

MODUL PRAKTIKUM

PEMROGRAMAN PERANGKAT BERGERAK

MODUL VII

DESAIN USER INTERFACE DENGAN VIEW



Disusun Oleh :

Munengsih Sari Bunga

PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU

2017

MODUL VII

DESAIN USER INTERFACE DENGAN VIEW

A. TUJUAN

Pada modul 7, akan membahas secara detail hal-hal yang diperlukan dalam rangka membuat aplikasi smartphone berbasis Android, dengan tujuan agar mahasiswa :

- Dapat menggunakan view dasar pada Android untuk merancang UI.
- Dapat menggunakan Picker View untuk menampilkan daftar item.
- Dapat menggunakan List View untuk menampilkan daftar item.
- Dapat menggunakan fragmen khusus.

B. TEORI & PRAKTEK

- Dalam modul ini mencakup ViewGroup berikut:
 - Basic View → pada umumnya view yang digunakan seperti TextView, EditText dan Button view.
 - Picker View → View yang memungkinkan user untuk memilih daftar, seperti TimePicker dan DatePicker.
 - List View → View yang menampilkan daftar item yang panjang, seperti ListView dan SpinnerView.
 - Specialized Fragment → Fragmen khusus yang melakukan fungsi tertentu.

A. BASIC VIEW

- Basic view memungkinkan untuk menampilkan informasi teks, serta melakukan beberapa seleksi dasar.
- berikut adalah beberapa view dari basic view:
 - TextView
 - EditText
 - Button
 - ImageButton
 - CheckBox
 - ToggleButton
 - RadioButton
 - RadioGroup

1. TextView View

- Ketika membuat Project Android baru, Android Studio menciptakan file activity_main.xml (terletak di folder res/layout), yang berisi elemen <TextView>:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />
</LinearLayout>
```

- TextView digunakan untuk menampilkan teks ke user. merupakan view yang paling dasar dan salah satu view yang sering digunakan untuk pengembangan aplikasi Android.
- Atribut layout_width dapat diatur dengan beberapa cara sebagai berikut:
 - o fill_parent : yang membuat lebarnya menempati seluruh lebar layar.
 - o wrap_content : lebarnya akan menjadi lebar yang sesuai dengan isinya.

2. Button, ImageButton, EditText, CheckBox, ToggleButton, RadioButton, dan RadioGroup View

- View lain yang sering digunakan adalah sebagai berikut:
 - o Button → merupakan widget push-button.
 - o ImageButton → mirip dengan Button view, tetapi ImageButton dapat menampilkan gambar.
 - o EditText → sebuah subclass dari view TextView, yang memungkinkan user untuk mengedit konten teks-nya.
 - o CheckBox → Jenis tombol khusus yang memiliki dua keadaan, yaitu check atau uncheck.
 - o RadioGroup dan RadioButton → RadioButton memiliki dua keadaan, yaitu check atau uncheck. sebuah Radiogroup digunakan untuk mengelompokkan satu atau lebih view RadioButton sehingga memungkinkan hanya satu RadioButton yang di-check dalam RadioGroup.
 - o ToggleButton → menampilkan check/uncheck menggunakan lampu indikator.

LATIHAN 1 (PENGGUNAAN BASIC VIEW)

1. Buka Android Studio, buat project Android baru dan beri nama **BasicView1**.
2. modifikasi file activity_main.xml yang terletak di folder res/layout dengan menambahkan unsur-unsur berikut ditampilkan dalam huruf tebal berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <Button android:id="@+id/btnSave"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="save" />
    <Button android:id="@+id/btnOpen"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Open" />
    <ImageButton android:id="@+id/btnImg1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:src="@mipmap/ic_launcher" />
    <EditText android:id="@+id/txtName"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <CheckBox android:id="@+id/chkAutosave"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Autosave" />
    <CheckBox android:id="@+id/star"
        style="?android:attr/starStyle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <RadioGroup android:id="@+id/rdbGp1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

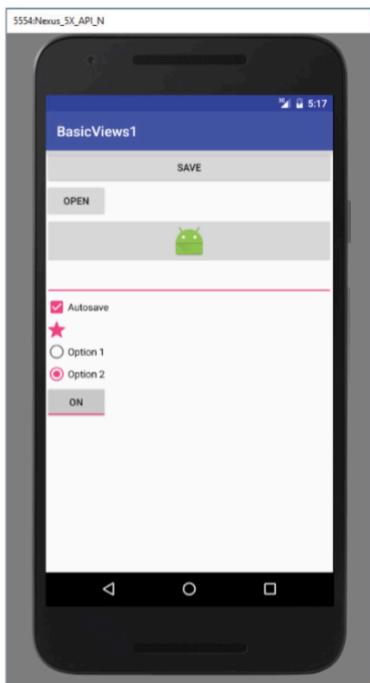
        <RadioButton android:id="@+id/rdb1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Option 1" />
        <RadioButton android:id="@+id/rdb2"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Option 2" />
    </RadioGroup>
    <ToggleButton android:id="@+id/toggle1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>
```

3. Untuk melihat view pada action, debug project di Android Studio dengan menekan Shift+F9. Gambar berikut menunjukkan berbagai view yang ditampilkan pada emulator Android.



4. Klik masing-masing view dan perhatikan bagaimana view memiliki tampilan dan nuansa yang berbeda. Gambar berikut menunjukkan perubahan tampilan tersebut.

- a. CheckBox view (autosave) pertama di-check.
- b. ChechkBox view kedua (star) dipilih.
- c. RadioButton kedua (Option 2) dipilih.
- d. ToggleButton diaktifkan.



LATIHAN 2 (HANDLING VIEW EVENT)

1. Gunakan project **BasicView1** yang sudah dibuat sebelumnya, modifikasi file `MainActivity.java` dengan menambahkan pernyataan huruf tebal berikut:

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.RadioButton;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //---Button view---
        Button btnOpen = (Button) findViewById(R.id.btnOpen);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                DisplayToast("You have clicked the Open button");
            }
        });
    }
}
```

```

    //---Button view---
    Button btnSave = (Button) findViewById(R.id.btnSave);
    btnSave.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v) {
            DisplayToast("You have clicked the Save button");
        }
    });
    //---CheckBox---
    CheckBox checkBox = (CheckBox) findViewById(R.id.chkAutosave);
    checkBox.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v) {
            if (((CheckBox)v).isChecked())
                DisplayToast("CheckBox is checked");
            else
                DisplayToast("CheckBox is unchecked");
        }
    });
    //---RadioButton---
    RadioGroup radioGroup = (RadioGroup) findViewById(R.id.rdbGp1);
    radioGroup.setOnCheckedChangeListener(
        new RadioGroup.OnCheckedChangeListener()
    {
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            RadioButton rbl = (RadioButton) findViewById(R.id.rdb1);
            if (rbl.isChecked()) {
                DisplayToast("Option 1 checked!");
            } else {
                DisplayToast("Option 2 checked!");
            }
        }
    });
    //---ToggleButton---
    ToggleButton toggleButton =
        (ToggleButton) findViewById(R.id.toggle1);
    toggleButton.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View v) {
            if (((ToggleButton)v).isChecked())
                DisplayToast("Toggle button is On");
            else
                DisplayToast("Toggle button is Off");
        }
    });
    private void DisplayToast(String msg)
    {
        Toast.makeText(getApplicationContext(), msg,
            Toast.LENGTH_SHORT).show();
    }
}

```

2. Tekan Shift+F9 untuk debug project pada emulator Android.
3. Klik masing-masing view dan amati pesan yang ditampilkan di jendela Toast.

LATIHAN 3 (PENGGUNAAN PROGRESSBAR VIEW)

1. Buka Android Studio, buat project Android baru dan beri nama **BasicView2**.
2. Modifikasi file activity_main.xml yang terletak di folder res/layout dengan menambahkan kode berikut dalam huruf tebal:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
<ProgressBar android:id="@+id/progressbar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>

```

3. Di file MainActivity.java, tambahkan pernyataan dengan huruf tebal berikut:

```

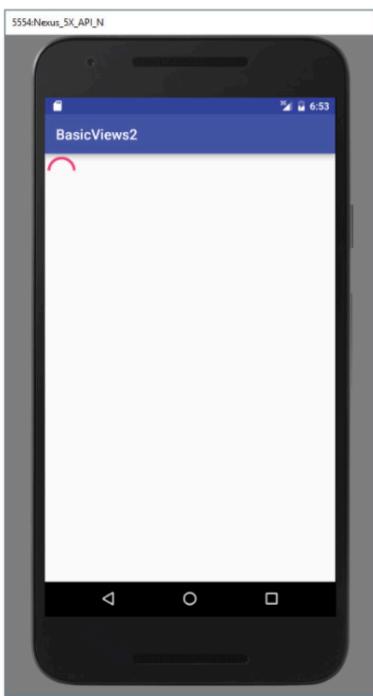
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ProgressBar;

public class MainActivity extends AppCompatActivity {
    static int progress;
    ProgressBar progressBar;
    int progressStatus = 0;
    Handler handler = new Handler();
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        progress = 0;
        progressBar = (ProgressBar) findViewById(R.id.progressbar);
        //---do some work in background thread---
        new Thread(new Runnable()
        {
            public void run()
            {
                //---do some work here---
                while (progressStatus < 10)
                {
                    progressStatus = doSomeWork();
                }
                //---hides the progress bar---
                handler.post(new Runnable()
                {
                    public void run()
                    {
                        //---0 - VISIBLE; 4 - INVISIBLE; 8 - GONE---
                        progressBar.setVisibility(View.GONE);
                    }
                });
            }
        });
        //---do some long running work here---
        private int doSomeWork()
        {
            try {
                //---simulate doing some work---
                Thread.sleep(500);
            } catch (InterruptedException e)
            {
                e.printStackTrace();
            }
            return ++progress;
        }
    }).start();
}

```

4. Tekan Shift+F9 untuk debug project pada emulator Android. Gambar di bawah ini menunjukkan animasi ProgressBar. setelah sekitar lima detik, akan hilang.



5. Ketika task selesai, ProgressBar akan disembunyikan dengan menetapkan property Visibility ke View.GONE (value 8). Ada dua perbedaan antara konstan INVISIBLE dan GONE, yaitu:
- a. INVISIBLE konstan hanya menyembunyikan ProgressBar (wilayah yang diduduki oleh ProgressBar masih mengambil ruang dalam Activity).
 - b. GONE konstan menghilangkan ProgressBar view dari activity dan tidak mengambil ruang apapun.

LATIHAN 4 (CUSTOMIZING PROGRESSBAR VIEW)

1. Gunakan project **BasicView2** yang dibuat sebelumnya, modifikasi file activity_main.xml seperti yang ditunjukkan di bawah ini:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <ProgressBar android:id="@+id/progressbar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        style="@android:style/Widget.ProgressBar.Horizontal" />
</LinearLayout>
```

2. Modifikasi file MainActivity.java dan tambahkan pernyataan yang ditebal berikut:

```
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ProgressBar;

public class MainActivity extends AppCompatActivity {
```

```

static int progress;
ProgressBar progressBar;
int progressStatus = 0;
Handler handler = new Handler();
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    progress = 0;
    progressBar = (ProgressBar) findViewById(R.id.progressbar);
    progressBar.setMax(200);

    //---do some work in background thread---
    new Thread(new Runnable()
    {
        public void run()
        {
            //---do some work here---
            while (progressStatus < 100)
            {
                progressStatus = doSomeWork();
                //---Update the progress bar---
                handler.post(new Runnable()
                {
                    public void run() {
                        progressBar.setProgress(progressStatus);
                    }
                });
            }
            //---hides the progress bar---
            handler.post(new Runnable()
            {
                public void run()
                {
                    //---0 - VISIBLE; 4 - INVISIBLE; 8 - GONE---
                    progressBar.setVisibility(View.GONE);
                }
            });
        }
    });
    //---do some long running work here---
    private int doSomeWork()
    {
        try {
            //---simulate doing some work---
            Thread.sleep(500);
        } catch (InterruptedException e)
        {
            e.printStackTrace();
        }
        return ++progress;
    }
}).start();
}
}

```

3. Tekan Shift+F9 untuk debug project pada emulator Android.
4. Gambar di bawah ini menunjukkan ProgressBar menampilkan progress. ProgressBar akan menghilang ketika progress mencapai 50 persen.

