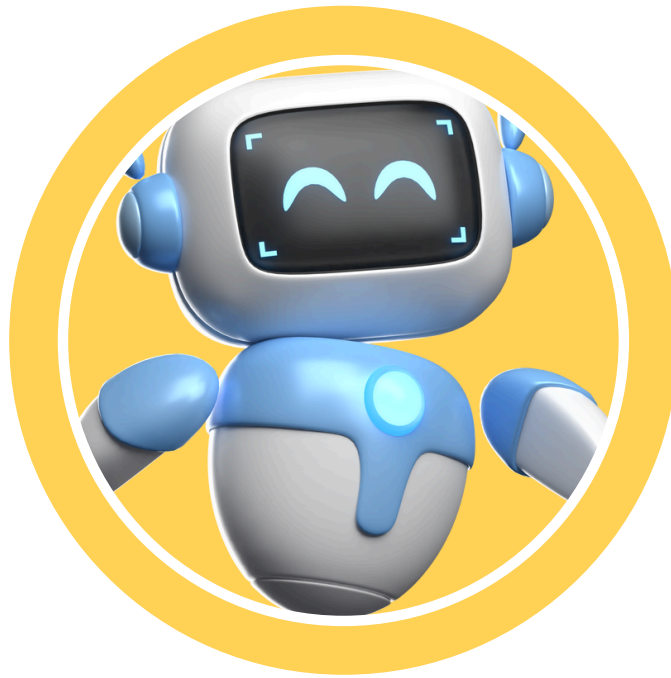




STMIK KAPUTAMA

PRAKTIKUM PEMROGRAMAN BERGERAK



Disusun Oleh:

Kristina Annatasia Br Sitepu, M.Kom
Ratih Puspadini, ST., M.Kom

PERTEMUAN 5 WIDGET VIEW (LANJUTAN)

5.1 TUJUAN PEMBELAJARAN :

- a. Menggunakan dan mengonfigurasi komponen UI (User Interface) lanjutan di Android dengan Java.
- b. Mengembangkan aplikasi Android yang interaktif dengan menggunakan berbagai widget view tersebut.

5.2 ALAT DAN BAHAN :

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

5.3 MATERI

Dalam pengembangan aplikasi Android, berbagai macam widget view atau komponen antarmuka disediakan oleh Android SDK. Widget lanjutan seperti `AutoCompleteTextView`, `MultiAutoCompleteTextView`, `ListView`, `GridView`, dan lainnya sangat membantu untuk membangun UI yang dinamis dan interaktif. Melalui praktikum ini, kita akan mempelajari cara menggunakan widget tersebut di Java.

1. `AutoCompleteTextView`

`AutoCompleteTextView` adalah subclass dari `EditText` yang menyediakan kemampuan untuk menampilkan daftar saran otomatis ketika pengguna mengetikkan karakter di dalamnya. Widget ini sangat berguna ketika kita ingin memberikan saran atau masukan yang disarankan berdasarkan input pengguna, seperti nama negara, kota, atau produk.

`AutoCompleteTextView` memudahkan pengguna untuk mengisi teks secara otomatis dari pilihan yang disediakan dalam daftar. Hal ini meningkatkan efisiensi pengguna saat memasukkan data.

Atribut yang Digunakan:

1. `android:completionThreshold`

Menentukan berapa banyak karakter yang harus diketik sebelum daftar saran muncul.

Contoh: `android:completionThreshold="1"` berarti saran akan muncul setelah pengguna mengetik 1 karakter.

2. android:dropDownWidth

Menentukan lebar dropdown yang muncul.

Contoh: android:dropDownWidth="match_parent" untuk menyesuaikan lebar dropdown dengan lebar parent.

3. android:dropDownHeight

Menentukan tinggi dropdown yang muncul.

Contoh: android:dropDownHeight="200dp" untuk membatasi tinggi dropdown.

4. android:hint

Menampilkan teks petunjuk di dalam input ketika field kosong.

Contoh: android:hint="Ketik nama negara"

Cara Kerja:

AutoCompleteTextView membutuhkan adapter (ArrayAdapter atau CursorAdapter) untuk memasok daftar pilihan. Saat pengguna mulai mengetik, widget akan menampilkan item yang cocok berdasarkan input dari pengguna.

Contoh penerapan :

XML Layout:

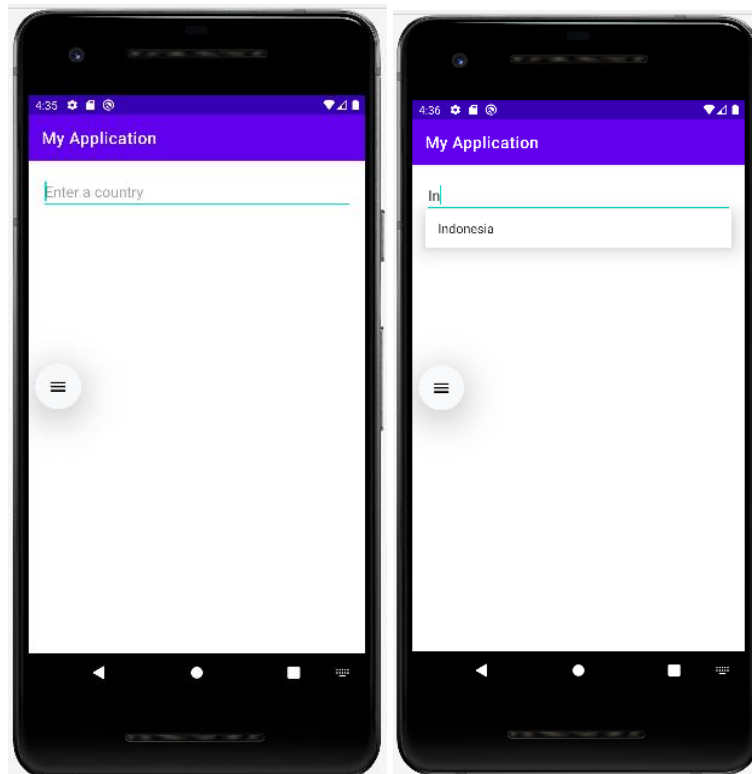
```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <AutoCompleteTextView
    android:id="@+id/autoCompleteTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter a country"
    android:completionThreshold="1" />
</LinearLayout>
```

Java Code :

```
AutoCompleteTextView autoCompleteTextView = findViewById(R.id.autoCompleteTextView);
String[] countries = {"Indonesia", "Malaysia", "Singapore", "Thailand", "Vietnam"};
ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, countries);
autoCompleteTextView.setAdapter(adapter);
```

Output :



2. MultiAutoCompleteTextView

MultiAutoCompleteTextView adalah versi lanjutan dari AutoCompleteTextView, yang memungkinkan pengguna memasukkan beberapa item yang dipisahkan dengan tanda pemisah, seperti koma. Setiap item yang dimasukkan oleh pengguna akan disarankan dari daftar pilihan yang disediakan.

Digunakan ketika pengguna diharapkan untuk memasukkan lebih dari satu nilai, misalnya beberapa email, tag, atau kata kunci.

Atribut yang Digunakan:

1. android:completionThreshold

Sama seperti AutoCompleteTextView, menentukan berapa banyak karakter yang harus diketik sebelum saran muncul.

2. android:inputType

Menentukan tipe input yang diharapkan, misalnya teks, angka, atau email.

Contoh: android:inputType="text"

Cara Kerja:

- Setiap kali pengguna mengetikkan teks yang dipisahkan dengan koma (atau delimiter lain yang ditentukan), widget akan menampilkan daftar saran untuk item berikutnya.

Contoh penerapan :

XML Layout :

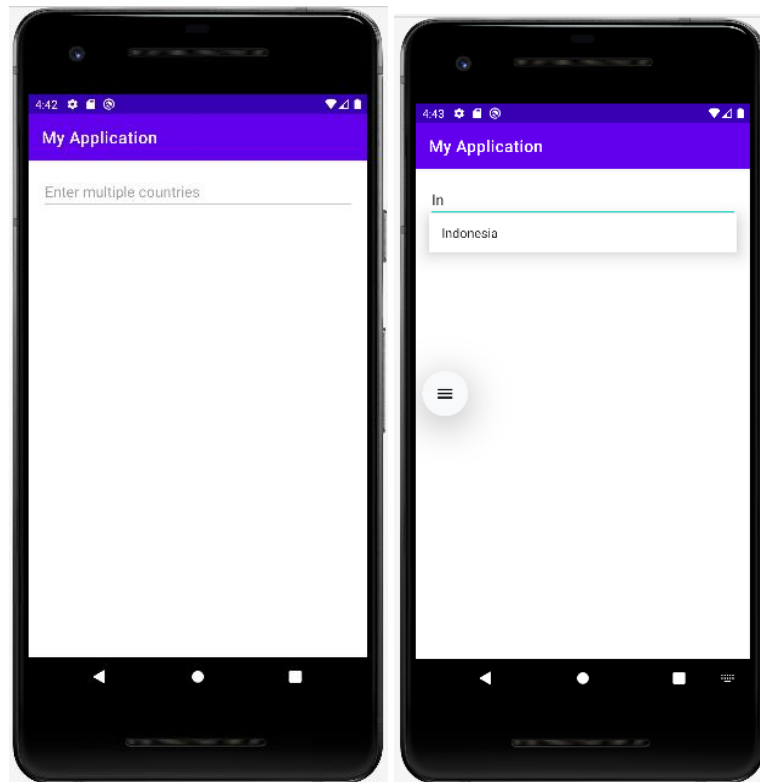
```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <MultiAutoCompleteTextView
    android:id="@+id/multiAutoCompleteTextView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter multiple countries"
    android:completionThreshold="1" />
</LinearLayout>
```

Java Code :

```
MultiAutoCompleteTextView multiAutoCompleteTextView =
findViewById(R.id.multiAutoCompleteTextView);
String[] countries = {"Indonesia", "Malaysia", "Singapore", "Thailand", "Vietnam"};
ArrayAdapter<String> adapterMulti = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, countries);
multiAutoCompleteTextView.setAdapter(adapterMulti);
multiAutoCompleteTextView.setTokenizer(new
MultiAutoCompleteTextView.CommaTokenizer());
```

Output :



3. ListView

ListView adalah widget yang menampilkan daftar item secara vertikal. Setiap item dalam daftar dapat di-scroll jika jumlah item lebih banyak dari tinggi tampilan.

ListView digunakan untuk menampilkan data dalam bentuk daftar yang dinamis, misalnya daftar kontak, daftar produk, atau daftar pesan.

Atribut yang Digunakan:

1. `android:divider`
Menentukan elemen pembatas antara item dalam daftar.
Contoh: `android:divider="@android:color/darker_gray"`
2. `android:dividerHeight`
Menentukan tinggi pembatas antar item.
Contoh: `android:dividerHeight="2dp"`
3. `android:choiceMode`
Menentukan apakah pengguna dapat memilih satu atau lebih item dari daftar.
Contoh: `android:choiceMode="singleChoice"` untuk memilih satu item atau `android:choiceMode="multipleChoice"` untuk memilih banyak item.

Cara Kerja:

- ListView memerlukan adapter seperti ArrayAdapter atau CustomAdapter untuk menghubungkan data ke tampilan daftar. Setiap item dalam daftar diwakili oleh elemen tampilan yang dapat berupa teks, gambar, atau kombinasi keduanya.

Contoh penerapan :

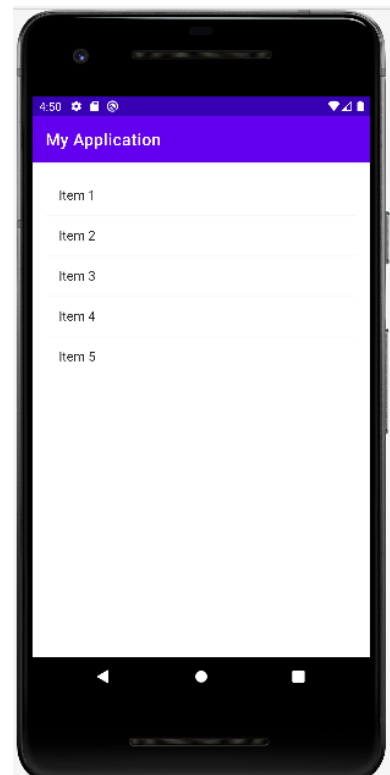
XML Layout :

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Java Code :

```
ListView listView = findViewById(R.id.listView);
String[] items = {"Item 1", "Item 2", "Item 3", "Item 4",
"Item 5"};
ArrayAdapter<String> adapterList = new
ArrayAdapter<>(this, android.R.layout.simple_list_item_1,
items);
listView.setAdapter(adapterList);
```



4. GridView

GridView adalah widget yang menampilkan kumpulan data dalam bentuk grid (tabel) yang dapat di-scroll. Item ditampilkan dalam beberapa kolom.

GridView cocok digunakan untuk menampilkan data dalam bentuk tabel, seperti galeri gambar atau daftar produk dengan gambar dan teks.

Atribut yang Digunakan:

1. android:numColumns

Menentukan jumlah kolom yang ditampilkan dalam grid.

Contoh: android:numColumns="3" untuk tiga kolom, atau android:numColumns="auto_fit" agar menyesuaikan secara otomatis.

2. android:columnWidth

Menentukan lebar setiap kolom dalam grid.

Contoh: android:columnWidth="100dp"

3. android

Menentukan jarak vertikal antar baris dalam grid.

Contoh: android:verticalSpacing="10dp"

4. android:horizontalSpacing

Menentukan jarak horizontal antar kolom dalam grid.

Contoh: android:horizontalSpacing="10dp"

Cara Kerja:

- Sama seperti ListView, GridView membutuhkan adapter (seperti ArrayAdapter) untuk menghubungkan data ke grid. Setiap item dalam grid diwakili oleh tampilan seperti gambar atau teks.

Contoh penerapan :

XML Layout :

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

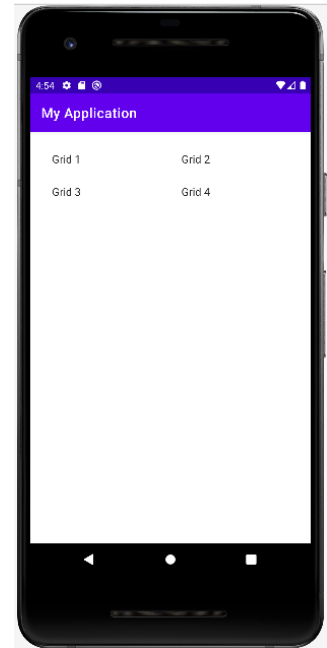
  <GridView
```



```
android:id="@+id/gridView"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:numColumns="2" />  
</LinearLayout>
```

Java Code :

```
GridView gridView = findViewById(R.id.gridView);  
String[] gridItems = {"Grid 1", "Grid 2", "Grid 3", "Grid 4"};  
ArrayAdapter<String> adapterGrid = new ArrayAdapter<>(this,  
android.R.layout.simple_list_item_1, gridItems);  
gridView.setAdapter(adapterGrid);
```



5. Date and Time

Pengertian:

- DatePicker: Widget yang memungkinkan pengguna untuk memilih tanggal.
- TimePicker: Widget yang memungkinkan pengguna untuk memilih waktu (jam dan menit).

Fungsi: Digunakan untuk input tanggal dan waktu dalam aplikasi yang memerlukan penjadwalan atau pencatatan waktu.

Atribut yang Digunakan (DatePicker):

1. android:calendarViewShown
Menentukan apakah kalender visual ditampilkan atau tidak.
Contoh: android:calendarViewShown="true"
2. android:minDate
Menentukan tanggal minimal yang dapat dipilih.
Contoh: android:minDate="01/01/2022"
3. android:maxDate
Menentukan tanggal maksimal yang dapat dipilih.
Contoh: android:maxDate="12/31/2022"

Atribut yang Digunakan (TimePicker):

1. android:is24HourView

Menentukan apakah format waktu adalah 24 jam atau 12 jam (AM/PM).

Contoh: android:is24HourView="true"

Cara Kerja:

- DatePicker dan TimePicker menyediakan interface yang memungkinkan pengguna memilih tanggal dan waktu. Anda dapat menangkap input pengguna menggunakan listener untuk mendapatkan nilai yang dipilih.

Contoh penerapan :

XML Layout :

```
<LinearLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"
  android:padding="16dp">

  <DatePicker
    android:id="@+id/datePicker"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

  <TimePicker
    android:id="@+id/timePicker"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
</LinearLayout>
```

Java Code :

```
DatePicker datePicker = findViewById(R.id.datePicker);
TimePicker timePicker = findViewById(R.id.timePicker);
```

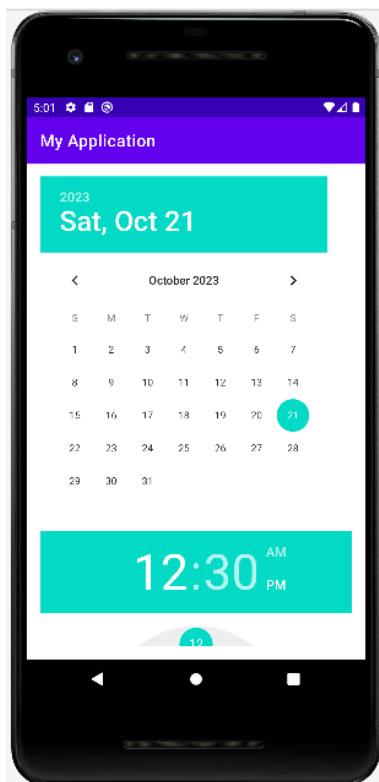
```

// Mengatur tanggal awal pada DatePicker
datePicker.init(2023, 9, 21, new DatePicker.OnDateChangedListener() {
    @Override
    public void onDateChanged(DatePicker view, int year, int monthOfYear, int dayOfMonth)
    {
        // Aksi saat tanggal diubah
    }
});

// Mengatur waktu awal pada TimePicker
timePicker.setHour(12);
timePicker.setMinute(30);

```

Output :



6. MapView and WebView

Pengertian:

- MapView: Widget yang menampilkan peta Google Maps dalam aplikasi.
- WebView: Widget yang memungkinkan aplikasi menampilkan halaman web secara langsung di dalam aplikasi, tanpa harus membuka browser eksternal.

Fungsi:

- MapView digunakan untuk menampilkan peta, menandai lokasi, atau menyediakan navigasi berbasis peta.
- WebView digunakan untuk menampilkan halaman web atau konten HTML di dalam aplikasi.

Atribut yang Digunakan (WebView):

1. `android:layout_width` dan `android:layout_height`
Menentukan ukuran tampilan WebView. Biasanya menggunakan `match_parent` agar memenuhi layar.

Cara Kerja (WebView):

- WebView menggunakan metode seperti `loadUrl()` untuk memuat halaman web, dan pengaturan seperti `setJavaScriptEnabled(true)` untuk mendukung JavaScript.

7. ImageView

ImageView adalah widget yang digunakan untuk menampilkan gambar di dalam aplikasi Android. ImageView berfungsi menampilkan gambar statis dari resource drawable atau dari URL jika ditampilkan secara dinamis.

Atribut yang Digunakan:

1. `android:src`
Menentukan sumber gambar yang akan ditampilkan.
Contoh: `android:src="@drawable/sample_image"`
2. `android:scaleType`
Menentukan bagaimana gambar akan ditampilkan dalam area ImageView (`fit`, `crop`, `center`, dll). Contoh: `android:scaleType="centerCrop"` untuk memotong gambar sehingga memenuhi tampilan.

8. Spinner

Spinner adalah widget yang menampilkan dropdown list dari mana pengguna dapat memilih satu item. Adapun spinner digunakan untuk memilih satu item dari daftar pilihan, seperti memilih kategori, jenis kelamin, atau ukuran produk.

Atribut yang Digunakan:

1. `android:prompt`
Menampilkan teks petunjuk yang akan muncul sebelum pengguna memilih item.
Contoh: `android:prompt="Pilih opsi"`

2. android:entries

Menentukan daftar pilihan yang akan dimasukkan langsung dari resource array.

Contoh: `android:entries="@array/item_list"`

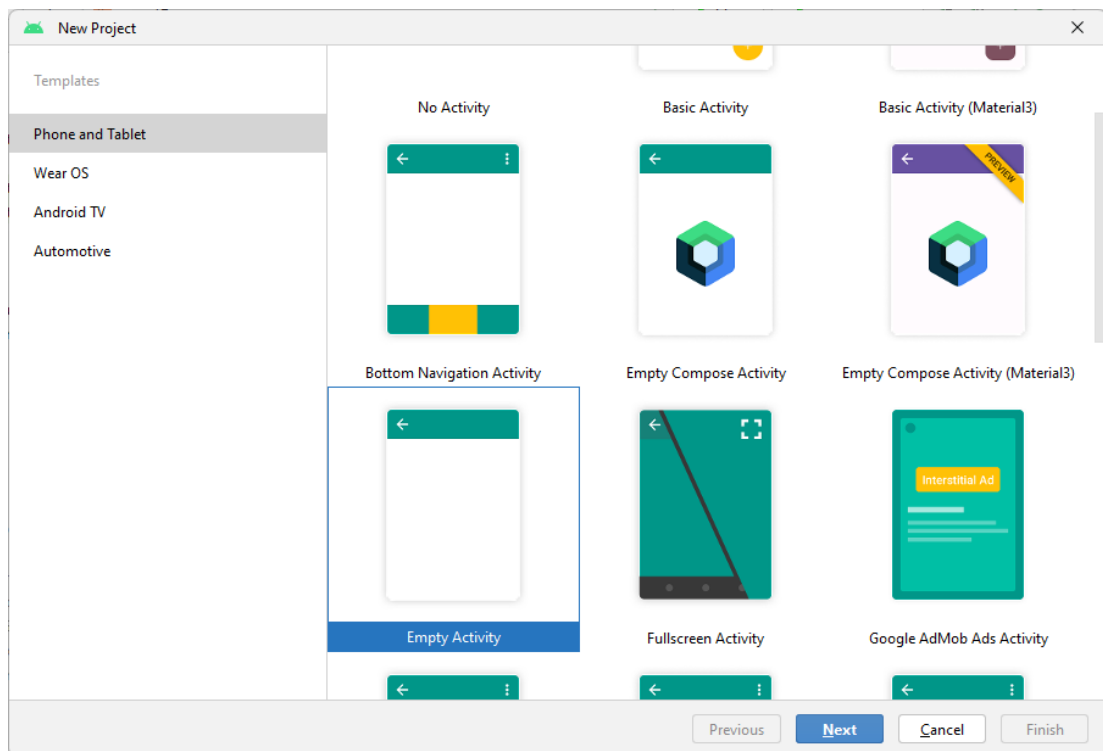
Cara Kerja:

- Sama seperti ListView, Spinner membutuhkan adapter untuk menghubungkan data. ArrayAdapter sering digunakan untuk mengisi Spinner dengan data statis atau dinamis.

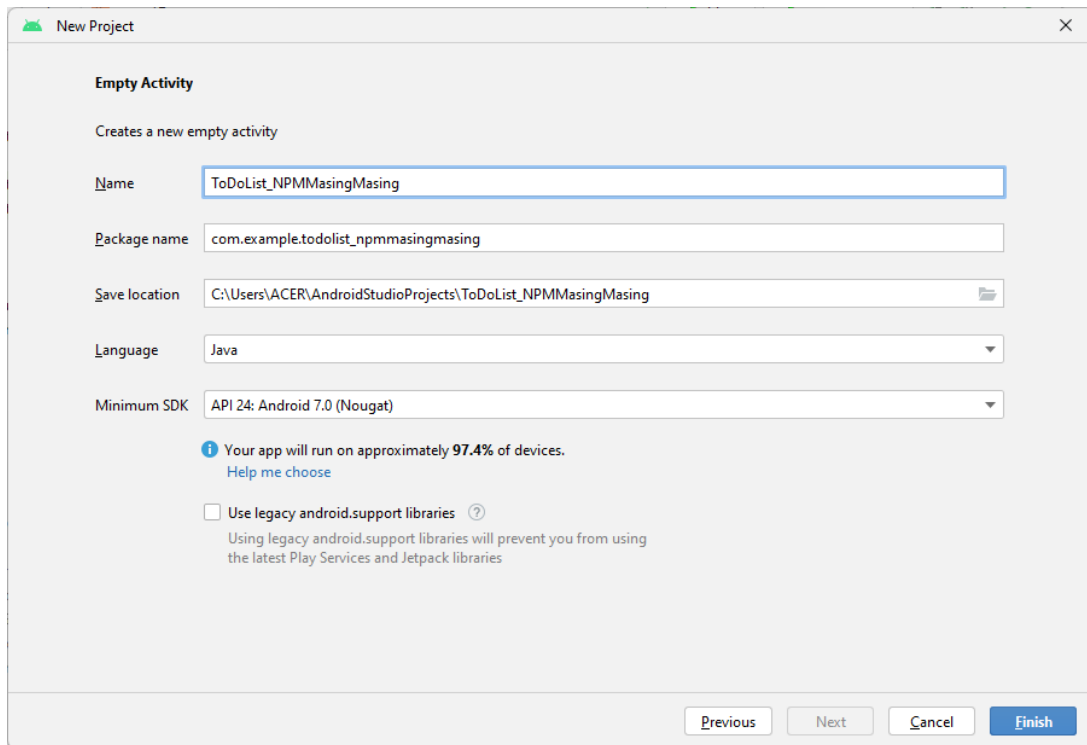
5.4 PRAKTIKUM

1) Buat Proyek Baru:

- Buka Android Studio, pilih "New Project".
- Pilih template "Empty Activity" dan klik "Next".



- Masukkan nama proyek : ToDoList_NPMMasingMasing
- Set language ke Java dan Minimum API Level sesuai kebutuhan
- Klik Finish



2) Menambahkan Dependensi di build.gradle

```
dependencies {  
    implementation 'com.google.android.material:material:1.6.1'  
}
```

Setelah menambahkannya, jangan lupa untuk **Sync Now**.

3) Buat layout utama (activity_main.xml) yang akan menampilkan semua komponen yang kita butuhkan.

```
<?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:padding="16dp">  
  
    <!-- AutoCompleteTextView untuk menambahkan catatan -->  
    <AutoCompleteTextView
```

```
android:id="@+id/autoCompleteTextView"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:hint="Enter task name" />
```

<!-- MultiAutoCompleteTextView untuk menambahkan detail catatan -->

```
<MultiAutoCompleteTextView  
    android:id="@+id/multiAutoCompleteTextView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Enter details" />
```

<!-- DatePicker untuk memilih tanggal -->

```
<DatePicker  
    android:id="@+id/datePicker"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

<!-- TimePicker untuk memilih waktu -->

```
<TimePicker  
    android:id="@+id/timePicker"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

<!-- Spinner untuk memilih kategori -->

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

<!-- Button untuk menambahkan catatan -->

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

```
        android:text="Add Task" />

<!-- ListView untuk menampilkan daftar catatan -->
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1" />

</LinearLayout>
```

4) Kode Aktivitas Utama (MainActivity.java)

```
package com.example.todolistapp;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.ListView;
import android.widget.MultiAutoCompleteTextView;
import android.widget.Spinner;
import android.widget.TimePicker;
import android.widget.AutoCompleteTextView;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private AutoCompleteTextView autoCompleteTextView;
    private MultiAutoCompleteTextView multiAutoCompleteTextView;
    private DatePicker datePicker;
```



```

private TimePicker timePicker;
private Spinner spinner;
private ListView listView;

private ArrayAdapter<String> adapter;
private ArrayList<String> taskList;

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

    autoCompleteTextView = findViewById(R.id.autoCompleteTextView);
    multiAutoCompleteTextView = findViewById(R.id.multiAutoCompleteTextView);
    datePicker = findViewById(R.id.datePicker);
    timePicker = findViewById(R.id.timePicker);
    spinner = findViewById(R.id.spinner);
    listView = findViewById(R.id.listView);
    Button addButton = findViewById(R.id.addButton);

    // Mengisi data Spinner
    String[] categories = {"Work", "Personal", "Shopping", "Other"};
    ArrayAdapter<String> categoryAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_spinner_item, categories);

categoryAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_
item);
    spinner.setAdapter(categoryAdapter);

    // Mengisi data untuk AutoCompleteTextView
    String[] tasks = {"Buy groceries", "Call mom", "Finish report", "Attend meeting"};
    ArrayAdapter<String> autoCompleteAdapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, tasks);
    autoCompleteTextView.setAdapter(autoCompleteAdapter);

```

```

// Menyiapkan ArrayList untuk daftar tugas
taskList = new ArrayList<>();
adapter = new ArrayAdapter<>(this, android.R.layout.simple_list_item_1, taskList);
listView.setAdapter(adapter);

// Menangani tombol add
addButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String taskName = autoCompleteTextView.getText().toString();
        String taskDetails = multiAutoCompleteTextView.getText().toString();
        String category = spinner.getSelectedItem().toString();
        int day = datePicker.getDayOfMonth();
        int month = datePicker.getMonth() + 1; // month index starts from 0
        int year = datePicker.getYear();
        String time = timePicker.getHour() + ":" + timePicker.getMinute();

        // Membuat catatan baru
        if (!taskName.isEmpty()) {
            String task = "Task: " + taskName + "\nDetails: " + taskDetails +
                "\nCategory: " + category +
                "\nDate: " + day + "/" + month + "/" + year + "\nTime: " +
time;

            taskList.add(task);
            adapter.notifyDataSetChanged();
            autoCompleteTextView.setText("");
            multiAutoCompleteTextView.setText("");
        } else {
            Toast.makeText(MainActivity.this, "Please enter a task name",
Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

Penjelasan kode :

a. **Deklarasi Komponen UI:**

- o Menggunakan berbagai widget seperti `AutoCompleteTextView`, `MultiAutoCompleteTextView`, `DatePicker`, `TimePicker`, `Spinner`, dan `ListView`.

b. **Mengisi Data untuk Spinner dan `AutoCompleteTextView`:**

- o Menggunakan `ArrayAdapter` untuk mengisi data ke dalam `Spinner` dan `AutoCompleteTextView`.

c. **`ArrayList` untuk Menyimpan Tugas:**

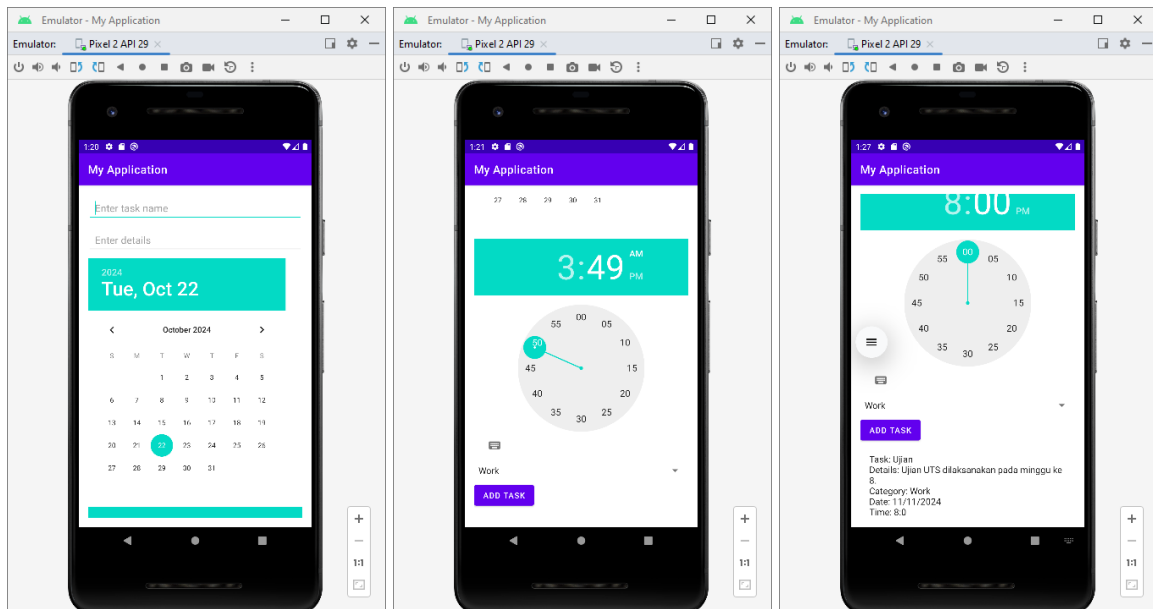
- o Digunakan untuk menyimpan catatan yang ditambahkan. Setiap kali tugas ditambahkan, kita memperbarui adapter untuk `ListView`.

d. **Menangani Tombol Add:**

- o Ketika tombol "Add Task" diklik, kita mengambil input dari pengguna dan membentuk string untuk setiap catatan, yang kemudian ditambahkan ke dalam `taskList`.

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 *to do list* sederhana akan ditampilkan.



5.5 TUGAS

1. Demokan project tersebut hingga muncul di emulator!
2. Tugas dikumpul dalam format pdf dengan nama file : NPM_Tugaske?
3. Didalam file tersebut dituliskan :
 - NPM
 - Nama Lengkap
 - Program Studi
 - Kelas