannya. Pemisahan ini adalah implementasi dari pola MVP (Model-View-Presenter).

❖ LATIHAN PRAKTIKUM

Latihan 1: Aplikasi Hello Toast

1. Buat proyek baru yang bernama **Hello Toast**, dan pilih template Empty Activity.
2. Buka file layout **activity_main.xml** untuk mengedit kode 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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button_toast"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hello Toast"
        android:background="#3F51B5"
        android:onClick="showToast"
        android:textColor="@android:color/white" />

    <TextView
        android:id="@+id/show_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="0"
        android:textSize="160sp"
        android:textStyle="bold"
        android:textColor="#3F51B5"
        android:background="#FFF043"
        android:layout_weight="2" />

    <Button
        android:id="@+id/button_count"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Count"
        android:background="#3F51B5"
        android:onClick="countUp"
```

```
        android:textColor="@android:color/white" />
    </LinearLayout>
```

3. Edit MainActivity.java

```
package com.example.hades.hellotoast;

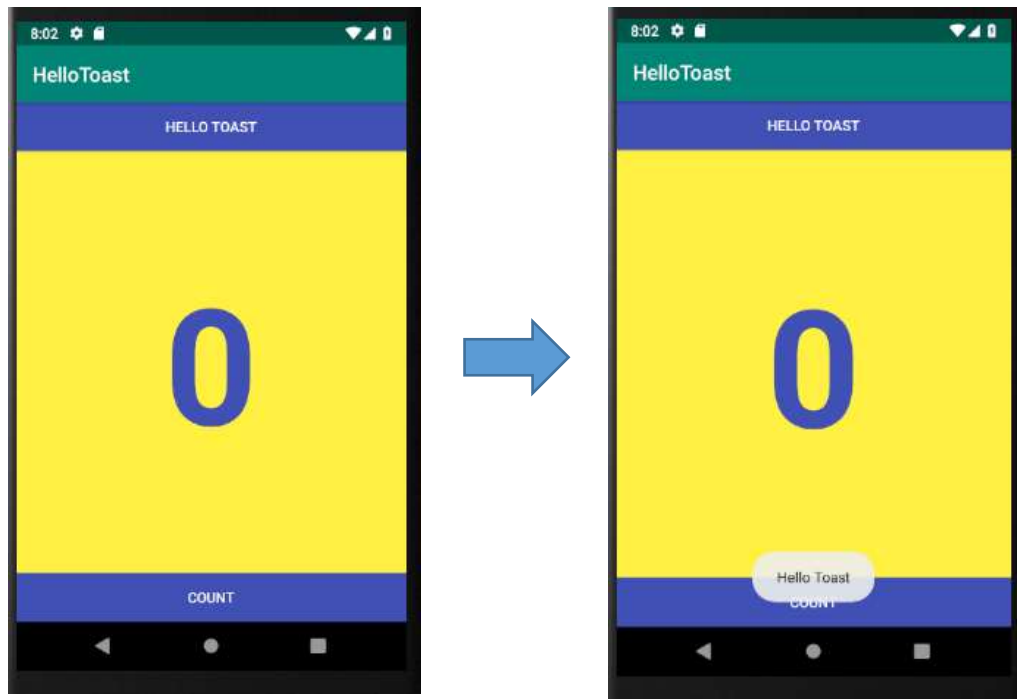
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private int mCount = 0;
    private TextView mShowCount;

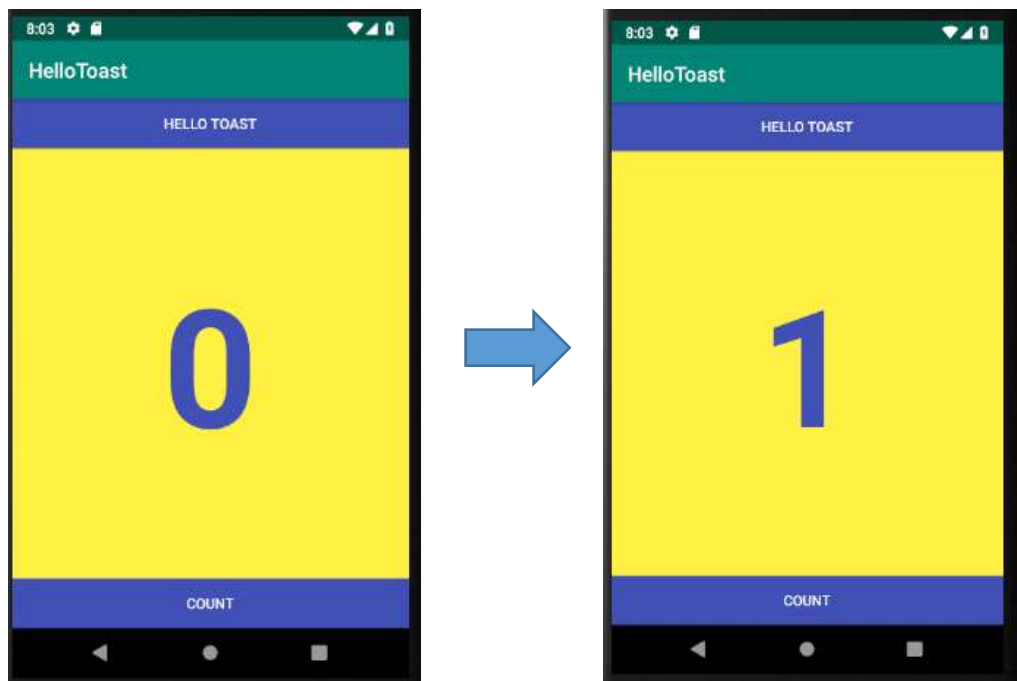
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mShowCount = (TextView) findViewById(R.id.show_count);
    }

    public void countUp(View view) {
        mCount++;
        if (mShowCount != null)
            mShowCount.setText(Integer.toString(mCount));
    }

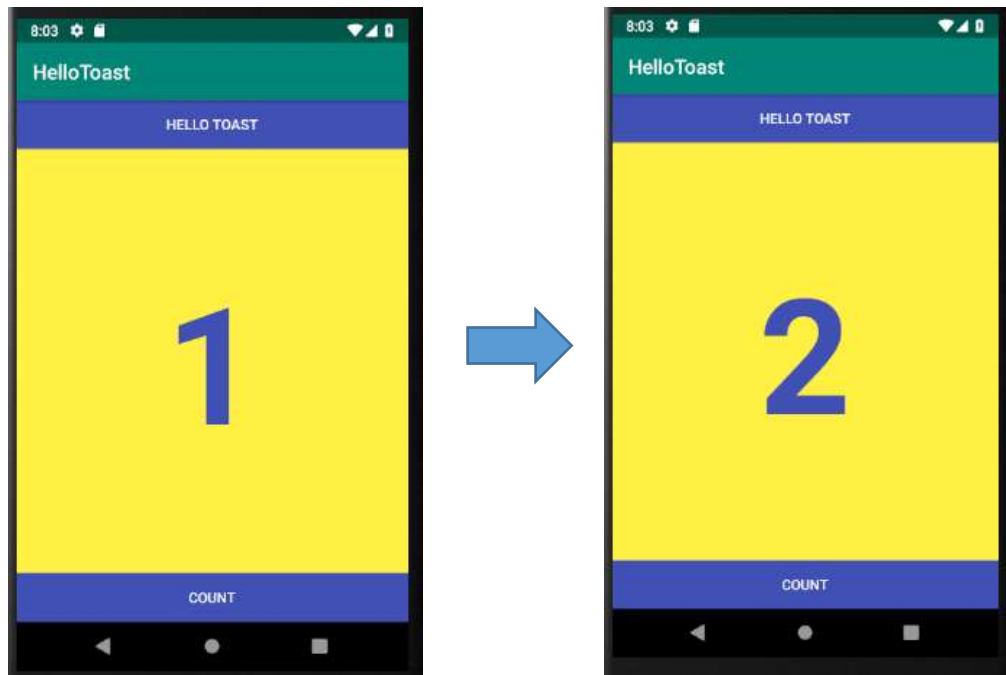
    public void showToast(View view) {
        Toast toast = Toast.makeText(this, "Hello Toast",
        Toast.LENGTH_LONG);
        toast.show();
    }
}
```



Gambar 2.1 Tombol Hello Toast ditekan



Gambar 2.2 Tombol Count ditekan



Gambar 2.3 Tombol Count ditekan

Latihan 2 : Mengubah Tampilan Text View

1. Buat proyek baru yang bernama **Keyboard Samples**, dan pilih template Empty Activity.
2. Buka file layout **activity_main.xml** untuk mengedit kode XML.

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

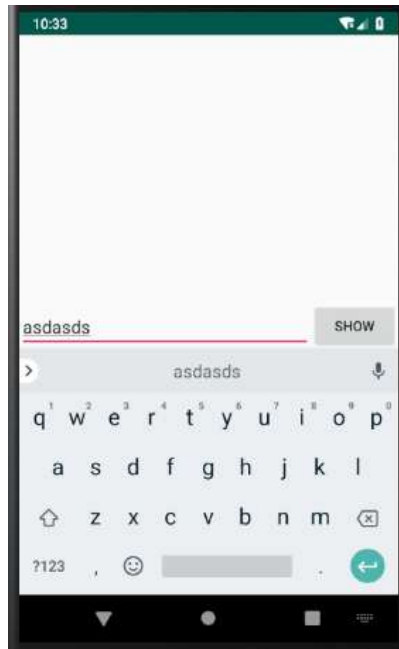
    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="showText"
        android:text="Show" />

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_toLeftOf="@id/button_main"
```

```

        android:hint="Masukkan text" />
    </RelativeLayout>

```



Gambar 2.4 Tipe input text normal

3. Ketika InputType pada TextView diubah ke bentuk password

```

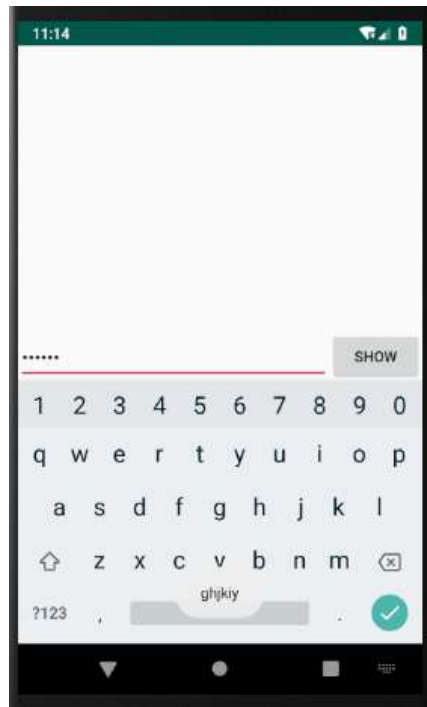
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"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

    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="showText"
        android:text="Show" />

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:inputType="textPassword"
        android:layout_toLeftOf="@id/button_main"
        android:hint="Masukkan text" />

</RelativeLayout>

```



Gambar 2.5 Tipe input text nerupa password

Perhatikan perbedaan tampilan TextView keduanya.

Latihan 3: Mengubah Tampilan Keyboard

3.1. Keyboard Email

Ubah InputType pada TextView diubah ke bentuk textEmailAddress

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

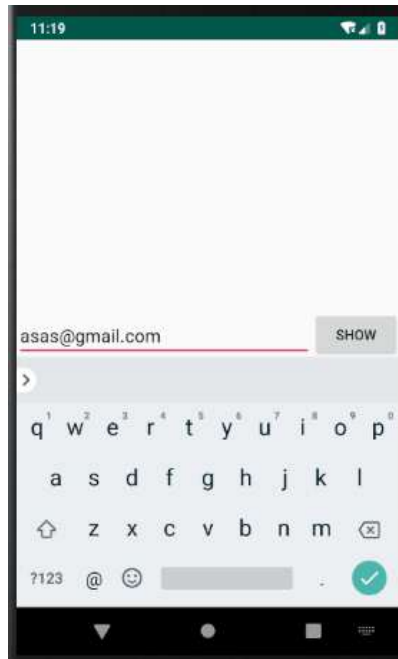
    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="showText"
        android:text="Show" />

    <EditText
        android:id="@+id/editText_main"
        android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:inputType="textEmailAddress"
        android:layout_toLeftOf="@id/button_main"
        android:hint="Masukkan text" />
    </RelativeLayout>

```



Gambar 2.6 Keyboard memudahkan input email

3.2. Keyboard Phone

Ubah InputType pada TextView diubah ke bentuk textEmailAddress

```

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

    <Button
        android:id="@+id/button_main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentRight="true"
        android:onClick="showText"

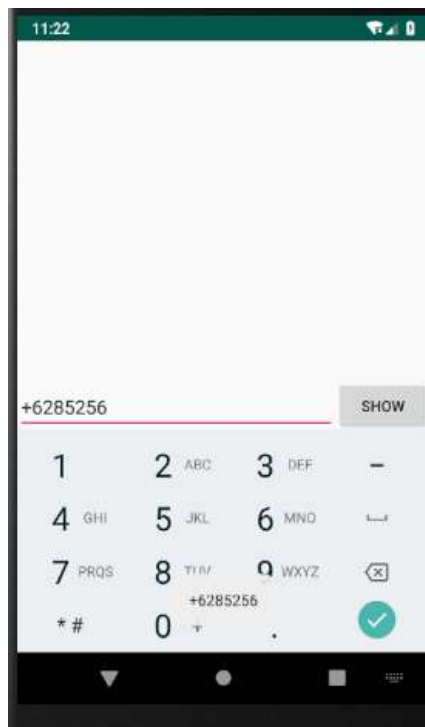
```



```

        android:text="Show" />
<EditText
    android:id="@+id/editText_main"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:inputType="phone"
    android:layout_toLeftOf="@id/button_main"
    android:hint="Masukkan text" />
</RelativeLayout>

```



Gambar 2.7 Keyboard angka

Perhatikan perbedaan tampilan Keyboard keduanya.

❖ TUGAS PENDAHULUAN

Jelaskan apa yang dimaksud dengan user interface.

❖ TUGAS PRAKTIKUM

Buatlah kalkulator sederhana

Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu memanggil halaman lain dalam satu aplikasi yang sama.
2. Mahasiswa mampu menjelaskan aktivitas dan intents pada Android

❖ DASAR TEORI

Sebuah aktivitas mewakili satu layar di aplikasi, yang digunakan pengguna untuk melakukan satu tugas terfokus seperti menelepon, mengambil foto, mengirim email, atau melihat peta. Aktivitas biasanya ditampilkan ke pengguna dalam jendela layar penuh.

Sebuah aplikasi biasanya terdiri atas beberapa aktivitas yang terikat secara longgar satu sama lain. Biasanya, satu aktivitas dalam aplikasi ditetapkan sebagai aktivitas "utama", yang ditampilkan kepada pengguna saat membuka aplikasi. Tiap aktivitas kemudian bisa memulai aktivitas lain untuk melakukan berbagai tindakan.

Tiap kali aktivitas baru dimulai, aktivitas sebelumnya akan dihentikan, namun sistem mempertahankan aktivitas dalam sebuah tumpukan ("back stack"). Saat sebuah aktivitas baru dimulai, aktivitas baru itu akan didorong ke atas back-stack dan mengambil fokus pengguna. Back-stack mematuhi mekanisme dasar tumpukan "masuk terakhir, keluar pertama", jadi jika pengguna selesai dengan aktivitas saat ini dan menekan tombol Back, aktivitas saat ini akan dikeluarkan dari tumpukan (dan dimusnahkan) dan aktivitas sebelumnya akan dilanjutkan.

Aktivitas Android dimulai atau diaktifkan dengan *intent*. Intent adalah pesan asinkron yang bisa Anda gunakan dalam aktivitas untuk meminta tindakan dari aktivitas lain (atau komponen aplikasi lainnya). Anda menggunakan intent untuk

memulai satu aktivitas dari aktivitas lainnya dan meneruskan data di antara aktivitas.

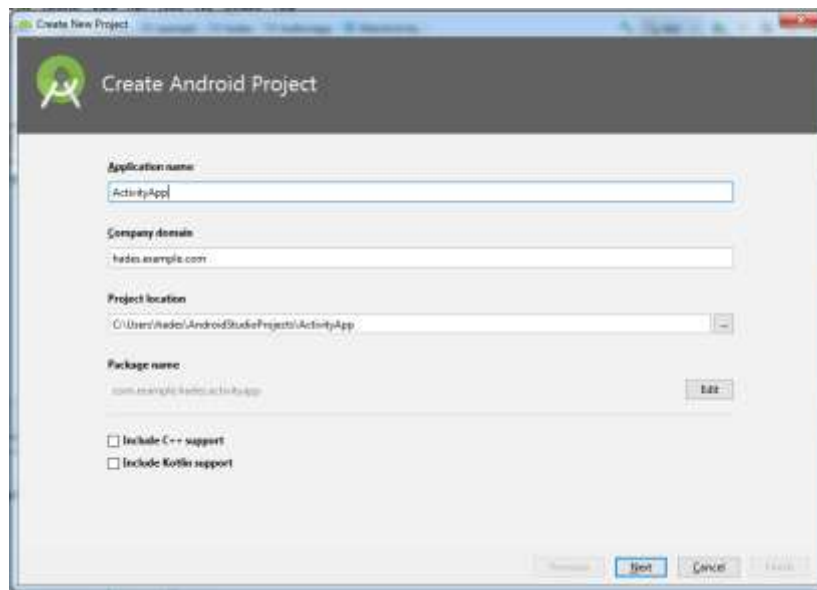
Ada dua jenis intent: *eksplisit* dan *implisit*. Intent eksplisit adalah intent yang targetnya Anda ketahui. Anda sudah mengetahui nama kelas yang sepenuhnya memenuhi syarat dari aktivitas spesifik tersebut. Intent implisit adalah intent yang nama komponen targetnya tidak Anda ketahui, namun memiliki tindakan umum untuk dikerjakan.

❖ LATIHAN PRAKTIKUM

Latihan 1 : Memanggil Halaman Lain

Sejauh ini kita belajar hanya menggunakan satu activity saja, nah sekarang saatnya kita melibatkan dua halaman saling memanggil. Siapkan dulu dua buah layout halaman satu dan dua masing-masing berisi satu Button, satu EditText dan satu TextView. Skenarionya, kita bisa menuliskan sebuah kata pada halaman satu. Kemudian jika button pada halaman satu diklik maka aplikasi akan berpindah ke halaman dua dengan menampilkan kata yang telah dimasukkan sebelumnya

1. Siapkan project baru dengan parameter sebagai berikut (nama aplikasi dan project sesuai selera masing-masing, tetapi perhatikan saat penyalinan kode)



Gambar 4.1 Jendela Create Android Project

2. Salinlah kode activity_main.xml seperti di bawah

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

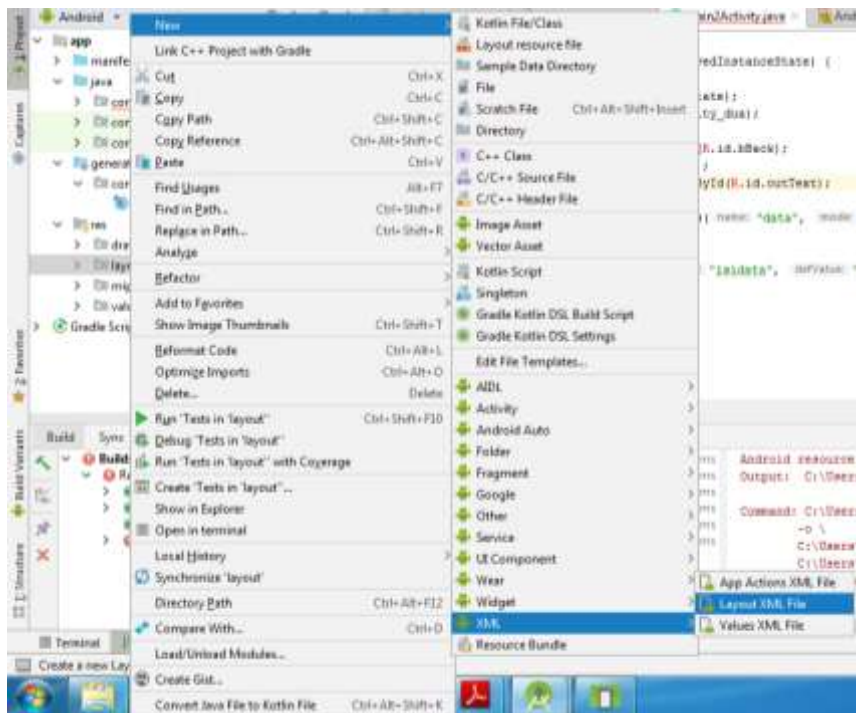
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Page-1" />

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

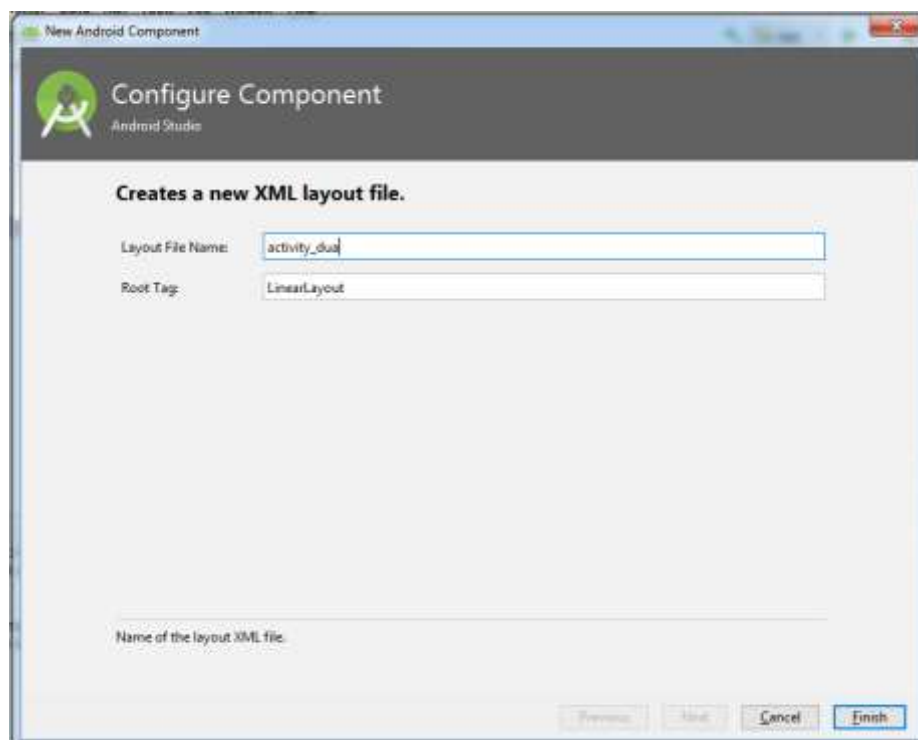
    <Button
        android:id="@+id/bNext"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Next" />

</LinearLayout>
```

3. Buat layout baru activity_dua.xml. Klik kanan folder res/layout > new > XML > Layout XML file.



Gambar 4.2 Buat layout baru



Gambar 4.3 Jendela new xml

4. Pada activity_dua.xml, tambahkan komponen yang sama dengan activity_main.xml. Ubah Textview menjadi tulisan “Page-2”, ganti background serta id pada widgetnya. Berikut code activity_dua.xml.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/colorAccent">

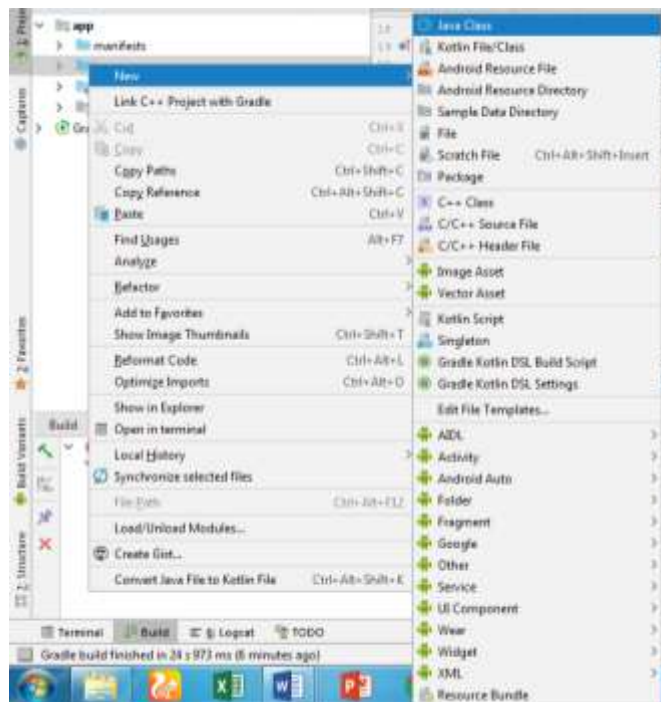
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Page-2" />

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

    <Button
        android:id="@+id/bBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Back" />

</LinearLayout>
```

5. Buat class baru DuaActivity.java. Caranya klik kanan package > new > Java class.



Gambar 4.4 Membuat class baru

6. Salin code MainActivity.java berikut.

```
package com.example.hades.activityapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener
{
    Button bNext;
    EditText inText;
    SharedPreferences sp;
    SharedPreferences.Editor spe;

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

        bNext = (Button) findViewById(R.id.bNext);
        bNext.setOnClickListener(this);
    }
}
```

```

        inText = (EditText) findViewById(R.id.inText);

        sp = this.getSharedPreferences("data", 0);
        spe = sp.edit();
    }
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub

        String input = inText.getText().toString();

        spe.putString("isidata", input);
        spe.commit();

        Intent i = new Intent(MainActivity.this, DuaActivity.class);
        startActivity(i);
    }
}

```

7. Salin kode DuaActivity.java berikut

```

package com.example.hades.activityapp;

import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class DuaActivity extends Activity implements OnClickListener {
    Button bBack;
    EditText outText;
    SharedPreferences sp;
    SharedPreferences.Editor spe;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dua);

        bBack = (Button) findViewById(R.id.bBack);
        bBack.setOnClickListener(this);
        outText = (EditText) findViewById(R.id.outText);

        sp = this.getSharedPreferences("data", 0);
        spe = sp.edit();

        String out = sp.getString("isidata", "");
        outText.setText(out);
    }

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        Intent i = new Intent(DuaActivity.this, MainActivity.class);
        startActivity(i);
    }
}

```


8. Daftarkan DuaActivity.java pada AndroidManifest > application (baris 19).
Berikut ini AndroidManifest keseluruhan.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.hades.activityapp">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

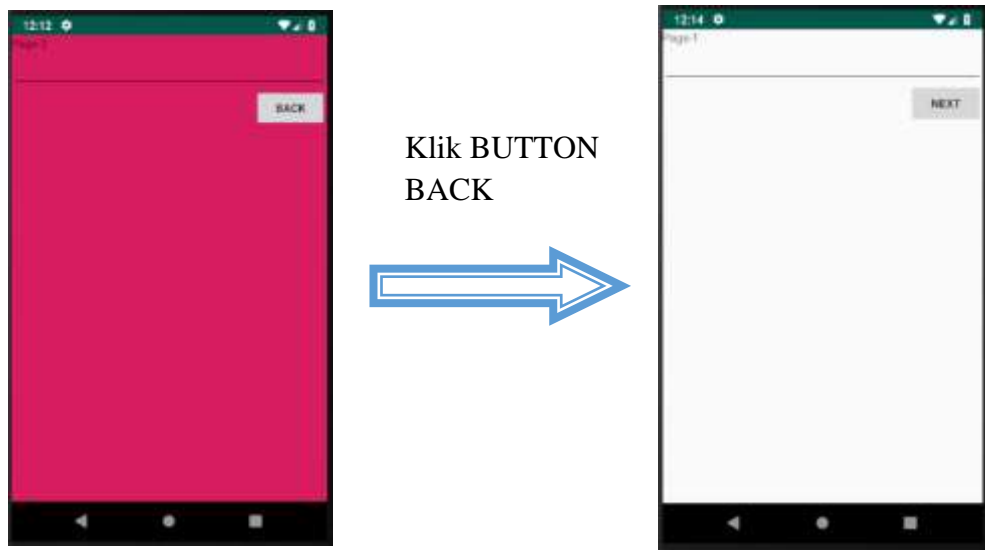
                <category
                    android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name="DuaActivity"></activity>
    </application>

</manifest>
```



Klik BUTTON
NEXT





Gambar 4.5 Memanggil halaman lain

❖ TUGAS PENDAHULUAN

Jelaskan perbedaan aktifitas dan intents!

❖ TUGAS PRAKTIKUM

Buat project baru. Apabila form ditekan yang muncul adalah layout sebelah kiri dan apabila biodata ditekan yang muncul adalah layout sebelah kanan

<div>Nama</div> <div>NIM</div> <div>TTL</div> <div>Biodata</div> <div>Form</div>	<div>Nama saya "Baco"</div> <div>NIM saya "H0711111"</div> <div>Tempat tanggal lahir saya "Maros, 3 Oktober 1991"</div> <div>Biodata</div> <div>Form</div>
--	--

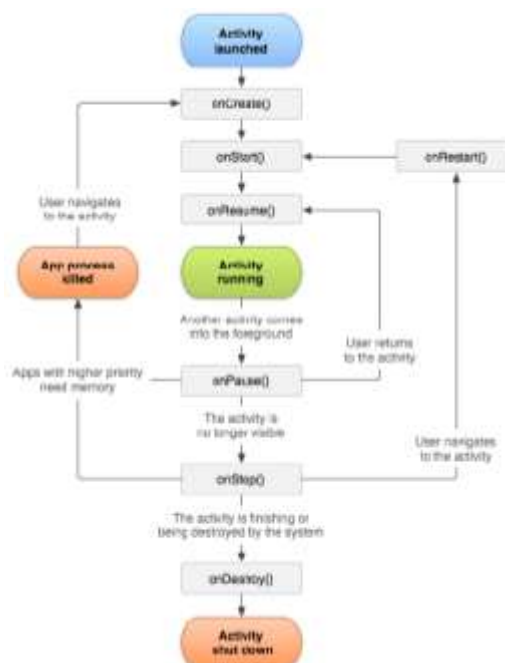
Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu menjelaskan siklus hidup aktivitas

❖ DASAR TEORI

Salah satu kelebihan smartphone adalah kehebatannya mengelola sendiri memori atau aktifitas prosesor. Dalam rangka mengatur penggunaan memori, aplikasi android mempunyai cara unik untuk mengendalikan status. Se jauh ini kita membuat aplikasi, pasti menjumpai method `onCreate()` pada `MainActivity.java`. Method `onCreate()` adalah salah satu method untuk mengontrol siklus hidup aplikasi, di eksekusi pertama kali saat aplikasi dijalankan. Masih ada method-method lain yang dieksekusi pada kondisi tertentu. Gambar berikut menjelaskan urutan terjadinya siklus hidup sebuah aplikasi, menunjukkan saat method-methodnya dieksekusi.

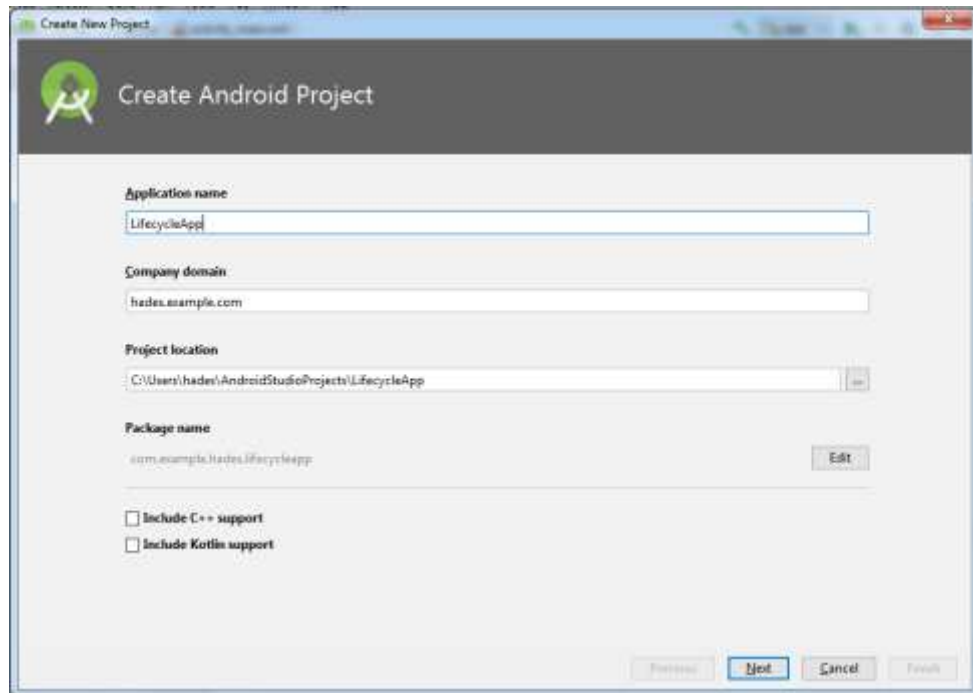


Gambar 5.1 Diagram siklus hidup aplikasi

❖ LATIHAN PRAKTIKUM

Latihan 1: Siklus Hidup Aplikasi

1. Buat project baru dengan parameter



Gambar 5.2 Create Android Project

2. Tambahkan method-method seperti code dibawah ini pada MainActivity.java

```
package com.example.hades.lifecycleapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.Toast;

public class MainActivity extends Activity {

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

        Toast.makeText(this, "onCreate executed",
            Toast.LENGTH_SHORT).show();
    }
}
```

```

@Override
protected void onPause() {
    // TODO Auto-generated method stub
    super.onPause();
    Toast.makeText(this, "onStop executed",
Toast.LENGTH_SHORT).show();
}

@Override
protected void onRestart() {
    // TODO Auto-generated method stub
    super.onRestart();
    Toast.makeText(this, "onRestart executed",
Toast.LENGTH_SHORT).show();
}

@Override
protected void onStart() {
    // TODO Auto-generated method stub
    super.onStart();
    Toast.makeText(this, "onRestart executed",
Toast.LENGTH_SHORT).show();
}

@Override
protected void onStop() {
    // TODO Auto-generated method stub
    super.onStop();
    Toast.makeText(this, "onStop executed",
Toast.LENGTH_SHORT).show();
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.layout.activity_main, menu);
    return true;
}

@Override
protected void onDestroy() {
    // TODO Auto-generated method stub
    super.onDestroy();
    Toast.makeText(this, "onDestroy executed",
Toast.LENGTH_SHORT).show();
}
}

```

3. Selesai dan jalankan, hasilnya tampak seperti Gambar

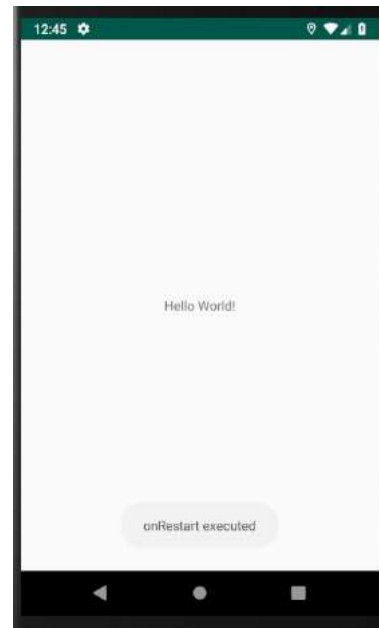
OnStop



OnDestroy



OnRestart



Gambar 5.3 Toast yang memperlihatkan siklus hidup aplikasi

Lakukan ujicoba dengan menjalankan aplikasi, memencet tombol Home atau memencet tombol Back secara acak atau bergantian. Toast akan muncul pada setiap method dieksekusi.

❖ TUGAS PENDAHULUAN

Apa yang dimaksud siklus hidup aktivitas?

❖ TUGAS PRAKTIKUM

Amati `onDestroy()` pada khususnya. Mengapa `onDestroy()` kadang dipanggil (setelah mengeklik tombol kembali, atau saat perangkat diputar) dan bukan yang lainnya (menghentikan dan memulai ulang aplikasi secara manual)?

Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu menggunakan fitur dialog pada Android Studio.

❖ DASAR TEORI**DIALOG**

Seringkali app perlu menyampaikan pesan sementara kepada pengguna, misalnya jika menampilkan informasi, melaporkan kesalahan atau memperlihatkan proses yang memakan waktu lama. Dialog adalah jendela kecil yang meminta pengguna untuk membuat keputusan atau memasukkan informasi tambahan. Dialog tidak mengisi layar dan biasanya digunakan untuk kejadian modal yang mengharuskan pengguna untuk melakukan aksi sebelum bisa melanjutkan.

TOAST

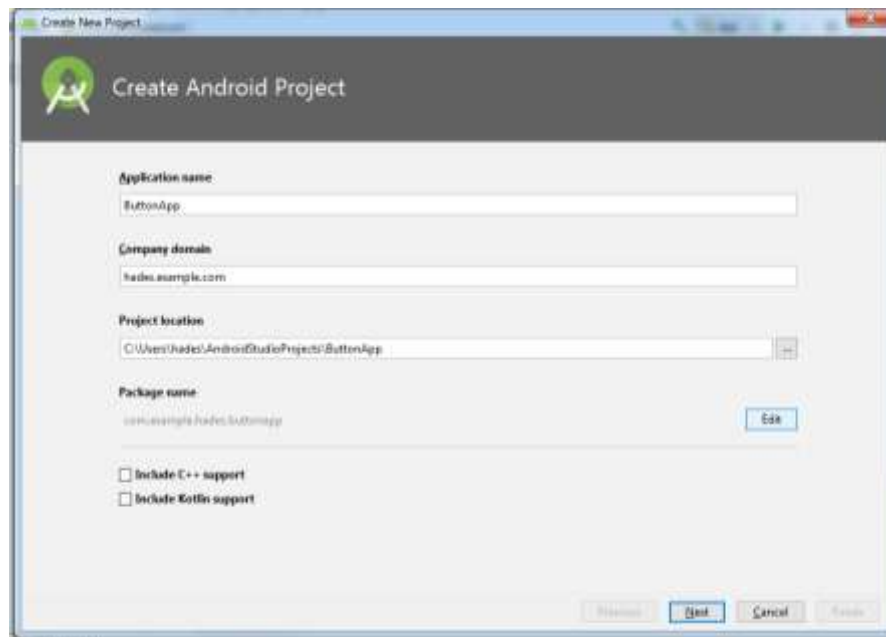
Toast digunakan untuk menampilkan pesan yang kemudian akan menghilang dengan sendirinya tanpa interaksi dengan pengguna. Fokus juga tetap pada aplikasi dan tidak berpindah ke Toast. Kelebihan Toast adalah mudah untuk dibuat, sedangkan kelemahannya adalah pengguna dapat saja tidak melihat pesan yang dikandung Toast.

ALERT

Jika yang diinginkan adalah dialog yang muncul dan harus ditutup secara manual oleh pengguna maka dapat digunakan AlertDialog.

❖ LATIHAN PRAKTIKUM**Latihan 1: Dialog**

1. Siapkan project baru dengan parameter sebagai berikut (nama aplikasi dan project sesuai selera masing-masing, tetapi perhatikan saat penyalinan kode)



Gambar 6.1 Create Android Project

2. Salinlah kode activity_main.xml seperti di bawah

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/LinearLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center_horizontal"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="hello_world" />

    <EditText
        android:id="@+id/inName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Masukkan Nama Anda"
        android:ems="10" />

    <Button
        android:id="@+id/bSatu"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button Toast" />

    <Button
        android:id="@+id/bDua"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button Ambil Text" />

<Button
    android:id="@+id/bTiga"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button Dialog" />

<Button
    android:id="@+id/bEmpat"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button Exit" />

</LinearLayout>

```

3. Salinlah kode MainActivity.java di bawah

```

package com.example.hades.buttonapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity implements OnClickListener{
    Button btnSatu, btnDua, btnTiga, btnEmpat;
    EditText nama;
    String tempNama;

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

        btnSatu = (Button) findViewById(R.id.bSatu);
        btnSatu.setOnClickListener(this);

        btnDua = (Button) findViewById(R.id.bDua);
        btnDua.setOnClickListener(this);

        btnTiga = (Button) findViewById(R.id.bTiga);
        btnTiga.setOnClickListener(this);

        btnEmpat = (Button) findViewById(R.id.bEmpat);
        btnEmpat.setOnClickListener(this);

        nama = (EditText) findViewById(R.id.inName);
    }

    @Override

```

```

    public void onClick(View v) {
// TODO Auto-generated method stub
        switch (v.getId()) {
            case R.id.bSatu:
                Toast.makeText(this, "Tombol toast terpencet",
                    Toast.LENGTH_SHORT).show();
                break;

            case R.id.bDua:
                tempNama = nama.getText().toString();
                Toast.makeText(this, "nama saya : " + tempNama,
                    Toast.LENGTH_SHORT).show();
                break;

            case R.id.bTiga:
                AlertDialog.Builder ad = new
AlertDialog.Builder(this);
                ad.setTitle("PERHATIAN");
                ad.setMessage("Tombol Dialog dipencet");
                ad.show();
                break;

            case R.id.bEmpat:
                AlertDialog.Builder d = new AlertDialog.Builder(this);
                d.setTitle("PERHATIAN");
                d.setMessage("Mau keluar?");
                d.setPositiveButton("yes", new
                    DialogInterface.OnClickListener() {

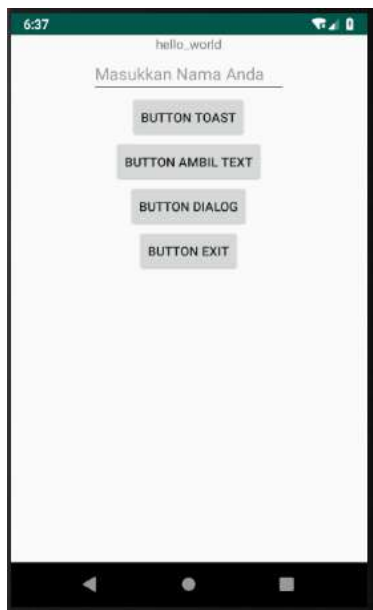
                        @Override
                        public void onClick(DialogInterface
dialog, int

                            which) {
                                // TODO Auto-generated method stub
                                finish();
                            }
                    });
                d.setNegativeButton("no", new
                    DialogInterface.OnClickListener() {

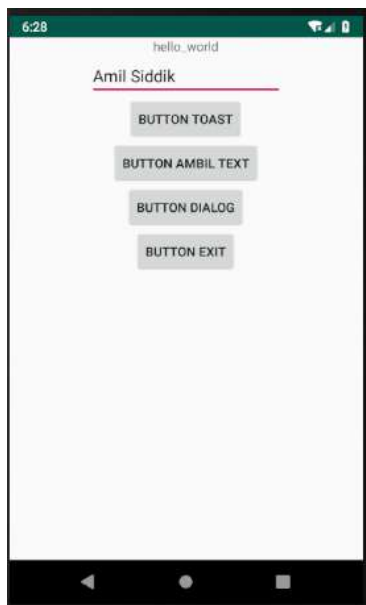
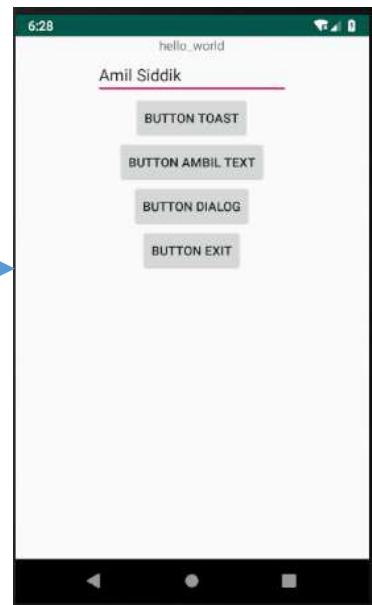
                        @Override
                        public void onClick(DialogInterface
dialog, int

                            which) {
                                // TODO Auto-generated method stub
                                dialog.dismiss();
                            }
                    });
                d.show();
                break;
        }
    }
}

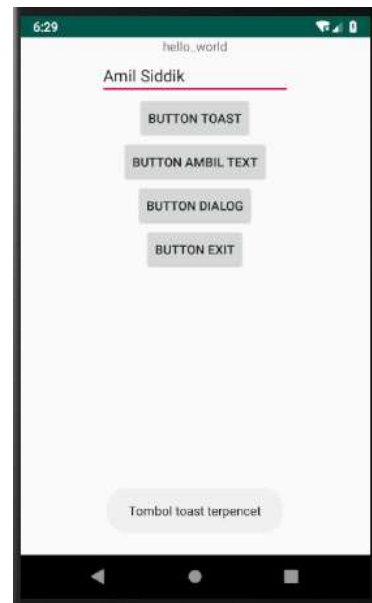
```

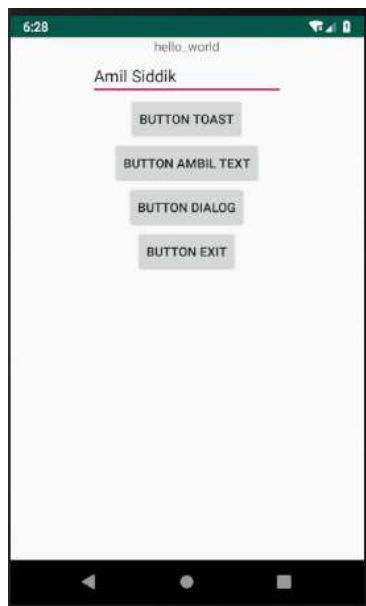


Insert nama
pada EditText.
Contoh: “Amil
Siddik”

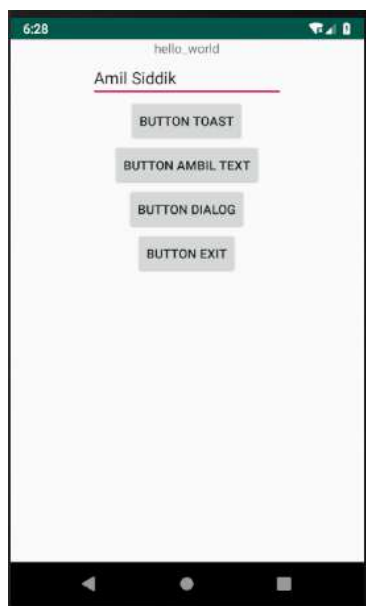
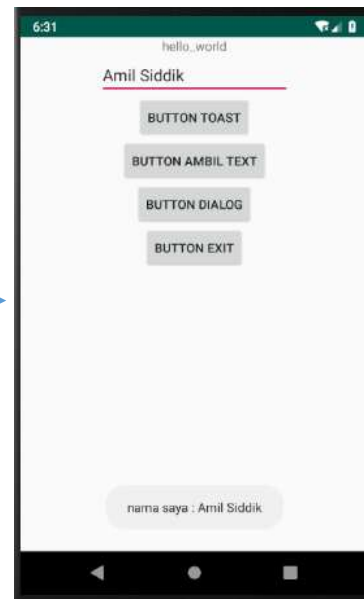


Klik BUTTON
TOAST



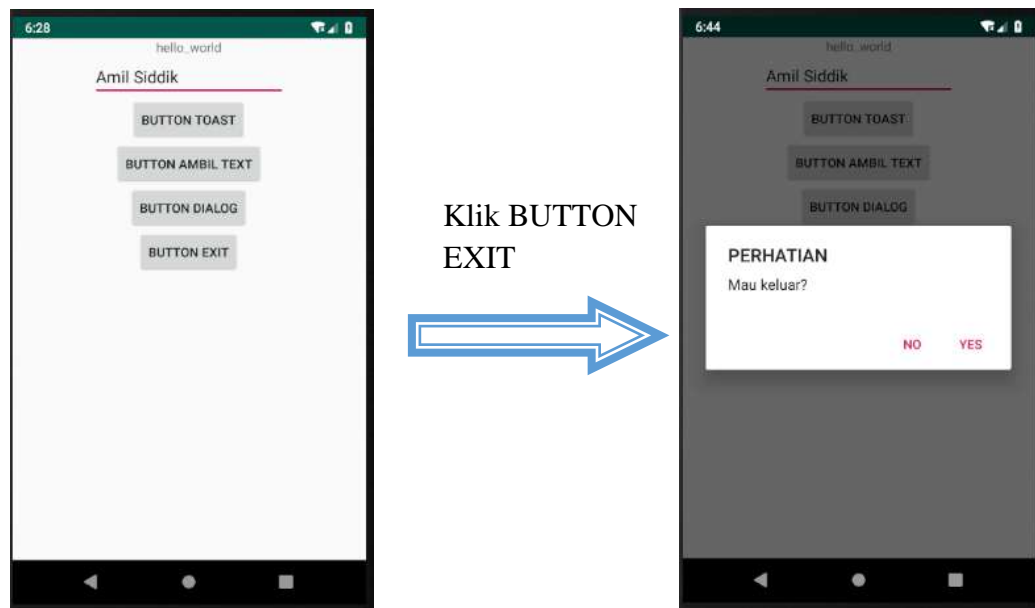


Klik BUTTON
AMBIL TEXT



Klik BUTTON
DIALOG





Gambar 6.2 Hasil run aplikasi

❖ TUGAS PENDAHULUAN

1. Jelaskan definisi dialog!
2. Apa fungsi dari dialog?

❖ TUGAS PRAKTIKUM

Buat program dengan tampilan seperti berikut apabila saat tombol button diklik, maka akan tampil AlertDialog seperti gambar dibawah ini. Jika di tekan tombol “OK” maka akan menampilkan dialog yang sesuai apa yang di inputkan. Jika di tekan tombol “CANCEL” maka akan kembali. Tip : Untuk menampilkan agama gunakan “Spinner”

ALERT WIDGET

NIM

NAMA

☐ Pria

☐ Wanita

ISLAM

CANCEL OK

Hasilnya

NIM = 1111
NAMA = Puji
Jenis Kelamin = Pria
Agama = ISLAM

Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu menggunakan fitur multimedia pada Android Studio.

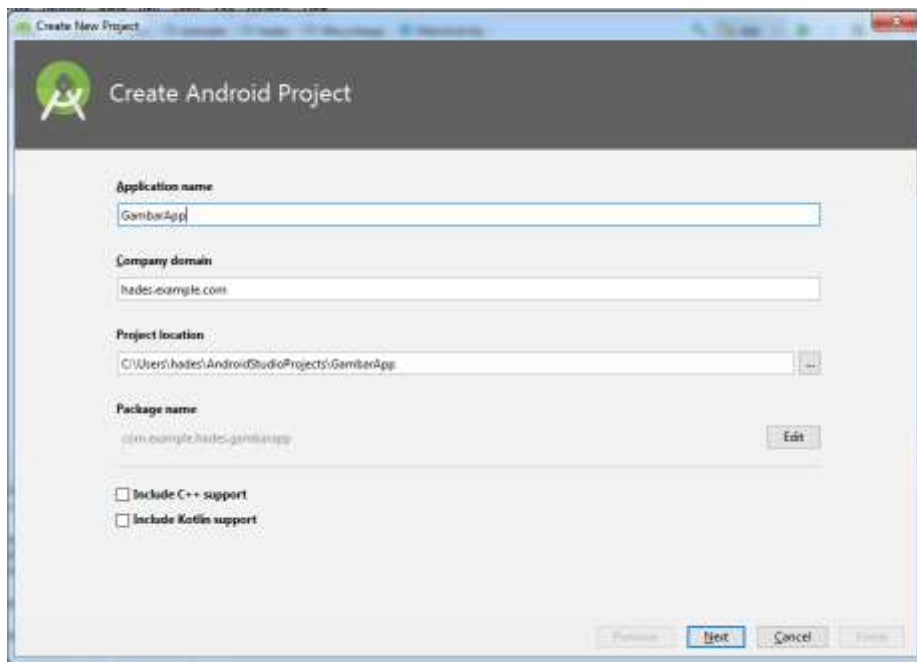
❖ DASAR TEORI

Multimedia berasal daripada kata ‘multi’ dan ‘media’. Multi bererti banyak, dan media bererti tempat, sarana atau alat yang digunakan untuk menyampaikan informasi. Jadi berdasarkan kata ‘multimedia’ dapat dirumuskan sebagai wadah atau penyatuan beberapa media yang kemudian didefinisikan sebagai elemen-elemen pembentukan multimedia. Elemen-elemen tersebut seperti teks, gambar, suara, animasi, dan video. Multimedia merupakan suatu konsep dan teknologi baru bidang teknologi informasi, di mana informasi dalam bentuk teks, gambar, suara, animasi, dan video disatukan dalam computer untuk di simpan, diproses dan disajikan baik secara liner mahupun interaktif.

Oleh itu, dengan menggabungkan seluruh elemen multimedia tersebut menjadikan informasi dalam bentuk multimedia yang dapat diterima oleh indera penglihatan dan pendengaran, lebih mendekati bentuk aslinya dalam dunia sebenarnya. Multimedia enteraktif adalah apabila suatu aplikasi terdapat seluruh elemen multimedia yang ada dan pemakai (user) diberi keputusan atau kemampuan untuk mengawal dan menghidupkan elemen-elemen tersebut.

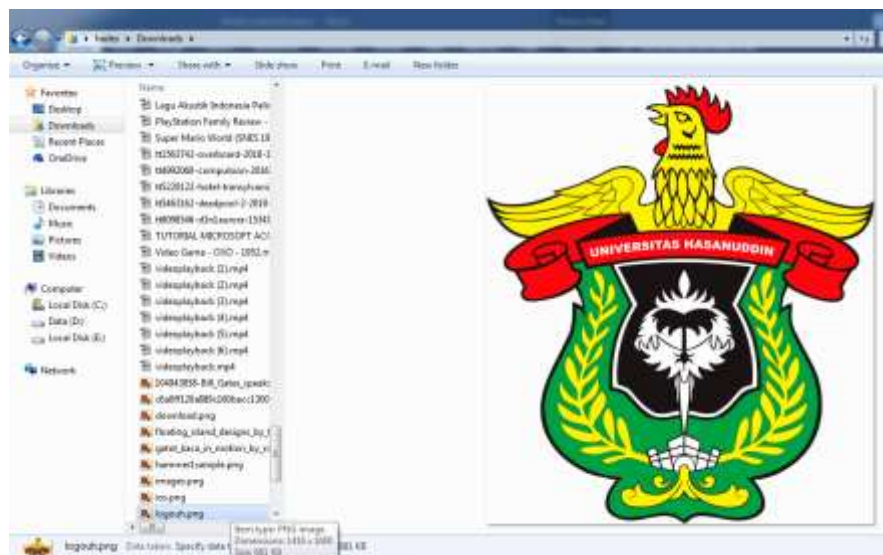
❖ LATIHAN PRAKTIKUM**Latihan 1: Menampilkan Gambar**

1. Buat project baru dengan parameter

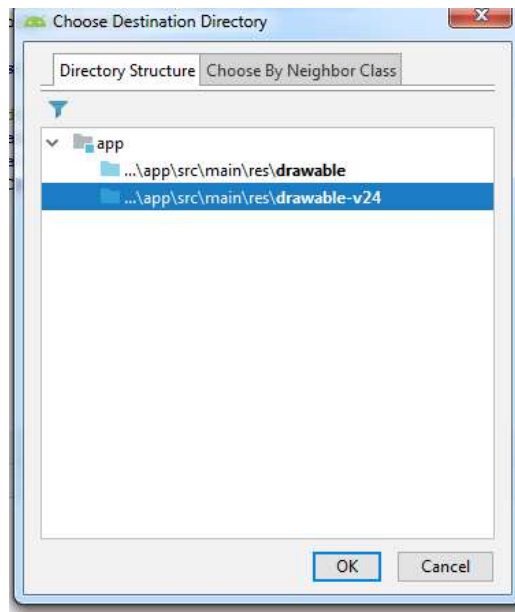


Gambar 7.1 Create Android Project

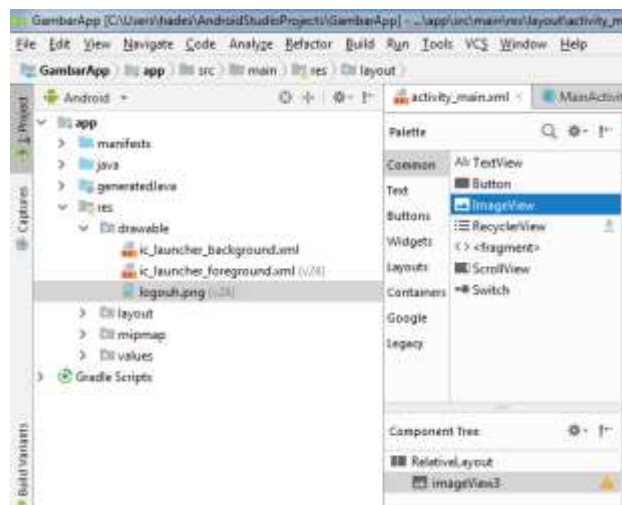
2. Masukkan sebuah gambar PNG atau JPG ke dalam folder drawable. **Klik kanan folder drawable > paste.**



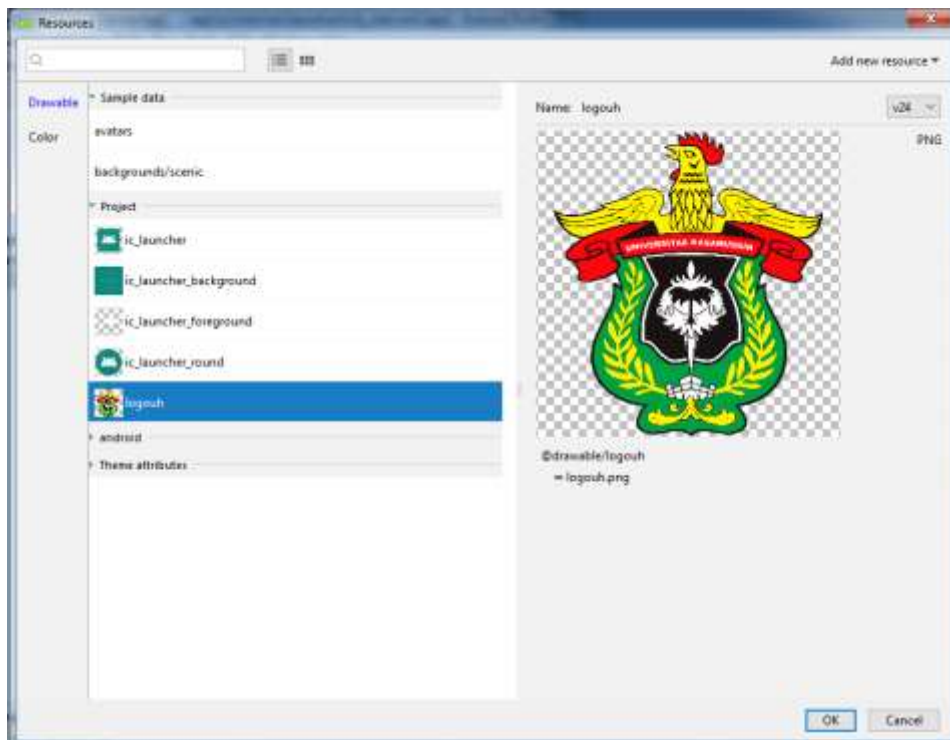
Gambar 7.2 Cari gambar kemudian copy



Gambar 7.3 Dialog yang muncul setelah gambar dipastekan



Gambar 7.4 Seret Image View ke Component Tree



Gambar 7.5 Source gambar

3. Tambahkan widget ImageView pada activity_main.xml. Pada saat menaruh ImageView pada layout, akan muncul kotak dialog seperti Gambar 7.5. Pilih gambar yang telah disiapkan. Berikut code untuk activity_main.xml.

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

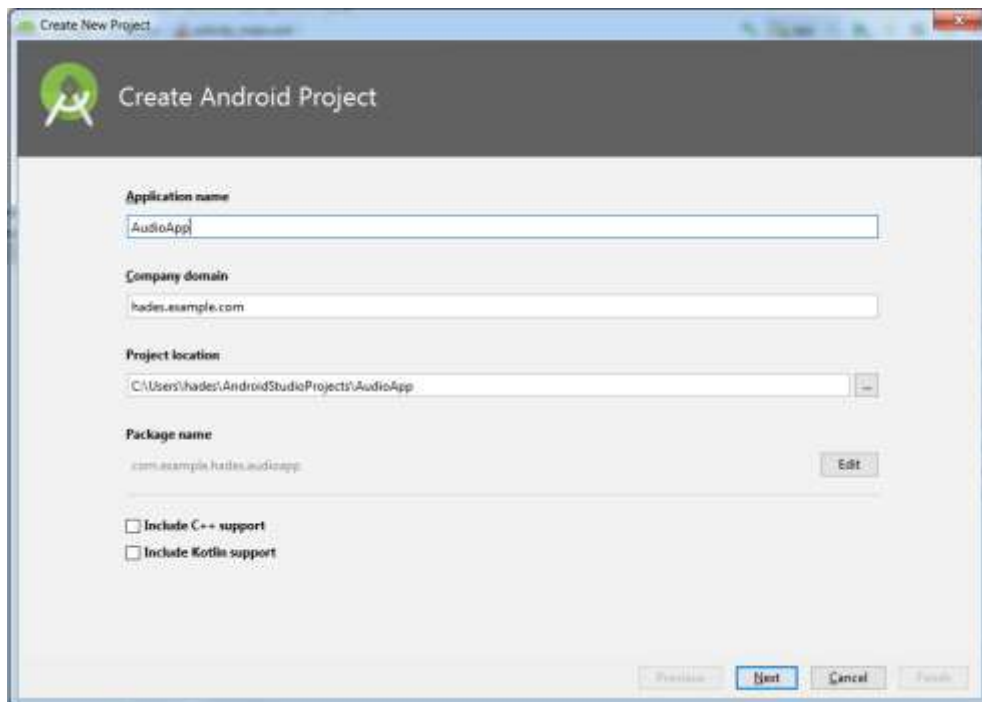
    <ImageView
        android:id="@+id/imageView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/logouh" />
</RelativeLayout>
```



Gambar 7.6 Hasil ImageView

Latihan 2 : Memanggil Audio

1. Buat project baru dengan parameter



Gambar 7.7 Create Android Project

2. Pada activity_main.xml, tambahkan ImageButton dua buah dengan resource gambar ic_media_play dan ic_media_pause. Berikut code untuk activity_main.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"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="AudioApp"/>
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <ImageButton
            android:id="@+id/bPlay"
            android:layout_width="150dp"
            android:layout_height="75dp"
            android:src="@android:drawable/ic_media_play" />
    </LinearLayout>
</LinearLayout>
```

```

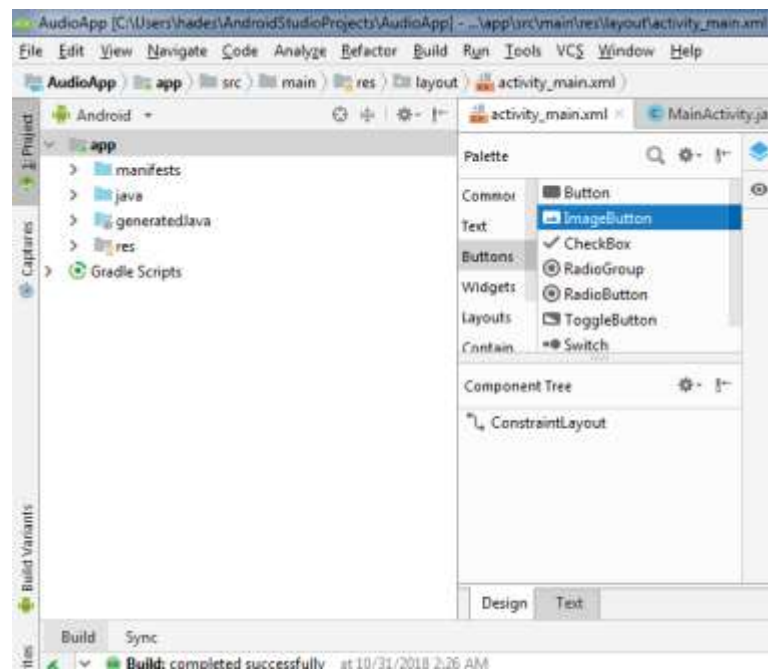
<ImageButton
    android:id="@+id/bPause"
    android:layout_width="150dp"
    android:layout_height="75dp"
    android:src="@android:drawable/ic_media_pause" />

</LinearLayout>

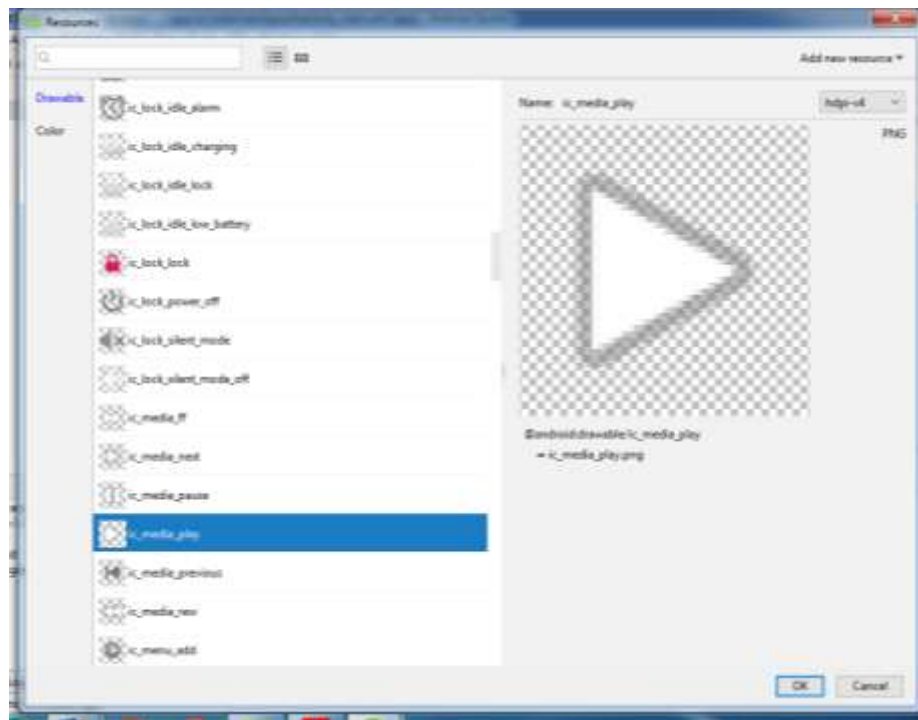
<TextView
    android:id="@+id/tKet"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView" />

</LinearLayout>

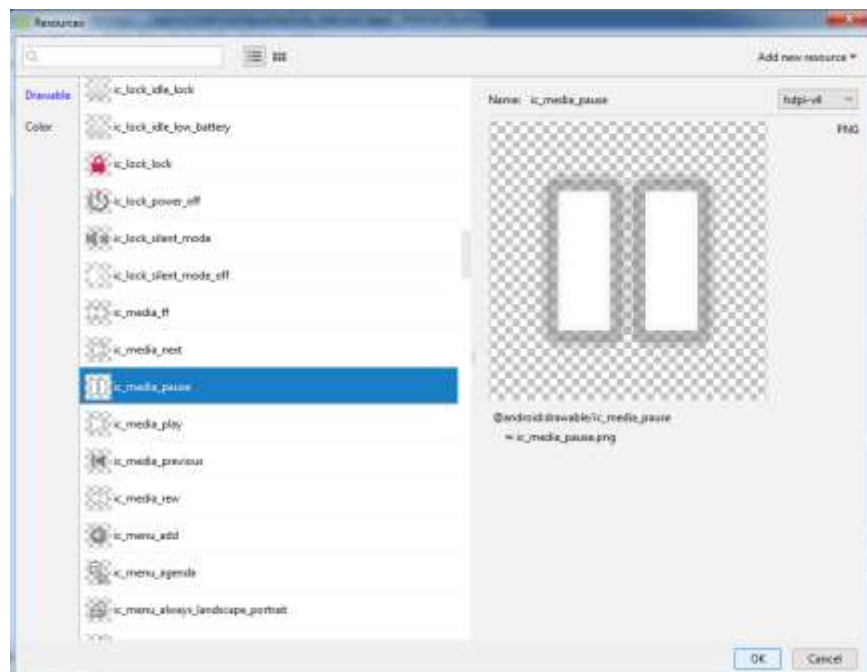
```



Gambar 7.8 ImageButton pada Palette Button

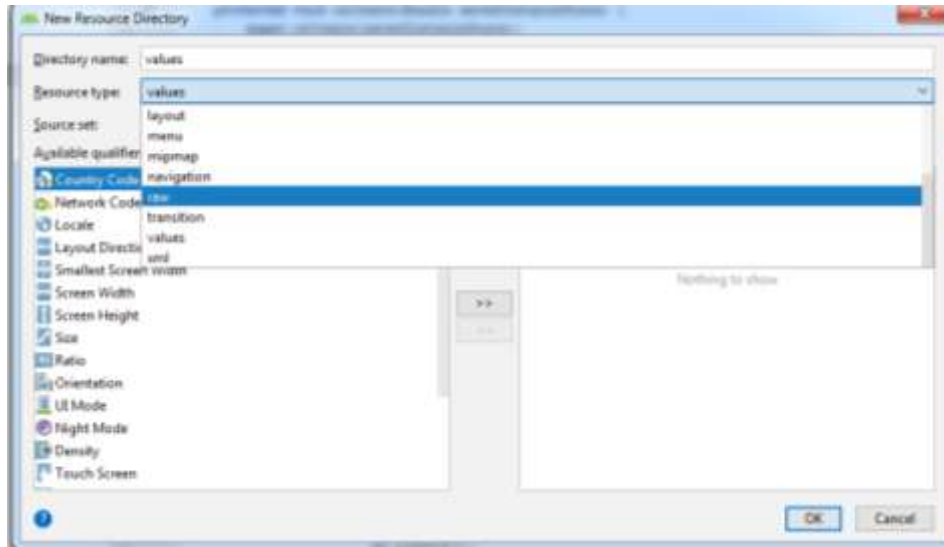


Gambar 7.9 logo ic_media_play



Gambar 7.10 logo ic_media_pause

3. Buat folder raw di res dan masukkan lagu atau music ke res > raw. Pada buku ini lagu yang dipilih adalah sempurna.mp3



Gambar 7.11 Dialog pembuatan folder raw

4. Salin code MainActivity.java berikut ini.

```
package com.example.hades.audioapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import java.io.IOException;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnCompletionListener;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ImageButton;
import android.widget.TextView;

public class MainActivity extends Activity implements OnClickListener
{
    ImageButton bplay, bpause;
    TextView tket;
    MediaPlayer mp;

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

        bplay = (ImageButton) findViewById(R.id.bPlay);
        bpause = (ImageButton) findViewById(R.id.bPause);
        tket = (TextView) findViewById(R.id.tKet);

        bplay.setOnClickListener(this);
        bpause.setOnClickListener(this);
    }
}
```



```

        mp = MediaPlayer.create(this, R.raw.sempurna);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.layout.activity_main, menu);
        return true;
    }

    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        switch (v.getId()) {
            case R.id.bPlay:
                bplay.setEnabled(false);
                tket.setText("playing...");
                if (!mp.isPlaying()) {
                    try {
                        mp.prepare();
                    } catch (IllegalStateException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    } catch (IOException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }

                mp.start();
                mp.setOnCompletionListener(new OnCompletionListener() {

                    public void onCompletion(MediaPlayer arg0) {
                        bplay.setEnabled(true);
                        tket.setText("play again..");
                    }

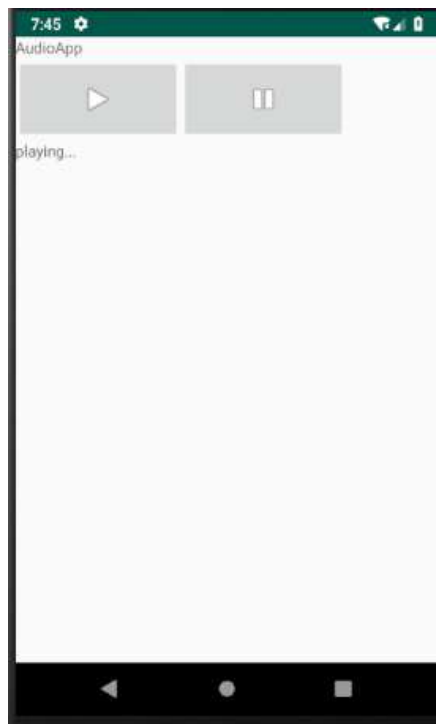
                });

                break;

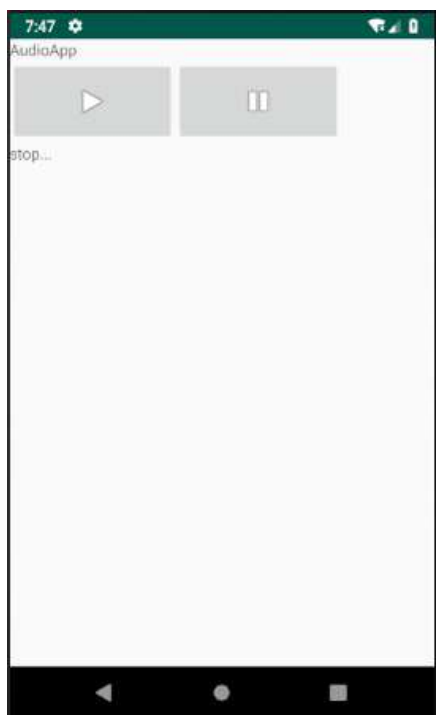
            case R.id.bPause:
                tket.setText("stop...");
                mp.pause();
                bplay.setEnabled(true);
                break;

        }
    }
}

```



Gambar 7.12 Tombol play ditekan dan musik terputar



Gambar 7.13 Tombol pause ditekan dan musik berhenti

❖ **TUGAS PENDAHULUAN**

Jelaskan pentingnya multimedia bagi Android!

❖ **TUGAS PRAKTIKUM**

Buat aplikasi baru, masukkan logo Unhas, dan ketika logo Unhas ditekan mars Unhas akan didengarkan.

Sasaran Pembelajaran

Setelah menyelesaikan praktikum ini :

1. Mahasiswa mampu membuat file apk dari aplikasi dan menginstall aplikasi tersebut pada perangkat Android.

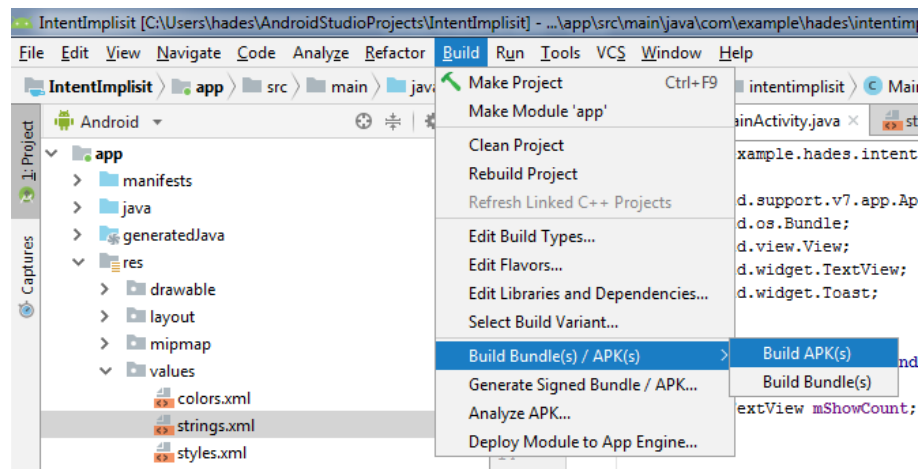
❖ DASAR TEORI

Secara default, Android Studio menyiapkan proyek baru untuk diluncurkan ke Emulator atau perangkat fisik hanya dengan beberapa klik. Dengan Instant Run, Anda bisa mendorong perubahan ke metode dan sumber daya aplikasi yang sudah ada ke aplikasi yang berjalan tanpa membuat APK baru, sehingga perubahan kode terlihat hampir seketika.

Saat Anda mengeklik **Run**, Android Studio akan menghasilkan APK debug dan menerapkannya ke perangkat target Anda. Sebelum Anda bisa menghasilkan aplikasi versi rilis untuk distribusi publik, Anda harus terlebih dahulu belajar cara menandai aplikasi Anda. Anda kemudian bisa mengeluarkan beberapa APK yang sudah ditandai dari debug atau merilis varian versi.

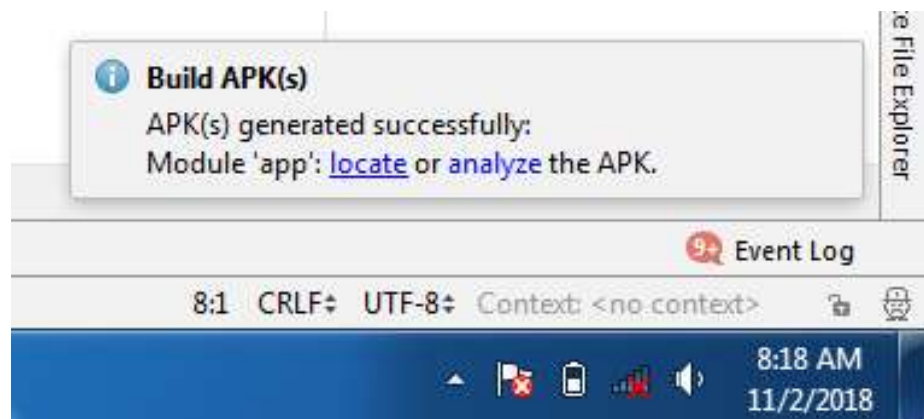
❖ LATIHAN PRAKTIKUM

1. Pilih **Build > Build Bundle(s)/APK(s) > Build APK(s)**



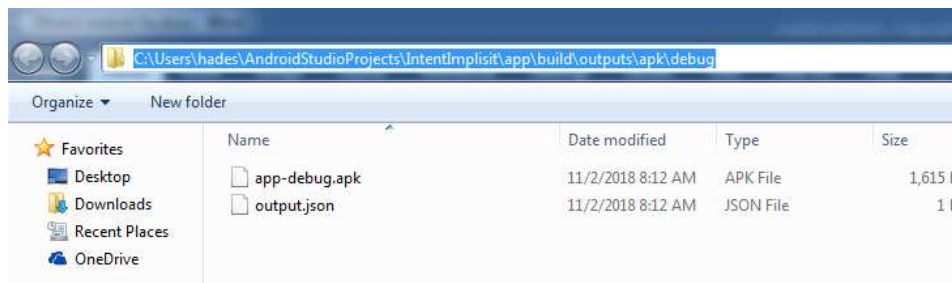
Gambar 8.1 Tahap awal

2. Ketika proses selesai akan muncul dialog di sebelah kanan bawah klik locate.



Gambar 8.2 Dialog Build APK

3. Secara otomatis anda akan diarahkan ke directory penyimpanan apk.



Gambar 8.3 Directory tempat file apk

4. Copy file apk tersebut ke hp android Anda kemudian install

❖ **TUGAS PENDAHULUAN**

Jelaskan proses RUN pada Android Studio!

❖ **TUGAS PRAKTIKUM**

Buat semua hasil praktikum sebelumnya ke dalam file apk.