

## Praktikum #2

### Input Controls, Menus and Pickers, RecyclerView

#### 1. Input Controls

##### TASK 1 : EXPERIMENT WITH TEXT ENTRY ATTRIBUTES

##### 1.1 ADD AN EDIT TEXT FOR ENTERING A NAME

Pertama, kita membuat label nama. pada **activity\_order.xml** tambahkan atribut seperti tabel dibawah ini :

TextView attribute	Value
<code>android:id</code>	<code>"@+id/name_label"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:layout_marginStart</code>	<code>"24dp"</code>
<code>android:layout_marginLeft</code>	<code>"24dp"</code>
<code>android:layout_marginTop</code>	<code>"32dp"</code>
<code>android:text</code>	<code>"Name"</code>
<code>app:layout_constraintStart_toStartOf</code>	<code>"parent"</code>
<code>app:layout_constraintTop_toBottomOf</code>	<code>"@+id/order_textview"</code>

Selanjutnya, kita membuat field untuk pengisian nama. Pada **activity\_order.xml** tambahkan atribut sesuai dengan ketentuan tabel dibawah ini :

EditText attribute	Value
android:id	"@+id/name_text"
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_marginStart	8dp
android:layout_marginLeft	8dp
android:ems	"10"
android:hint	"@string/enter_name_hint"
android:inputType	"textPersonName"
app:layout_constraintBaseline_toBaselineOf	"@+id/name_label"
app:layout_constraintStart_toEndOf	"@+id/name_label"

Tampilan :

Order Activity

Name Enter your name

Address Enter address

Phone Enter phone

Note: Enter note

Choose a delivery method:

☐ Same day messenger service

☒ Next day ground delivery

☐ Pick up

## 1.2 ADD A MULTIPLE-LINE EDIT TEXT

Pada langkah ini kita akan menambahkan EditText ke OrderActivity sehingga pengguna dapat memasukkan alamat menggunakan beberapa baris.

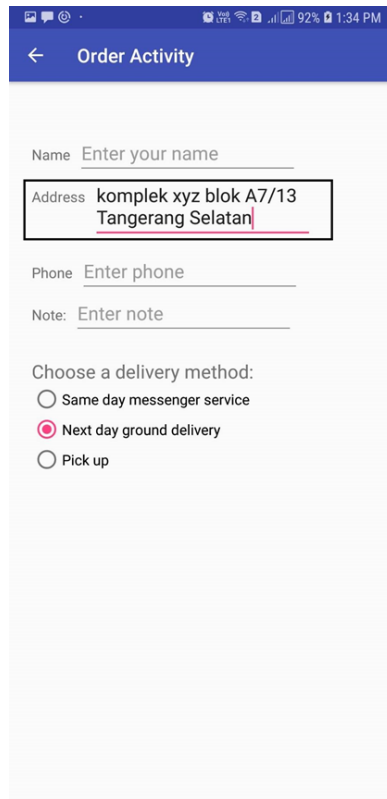
Pertama, kita membuat label untuk address :

TextView attribute	Value
android:id	"@+id/address_label"
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_marginStart	"24dp"
android:layout_marginLeft	"24dp"
android:layout_marginTop	"24dp"
android:text	"Address"
app:layout_constraintStart_toStartOf	"parent"
app:layout_constraintTop_toBottomOf	"@+id/name_label"

Selanjutnya, kita akan membuat field untuk address

EditText attribute	Value
android:id	"@+id/address_text"
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_marginStart	8dp
android:layout_marginLeft	8dp
android:ems	"10"
android:hint	"@string/enter_address_hint"
android:inputType	"textMultiLine"
app:layout_constraintBaseline_toBaselineOf	"@+id/address_label"
app:layout_constraintStart_toEndOf	"@+id/address_label"

Ketika dijalankan :



### 1.3 USE A KEYPAD FOR PHONE NUMBER

Pada langkah ini Anda menambahkan EditText ke layout **OrderActivity** di aplikasi **DroidCafe** sehingga pengguna dapat memasukkan nomor telepon pada keypad numerik.

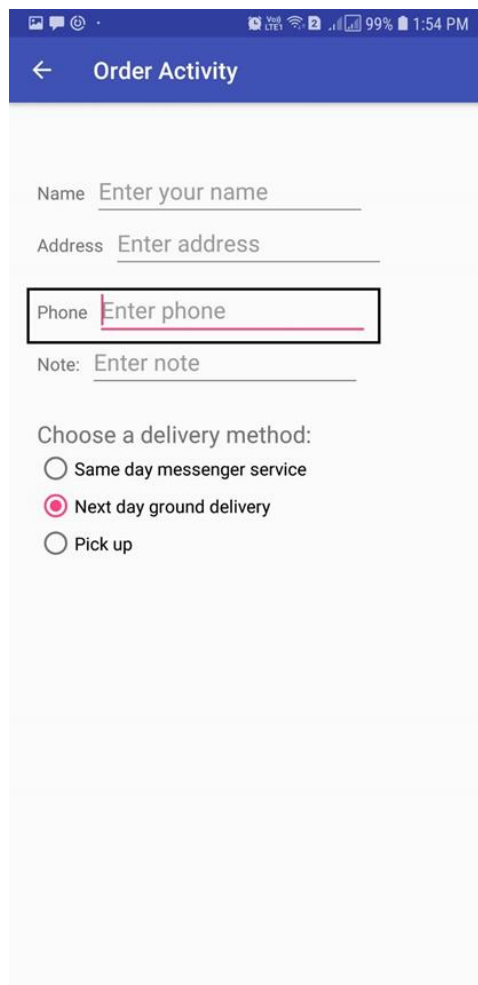
Pertama, kita membuat label Phone

TextView attribute	Value
android:id	"@+id/phone_label"
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_marginStart	"24dp"
android:layout_marginLeft	"24dp"
android:layout_marginTop	"24dp"
android:text	"Phone"
app:layout_constraintStart_toStartOf	"parent"
app:layout_constraintTop_toBottomOf	"@+id/address_text"

Selanjutnya, kita akan membuat field untuk memasukkan nomor handphone

EditText attribute	Value
android:id	"@+id/phone_text"
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_marginStart	8dp
android:layout_marginLeft	8dp
android:ems	"10"
android:hint	"@string/enter_phone_hint"
android:inputType	"phone"
app:layout_constraintBaseline_toBaselineOf	"@+id/phone_label"
app:layout_constraintStart_toEndOf	"@+id/phone_label"

Ketika dijalankan :



Order Activity

Name

Address

Phone

Note:

Choose a delivery method:

☐ Same day messenger service

☒ Next day ground delivery

☐ Pick up

## 1.4 COMBINE INPUT TYPES IN ONE EDIT TEXT

Anda dapat menggabungkan nilai atribut `inputType` yang tidak saling bertentangan. Misalnya, Anda bisa menggabungkan nilai atribut `textMultiLine` dan `textCapSentences` untuk beberapa baris teks di mana setiap kalimat dimulai dengan huruf kapital.

TextView attribute	Value
<code>android:id</code>	<code>"@+id/note_label"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:layout_marginStart</code>	<code>"24dp"</code>
<code>android:layout_marginLeft</code>	<code>"24dp"</code>
<code>android:layout_marginTop</code>	<code>"24dp"</code>
<code>android:text</code>	<code>"Note"</code>
<code>app:layout_constraintStart_toStartOf</code>	<code>"parent"</code>
<code>app:layout_constraintTop_toBottomOf</code>	<code>"@+id/phone_label"</code>

Selanjutnya, kita akan membuat field untuk Note

EditText attribute	Value
<code>android:id</code>	<code>"@+id/note_text"</code>
<code>android:layout_width</code>	<code>"wrap_content"</code>
<code>android:layout_height</code>	<code>"wrap_content"</code>
<code>android:layout_marginStart</code>	<code>8dp</code>
<code>android:layout_marginLeft</code>	<code>8dp</code>
<code>android:ems</code>	<code>"10"</code>
<code>android:hint</code>	<code>"@string/enter_note_hint"</code>
<code>android:inputType</code>	<code>"textCapSentences textMultiLine"</code>
<code>app:layout_constraintBaseline_toBaselineOf</code>	<code>"@+id/note_label"</code>
<code>app:layout_constraintStart_toEndOf</code>	<code>"@+id/note_label"</code>

Ketika dijalankan :

The screenshot shows a mobile application interface titled "Order Activity". At the top, there is a blue header bar with a back arrow and the title. Below the header, the screen contains several input fields: "Name" with the placeholder "Enter your name", "Address" with "Enter address", "Phone" with "Enter phone", and "Note" with "Enter note". The "Note" field is highlighted with a black border. Below these fields, there is a section titled "Choose a delivery method:" followed by three radio button options: "Same day messenger service", "Next day ground delivery", and "Pick up". The "Pick up" option is selected, indicated by a red dot. The status bar at the top of the phone shows various icons including signal strength, Wi-Fi, battery level at 99%, and the time 2:08 PM.

Order Activity

Name: Enter your name

Address: Enter address

Phone: Enter phone

Note: Enter note

Choose a delivery method:

☐ Same day messenger service

☐ Next day ground delivery

☒ Pick up

## TASK 2 : USE RADIO BUTTONS\

### 2.1 ADD A RADIO GROUP AND RADIO BUTTONS

Buka tab **activity\_order.xml**, kemudian tambahkan atribut sesuai tabel dibawah ini :

TextView attribute	Value
android:id	"@+id/delivery_label"
android:layout_width	"wrap_content"
android:layout_height	"wrap_content"
android:layout_marginStart	"24dp"
android:layout_marginLeft	"24dp"
android:layout_marginTop	"24dp"
android:text	"Choose a delivery method: "
android:textSize	"18sp"
app:layout_constraintStart_toStartOf	"parent"
app:layout_constraintTop_toBottomOf	"@+id/note_text"

Selanjutnya, menambahkan Radio Button :

<RadioGroup

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="24dp"
    android:layout_marginStart="24dp"
    android:orientation="vertical"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/delivery_label">
```

<RadioButton

```
    android:id="@+id/sameday"
    android:layout_width="wrap_content"
```



```
        android:layout_height="wrap_content"
        android:onClick="onRadioButtonClicked"
        android:text="Same day messenger service" />
```

```
<RadioButton
```

```
    android:id="@+id/nextday"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onRadioButtonClicked"
    android:text="Next day ground delivery" />
```

```
<RadioButton
```

```
    android:id="@+id/pickup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="onRadioButtonClicked"
    android:text="Pick up" />
```

```
</RadioGroup>
```

## 2.1 ADD THE RADIO BUTTON CLICK HANDLER

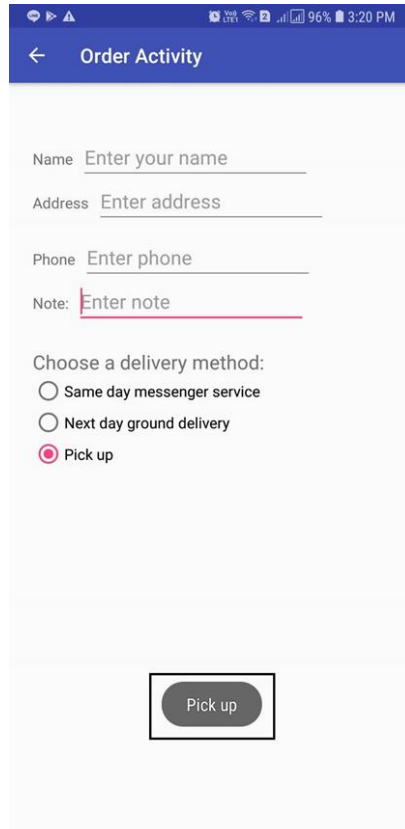
Pada **activity\_order.java** tambahkan Source Code berikut :

```
public void onRadioButtonClicked(View view) {  
    // Is the button now checked?  
    boolean checked = ((RadioButton) view).isChecked();  
    // Check which radio button was clicked.  
    switch (view.getId()) {  
        case R.id.sameday:  
            if (checked)  
                // Same day service  
                displayToast(getString(R.string.same_day_messenger_service));  
            break;  
        case R.id.nextday:  
            if (checked)  
                // Next day delivery  
                displayToast(getString(R.string.next_day_ground_delivery));  
            break;  
        case R.id.pickup:  
            if (checked)  
                // Pick up  
                displayToast(getString(R.string.pick_up));  
            break;  
        default:  
            // Do nothing.  
            break;  
    }  
}
```

Tambahkan Method displayToast :

```
public void displayToast(String message) {  
    Toast.makeText(getApplicationContext(), message,  
        Toast.LENGTH_SHORT).show();  
}
```

Ketika dijalankan :



The screenshot shows an Android application interface titled "Order Activity". The interface includes a blue header bar with a back arrow and the title. Below the header, there is a form with the following fields:

- Name: Enter your name
- Address: Enter address
- Phone: Enter phone
- Note: Enter note

Below the form, there is a section titled "Choose a delivery method:" with three radio button options:

- ☐ Same day messenger service
- ☐ Next day ground delivery
- ☒ Pick up

At the bottom of the screen, there is a button labeled "Pick up".

## TASK 3 : USE A SPINNER FOR USER CHOICES

### 3 .1 ADD A SPINNER TO LAYOUT

Buka **activity\_order.xml**, lalu drag **Spinner** ke layout

Source Code untuk Spinner adalah sebagai berikut :

<Spinner

```
    android:id="@+id/label_spinner"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginEnd="24dp"
    android:layout_marginRight="24dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="24dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toEndOf="@+id/phone_text"
    app:layout_constraintTop_toBottomOf="@+id/address_text" />
```

### 3.2 ADD CODE TO ACTIVATE THE SPINNER AND ITS LISTENER

Kali ini kita menambahkan nilai untuk Spinner yang telah kita buat sebelumnya,

Buka **strings.xml** lalu tambahkan Source Code berikut :

```
<string-array name="labels_array">
    <item>Home</item>
    <item>Work</item>
    <item>Mobile</item>
    <item>Other</item>
</string-array>
```

- Untuk mendefinisikan **Spinner** yang telah kita buat, implementasikan `AdapterView.OnItemSelectedListener` pada **OrderActivity**

```
public class OrderActivity extends AppCompatActivity implements
    AdapterView.OnItemSelectedListener {
```

- Lakukan import **AdapterView**

```
import android.widget.AdapterView;
```

- Tambahkan Source Code dibawah ini untuk mengaktifkan **Spinner**

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    // ... Rest of onCreate code ...
    // Create the spinner.
    Spinner spinner = findViewById(R.id.label_spinner);
    if (spinner != null) {
        spinner.setOnItemSelectedListener(this);
    }
}
```

- Pada method **onCreate()** tambahkan Source Code dibawah ini :

```
ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,  
    R.array.labels_array, android.R.layout.simple_spinner_item);
```

- Tambahkan Source Code dibawah ini untuk mengaktifkan Adapter :

```
adapter.setDropDownViewResource  
    (android.R.layout.simple_spinner_dropdown_item);  
  
// Apply the adapter to the spinner.  
if (spinner != null) {  
    spinner.setAdapter(adapter);  
}
```

### 3.3 ADD CODE TO RESPOND TO SPINNER SELECTIONS

Kali ini kita akan menambahkan Code, agar **Spinner** yang kita tambahkan sebelumnya dapat merespon, Buka **OrderActivity**, lalu tambahkan :

```
onItemSelected()  
  
public void onItemSelected(AdapterView<?> adapterView, View view, int  
    i, long l) {  
    spinnerLabel = adapterView.getItemAtPosition(i).toString();  
    displayToast(spinnerLabel);  
}
```

Ketika dijalankan :

**Droid Cafe**

You ordered a donut.

Name

Address

Phone

Note:

Choose a delivery method

☐ Same day messenger service

☒ Next day ground delivery

☐ Pick up

Home  
Home  
Work  
Mobile  
Service  
Other

## 2. Option Menu and Pickers

### Task 1. Add items to the options menu

#### 1.1 menambahkan item pada menu option

- Lihat **menu\_main.xml** (luaskan **res > menu** di tampilan Project).

```
<item  
    android:id="@+id/action_settings"  
    android:orderInCategory="100"  
    android:title="@string/action_settings"  
    app:showAsAction="never" />
```

- Ubah atribut item **action\_settings** berikut untuk membuatnya sebagai item **action\_contact** (jangan ubah atribut **android:orderInCategory** yang sudah ada):

Atribut	Nilai
android:id	"@+id/action_contact"
android:title	"Contact"
app:showAsAction	"never"

- Tambahkan item menu baru menggunakan blok **<item>** **</item>** di dalam **<menu>** **</menu>**, dan berikan atribut berikut ke item:I

Atribut	Nilai
android:id	"@+id/action_order"
android:orderInCategory	"10"
android:title	"Order"
app:showAsAction	"never"

- Tambahkan dua item menu lagi dengan cara yang sama
- Ubah *"title"*, *"id"*, dan *"showAction"* dengan nama "Item Menu1"
- Tambahkan nama dan nilai string berikut di **strings.xml**



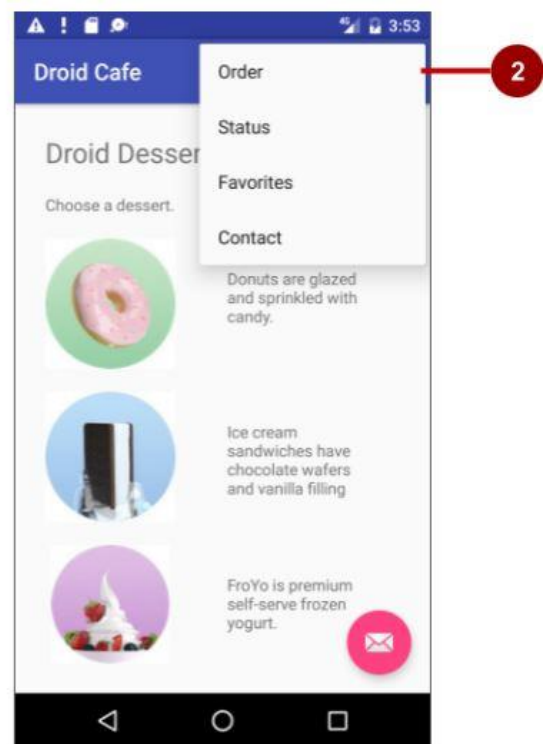
<string name="action\_order\_message">You selected Order.</string>

<string name="action\_status\_message">You selected Status.</string>

<string name="action\_favorites\_message">You selected Favorites.</string>

<string name="action\_contact\_message">You selected Contact.</string>

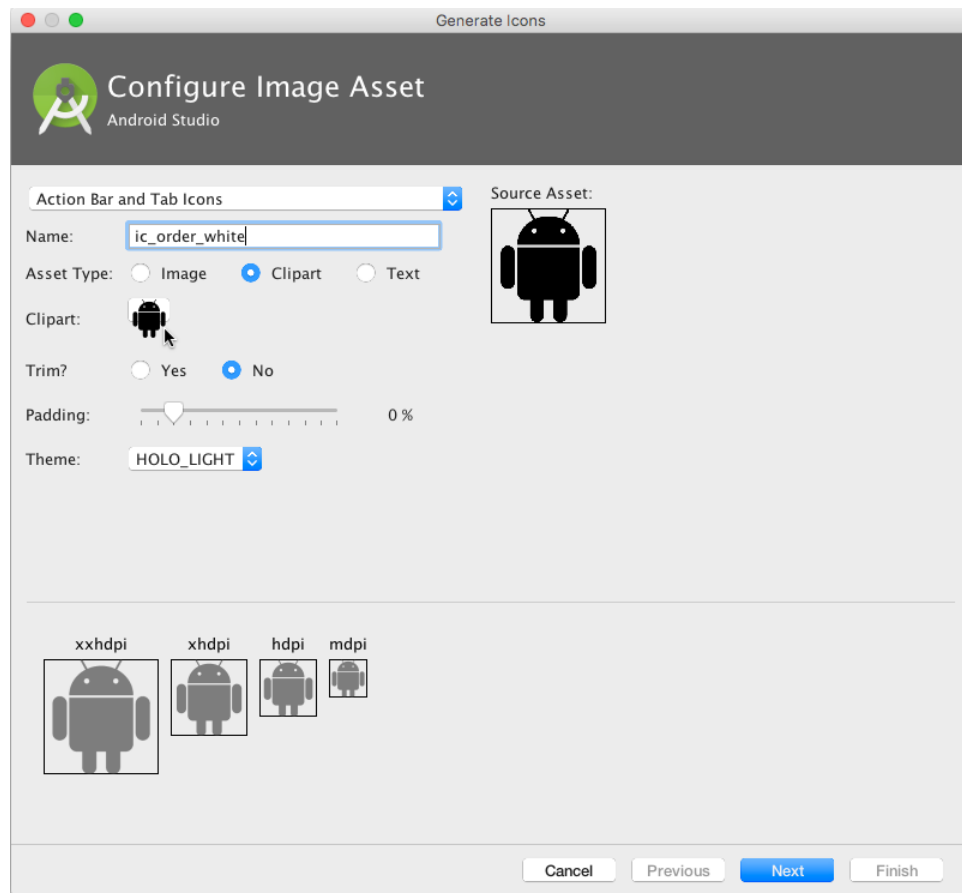
- Buka **MainActivity** dan ubah pernyataan `if` di metode `onOptionsItemSelected()` dengan mengganti id `action_settings` dengan id `action_order` baru
- Jalankan aplikasi.



## Task 2. Menambah ikon untuk item menu

### 2.1 Menambahkan ikon untuk item menu

- Luaskan **res** di tampilan Project, dan klik kanan (atau Kontrol-klik) **drawable**.
- Pilih **New > Image Asset**. Dialog Configure Image Asset akan muncul.
- Pilih **Action Bar and Tab Items** di menu tarik-turun.
- Ubah **ic\_action\_name** ke **ic\_order\_white** (untuk tindakan **Order**). Layar Configure Image Asset akan tampak seperti berikut



- Klik gambar clipart (logo Android di sebelah "Clipart") untuk memilih gambar clipart sebagai ikon. Laman ikon akan muncul. Klik ikon yang ingin Anda gunakan untuk tindakan **Order** (misalnya, ikon keranjang belanja mungkin sesuai).
- Pilih **HOLO\_DARK** dari menu tarik-turun Theme. Ini akan menyetel ikon menjadi putih dengan latar belakang berwarna gelap (atau hitam). Klik **Next**.
- Klik **Finish**, dalam dialog Confirm Icon Path.

- Ulang langkah-langkah di atas untuk ikon lainnya.

## 2.2 Menampilkan item menu sebagai ikon di layer

- Buka **menu\_main.xml** lagi dan tambahkan atribut berikut ke item **Order**, **Status**, dan **Favorites**
- kemudian tambahkan **“android:icon ”@drawable/ic\_order\_white”** untuk setiap item dan sesuaikan dengan namanya.

## Task 3. Menangani item menu yang di pilih

### 3.1 untuk menampilkan pilihan menu

- Buat Buka **MainActivity**.
- Tambah kode berikut ini (anda telah mempelajari sebelumnya untuk menampilkan pesan toast, tambahkan sekarang)

```
public void displayToast(String message) {

    Toast.makeText(getApplicationContext(), message,

        Toast.LENGTH_SHORT).show();

}
```

### 3.2 Tambahkan action dengan onOptionsItemSelected

- Buatlah kode berikut pada layer anda
- ```
@Override

public boolean onOptionsItemSelected(MenuItem item) {

    int id = item.getItemId();

    if (id == R.id.action_order) {

        return true;
```

```

    }

    return super.onOptionsItemSelected(item);
}

```

- Ganti pernyataan if dan penetapannya ke id dengan blok switch case berikut yang menyetel message yang sesuai berdasarkan id item menu.

@Override

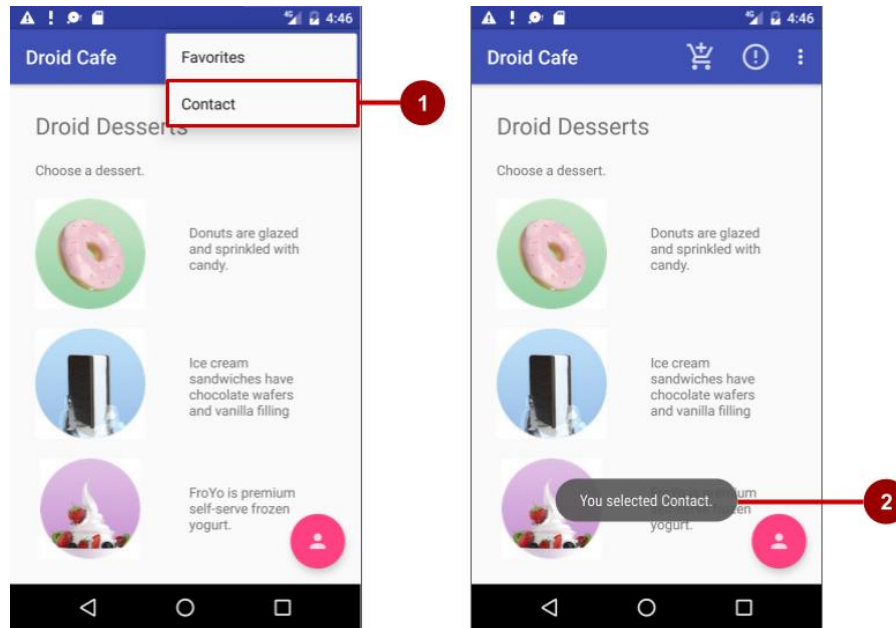
```

public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.action_order:
            displayToast(getString(R.string.action_order_message));
            return true;
        case R.id.action_status:
            displayToast(getString(R.string.action_status_message));
            return true;
        case R.id.action_favorites:
            displayToast(getString(R.string.action_favorites_message));
            return true;
        case R.id.action_contact:
            displayToast(getString(R.string.action_contact_message));
            return true;
        default:
            // Do nothing
    }

    return super.onOptionsItemSelected(item);
}

```

- Jalankan aplikasi



## Task 4. Menggunakan picker

### 4.1 membuat layer utama

- **Membuat layout utama**
- Ubah induk `RelativeLayout` agar menjadi `LinearLayout` dan tambahkan `android:orientation="vertical"` untuk mengatur orientasi layout menjadi vertical
- Ubah teks elemen `TextView` pertama ke "**Choose the date and time:** " dan ekstrak teks ke sumber daya string `choose_datetime`.
- Tambahkan atribut `android:textSize` dan masukkan ukuran teks **20sp**. Ekstrak dimensi `android:textSize` ke `text_size`.
- Tambahkan turunan `RelativeLayout` di dalam `LinearLayout` untuk menampung elemen `Button`, dan terima lebar dan tinggi default "`match parent`".
- Tambahkan elemen `Button` pertama di dalam `RelativeLayout` dengan atribut berikut:

| Atribut Tombol Pertama   | Nilai                  |
|--------------------------|------------------------|
| android:layout_width     | "wrap_content"         |
| android:layout_height    | "wrap_content"         |
| android:id               | "@+id/button_date"     |
| android:layout_marginTop | "12dp"                 |
| android:text             | "Date"                 |
| android:onClick          | "showDatePickerDialog" |

- Ubah string "Date" ke dalam sumber daya string date\_button.
- Ubah dimensi "12dp" untuk android:layout\_marginTop ke button\_top\_margin.
- Tambahkan elemen Button kedua di dalam turunan RelativeLayout atribut berikut:

| Atribut Tombol Kedua       | Nilai                      |
|----------------------------|----------------------------|
| android:layout_width       | "wrap_content"             |
| android:layout_height      | "wrap_content"             |
| android:id                 | "@+id/button_time"         |
| android:layout_marginTop   | "@dimen/button_top_margin" |
| android:layout_alignBottom | "@id/button_date"          |
| android:layout_toRightOf   | "@id/button_date"          |
| android:text               | "Time"                     |
| android:onClick            | "showTimePickerDialog"     |

- Kode solusi
- Ubah string "Time" ke dalam sumber daya string time\_button

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:orientation="vertical"
```

```
        android:layout_width="match_parent"

        android:layout_height="match_parent"

        android:paddingBottom="@dimen/activity_vertical_margin"

        android:paddingLeft="@dimen/activity_horizontal_margin"

        android:paddingRight="@dimen/activity_horizontal_margin"

        android:paddingTop="@dimen/activity_vertical_margin"

        tools:context="com.example.android.DateTimePickers.MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:textSize="@dimen/text_size"

    android:text="@string/choose_datetime"/>
```

```
<RelativeLayout
```

```
    android:layout_width="match_parent"

    android:layout_height="match_parent">
```

```
<Button
```

```
    android:layout_width="wrap_content"

    android:layout_height="wrap_content"

    android:id="@+id/button_date"
```

```
android:layout_marginTop="@dimen/button_top_margin"

android:text="@string/date_button"

android:onClick="showDatePickerDialog"/>
```

```
<Button
```

```
android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:id="@+id/button_time"

android:layout_marginTop="@dimen/button_top_margin"

android:layout_alignBottom="@id/button_date"

android:layout_toRightOf="@id/button_date"

android:text="@string/time_button"

android:onClick="showTimePickerDialog"/>
```

```
</RelativeLayout>
```

```
</LinearLayout>
```

#### 4.2 Buat fragmen baru untuk picker tanggal

- Luaskan **app** > **java** > **com.example.android.DateTimePickers** dan pilih **MainActivity**.
- Pilih **File** > **New** > **Fragment** > **Fragment (Blank)**, dan beri nama fragmen **DatePickerFragment**. Kosongkan ketiga opsi kotak centang sehingga Anda *tidak* membuat XML layout, *jangan* sertakan metode pabrik fragmen, dan *jangan* sertakan callback antarmuka. Anda tidak perlu membuat layout untuk picker standar. Klik **Finish** untuk membuat fragmen.



- Buka **DatePickerFragment** dan edit definisi kelas `DatePickerFragment` untuk memperluas `DialogFragment` dan dan implementasikan `DatePickerDialog.OnDateSetListener` untuk membuat picker tanggal standar dengan sebuah listener.

```
public class DatePickerFragment extends DialogFragment
    implements DatePickerDialog.OnDateSetListener {
```

- Selagi anda mengetikkan kode di atas maka akan otomatis akan menambahkan yang berikut di blok `import` di atas:

```
import android.app.DatePickerDialog;
import android.support.v4.app.DialogFragment;
```

- Klik ikon bola lampu merah dan pilih **Implement methods** dari menu munculan.
- Buang konstruktor publik kosong untuk `DatePickerFragment`.
- Ganti `onCreateView()` dengan `onCreateDialog()` yang mengembalikan `Dialog`, dan anotasikan `onCreateDialog()` dengan `@NonNull` untuk menunjukkan bahwa nilai pengembalian `Dialog` tidak boleh null—upaya apa pun untuk mengacu pada nilai pengembalian `Dialog` harus dicentang null.

```
@NonNull
```

```
@Override
```

```
public Dialog onCreateDialog(Bundle savedInstanceState) {
```

```
...  
  
}
```

- Tambahkan kode berikut ke `onCreateDialog()` untuk membuat instance `year`, `month`, dan `day` dari `Calendar`, dan mengembalikan dialog dan nilai ini ke aktivitas utama. Saat Anda memasukkan `Calendar`, tetapkan import menjadi `java.util.Calendar`

`@NonNull`

`@Override`

```
public Dialog onCreateDialog(Bundle savedInstanceState) {  
  
    // Use the current date as the default date in the picker.  
  
    final Calendar c = Calendar.getInstance();  
  
    int year = c.get(Calendar.YEAR);  
  
    int month = c.get(Calendar.MONTH);  
  
    int day = c.get(Calendar.DAY_OF_MONTH);  
  
  
    // Create a new instance of DatePickerDialog and return it.  
  
    return new DatePickerDialog(getActivity(), this, year, month, day);  
  
}
```

Kode Solusinya :

`Public class DatePickerFragment extends DialogFragment`

```

        implements DatePickerDialog.OnDateSetListener {

        @NonNull

        @Override

        public Dialog onCreateDialog(Bundle savedInstanceState) {

            // Use the current date as the default date in the picker.

            final Calendar c = Calendar.getInstance();

            int year = c.get(Calendar.YEAR);

            int month = c.get(Calendar.MONTH);

            int day = c.get(Calendar.DAY_OF_MONTH);


            // Create a new instance of DatePickerDialog and return it.

            return new DatePickerDialog(getActivity(), this, year, month, day);

        }


        public void onDateSet(DatePicker view, int year, int month, int day) {

            // Do something with the date chosen by the user.

        }

    }

```

### 4.3 membuat fragme untuk picker waktu

- Pilih **MainActivity** lagi.

- Pilih **File > New > Fragment > Fragment (Blank)**, dan beri nama fragmen **TimePickerFragment**. Kosongkan ketiga opsi sehingga Anda *tidak* membuat XML layout, *jangan* sertakan metode pabrik fragmen, dan *jangan* sertakan callback antarmuka. Klik **Finish** untuk membuat fragmen.
- Bukan *\*TimePickerFragment* dan ikuti prosedur yang sama dengan `DatePickerFragment`, mengimplementasikan metode kosong `onTimeSet()`, mengganti `onCreateView()` dengan `onCreateDialog()`, dan menghapus konstruktor publik untuk `TimePickerFragment`. `TimePickerFragment` melakukan tugas yang sama dengan `DatePickerFragment`, tetapi dengan nilai waktu

Kode solusi untuk waktu:

```
public class TimePickerFragment extends DialogFragment

    implements TimePickerDialog.OnTimeSetListener {

    @NonNull

    @Override

    public Dialog onCreateDialog(Bundle savedInstanceState) {

        // Use the current time as the default values for the picker.

        final Calendar c = Calendar.getInstance();

        int hour = c.get(Calendar.HOUR_OF_DAY);

        int minute = c.get(Calendar.MINUTE);

        // Create a new instance of TimePickerDialog and return it.

        return new TimePickerDialog(getActivity(), this, hour, minute,
```

```

        DateFormat.is24HourFormat(getActivity()));

    }

    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

        // Do something with the time chosen by the user.

    }

}

```

#### 4.4 memodifikasi layer utama

- Buat sumber daya string di **strings.xml**:
- Buka **MainActivity**.
- Tambahkan metode `showDatePickerDialog()` dan `showTimePickerDialog()`, dengan merujuk ke kode di bawah ini. Ini membuat instance `FragmentManager` untuk mengelola fragmen secara otomatis, dan untuk menampilkan picker

```

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

    }

    public void showDatePickerDialog(View v) {

        DialogFragment newFragment = new DatePickerFragment();

```

```

        newFragment.show(getSupportFragmentManager(),
                        getString(R.string.date_picker));
    }

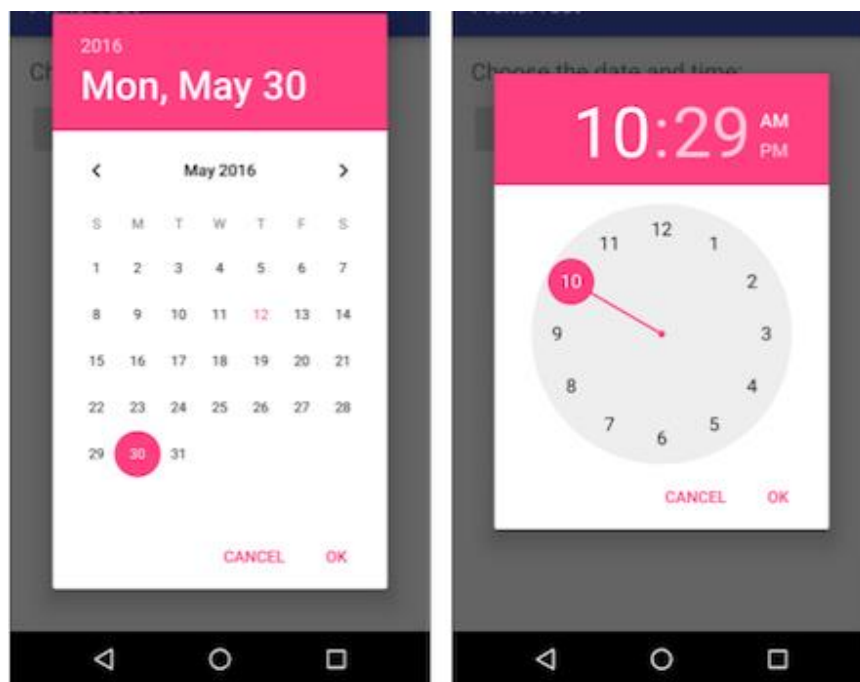
    public void showTimePickerDialog(View view) {

        DialogFragment newFragment = new TimePickerFragment();

        newFragment.show(getSupportFragmentManager(),
                        getString(R.string.time_picker));
    }
}

```

- Jalankan aplikasi.



#### 4.5 menggunakan tanggal dan waktu yang di pilih

- Buka **MainActivity** dan tambahkan tanda tangan metode `processDatePickerResult()` yang mengambil `year`, `month`, dan `day` sebagai argumen:

```
public void processDatePickerResult(int year, int month, int day) {  
  
}
```

- Tambahkan kode berikut ke metode `processDatePickerResult()` untuk mengonversi `month`, `day`, dan `year` ke string yang terpisah

```
String month_string = Integer.toString(month+1);
```

```
String day_string = Integer.toString(day);
```

```
String year_string = Integer.toString(year);
```

- Tambahkan yang berikut ini setelah kode di atas untuk menggabungkan tiga string dan menyertakan garis miring untuk format tanggal A.S.

```
String dateMessage = (month_string + "/" +  
  
day_string + "/" + year_string);
```

- Tambahkan yang berikut ini setelah pernyataan di atas untuk menampilkan pesan toast:

```
Toast.makeText(this, "Date: " + dateMessage,  
  
Toast.LENGTH_SHORT).show();
```

- Ubah string `"Date: "` hard-code ke dalam sumber daya string bernama `date`

```
public void processDatePickerResult(int year, int month, int day) {  
  
String month_string = Integer.toString(month + 1);  
  
String day_string = Integer.toString(day);  
  
String year_string = Integer.toString(year);
```

```

// Assign the concatenated strings to dateMessage.

String dateMessage = (month_string + "/" +

                        day_string + "/" + year_string);

Toast.makeText(this, getString(R.string.date) + dateMessage,

                Toast.LENGTH_SHORT).show();

}

```

- Buka **DatePickerFragment** dan tambahkan yang berikut ini ke metode `onDateSet()` untuk menjalankan metode `processDatePickerResult()` di **MainActivity** dan meneruskan `year`, `month`, dan `day` ke metode:

```

public void onDateSet(DatePicker view, int year, int month, int day) {

    // Set the activity to the Main Activity.

    MainActivity activity = (MainActivity) getActivity();

    // Invoke Main Activity's processDatePickerResult() method.

    activity.processDatePickerResult(year, month, day);

}

```

- **TimePickerFragment** menggunakan logika yang sama.  
1. Buka **MainActivity** dan tambahkan tanda tangan metode `processTimePickerResult()` yang mengambil `hourOfDay` dan `minute` sebagai argument

```

public void processTimePickerResult(int hourOfDay, int minute) {

}

```

- Tambahkan kode berikut ke metode `processTimePickerResult()` untuk mengonversi `hourOfDay` dan `minute` ke string yang terpisah



```
String hour_string = Integer.toString(hourOfDay);
```

```
String minute_string = Integer.toString(minute);
```

- Tambahkan yang berikut ini setelah kode di atas untuk menggabungkan string dan menyertakan titik dua untuk format waktu:

```
String timeMessage = (hour_string + ":" + minute_string);
```

- Tambahkan yang berikut ini setelah pernyataan di atas untuk menampilkan pesan toast:

```
Toast.makeText(this, "Time: " + timeMessage,
```

```
                Toast.LENGTH_SHORT).show();
```

- Ekstrak string "Time: " hard-code ke dalam sumber daya string bernama `time`. Hal ini secara otomatis mengganti string hard-code dengan `getString(R.string.time)`. Kode untuk metode `processDatePickerResult()` sekarang harus terlihat seperti ini:

```
public void processTimePickerResult(int hourOfDay, int minute) {
```

```
    // Convert time elements into strings.
```

```
    String hour_string = Integer.toString(hourOfDay);
```

```
    String minute_string = Integer.toString(minute);
```

```
    // Assign the concatenated strings to timeMessage.
```

```
    String timeMessage = (hour_string + ":" + minute_string);
```

```
    Toast.makeText(this, getString(R.string.time) + timeMessage,
```

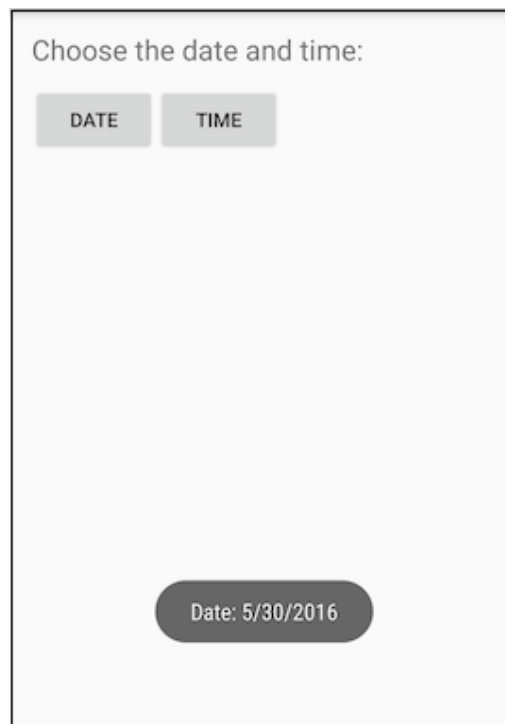
```
                    Toast.LENGTH_SHORT).show();
```

```
}
```

- Buka **TimePickerFragment** dan tambahkan yang berikut ini ke metode `onTimeSet()` untuk memanggil metode `processTimePickerResult()` di `MainActivity` dan meneruskan `hourOfDay` dan `minute` ke metode:

```
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
  
    // Set the activity to the Main Activity.  
  
    MainActivity activity = (MainActivity) getActivity();  
  
    // Invoke Main Activity's processTimePickerResult() method.  
  
    activity.processTimePickerResult(hourOfDay, minute);  
  
}
```

- Jalankan aplikasi



### 3. RecyclerView

#### TASK 1 : CREATE A NEW PROJECT AND DATASET

##### 1.1 CREATE THE PROJECT AND LAYOUT

- Buat Project baru dengan nam **RecyclerView**, lalu pilih **Basic Activity**

##### 1.2 ADD CODE TO CREATE DATA

- Pada **MainActivity** tambahkan private variable untuk **mWordList**

```
public class MainActivity extends AppCompatActivity {  
    private final LinkedList<String> mWordList = new LinkedList<>();  
    // ... Rest of MainActivity code ...  
}
```

- Tambahkan Source Code di dalam method **onCreate()** yang akan mengisi **mWordList**

**// Put initial data into the word list.**

```
for (int i = 0; i < 20; i++) {  
    mWordList.addLast("Word " + i);  
}
```

##### 1.3 CHANGE THE FAB ICON

- Masuk ke folder **Drawable**
- Klik **New>Image Asset**
- Pilih **Action Bar and Tab Items** pada drop-down menu
- Ubah **ic\_action\_name** menajdi **ic\_add\_for\_fab**

## TASK 2 : CREATE A RECYCLER VIEW

### 2.1 MODIFY THE LAYOUT IN CONTENT\_MAIN.XML

- Pada **content\_main.xml**
- Lalu ubah keseluruhan **TextView** menjadi

```
<android.support.v7.widget.RecyclerView  
    android:id="@+id/recyclerview"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</android.support.v7.widget.RecyclerView>
```

### 2.2. CREATE THE LAYOUT FOR ONE LIST ITEM

- Klik **App>res>layout**. Lalu pilih **New>Layout Resource File**
- Beri nama file dengan **wordlist\_item**
- Ubah **ConstraintLayout** menjadi **LinearLayout** dengan atribut sebagai berikut

LinearLayout attribute	Value
android:layout_width	"match_parent"
android:layout_height	"wrap_content"
android:orientation	"vertical"
android:padding	"6dp"

- Tambahkan **TextView** dengan atribut sebagai berikut

Attribute	Value
android:id	"@+id/word"
android:layout_width	"match_parent"
android:layout_height	"wrap_content"
android:textSize	"24sp"
android:textStyle	"bold"

## 2.3 CREATE A STYLE FROM THE TEXTVIEW ATTRIBUTES

- Klik kanan pada **TextView** yang terdapat pada **wordlist\_item.xml**.
- Pilih **Refactor>Extract>Style**.

```
<?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="wrap_content"
```

```
    android:orientation="vertical"
```

```
    android:padding="6dp">
```

```
    <TextView
```

```
        android:id="@+id/word"
```

```
        style="@style/word_title" />
```

```
</LinearLayout>
```

## 2.4 CREATE AN ADAPTER

- Klik kanan **java/com.android.example.recyclerview**. Klik **New>Java Class**
- Beri nama kelas **WordListAdapter**
- Tambahkan Source Code berikut

```
public class WordListAdapter extends
```

```
    RecyclerView.Adapter<WordListAdapter.WordViewHolder> { }
```

## 2.5 CREATE THE VIEW HOLDER FOR THE ADAPTER

- Untuk membuat **ViewHolder** berikut merupakan langkah-langkahnya :
- Didalam kelas **WordListAdapter**, tambahkan inner class **WordViewHolder**

```
class WordViewHolder extends RecyclerView.ViewHolder { }
```

- Tambahkan variabel pada inner class **WordViewHolder** untuk **TextView**

```
public final TextView wordItemView;
```

```
final WordListAdapter mAdapter;
```

- Pada inner class **WordViewHolder**, tambahkan Constructor yang menginisialisasi **TextView** pada **ViewHolder**

## 2.6 STORING YOUR DATA IN THE ADAPTER

- Untuk menyimpan data pada sebuah Adapter, **WordListAdapter** membutuhkan sebuah Constructor untuk menginisialisasi daftar kata dari data.
- Buat private linked list

```
private final LinkedList<String> mWordList;
```

- Isi method **getItemCount()** untuk mengembalikan ukuran **mWordList**

```
@Override
```

```
public int getItemCount() {  
    return mWordList.size();  
}
```

- **WordListAdapter** membutuhkan konstruktor yang menginisialisasi daftar kata dari data. **LayoutInflater** membaca deskripsi layout XML dan akan melakukan konversi menjadi item Tampilan yang sesuai.

```
private LayoutInflater mInflater;
```

- Menerapkan konstruktor untuk **WordListAdapter**. Metode ini perlu instansiasi **LayoutInflater** untuk **mInflater** dan mengatur **mWordList** ke data yang diteruskan.

```
public WordListAdapter(Context context,
                        LinkedList<String> wordList) {

    mInflater = LayoutInflater.from(context);
    this.mWordList = wordList;
}
```

- Tambahkan Source Code untuk method **onCreateViewHolder()**
- Method **onCreateViewHolder()** mirip dengan method **onCreate()**.

@Override

```
public WordViewHolder onCreateViewHolder(ViewGroup parent,
  int viewType) {

    View mView = mInflater.inflate(R.layout.wordlist_item,
                                    parent, false);

    return new WordViewHolder(mView, this);
}
```

- Tambahkan Source Code pada method **onBindViewHolder()**. Method ini menghubungkan data dengan View Holder

@Override

```
public void onBindViewHolder(WordViewHolder holder, int position) {

    String mCurrent = mWordList.get(position);
    holder.wordItemView.setText(mCurrent);
}
```

## 2.7 CREATE THE RECYCLER VIEW IN THE ACTIVITY

- Buka **MainActivity**, tambahkan member variabel untuk **RecyclerView** dan **Adapter**

```
private RecyclerView mRecyclerView;
private WordListAdapter mAdapter;
```

- Pada method **onCreate()**, tambahkan Source Code berikut yang akan membuat **RecyclerView** dan menghubungkan Adapter dengan data.

// Get a handle to the RecyclerView.

```
mRecyclerView = findViewById(R.id.recyclerview);
```

// Create an adapter and supply the data to be displayed.

```
mAdapter = new WordListAdapter(this, mWordList);
```

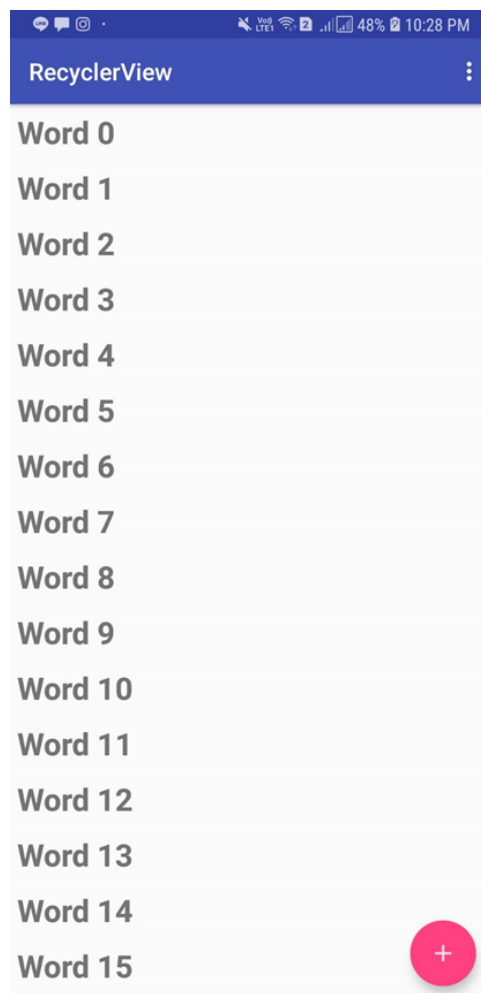
// Connect the adapter with the RecyclerView.

```
mRecyclerView.setAdapter(mAdapter);
```

// Give the RecyclerView a default layout manager.

```
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```

- Aplikasi ketika dijalankan :





## TASK 3 : MAKE THE LIST INTERACTIVE

### 3.1 MAKE ITEMS RESPOND TO CLICKS

- Buka **WordListAdapter**
- Tambahkan **View.OnClickListener**
- Tambahkan Source Code berikut pada method **onClick()**

// Get the position of the item that was clicked.

**int mPosition = getLayoutPosition();**

// Use that to access the affected item in mWordList.

**String element = mWordList.get(mPosition);**

// Change the word in the mWordList.

**mWordList.set(mPosition, "Clicked! " + element);**

// Notify the adapter, that the data has changed so it can

// update the RecyclerView to display the data.

**mAdapter.notifyDataSetChanged();**

- Hubungkan **onClickListner** dengan **View**. Tambahkan Source Code ke **WordViewHolder** constructor

**itemView.setOnClickListener(this);**

### 3.3 ADD BEHAVIOR TO THE FAB

- Pada method **onCreate()** di **MainActivity**,
- Tambahkan **OnClickListener()** pada **FloatingActionButton**

**@Override**

**public void onClick(View view) {**

**int wordListSize = mWordList.size();**

**// Add a new word to the wordList.**

**mWordList.addLast(" Word " + wordListSize);**

**// Notify the adapter, that the data has changed.**

**mRecyclerView.getAdapter().notifyItemInserted(wordListSize);**

**// Scroll to the bottom.**

**mRecyclerView.smoothScrollToPosition(wordListSize);**

**}**

- Aplikasi ketika dijalankan :

