



Kampus
Merdeka
INDONESIA JAYA

MODUL PRAKTIKUM PEMOGRAMAN BERGERAK



PEMBELAJARAN
DARING KOLABORATIF
(PDK)



TIM Penyusun :

**KRISTINA ANNATASIA BR SITEPU
HERU SAPUTRA**

PERTEMUAN 4

WIDGET VIEW

4.1 TUJUAN PEMBELAJARAN :

- a. Mahasiswa mampu menggunakan Widget View untuk membuat aplikasi sederhana.

4.2 ALAT DAN BAHAN :

- a. Laptop/PC
- b. Android Studio
- c. Sistem Operasi Windows

4.3 MATERI

Paket widget pada dasarnya merupakan visualisasi dari elemen user interface (UI) yang digunakan pada layar aplikasi Android di mana kita dapat merancang sendiri sesuai kebutuhan.

Widget di dalam Android ditampilkan dengan konsep *View*. Di mana aplikasi Android pada umumnya menggunakan widget sebagai Layout XML. Widget ditempatkan di dalam layout yang bisa berupa linear, relative, atau constraint layout. Untuk mengimplementasikan widget, selain file java/kotlin kita juga membutuhkan tambahan dua file. Berikut ini adalah file-file yang umumnya kita butuhkan apabila kita membuat widget:

- 1) File Java/Kotlin. Berupa file yang mengimplementasikan aksi dari widget. Jika kita mendefinisikan suatu widget beserta posisinya di layar yang didefinisikan dari file XML, kita harus melakukan coding di file java/kotlin yang dapat mengambil semua nilai atribut dari file layout XML yang didefinisikan.
- 2) File XML. Sebuah file yang mendefinisikan komponen elemen-elemen XML yang digunakan untuk inialisasi widget serta atribut yang mendukungnya.
- 3) Layout XML. File XML menggambarkan atau penambahan keterangan pada layout widget kita.

1. Text View

TextView adalah widget yang digunakan untuk menampilkan teks di layar aplikasi. Ini adalah widget yang paling dasar untuk menampilkan informasi kepada pengguna. Untuk menggunakan TextView kita bisa mengaplikasikan atribut-atribut penting yang sering digunakan pada TextView diantaranya, sebagai berikut :

1) TextView – Id

Atribut TextView – Id (`android:id`) ini digunakan untuk memberikan nama pada TextView. Biasanya pada aplikasi ada beberapa teks yang ingin di tampilkan dalam satu activity. Agar tidak terjadi error maka kita harus memberikan nama unik atau Id pada masing-masing teks tersebut. Dengan menggunakan Id bisa memudahkan kita saat mengatur layout untuk teks yang akan ditampilkan di aplikasi.

2) TextView – Text

Untuk menampilkan teks pada aplikasi kita membutuhkan atribut TextView – Text ini, berada pada atribut `android:text`.

3) TextView – Color

Warna teks sangat berpengaruh pada tampilan UI dan UX pada aplikasi android. Untuk memberikan warna teks kita menggunakan atribut TextView – Color (`android:textColor`) ini. Untuk warna sendiri kita bisa mengaturnya pada `color.xml` atau juga langsung dengan cara memasukan kode-kode warna seperti : `#ffff` dan lainnya.

4) TextView – Size

Atribut TextView – Size (`android:textSize`) ini digunakan untuk memberikan ukuran pada teks yang ingin kita tampilkan pada aplikasi.

5) TextView – Appearance

Atribut TextView – Appearance (`android:textAppearance`) ini digunakan untuk mengubah ukuran teks yang digunakan untuk heading atau sub heading.

6) TextView – Style

Beberapa style yang ingin kita gunakan saat menampilkan teks, yaitu : bold dan italic dengan menggunakan Atribut TextView – Style (`android:textStyle`) ini.

7) TextView – Type Face

Dengan Atribut TextView – Type Face (`android:typeface`) ini kita bisa memilih tipe font bawaan yaitu : serif, monospace, dan sans.

8) TextView – Aligment

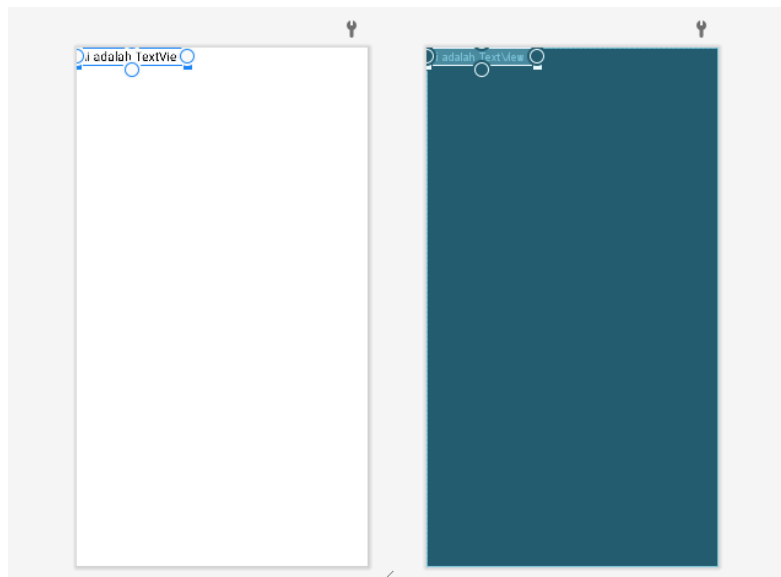
Atribut TextView – Aligment (`android:textAlignment`) digunakan untuk melakukan pengaturan posisi teks (perataan).

9) Menambahkan Gambar pada TextView

Kita juga bisa menambahkan gambar pada TextView. Caranya gampang silahkan copykan gambar / icon pada folder drawable. Atau bisa juga menggunakan icon default yang sudah tersedia pada android studio. Atribut `android:drawableLeft` ini digunakan untuk memasukan gambar tinggal sesuaikan saja iconnya akan ditempatkan sebelah mana.

Contoh XML untuk TextView:

```
<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ini adalah TextView"
    android:textSize="18sp"
    android:textColor="#000000"/>
```

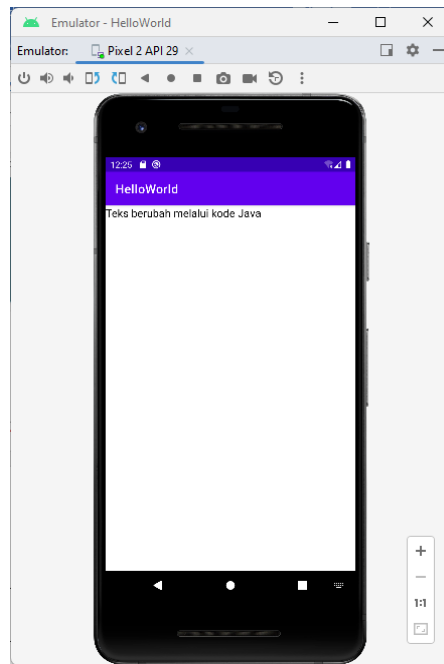


Gambat 4.1 Rancangan TextView

Penggunaan TextView di Java:

```
TextView textView = findViewById(R.id.textView);
textView.setText("Teks berubah melalui kode Java");
```

Dalam kode di atas, kita mengubah teks yang ditampilkan dalam TextView melalui program Java.



Gambar 4.2 Tampilan output TextView

2. Edit Text

EditText digunakan untuk menerima input dari pengguna dalam bentuk teks. Beberapa atribut penting yang sering digunakan:

- a. `android:hint`: Menampilkan teks petunjuk saat EditText kosong.

Contoh:

```
android:hint="Masukkan nama"
```

- b. `android:inputType`: Menentukan jenis input (misalnya teks, angka, email, password, dll.). Ini akan mengubah tampilan keyboard yang muncul.

Contoh:

```
android:inputType="text" atau android:inputType="number"
```

- c. `android:maxLength`: Membatasi jumlah karakter maksimum yang bisa dimasukkan pengguna.

Contoh:

```
android:maxLength="10"
```

- d. `android:text`: Menampilkan teks yang sudah ada di dalam EditText.

Contoh:

```
android:text="Nama Awal"
```

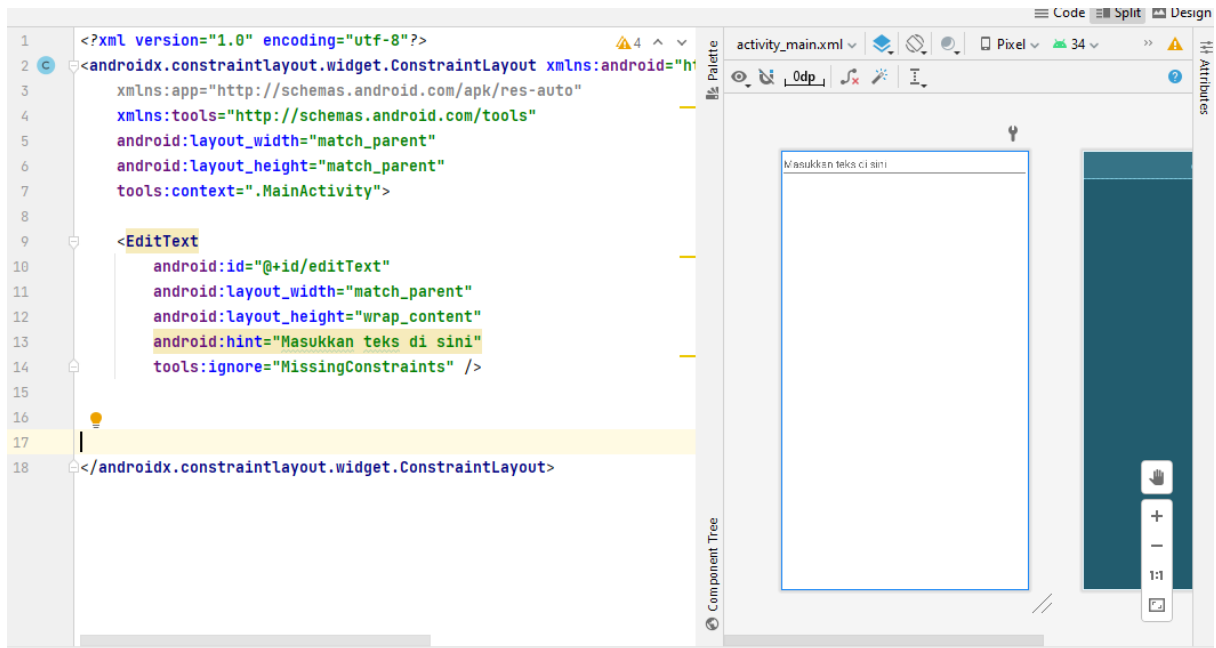
- e. `android:singleLine`: Mengatur apakah input akan ditampilkan dalam satu baris saja.

Contoh:

```
android:singleLine="true"
```

Contoh XML untuk EditText:

```
<EditText
    android:id="@+id/editText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Masukkan teks di sini" />
```

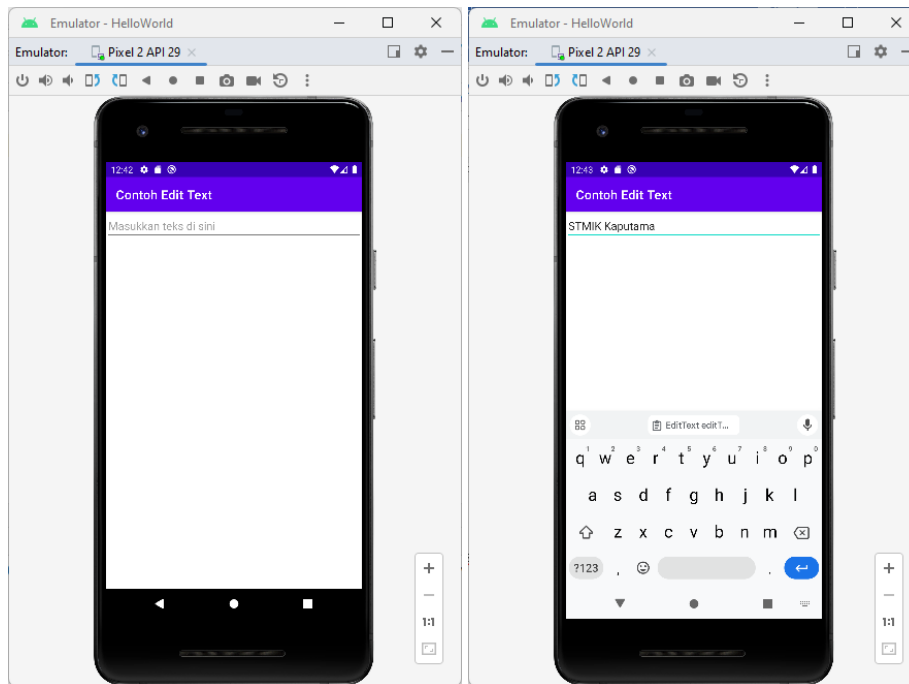


Gambar 4.3 Tampilan Edit Text pada xml

Penggunaan EditText di Java:

```
EditText editText = findViewById(R.id.editText);
String userInput = editText.getText().toString();
```

Dengan kode di atas, kita bisa mendapatkan teks yang dimasukkan oleh pengguna ke dalam EditText.



Gambar 4.4 Contoh Edit Text

3. Button

Button adalah widget yang digunakan untuk membuat tombol interaktif. Ketika pengguna mengklik tombol, aksi tertentu dapat dijalankan. Atribut-atribut penting untuk Button:

- a. `android:text`: Menentukan teks yang ditampilkan pada tombol.

Contoh:

```
android:text="Klik Saya"
```

- b. `android:onClick`: Menentukan metode (fungsi) yang dipanggil saat tombol diklik. Fungsi ini harus didefinisikan di file Java. Contoh:

```
android:onClick="tombolDiklik"
```

- c. `android:background`: Menentukan latar belakang tombol. Bisa berupa warna, gambar, atau drawable XML.

Contoh:

```
android:background="#FF0000"
```

- d. `android:textColor`: Mengatur warna teks di dalam tombol.

Contoh:

```
android:textColor="#FFFFFF"
```

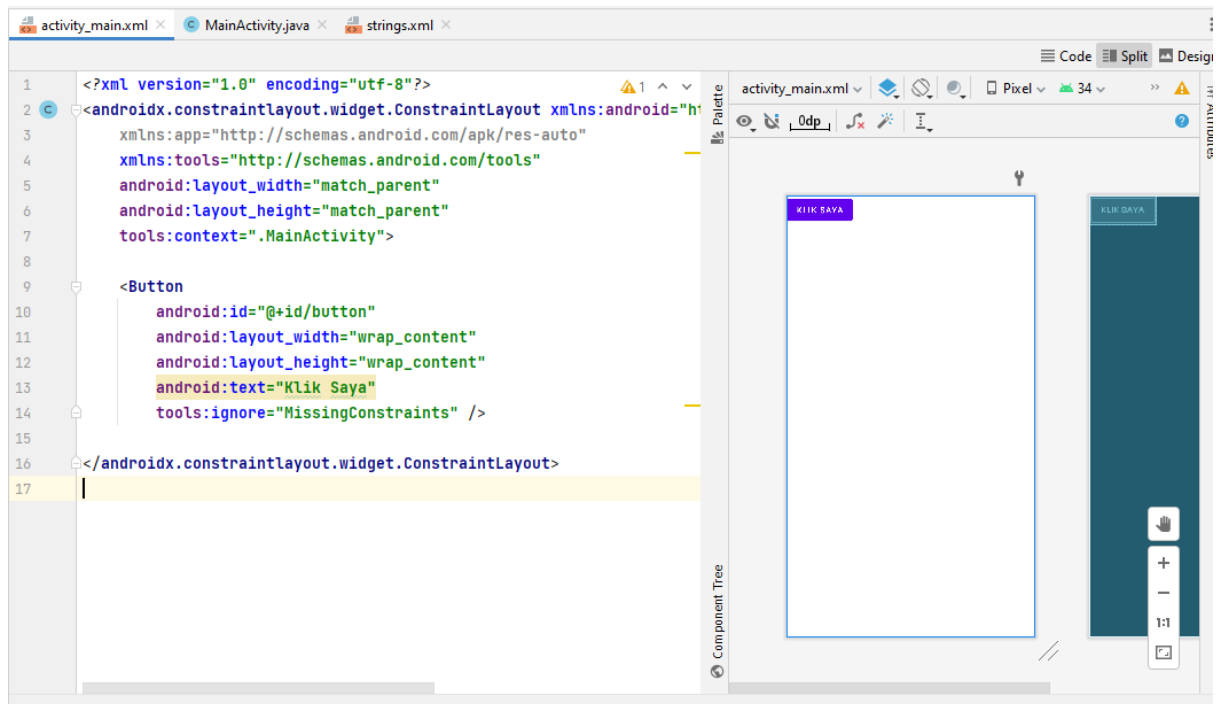
- e. `android:enabled`: Mengatur apakah tombol dapat diklik atau tidak.

Contoh:

```
android:enabled="false"
```

Contoh XML untuk Button:

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

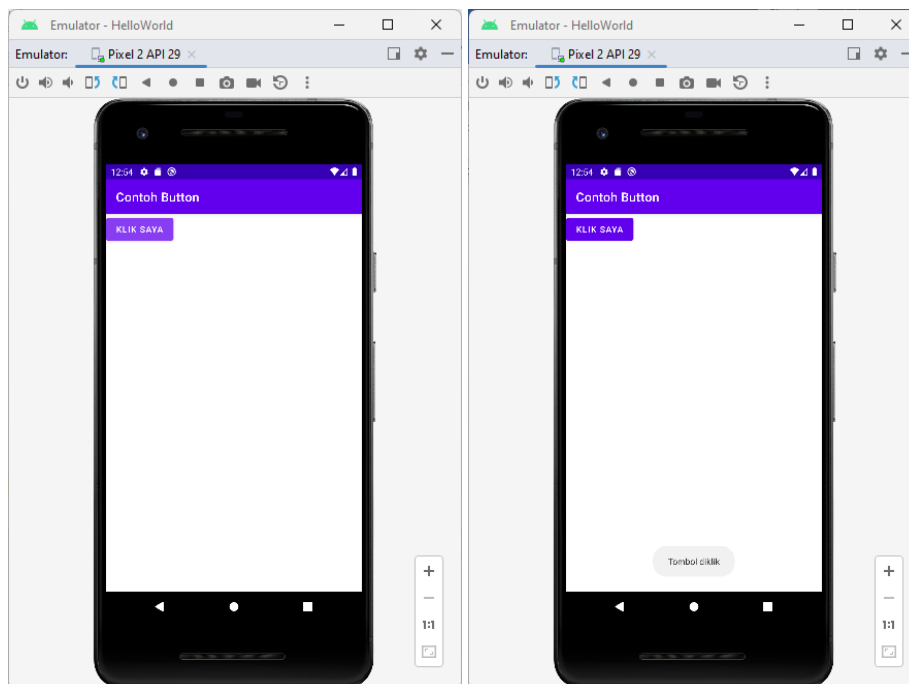


Gambar 4.5 Contoh Desain Button

Penggunaan Button di Java:

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Aksi ketika tombol diklik
        Toast.makeText(getApplicationContext(), "Tombol diklik",
        Toast.LENGTH_SHORT).show();
    }
});
```


Kode di atas membuat Button dan menetapkan listener untuk mendeteksi klik, lalu menjalankan aksi berupa menampilkan toast saat tombol diklik.



Gambar 4.6 Contoh penggunaan Button

4. Radio Button

RadioButton digunakan ketika pengguna harus memilih satu dari beberapa opsi yang tersedia. Biasanya ditempatkan dalam RadioGroup. Atribut-atribut yang penting pada RadioButton:

- a. `android:text`: Menentukan teks yang ditampilkan di samping RadioButton.
Contoh: `android:text="Pilihan 1"`
- b. `android:checked`: Menentukan apakah RadioButton sudah terpilih atau tidak.
Contoh: `android:checked="true"`
- c. `android:buttonTint`: Mengatur warna tombol RadioButton.
Contoh: `android:buttonTint="#0000FF"`
- d. `android:onClick`: Menentukan metode yang dipanggil saat RadioButton diklik.
Contoh: `android:onClick="radioButtonDiklik"`

Contoh XML untuk RadioButton:

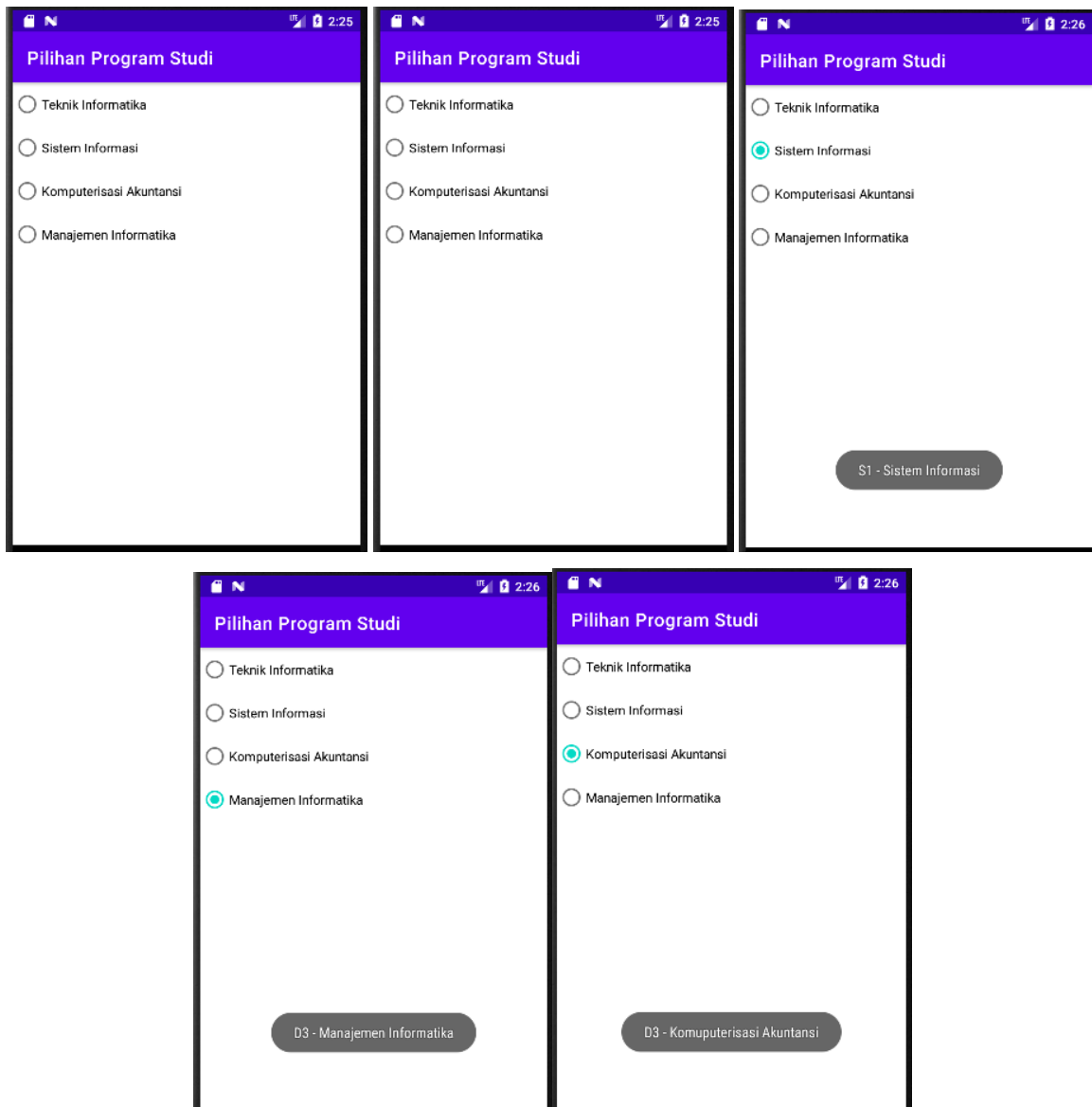
```
activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="wrap_content"
6     android:orientation="vertical"
7     tools:context=".MainActivity">
8
9     <RadioButton
10        android:layout_width="wrap_content"
11        android:layout_height="wrap_content"
12        android:id="@+id/radio_ti"
13        android:text="Teknik Informatika"
14        android:onClick="onRadioButtonClicked"
15        tools:ignore="UsingOnClickInXml" />
16
17    <RadioButton
18        android:layout_width="wrap_content"
19        android:layout_height="wrap_content"
20        android:id="@+id/radio_si"
21        android:text="Sistem Informasi"
22        android:onClick="onRadioButtonClicked"
23        tools:ignore="UsingOnClickInXml" />
24
25    <RadioButton
26        android:layout_width="wrap_content"
27        android:layout_height="wrap_content"
28        android:id="@+id/radio_ka"
29        android:text="Komputerisasi Akuntansi"
30        android:onClick="onRadioButtonClicked"
31        tools:ignore="UsingOnClickInXml" />
32
33    <RadioButton
34        android:layout_width="wrap_content"
35        android:layout_height="wrap_content"
36        android:id="@+id/radio_mi"
37        android:text="Manajemen Informatika"
38        android:onClick="onRadioButtonClicked"
39        tools:ignore="UsingOnClickInXml" />
40
41 </RadioGroup>
```

Gambar 4.7 Tampilan coding Radio Button

Penggunaan RadioButton di Java:

```
MainActivity.java x
1 package com.ratih.radiobutton;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.annotation.SuppressLint;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.RadioButton;
8 import android.widget.Toast;
9
10 public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16     }
17
18     @SuppressLint("NonConstantResourceId")
19     public void onRadioButtonClicked(View view) {
20         boolean checked = ((RadioButton) view).isChecked();
21         switch(view.getId()) {
22             case R.id.radio_ti:
23                 if (checked)
24                     Toast.makeText(getApplicationContext(), "S1 - Teknik Informatika",
25                         Toast.LENGTH_SHORT).show();
26                 break;
27             case R.id.radio_si:
28                 if (checked)
29                     Toast.makeText(getApplicationContext(), "S1 - Sistem Informasi",
30                         Toast.LENGTH_SHORT).show();
31                 break;
32             case R.id.radio_ka:
33                 if (checked)
34                     Toast.makeText(getApplicationContext(), "D3 - Komputerisasi Akuntansi",
35                         Toast.LENGTH_SHORT).show();
36                 break;
37             case R.id.radio_mi:
38                 if (checked)
39                     Toast.makeText(getApplicationContext(), "D3 - Manajemen Informatika",
40                         Toast.LENGTH_SHORT).show();
41                 break;
42             }
43         }
44     }
45 }
```

Dengan kode di atas, kita mendapatkan nilai dari opsi yang dipilih oleh pengguna di dalam RadioGroup yang terdiri dari beberapa RadioButton.



Gambar 4.8 Contoh Radio Button

5. Rating Bar

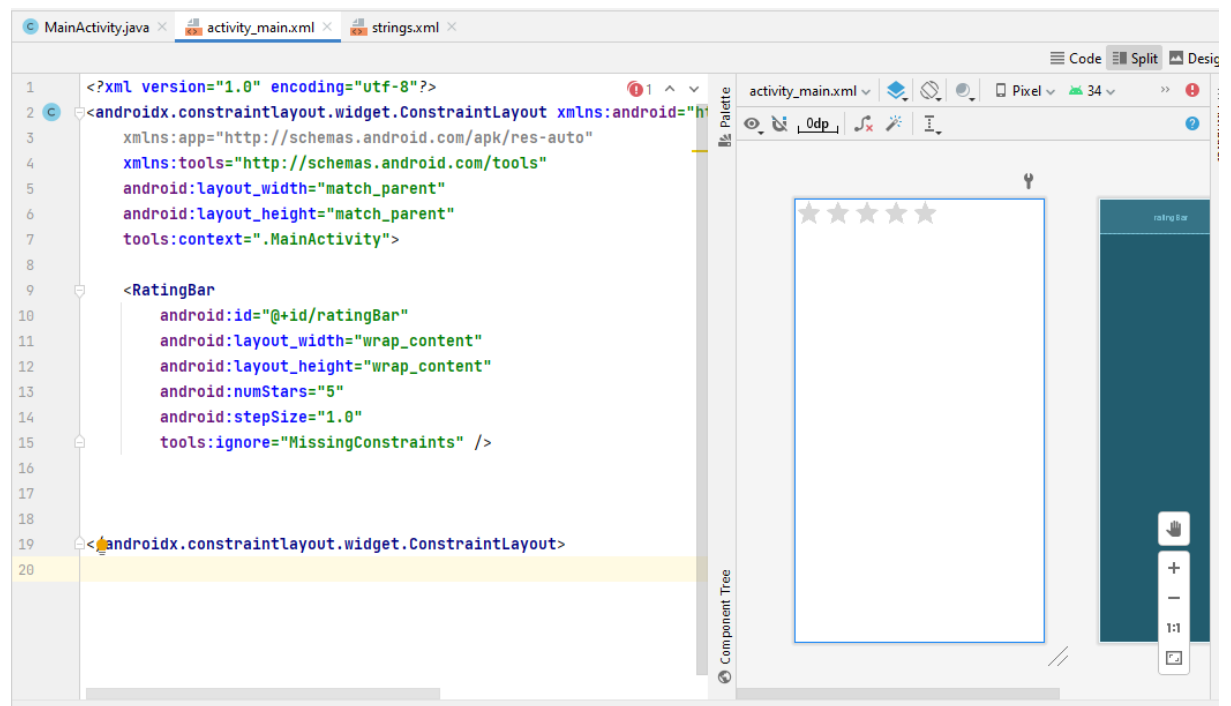
RatingBar memungkinkan pengguna memberikan rating atau penilaian bintang. Atribut-atribut yang sering digunakan:

- `android:numStars`: Menentukan jumlah bintang yang ditampilkan.
Contoh: `android:numStars="5"`
- `android:rating`: Menentukan nilai rating awal yang ditampilkan (dalam satuan bintang).
Contoh: `android:rating="3.5"`
- `android:stepSize`: Menentukan langkah perubahan rating (misalnya 0.5 untuk setengah bintang).
Contoh: `android:stepSize="0.5"`

- d. `android:isIndicator`: Menentukan apakah RatingBar hanya sebagai indikator (tidak bisa diubah oleh pengguna).
Contoh: `android:isIndicator="true"`
- e. `android:progressTint`: Menentukan warna bintang yang sudah diisi (berdasarkan rating).
Contoh: `android:progressTint="#FFEB3B"`

Contoh XML untuk RatingBar:

```
<RatingBar
    android:id="@+id/ratingBar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:numStars="5"
    android:stepSize="1.0"/>
```

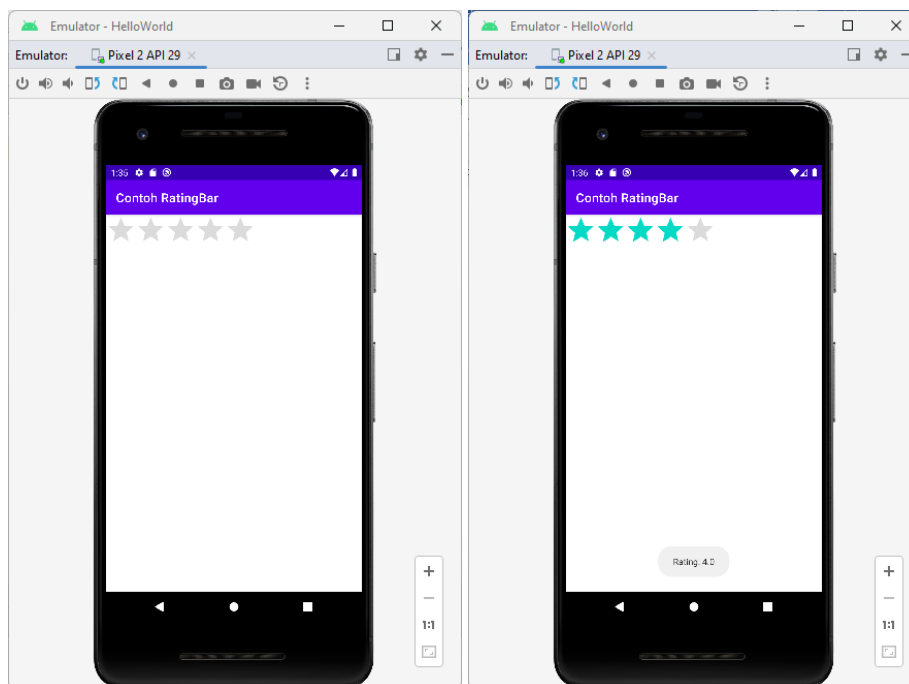


Gambar 4.9 Desain Rating Bar

Penggunaan RatingBar di Java:

```
RatingBar ratingBar = findViewById(R.id.ratingBar);
float rating = ratingBar.getRating();
ratingBar.setOnRatingChangeListener(new RatingBar.OnRatingChangeListener() {
    @Override
    public void onRatingChanged(RatingBar ratingBar, float rating, boolean fromUser) {
        Toast.makeText(getApplicationContext(), "Rating: " + rating,
        Toast.LENGTH_SHORT).show();
    }
});
```

Kode di atas menampilkan rating yang dipilih oleh pengguna dan menampilkan toast setiap kali rating diubah.



Gambar 4.10 Contoh RatingBar

6. ImageButton

ImageButton adalah tombol yang menggunakan gambar sebagai tampilan, berbeda dari Button yang hanya menggunakan teks. Atribut penting pada ImageButton meliputi:

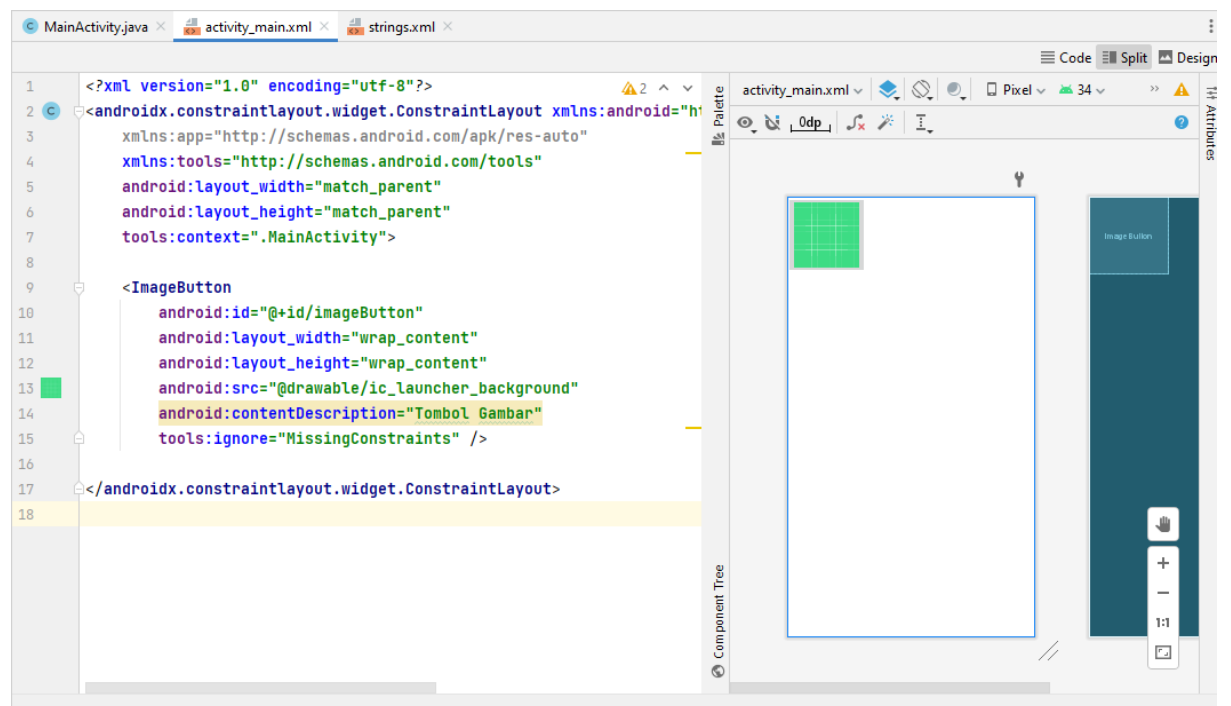
- android:src: Menentukan sumber gambar yang akan ditampilkan di dalam tombol. Ini bisa berupa file gambar di folder res/drawable.

Contoh: `android:src="@drawable/ic_launcher"`

- b. android:contentDescription: Menambahkan deskripsi teks yang akan digunakan oleh layanan aksesibilitas (misalnya pembaca layar).
Contoh: android:contentDescription="Tombol Gambar"
- c. android:background: Menentukan latar belakang tombol. Biasanya transparan agar hanya gambar yang terlihat.
Contoh: android:background="@android:color/transparent"
- d. android:onClick: Menentukan metode yang dipanggil saat ImageButton diklik.
Contoh: android:onClick="imageButtonDiklik"

Contoh XML untuk ImageButton:

```
<ImageButton
    android:id="@+id/imageButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_launcher_background"
    android:contentDescription="Tombol Gambar"/>
```

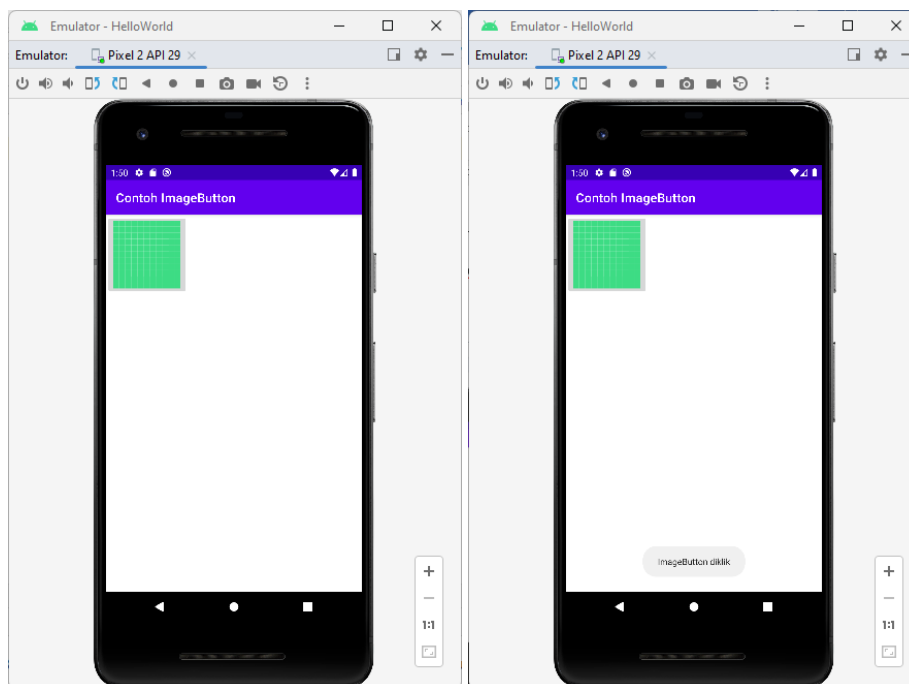


Gambar 4.11 Desain Image Button

Penggunaan ImageButton di Java:

```
ImageButton imageButton = findViewById(R.id.imageButton);
imageButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Aksi ketika ImageButton diklik
        Toast.makeText(getApplicationContext(), "ImageButton diklik",
        Toast.LENGTH_SHORT).show();
    }
});
```

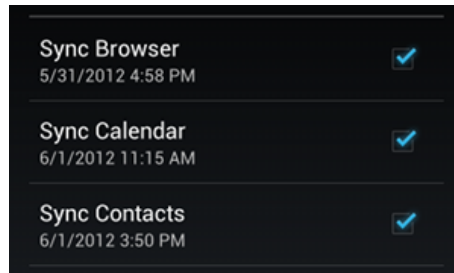
Pada kode di atas, ImageButton menggunakan gambar dari sumber daya drawable, dan ketika diklik, ia menjalankan aksi tertentu.



Gambar 4.12 Contoh ImageButton

7. CheckBox

CheckBox atau kotak centang pada android studio adalah palette pada kategori Buttons yang dapat dicentang atau tidak, dan bisa digunakan untuk pilihan dua keadaan atau lebih serta dapat memilih satu checkbox secara bersamaan. CheckBox (Kotak Centang) memungkinkan pengguna untuk memilih satu atau beberapa opsi dari suatu kumpulan. Biasanya, CheckBox harus menampilkan setiap opsi kotak centang dalam daftar vertikal.



Gambar 4.13 Contoh Penggunaan CheckBox

Untuk membuat setiap opsi kotak centang, buat CheckBox dalam layout (tata letak). Karena sekumpulan opsi kotak centang memungkinkan pengguna memilih beberapa item, setiap kotak centang akan dikelola secara terpisah dan harus mendaftarkan pemroses klik untuk masing-masing kotak centang. Atribut penting pada CheckBox meliputi:

- a. `android:text` untuk mengatur teks di sebelah Checkbox.
- b. `android:checked` untuk menentukan status awal (tercentang atau tidak).
- c. `android:onClick` untuk menentukan metode yang dipanggil ketika Checkbox diklik.
- d. `android:buttonTint` untuk mengubah warna tombol centang.
- e. `android:textColor`, `android:textSize`, `android:padding`, dan `android:gravity` untuk mengatur tampilan visual Checkbox.
- f. `android:enabled` dan `android:focusable` untuk mengontrol apakah Checkbox dapat berinteraksi dengan pengguna.

Contoh XML untuk CheckBox:

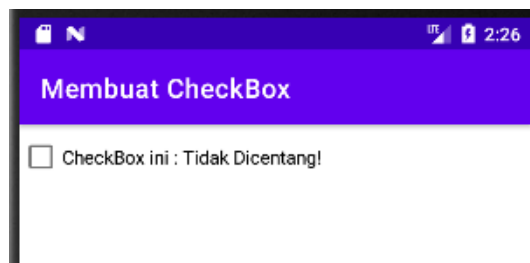
```
activity_main.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <CheckBox xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="wrap_content"
5   android:layout_height="wrap_content"
6   android:id="@+id/CheckBox01"
7   android:text="@string/checkbox"
8   tools:context=".MainActivity">
9 </CheckBox>
```

Penggunaan ImageButton di Java:

```
MainActivity.java x
1 package com.ratih.checkboxsederhana;
2
3 import android.os.Bundle;
4 import android.widget.CheckBox;
5 import android.widget.CompoundButton;
6 import android.widget.CompoundButton.OnCheckedChangeListener;
7 import androidx.appcompat.app.AppCompatActivity;
8
9 public class MainActivity extends AppCompatActivity implements
10     OnCheckedChangeListener {
11
12     6 usages
13     CheckBox cb;
14
15     /** Called when the activity is first created. */
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_main);
20
21         cb = (CheckBox) findViewById(R.id.CheckBox01);
22         cb.setOnCheckedChangeListener(this);
23     }
24
25     public void onCheckedChange(CompoundButton buttonView, boolean isChecked) {
26         if (isChecked) {
27             cb.setText("CheckBox ini : Dicentang!");
28         } else {
29             cb.setText("CheckBox ini : Tidak Dicentang!");
30         }
31     }
32
33     2 usages
34     @Override
35     public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
36         // TODO Auto-generated method stub
37         if (isChecked) {
38             cb.setText("CheckBox ini : Dicentang!");
39         } else {
40             cb.setText("CheckBox ini : Tidak Dicentang!");
41         }
42     }
43 }
```

Perhatikan dahulu `activity_main.xml`, disini kita memakai widget `CheckBox` yang diberi id "`CheckBox01`" (lihat baris 6). Widget `CheckBox` memiliki property `text`, isinya mengambil dari string "`checkbox`" pada `strings.xml` (baris 7). Widget `CheckBox` memiliki 2 status yaitu : `checked` dan `unchecked`. Dengan mengeklik `CheckBox` artinya kita mengubah status dari `checked` menjadi `unchecked` atau sebaliknya.

Sekarang kita lihat `activity CheckBoxSederhana (MainActivity.java)`, baris 20 adalah sinkronisasi antara object "`cb`" terhadap widget `CheckBox` dari layout `activity_main.xml`, sedangkan baris 21 untuk mengaktifkan widget `CheckBox` agar berfungsi saat diklik. Activity ini mengimplementasikan class `OnCheckedChangeListener`, agar sistem bisa mengenali apakah `CheckBox` diklik atau tidak. Aktivitas ini di baca oleh method `onCheckedChanged()` pada baris 24-30. Method ini lah yang bertugas mengganti tulisan pada widget ketika `CheckBox` diklik.



Gambar 4.14 Tampilan `CheckBox` Sederhana

Event Handling.

Android dapat menangani **event** dari interaksi dengan pengguna. Saat mempertimbangkan event dalam user interface, pendekatannya adalah menangkap event dari objek **View** tertentu yang digunakan pengguna untuk berinteraksi. Kelas `View` menyediakan sarana untuk melakukannya.

Dalam berbagai kelas `View` yang akan digunakan untuk menyusun layout, mungkin dapat dilihat beberapa method callback publik yang tampak berguna untuk kejadian UI. Method ini dipanggil oleh framework Android ketika masing-masing tindakan terjadi pada objek itu. Misalnya, jika `View` (seperti `Button`) disentuh, method `onTouchEvent()` akan dipanggil pada objek itu. Kelas `View` salah satunya berisi sekumpulan interface bertumpuk dengan callback yang mudah didefinisikan. Antarmuka ini, yang disebut event listener, digunakan untuk melakukan interaksi pengguna dengan UI.

Event listener

Event listener merupakan antarmuka di kelas View yang berisi method callback tunggal. Method ini akan dipanggil oleh framework Android jika View yang telah didaftarkan dengan listener dipicu oleh interaksi pengguna dengan item dalam UI.

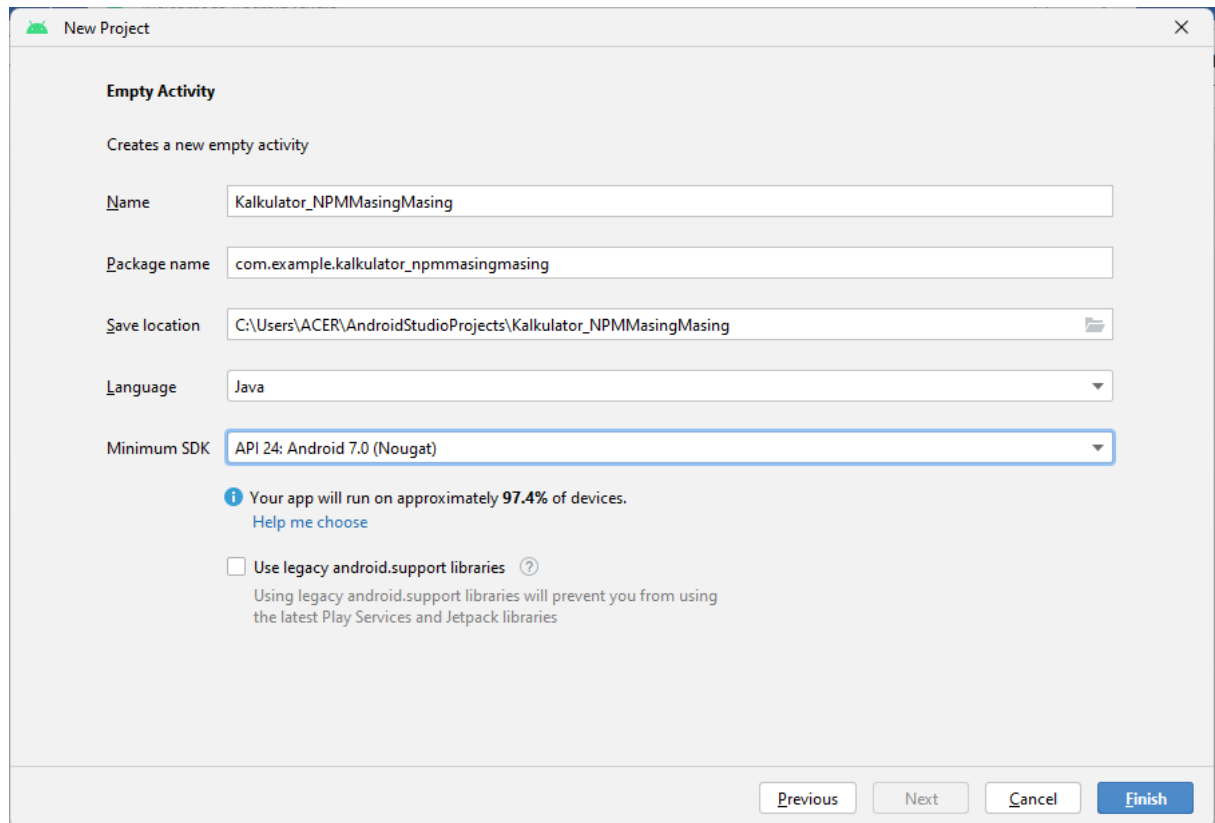
Yang juga disertakan dalam antarmuka event listener adalah method callback berikut ini:

- 1) Method `onClick()` dari `View.OnClickListener`. Ini dipanggil baik saat pengguna menyentuh item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan menekan tombol "enter" yang sesuai atau menekan trackball.
- 2) Method `onLongClick()` dari `View.OnLongClickListener`. Ini dipanggil baik saat pengguna menyentuh dan menahan item (jika dalam mode sentuh), maupun memfokuskan pada item dengan tombol navigasi atau trackball dan menekan serta menahan tombol "enter" yang sesuai atau menekan dan menahan trackball (selama satu detik).
- 3) Method `onFocusChange()` dari `View.OnFocusChangeListener`. Ini dipanggil saat pengguna menyusuri ke atau dari item, dengan menggunakan tombol navigasi atau trackball.
- 4) Method `onKey()` dari `View.OnKeyListener`. Ini dipanggil saat pengguna memfokuskan pada item dan menekan atau melepas tombol perangkat keras pada perangkat.
- 5) Method `onTouch()` dari `View.OnTouchListener`. Ini dipanggil saat pengguna melakukan tindakan yang digolongkan sebagai peristiwa sentuh, termasuk penekanan, pelepasan, atau isyarat perpindahan pada layar (dalam batasan item itu).
- 6) Method `onCreateContextMenu()` dari `View.OnCreateContextMenuListener`. Ini dipanggil saat Menu Konteks sedang dibuat (akibat "klik lama" terus-menerus).

4.4 PRAKTIKUM

1) Buat Proyek Baru:

- Buka Android Studio, pilih "New Project".
- Pilih template "Empty Activity" dan klik "Next".
- Masukkan nama proyek : KatalogBarang_NPMMasing-masing
- Set language ke Java dan Minimum API Level sesuai kebutuhan
- Klik Finish



2) Membuat Layout (activity_main.xml)

Kita akan menggunakan GridLayout untuk menampilkan angka dan tombol operasi. Berikut adalah layout yang kita buat:

- Buka file activity_main.xml kemudian ubah kode seperti berikut :

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:orientation="vertical"
7     android:padding="16dp">
8
9     <!-- Tampilan Hasil -->
10    <TextView
11        android:id="@+id/HasilText"
12        android:layout_width="match_parent"
13        android:layout_height="wrap_content"
14        android:background="#E0E0E0"
15        android:gravity="end"
16        android:padding="16dp"
17        android:text="0"
18        android:textSize="36sp"
19        android:layout_marginBottom="16dp"/>
20
21    <!-- Grid untuk Angka dan Operasi -->
22    <GridLayout
23        android:id="@+id/LayoutGrid"
24        android:layout_width="match_parent"
25        android:layout_height="wrap_content"
26        android:rowCount="5"
27        android:columnCount="4"
28        android:layout_gravity="center"
29        android:layout_marginBottom="16dp">
30
31        <!-- Tombol Angka dan Operasi -->
32        <!-- Baris 1 -->
33        <Button
34            android:id="@+id/btn7"
35            android:text="7"
36            android:layout_row="0"
37            android:layout_column="0"
38            android:layout_width="0dp"
39            android:layout_height="wrap_content"
40            android:layout_columnWeight="1"
41            android:layout_margin="4dp"/>
42
43        <Button
44            android:id="@+id/btn8"
45            android:text="8"
46            android:layout_row="0"
47            android:layout_column="1"
48            android:layout_width="0dp"
49            android:layout_height="wrap_content"
50            android:layout_columnWeight="1"
51            android:layout_margin="4dp"/>
52
53        <Button
54            android:id="@+id/btn9"
55            android:text="9"
56            android:layout_row="0"
57            android:layout_column="2"
58            android:layout_width="0dp"
59            android:layout_height="wrap_content"
60            android:layout_columnWeight="1"
61            android:layout_margin="4dp"/>
```

```
62
63
64 <Button
65     android:id="@+id/btnDivide"
66     android:text="/"
67     android:layout_row="0"
68     android:layout_column="3"
69     android:layout_width="0dp"
70     android:layout_height="wrap_content"
71     android:layout_columnWeight="1"
72     android:layout_margin="4dp"/>
73
74 <!-- Baris 2 -->
75 <Button
76     android:id="@+id/btn4"
77     android:text="4"
78     android:layout_row="1"
79     android:layout_column="0"
80     android:layout_width="0dp"
81     android:layout_height="wrap_content"
82     android:layout_columnWeight="1"
83     android:layout_margin="4dp"/>
84
85 <Button
86     android:id="@+id/btn5"
87     android:text="5"
88     android:layout_row="1"
89     android:layout_column="1"
90     android:layout_width="0dp"
91     android:layout_height="wrap_content"
92     android:layout_columnWeight="1"
93     android:layout_margin="4dp"/>
94
95 <Button
96     android:id="@+id/btn6"
97     android:text="6"
98     android:layout_row="1"
99     android:layout_column="2"
100    android:layout_width="0dp"
101    android:layout_height="wrap_content"
102    android:layout_columnWeight="1"
103    android:layout_margin="4dp"/>
```

```

104 <Button
105     android:id="@+id/btnMultiply"
106     android:text="*"
107     android:layout_row="1"
108     android:layout_column="3"
109     android:layout_width="0dp"
110     android:layout_height="wrap_content"
111     android:layout_columnWeight="1"
112     android:layout_margin="4dp"/>
113
114 <!-- Baris 3 -->
115 <Button
116     android:id="@+id/btn1"
117     android:text="1"
118     android:layout_row="2"
119     android:layout_column="0"
120     android:layout_width="0dp"
121     android:layout_height="wrap_content"
122     android:layout_columnWeight="1"
123     android:layout_margin="4dp"/>
124
125 <Button
126     android:id="@+id/btn2"
127     android:text="2"
128     android:layout_row="2"
129     android:layout_column="1"
130     android:layout_width="0dp"
131     android:layout_height="wrap_content"
132     android:layout_columnWeight="1"
133     android:layout_margin="4dp"/>
134
135 <Button
136     android:id="@+id/btn3"
137     android:text="3"
138     android:layout_row="2"
139     android:layout_column="2"
140     android:layout_width="0dp"
141     android:layout_height="wrap_content"
142     android:layout_columnWeight="1"
143     android:layout_margin="4dp"/>
144
145 <Button
146     android:id="@+id/btnSubtract"
147     android:text="-"
148     android:layout_row="2"
149     android:layout_column="3"
150     android:layout_width="0dp"
151     android:layout_height="wrap_content"
152     android:layout_columnWeight="1"
153     android:layout_margin="4dp"/>
154
155 <!-- Baris 4 -->
156 <Button
157     android:id="@+id/btn0"
158     android:text="0"
159     android:layout_row="3"
160     android:layout_column="0"
161     android:layout_width="0dp"
162     android:layout_height="wrap_content"
163     android:layout_columnWeight="1"
164     android:layout_margin="4dp"/>

```



```

165
166 <Button
167     android:id="@+id/btnPercent"
168     android:text="%"
169     android:layout_row="3"
170     android:layout_column="1"
171     android:layout_width="0dp"
172     android:layout_height="wrap_content"
173     android:layout_columnWeight="1"
174     android:layout_margin="4dp"/>
175
176 <Button
177     android:id="@+id/btnClear"
178     android:text="C"
179     android:layout_row="3"
180     android:layout_column="2"
181     android:layout_width="0dp"
182     android:layout_height="wrap_content"
183     android:layout_columnWeight="1"
184     android:layout_margin="4dp"/>
185
186 <Button
187     android:id="@+id/btnAdd"
188     android:text="+"
189     android:layout_row="3"
190     android:layout_column="3"
191     android:layout_width="0dp"
192     android:layout_height="wrap_content"
193     android:layout_columnWeight="1"
194     android:layout_margin="4dp"/>
195
196 <!-- Baris 5 -->
197 <Button
198     android:id="@+id/btnSquareRoot"
199     android:text="√"
200     android:layout_row="4"
201     android:layout_column="0"
202     android:layout_width="0dp"
203     android:layout_height="wrap_content"
204     android:layout_columnWeight="1"
205     android:layout_margin="4dp"/>
206
207 <Button
208     android:id="@+id/btnSquare"
209     android:text="x2"
210     android:layout_row="4"
211     android:layout_column="1"
212     android:layout_width="0dp"
213     android:layout_height="wrap_content"
214     android:layout_columnWeight="1"
215     android:layout_margin="4dp"/>
216
217 <Button
218     android:id="@+id/btnEquals"
219     android:text="="
220     android:layout_row="4"
221     android:layout_column="2"
222     android:layout_width="0dp"
223     android:layout_height="wrap_content"
224     android:layout_columnSpan="2"
225     android:layout_margin="4dp"
226     android:layout_gravity="fill"/>
227 </GridLayout>
228 </LinearLayout>

```

Penjelasan:

a) LinearLayout (Induk):

- `orientation="vertical"`: Mengatur orientasi layout agar vertikal (komponen satu di bawah yang lain).
- `padding="16dp"`: Memberikan padding di sekeliling layout.

b) TextView untuk Hasil:

- `android:gravity="end"`: Agar teks hasil rata kanan.
- `android:textSize="36sp"`: Ukuran teks besar agar jelas terbaca.
- `android:background="#E0E0E0"`: Warna latar abu-abu muda.

c) GridLayout untuk Tombol:

- `rowCount="5"`: GridLayout memiliki 5 baris (untuk tombol-tombol angka dan operasi).
- `columnCount="4"`: GridLayout memiliki 4 kolom.

d) Button:

Setiap tombol menggunakan `layout_width="0dp"` dan `layout_columnWeight="1"` untuk memastikan semua tombol memiliki ukuran yang sama dan mengisi ruang yang tersedia dengan merata.

e) Tombol Operasi:

Tombol untuk operasi matematika ditempatkan di baris dan kolom yang sesuai.

3) Membuat Membuat Kode Java (MainActivity.java)

Buka file MainActivity.java dan masukkan kode berikut. Kode ini akan menangani logika kalkulator dan interaksi pengguna.

```
package com.example.kalkulator_npmmasingmasing;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private TextView textResult;
    private String operator = "";
    private String currentInput = "";
    private double result = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

setContentView(R.layout.activity_main);

textResult = findViewById(R.id.HasilText);

// Inisialisasi tombol
Button btn0 = findViewById(R.id.btn0);
Button btn1 = findViewById(R.id.btn1);
Button btn2 = findViewById(R.id.btn2);
Button btn3 = findViewById(R.id.btn3);
Button btn4 = findViewById(R.id.btn4);
Button btn5 = findViewById(R.id.btn5);
Button btn6 = findViewById(R.id.btn6);
Button btn7 = findViewById(R.id.btn7);
Button btn8 = findViewById(R.id.btn8);
Button btn9 = findViewById(R.id.btn9);
Button btnAdd = findViewById(R.id.btnAdd);
Button btnSubtract = findViewById(R.id.btnSubtract);
Button btnMultiply = findViewById(R.id.btnMultiply);
Button btnDivide = findViewById(R.id.btnDivide);
Button btnPercent = findViewById(R.id.btnPercent);
Button btnClear = findViewById(R.id.btnClear);
Button btnSquare = findViewById(R.id.btnSquare);
Button btnSquareRoot = findViewById(R.id.btnSquareRoot);
Button btnEquals = findViewById(R.id.btnEquals);

// Set listener untuk setiap tombol angka
btn0.setOnClickListener(view -> appendNumber("0"));
btn1.setOnClickListener(view -> appendNumber("1"));
btn2.setOnClickListener(view -> appendNumber("2"));
btn3.setOnClickListener(view -> appendNumber("3"));
btn4.setOnClickListener(view -> appendNumber("4"));
btn5.setOnClickListener(view -> appendNumber("5"));
btn6.setOnClickListener(view -> appendNumber("6"));
btn7.setOnClickListener(view -> appendNumber("7"));
btn8.setOnClickListener(view -> appendNumber("8"));
btn9.setOnClickListener(view -> appendNumber("9"));

// Set listener untuk setiap operator
btnAdd.setOnClickListener(view -> setOperator("+"));
btnSubtract.setOnClickListener(view -> setOperator("-"));
btnMultiply.setOnClickListener(view -> setOperator("*"));
btnDivide.setOnClickListener(view -> setOperator("/"));
btnPercent.setOnClickListener(view ->
calculatePercent());
btnClear.setOnClickListener(view -> clear());
btnSquare.setOnClickListener(view -> calculateSquare());
btnSquareRoot.setOnClickListener(view ->
calculateSquareRoot());
btnEquals.setOnClickListener(view -> calculateResult());
}

private void appendNumber(String number) {
// Jika hasil sebelumnya adalah 0, reset currentInput
if (currentInput.equals("0")) {
    currentInput = number;
} else {

```

```

        currentInput += number;
    }
    textResult.setText(currentInput);
}

private void setOperator(String op) {
    // Jika ada input, hitung hasilnya sebelum mengatur
operator baru
    if (!currentInput.isEmpty()) {
        if (!operator.isEmpty()) {
            calculateResult(); // Hitung hasil jika operator
sudah ada
        } else {
            result = Double.parseDouble(currentInput);
        }
        currentInput = ""; // Reset currentInput setelah
menyimpan result
    }
    operator = op;
    textResult.setText(result + " " + operator + " ");
}

private void calculateResult() {
    if (!currentInput.isEmpty()) {
        double secondOperand =
Double.parseDouble(currentInput);
        switch (operator) {
            case "+":
                result += secondOperand;
                break;
            case "-":
                result -= secondOperand;
                break;
            case "*":
                result *= secondOperand;
                break;
            case "/":
                if (secondOperand != 0) {
                    result /= secondOperand;
                } else {
                    textResult.setText("Error");
                    return;
                }
                break;
        }
        textResult.setText(String.valueOf(result)); //
Tampilkan hasil
        currentInput = ""; // Reset input untuk operasi
berikutnya
        operator = ""; // Reset operator setelah perhitungan
    }
}

private void calculatePercent() {
    // Hitung persen dari currentInput
    if (!currentInput.isEmpty()) {

```

```

        double inputNumber =
Double.parseDouble(currentInput);
        result = inputNumber / 100; // Persen dihitung dari
input
        textResult.setText(String.valueOf(result)); //
Tampilkan hasil persen
        currentInput = ""; // Reset untuk input berikutnya
    }
}

private void clear() {
    textResult.setText("0");
    operator = "";
    currentInput = "";
    result = 0;
}

private void calculateSquare() {
    // Hitung kuadrat dari currentInput
    if (!currentInput.isEmpty()) {
        result = Math.pow(Double.parseDouble(currentInput),
2);
        textResult.setText(String.valueOf(result));
    }
    currentInput = ""; // Reset untuk input berikutnya
}

private void calculateSquareRoot() {
    // Hitung akar kuadrat dari currentInput
    if (!currentInput.isEmpty()) {
        result = Math.sqrt(Double.parseDouble(currentInput));
        textResult.setText(String.valueOf(result));
    }
    currentInput = ""; // Reset untuk input berikutnya
}
}

```

Penjelasan Kode:

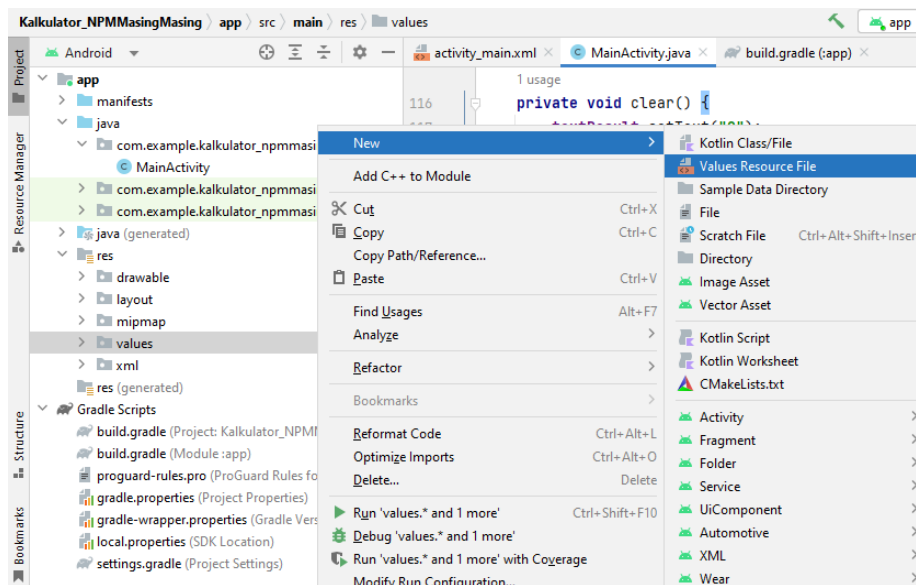
- Penanganan Angka : Metode appendNumber menambahkan angka ke tampilan hasil.
- Set Operator : Metode setOperator menyimpan operator dan operand pertama saat operator ditekan.
- Hitung Hasil : Metode calculateResult menghitung berdasarkan operator yang dipilih.
- Hitung Persen : Metode calculatePercent mengambil nilai dari tampilan hasil, membaginya dengan 100, dan menampilkan hasilnya.

- Tombol C : Metode clear mengatur ulang tampilan hasil dan variabel operator dan operand.
- Hitung Kuadrat dan Akar: Metode untuk menghitung kuadrat dan akar dari nilai yang ditampilkan.

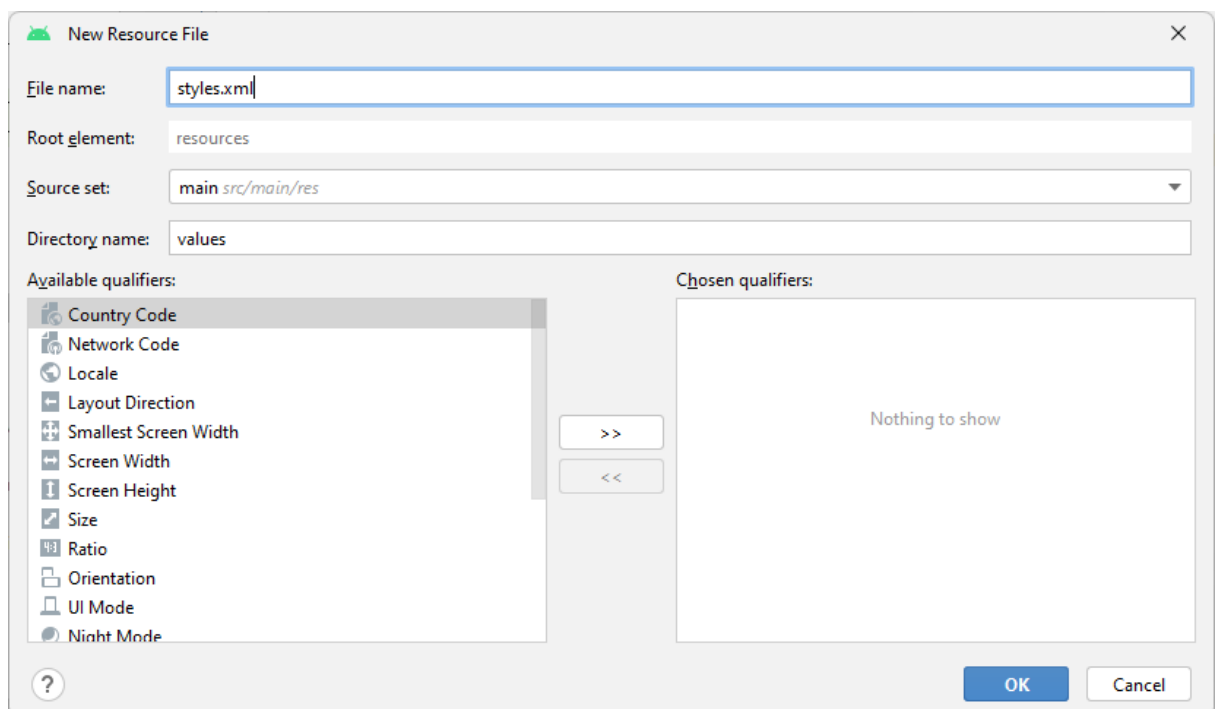
4) Masukkan Style ke dalam styles.xml

Adapun caranya sebagai berikut :

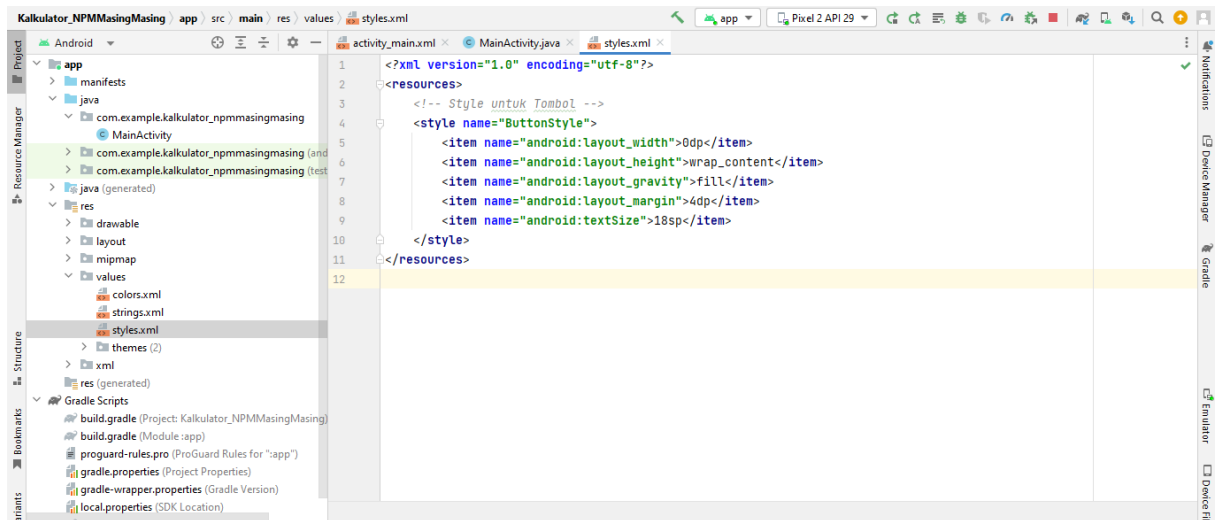
- ✓ Pada folder res/values/ klik kanan kemudian pilih New → Values Resource File



- ✓ Pada bagian File name isikan dengan styles.xml kemudian klik OK



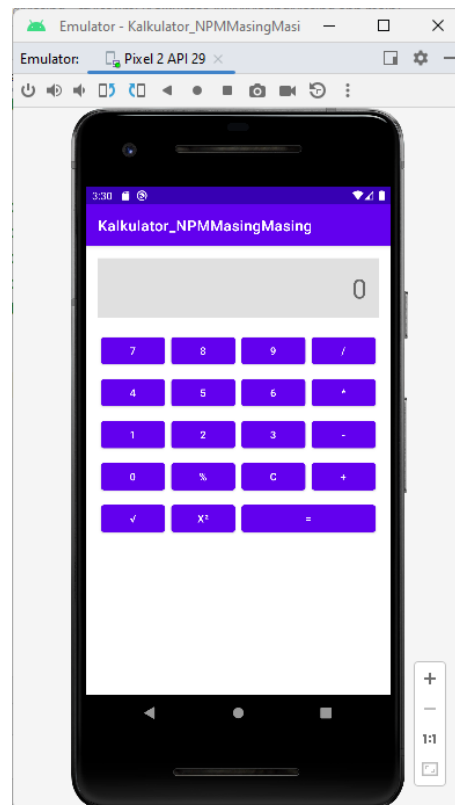
- ✓ Ubah kode program menjadi seperti berikut :



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <!-- Style untuk Tombol -->
  <style name="ButtonStyle">
    <item name="android:layout_width">0dp</item>
    <item name="android:layout_height">wrap_content</item>
    <item name="android:layout_gravity">fill</item>
    <item name="android:layout_margin">4dp</item>
    <item name="android:textSize">18sp</item>
  </style>
</resources>
```

5) Menjalankan Aplikasi

- Hubungkan perangkat Android atau buka Android Emulator.
- Klik tombol Run di Android Studio.
- Pilih perangkat yang ingin digunakan untuk menjalankan aplikasi.
- Aplikasi kalkulator sederhana akan ditampilkan.



4.5 TUGAS

1. Demokan project tersebut hingga muncul di emulator!
2. Tambahkan tombol aksi untuk menghitung :
 - a. Modulus
 - b. Sin, Cos, Tan
3. Tugas dikumpul dalam format pdf dengan nama file : NPM_Tugaske?
4. Didalam file tersebut dituliskan :
 - NPM
 - Nama Lengkap
 - Program Studi
 - Kelas