

# Pertemuan 4

## Selection Widget

### A. Materi

- ListView
- Spinner
- AutoComplete TextView
- GalleryView
- GridView

### B. Kompetensi Dasar

- Mengenal macam selection widget
- Membuat aplikasi dengan selection widget

### C. Tujuan

- Mahasiswa mengenal dan bisa menerapkan Selection widget pada aplikasi android

### D. Dasar Teori

Selection views adalah widget yang menyediakan list berupa pilihan-pilihan yang menggunakan framework dari class adapter. Android menyediakan framework data adapter yang menyediakan untuk membantu membangun selection widget dengan data yang berasal baik dari array maupun dari database.

Berikut cara menggunakan class adapter :

```
new ArrayAdapter<String>
(this, android.R.layout.simple_list_item_1, items);
```

Beberapa Selection Widgget antara lain :

- ListView
- Spinner
- AutoComplete TextView
- GalleryView
- GridView
- 

#### 1. ListView

ListView menampilkan daftar item vertical yang dapat di scroll apabila data yang ditampilkan melebihi besarnya layar Samahalnya komponen widget lainnya, membuat listView bisa dilakukan dengan men-drag komponen ke dalam layout atau dengan menuliskan source code pada file .xml seperti dibawah ini :

Pada source code dibawah ini simpan pada file *example\_listview.xml*

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/android:list"
        android:layout_weight="1" />
</LinearLayout>
```

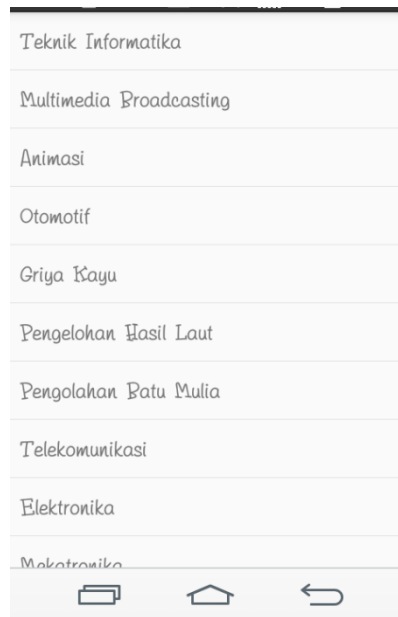
Kemudian tambahkan koding pada activity file java yang handle layout example\_listview.xml

```
import android.app.ListActivity;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends ListActivity {
    String[] about={"Teknik Informatika","Multimedia Broadcasting",
"Animasi","Otomotif","Griya Kayu","Pengelohan Hasil Laut",
"Pengolahan Batu Mulia","Telekomunikasi","Elektronika","Mekatronika",
"Kimia","Fisika","Matematika","Perkapalan"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.example_listview);
        setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, about));
        ListView lv = getListView();
        public void onItemClick(ListView parent, View v,int position, long id) {
            Toast.makeText(this, about[position], 5).show();
        }
    }
}
```

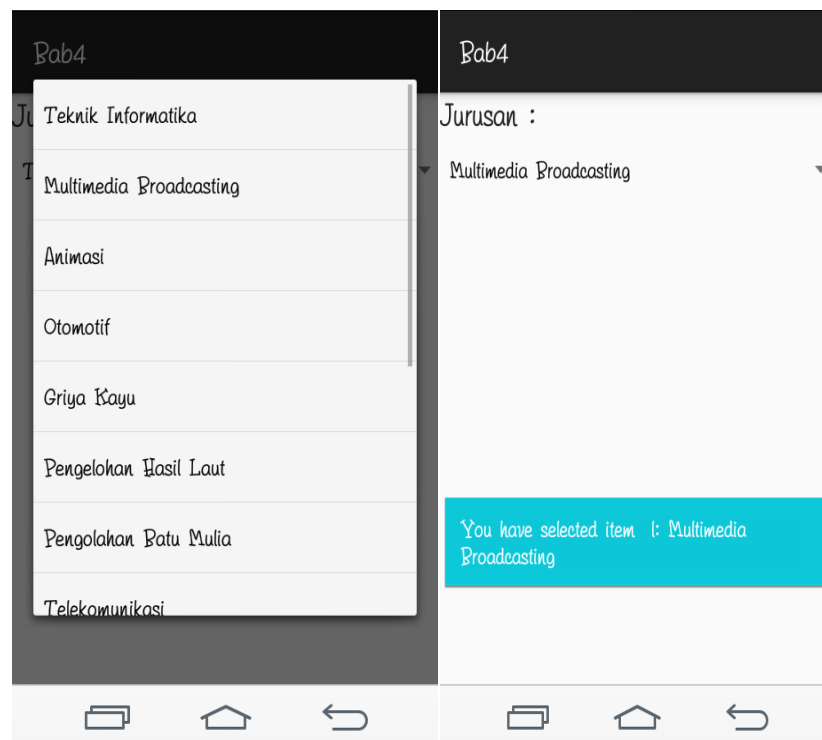
Hasil pada **Gambar 4.1.**



**Gambar 4.1.** ListView

## 2. Spinner

Spinner View menampilkan daftar item yang salah satunya bisa dipilih oleh user. Pada Spinner juga membutuhkan adapter untuk menghubungkan data menggunakan `setAdapter()`. Sedangkan untuk menambahkan action listener apabila elemen spinner dipilih dengan menggunakan `setOnItemSelectedListener()`. Tampilan spinner terlihat pada **Gambar 4.2.**



**Gambar 4.2.** Spinner

membuat Spinner bisa dilakukan dengan men-drag komponen ke dalam layout atau dengan menuliskan source code pada file .xml seperti dibawah ini :

Pada source code dibawah ini simpan pada file *examplespinner.xml*

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

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Jurusan :"
        android:id="@+id/textView" />

    <Spinner
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/spinner"
        android:spinnerMode="dialog"
        />

</LinearLayout>
```

Kemudian tambahkan koding pada activity file java yang handle layout *examplespinner.xml*

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
public class Spinnerku extends AppCompatActivity {
    Spinner s;
    String[] jurusan={"Teknik Informatika","Multimedia
Broadcasting","Animasi","Otomotif","Griya Kayu","Pengelohan Hasil
Laut","Pengolahan Batu
Mulia","Telekomunikasi","Elektronika","Mekatronika","Kimia","Fisika","Ma
tematika","Perkapalan"};

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout. examplespinner.xml);
        s=(Spinner) findViewById(R.id.spinner);
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_dropdown_item,
jurusan);
        s.setAdapter(adapter);
        s.setOnItemClickListener(new
AdapterView.OnItemClickListener()
        {
            public void onItemClick(AdapterView<?> arg0,View arg1,
int arg2, long arg3)
            {
                int index = s.getSelectedItemPosition();
                Toast.makeText(getApplicationContext(), "You have selected item
1: " + jurusan[index], Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

```

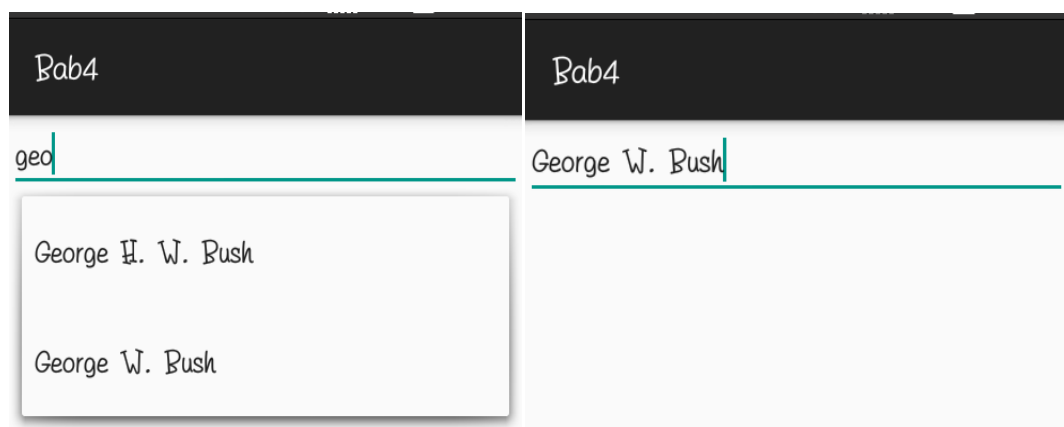
    }

    public void onNothingSelected(AdapterView<?> arg0) {}
    });
}
}

```

### 3. AutoComplete TextView

AutoComplete adalah text yang terlatih untuk menebak huruf pertama yang diinputkan user. Akan dibandingkan text yang diinputkan dengan daftar kandidat yang telah ada pada system. AutoCompleteTextView memiliki properti ***android:completionThreshold*** yang mengidentifikasi berapa banyak kata yang harus diinputkan sebelum dilakukan daftar filtering seperti **Gambar 4.3**.



**Gambar 4.3.** AutoComplete TextView

Membuat AutoComplete TextView bisa dilakukan dengan men-drag komponen ke dalam layout atau dengan menuliskan source code pada file .xml seperti dibawah ini :

Pada source code dibawah ini simpan pada file **exautocomplete.xml**

```

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

    <AutoCompleteTextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:id="@+id/autoCompleteTextView"
        android:completionThreshold="3"
    />
</LinearLayout>

```

Kemudian tambahkan koding pada activity file java yang handle layout **exautocomplete.xml**

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
public class AutoCompletekku extends AppCompatActivity {
    String[] presidents =
        {"Dwight D. Eisenhower", "John F. Kennedy",
         "Lyndon B. Johnson", "Richard Nixon",
         "Gerald Ford", "Jimmy Carter",
         "Ronald Reagan", "George H. W. Bush",
         "Bill Clinton", "George W. Bush",
         "Barack Obama"};
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.exautocomplete);
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(this, android.R.layout.simple_dropdown_item_1line,
presidents);
        AutoCompleteTextView textView = (AutoCompleteTextView)
            findViewById(R.id.autoCompleteTextView);
        textView.setAdapter(adapter);
    }
}
```

#### 4. GalleryView

Widget Gallery menyediakan set options untuk memilih image. Image yang dipilih disediakan dalam menu horizontal yang bisa digeser . Programmer harus menggunakan ImageAdapter untuk menandakan ketika user mengklik image.

Membuat AutoComplete TextView bisa dilakukan dengan men-drag komponen ke dalam layout atau dengan menuliskan source code pada file .xml seperti dibawah ini :

Pada source code dibawah ini simpan pada file **galleryku.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Images of San Francisco" />
    <Gallery
        android:id="@+id/gallery1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" />
    <ImageView
        android:id="@+id/image1"
        android:layout_width="320px"
        android:layout_height="250px"
        android:scaleType="fitXY" />
</LinearLayout>
```

Kemudian tambahkan koding pada activity file java yang menghandle layout galleryku.xml

```
import android.content.Context;
import android.content.res.TypedArray;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.Adapter;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

public class Galleryku extends AppCompatActivity {
    Integer[] imageIDs = {
        R.drawable.pic1,
        R.drawable.pic2,
        R.drawable.pic3,
        R.drawable.pic4,
        R.drawable.pic5,
        R.drawable.pic6,
        R.drawable.pic7
    };

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.galleryku);
        Gallery gallery = (Gallery) findViewById(R.id.gallery1);

        gallery.setAdapter(new ImageAdapter(this));
        gallery.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView parent,
                                   View v, int position, long id) {
                Toast.makeText(getApplicationContext(), "pic" + (position + 1) +
" selected", Toast.LENGTH_SHORT).show();
                //---display the images selected---
                ImageView imageView = (ImageView)
findViewById(R.id.image1);
                imageView.setImageResource(imageIDs[position]);
            }
        });
    }

    public class ImageAdapter extends BaseAdapter
    {
        private Context context;
        private int itemBackground;

        public ImageAdapter(Context c)
        {
            context = c;
            //---setting the style---
            TypedArray a = obtainStyledAttributes(R.styleable.Gallery1);
            itemBackground = a.getResourceId(
R.styleable.Gallery1_android_galleryItemBackground, 0);
            a.recycle();
        }
    }
}
```

```

//---returns the number of images---
public int getCount() {
    return imageIDs.length;
}

//---returns the ID of an item---
public Object getItem(int position) {
    return position;
}

public long getItemId(int position) {
    return position;
}

//---returns an ImageView view---
public View getView(int position, View convertView, ViewGroup
parent) {
    ImageView imageView = new ImageView(context);
    imageView.setImageResource(imageIDs[position]);
    imageView.setScaleType(ImageView.ScaleType.FIT_XY);
    imageView.setLayoutParams(new Gallery.LayoutParams(200,
120));
    imageView.setBackgroundResource(itemBackground);
    return imageView;
}
}

```

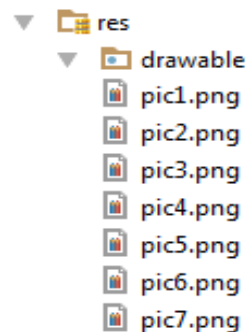
Menambahkan file baru **attrs.xml** pada folder res/value

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <declare-styleable name="Gallery1">
        <attr name="android:galleryItemBackground" />
    </declare-styleable>
</resources>

```

Kemudian tambahkan image-image yang akan ditampilkan pada folder /res/drawable. Seperti **Gambar 4.4**.



**Gambar 4.4.** Image pada folder drawable



## 5. GridView

GridView menampilkan view pada two-dimensional scrolling grid. Bisa menggunakan GridView bersama dengan ImageView untuk menampilkan image-image

Membuat GridView bisa dilakukan dengan men-drag komponen ke dalam layout atau dengan menuliskan source code pada file .xml seperti dibawah ini :

Simpan dengan nama **displayview.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <GridView
        android:id="@+id/gridview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:numColumns="auto_fit"
        android:verticalSpacing="2dp"
        android:horizontalSpacing="2dp"
        android:columnWidth="90dp"
        android:stretchMode="columnWidth"
        android:gravity="center"
    />
</LinearLayout>
```

Kemudian tambahkan koding pada activity file java yang menghandle layout galleryku.xml. Didalam GridViewku.java terdapat class ImageAdapter

```
import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.Toast;

public class GridViewku extends AppCompatActivity {
    Integer[] imageIDs = {
        R.drawable.android,
        R.drawable.download,
        R.drawable.images,
        R.drawable.images2,
        R.drawable.images3,
        R.drawable.pic1,
        R.drawable.pic2,
        R.drawable.pic3,
        R.drawable.pic4,
        R.drawable.pic5,
        R.drawable.pic6,
        R.drawable.pic7
    };

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

```

        setContentView(R.layout.displayview);
        GridView gridView = (GridView) findViewById(R.id.gridview);
        gridView.setAdapter(new ImageAdapter(this));

        gridView.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            public void onItemClick(AdapterView parent,
                                   View v, int position, long id) {
                Toast.makeText(getBaseContext(), "pic" + (position + 1) + ""
selected", Toast.LENGTH_SHORT).show();
            }
        });
    }

public class ImageAdapter extends BaseAdapter
{
    private Context context;
    public ImageAdapter(Context c)
    {
        context = c;
    }
    //---returns the number of images---
    public int getCount() {
        return imageIDs.length;
    }
    //---returns the ID of an item---
    public Object getItem(int position) {
        return position;
    }
    public long getItemId(int position) {
        return position;
    }

    //---returns an ImageView view---
    public View getView(int position, View convertView, ViewGroup
parent)
    {
        ImageView imageView;
        if (convertView == null) {
            imageView = new ImageView(context);
            imageView.setLayoutParams(new GridView.LayoutParams(130,
130));
            imageView.setScaleType(ImageView.ScaleType.CENTER_CROP);
            imageView.setPadding(2, 2, 2, 2);
        } else {
            imageView = (ImageView) convertView;
        }
        imageView.setImageResource(imageIDs[position]);
        return imageView;
    }
}

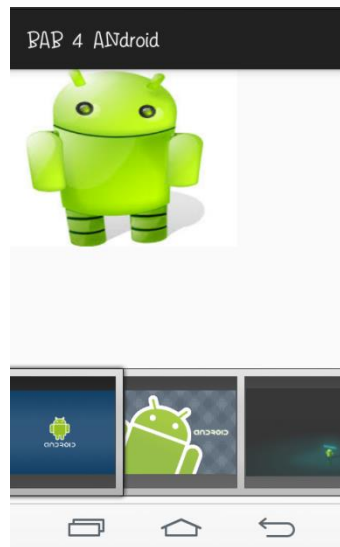
```

## E. Percobaan

Lakukan step-step pada dasar teori

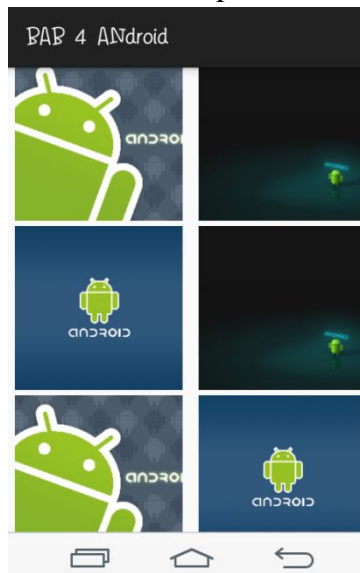
## F. Praktikum

1. Pada Gallery, edit file .xml anda. Jadikan tampilan Gallery berada pada posisi bottom seperti gambar berikut! Dan setting image supaya terlihat lebih besar! Hasil seperti **Gambar 4.5**.



**Gambar 4.5.** Hasil praktikum 4 no 1

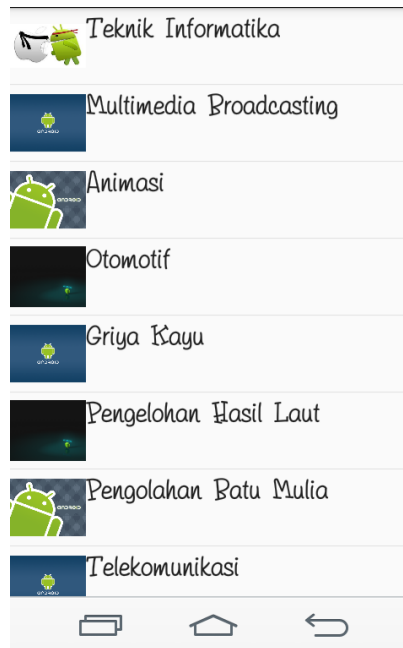
2. Rubah tampilan GridView pada dasar teori seperti **Gambar 4.6**.



**Gambar 4.6.** Hasil praktikum 4 no 2

## G. STUDI KASUS

Bagaimana membuat ListView dengan tampilan seperti **Gambar 4.7**.



**Gambar 4.7.** Hasil studi kasus praktikum 4