



MOBILE PROGRAMMING 1

Oleh: Niken Riyanti, S.T
Pertemuan 6

Table of contents

01 User Input

Mengetahui tools input untuk aplikasi

02 User Interaction

Membahas interaksi user dengan aplikasi

03 Gestures

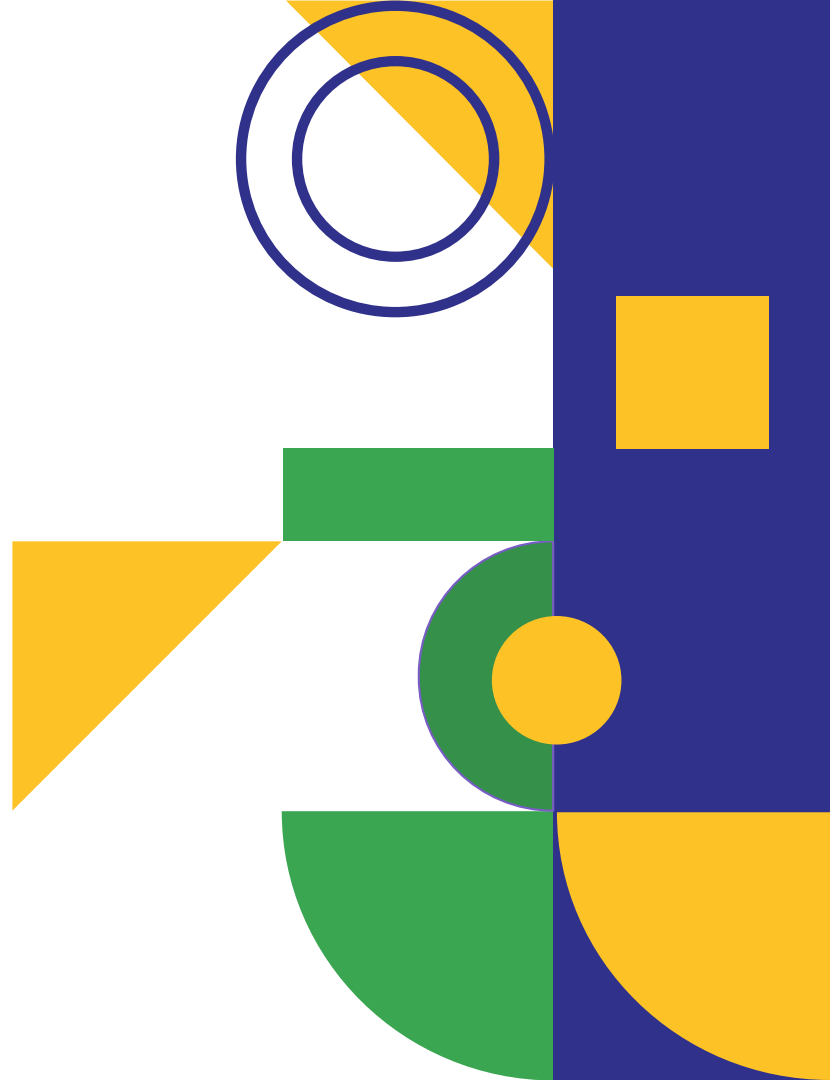
Mengetahui gestur yang terdapat pada kebiasaan user

04 Diskusi

Diskusi mengenai cara pengajaran Bersama Mahasiswa



User Interaction




Ekspektasi User dalam berinteraksi

mengetuk, menekan, berbicara, mengetik, dan mendengarkan menggunakan kontrol masukan pengguna seperti tombol, sssmenu, keyboard, teks box, dan mikrofon

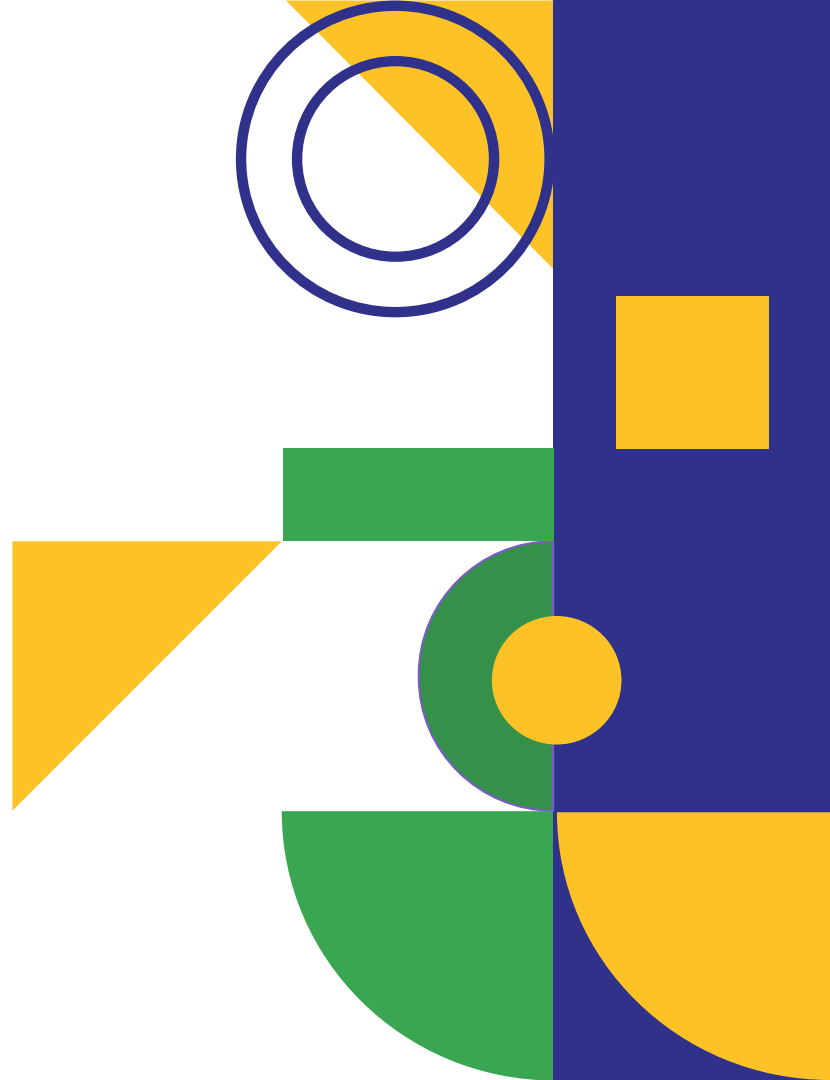
Selain itu, user juga berekspektasi untuk menavigasi antar Activity



Desain Interaksi User

- Penting untuk terlihat jelas, mudah, dan konsisten
 - Pikirkan tentang bagaimana pengguna akan menggunakan aplikasi Anda
 - Minimalkan langkah
 - Gunakan elemen UI yang mudah diakses, dipahami, digunakan
 - Ikuti praktik terbaik Android
 - Memenuhi ekspektasi pengguna
- 

Input Control

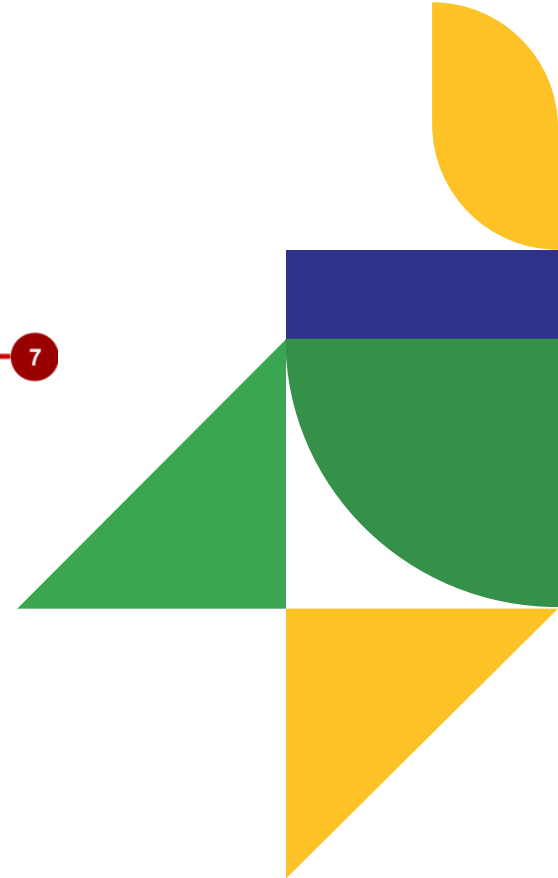
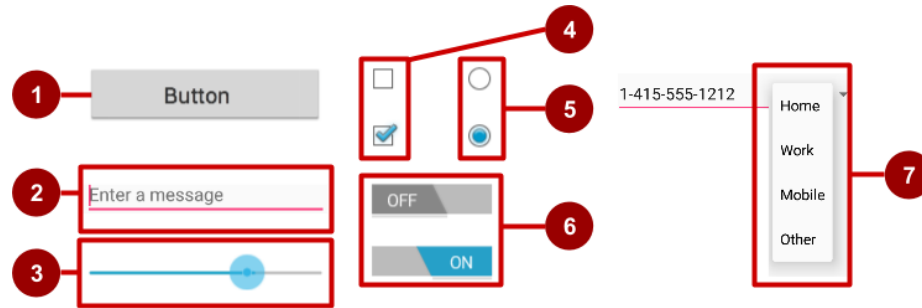


Cara untuk mendapatkan input user

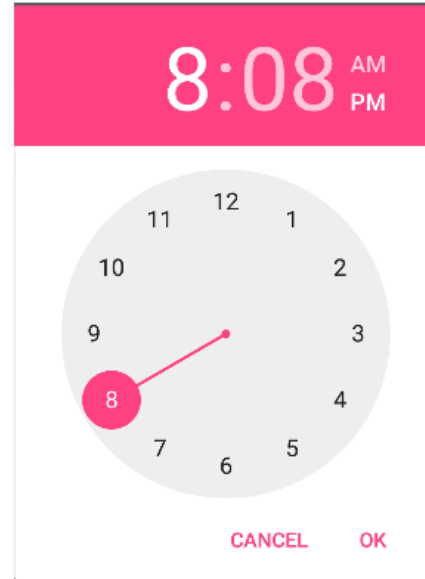
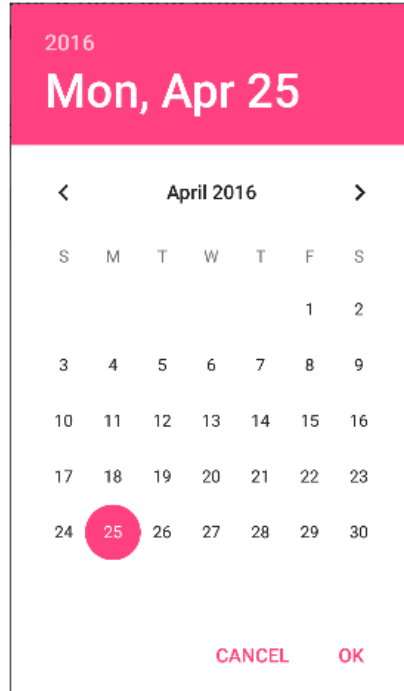
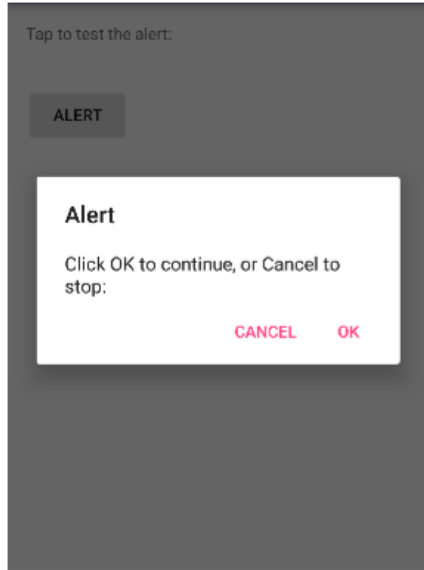
1. Free form
 1. Text and voice input
2. Actions
 1. Buttons
 2. Contextual menus
 3. Gestures
 4. Dialogs
3. Constrained choices
 1. Pickers
 2. Checkboxes
 3. Radio buttons
 4. Toggle buttons
 5. Spinners

Contoh User Input

1. Button
2. Text field
3. Seek bar
4. Checkboxes
5. Radio buttons
6. Toggle
7. Spinner

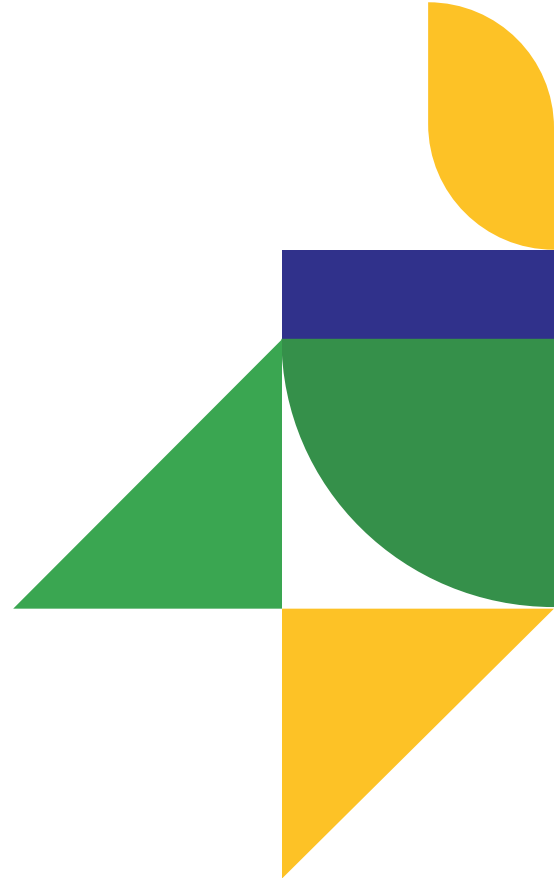


Alert Dialog, Date Picker, Time Picker

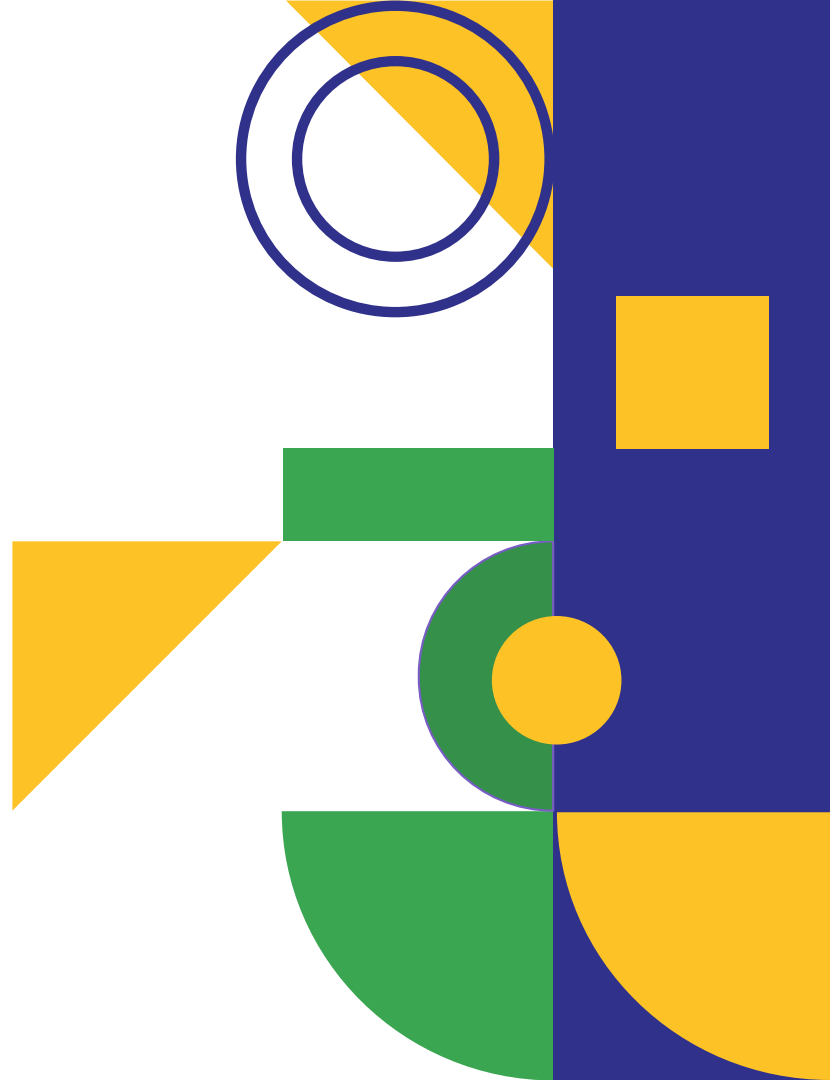


Kelas Dasar untuk mengimplementasikan input tools

View Class adalah blok penyusun dasar untuk semua komponen UI, termasuk kontrol input, View juga merupakan kelas dasar untuk kelas yang menyediakan komponen UI interaktif. View menyediakan interaksi dasar melalui android: onClick



Focus



Focus

Tampilan yang menerima masukan pengguna memiliki "Fokus".
Hanya satu tampilan yang dapat memiliki focus. Fokus membuatnya tidak ambigu tampilan mana yang mendapat masukan data.

Fokus ditetapkan oleh Pengguna :

- mengetuk tampilan
- Aplikasi memandu pengguna dari satu kontrol input teks ke yang berikutnya menggunakan tombol Return, Tab, atau panah
- Memanggil `requestFocus ()` pada tampilan apa pun yang dapat difokuskan



Perbedaan Click dan Focus

Clickable—View can respond to being clicked or tapped

Focusable—View can gain focus to accept input

Input controls such as keyboards send input to the view that has focus



Petunjuk untuk User

- Tunjukkan secara visual tampilan mana yang memiliki fokus sehingga pengguna tahu ke mana arah masukan mereka
- Menunjukkan secara visual tampilan mana yang dapat memiliki fokus membantu pengguna menavigasi melalui alur aplikasi
- Dapat diprediksi dan logis — tidak ada kejutan!



Penggunaan Focus

Arrange input controls in a layout from left to right and top to bottom in the order you want focus assigned

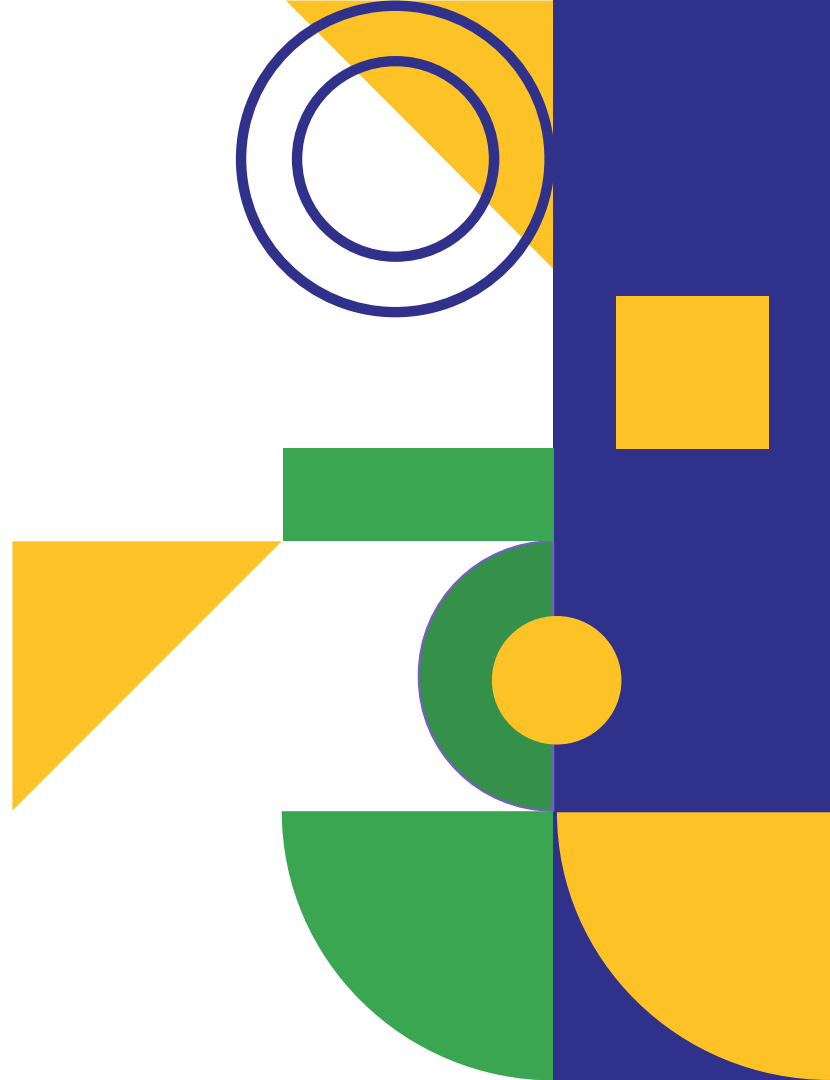
Place input controls inside a view group in your layout

Specify ordering in XML

```
android:id="@+id/top"  
android:focusable="true"  
android:nextFocusDown="@+id/bottom"
```



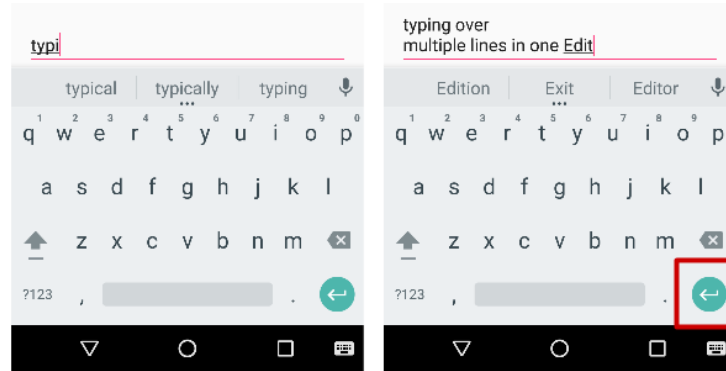
Text Input



Edit Text

EditText class

- Multiple lines of input
- Characters, numbers, and symbols
- Spelling correction
- Tapping the Return (Enter) key starts a new line
- Customizable



"Action" key

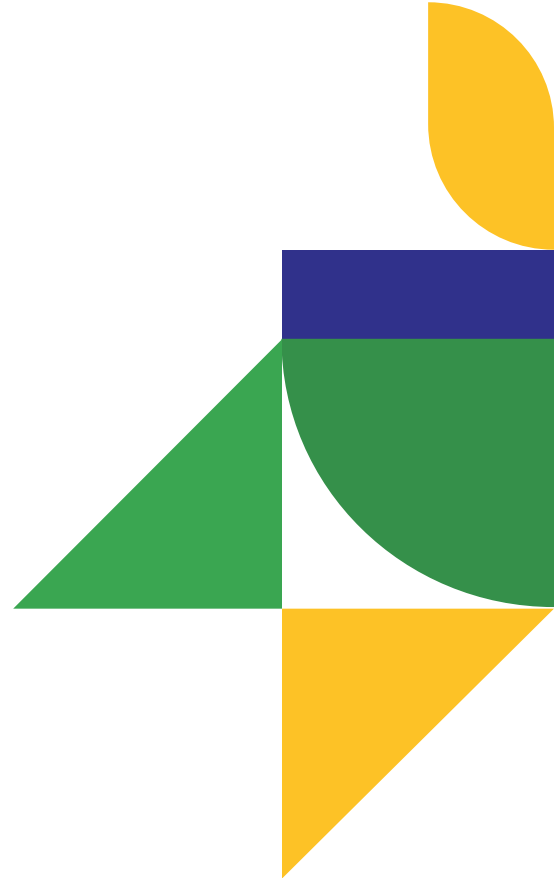
Mengambil teks

Get the EditText object for the EditText view

```
EditText simpleEditText =  
    (EditText) findViewById(R.id.edit_simple);
```

Retrieve the CharSequence and convert it to a string

```
String strValue =  
    simpleEditText.getText().toString();
```



Beberapa tipe inputan

textShortMessage—Limit input to 1 line

textCapSentences—Set keyboard to caps at beginning of sentences

textAutoCorrect—Enable autocorrecting

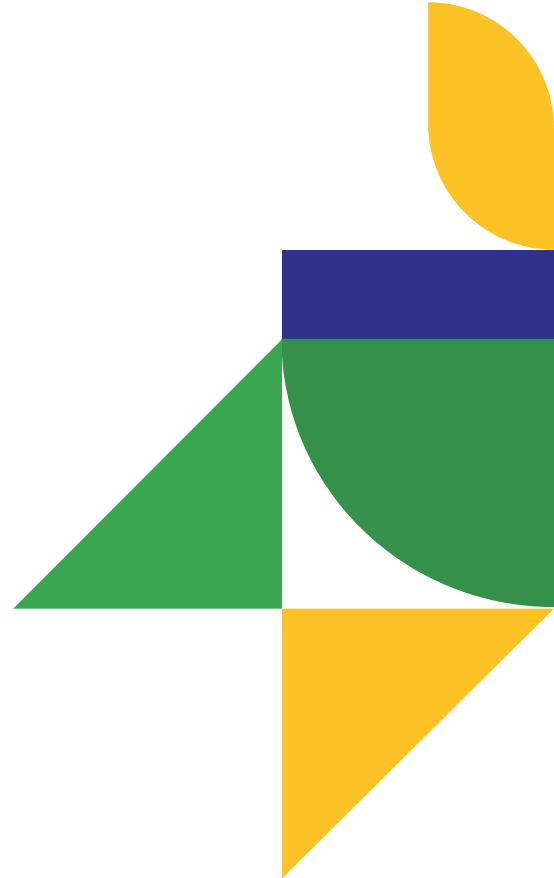
textPassword—Conceal typed characters

textEmailAddress—Show an @ sign on the keyboard

phone—numeric keyboard for phone numbers

```
android:inputType="phone"
```

```
android:inputType="textAutoCorrect|textCapSentences"
```



Button



Button

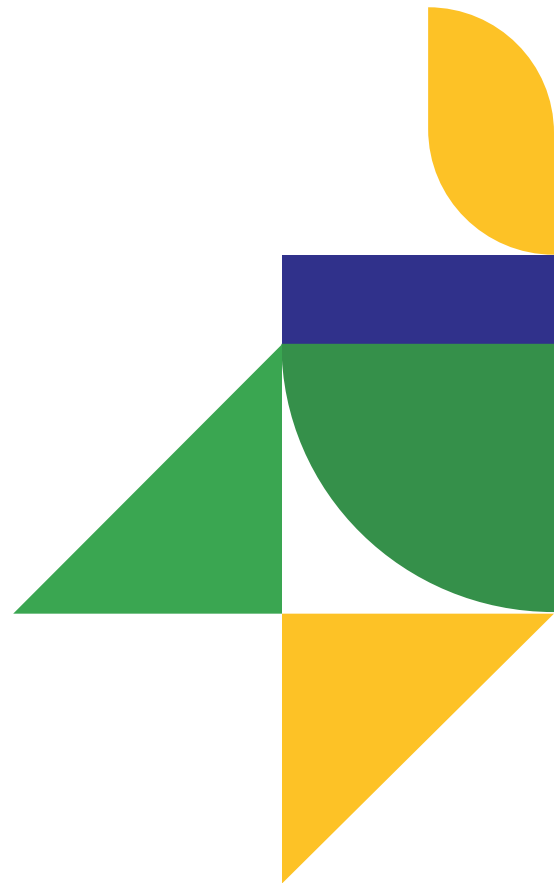
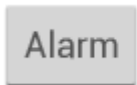
View that responds to clicking or pressing

Usually text or visuals indicate what will happen when it is pressed

Views: Button > ToggleButton, ImageView > FloatingActionButton (FAB)

State: normal, focused, disabled, pressed, on/off

Visuals: raised, flat, clipart, images, text

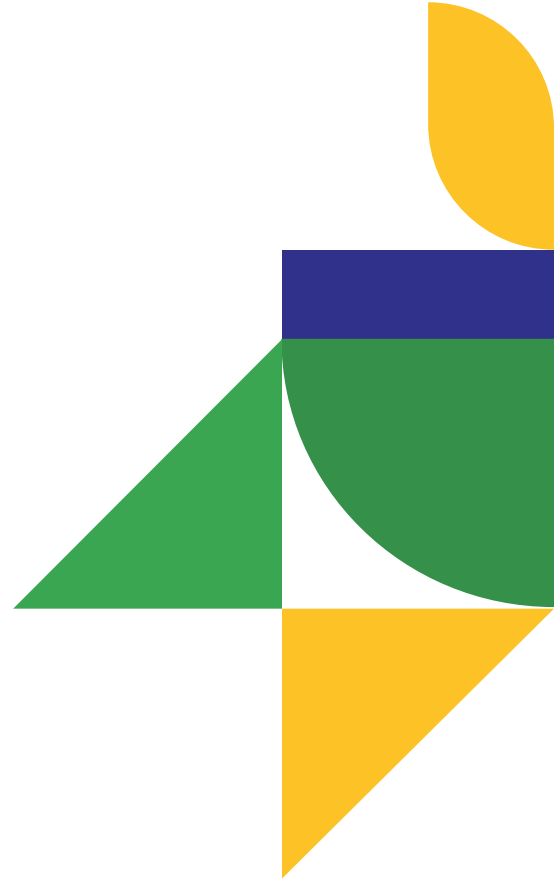


Merespon button tap/click

In your code: Use OnClickListener event listener.

In XML: use android:onClick attribute in the XML layout:

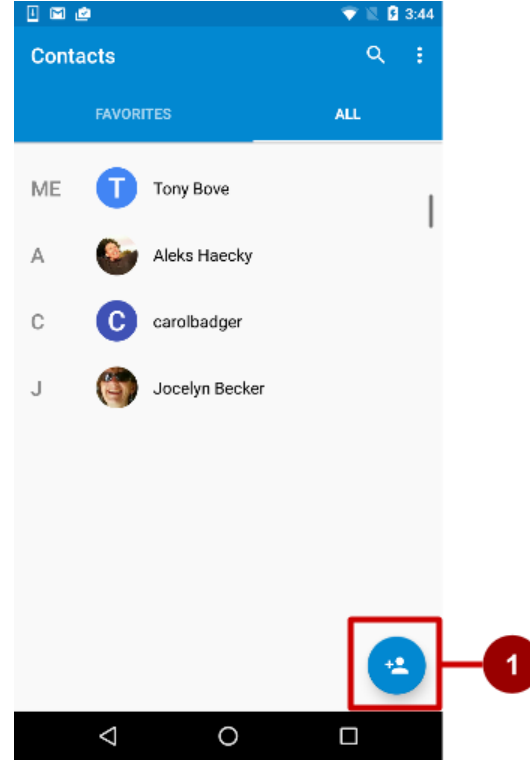
```
<Button  
    android:id="@+id/button_send"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/button_send"  
    android:onClick="sendMessage" />
```



Floating Action Button

Raised, circular, floats above layout
Primary or "promoted" action for a screen
One per screen

For example:
Add Contact button in Contacts app



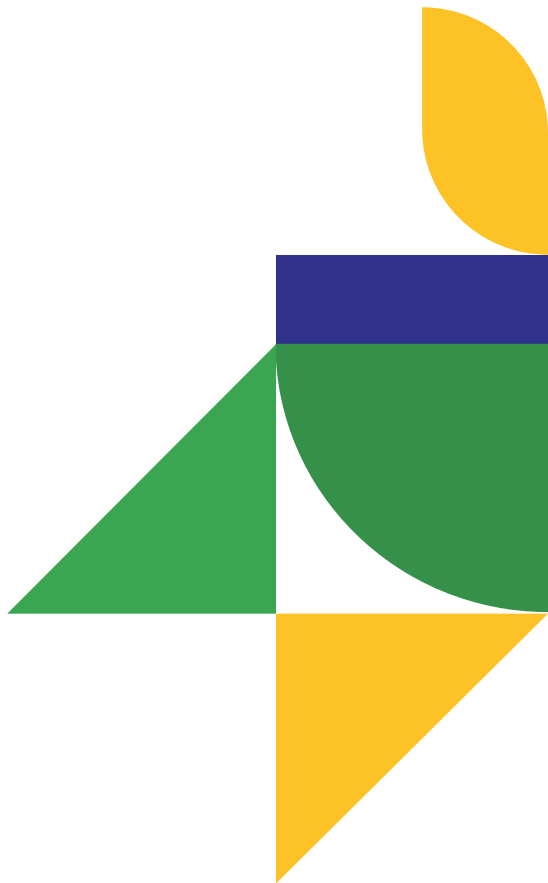
Membuat FAB

Add design support library to build.gradle

compile 'com.android.support:design:a.b.c'

Layout

```
<android.support.design.widget.FloatingActionButton  
    android:id="@+id/fab"  
    android:layout_gravity="bottom|end"  
    android:layout_margin="@dimen/fab_margin"  
    android:src="@drawable/ic_fab_chat_button_white"  
.../>
```



Asset Button Image

Right-click app/res/drawable

Choose **New > Image Asset**

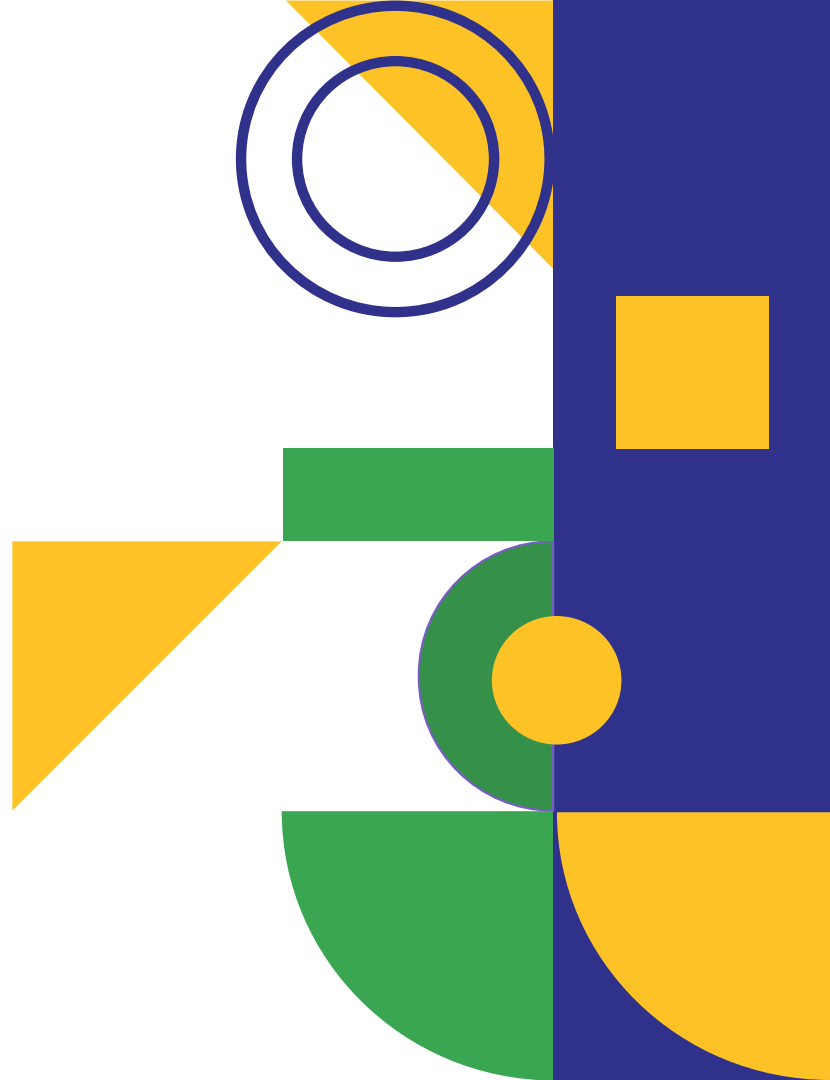
Choose **Action Bar and Tab Items** from drop down menu

Click the **Clipart:** image

(the Android logo)



Memilih Pilihan



Making Choice

- Checkboxes
- Radio buttons
- Toggles
- Spinner

Choose a delivery method:

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

- ☒ Chocolate Syrup
- ☒ Sprinkles
- ☐ Crushed Nuts

Turn on or off:

ON

Turn on or off:



Turn on or off:

OFF

Turn on or off:



1-415-555-1212

Home

Work

Mobile

Other



Checkbox

User can select any number of choices

Checking one box does not uncheck another

Users expect checkboxes in a vertical list

Commonly used with a submit button

Every checkbox is a view and can have an onClick handler

- ☒ Chocolate Syrup
- ☒ Sprinkles
- ☐ Crushed Nuts



RadioButton

User can select one of a number of choices

- Put radio buttons in a RadioGroup
- Checking one unchecks another
- Put radio buttons in a vertical list or horizontally if labels are short
- Every radio button can have an onClick handler
- Commonly used with a submit button for the RadioGroup
-

Choose a delivery method:

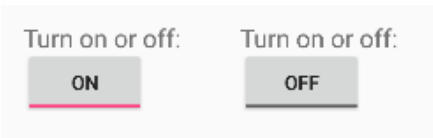
- ☒ Same day messenger service
- ☐ Next day ground delivery
- ☐ Pick up



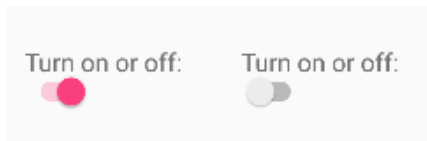
Toogle Botton & Switches

User can switch between 2 exclusive states (on/off)

Use android:onClick+callback—or handle clicks in code



Toggle buttons



Switches



Spinner

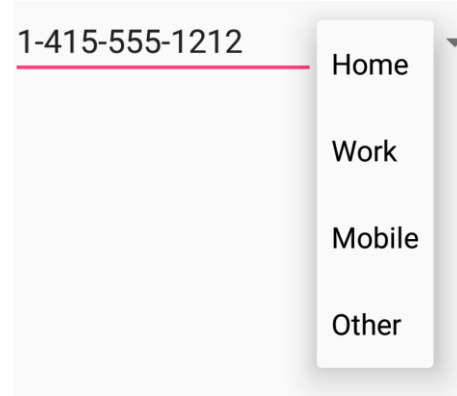
Quick way to select value from a set

Drop-down list shows all values,

user can select only one

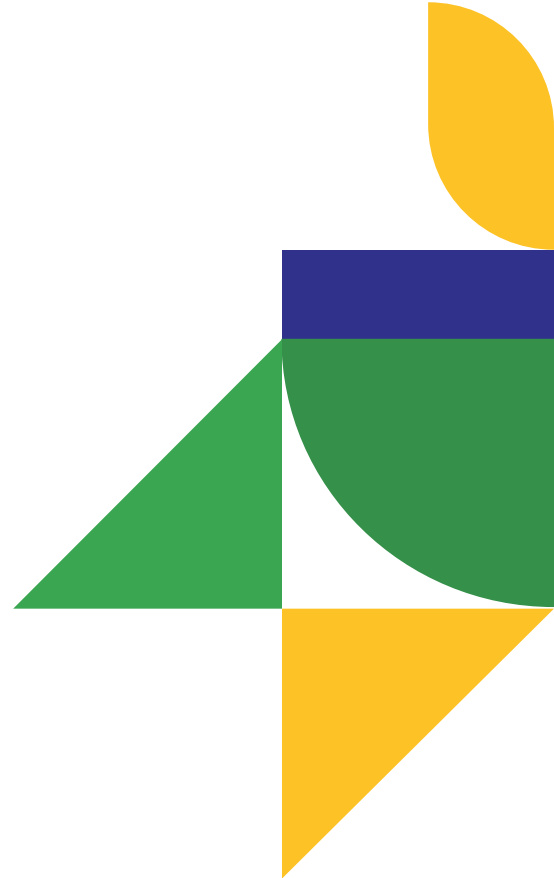
Spinners scroll automatically if necessary

Use the Spinner class.



Membuat Spinner

- Create Spinner UI element in the XML layout
- Define spinner choices in an array
- Create Spinner and set onItemSelectedListener
- Create an adapter with default spinner layouts
- Attach the adapter to the spinner
- Implement onItemSelectedListener method



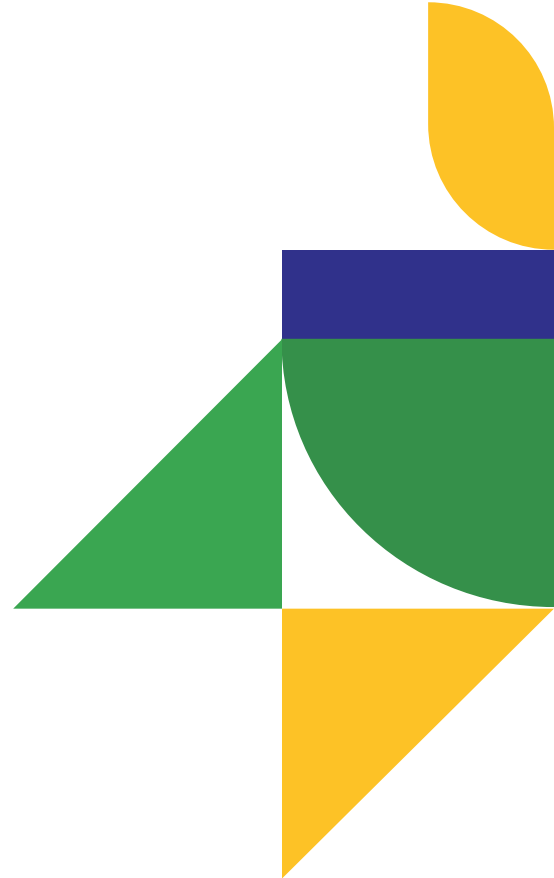
Membuat Spinner di XML

In layout XML file

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

Define Array pilihan ,
In arrays.xml resource file

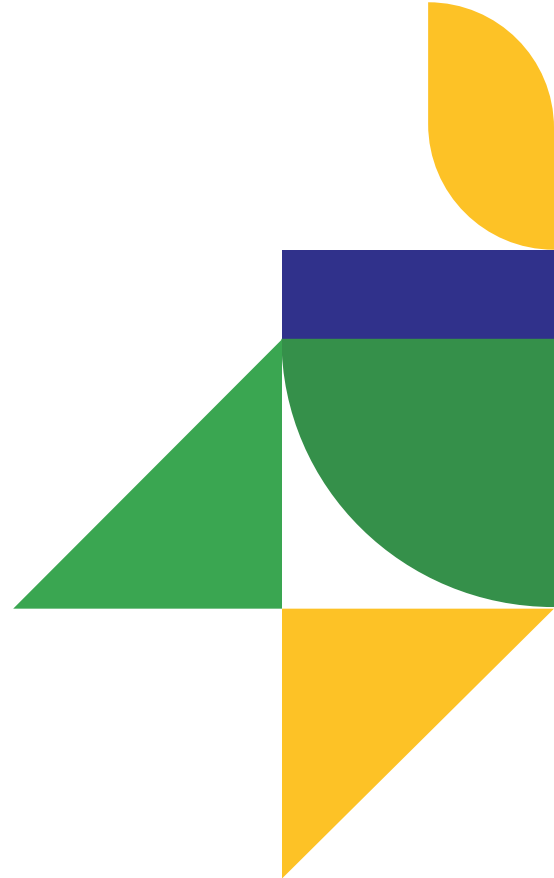
```
<string-array name="labels_array">  
    <item>Home</item>  
    <item>Work</item>  
    <item>Mobile</item>  
    <item>Other</item>  
</string-array>
```



Membuat Spinner dan melampirkan listener

```
public class MainActivity extends AppCompatActivity implements  
AdapterView.OnItemSelectedListener
```

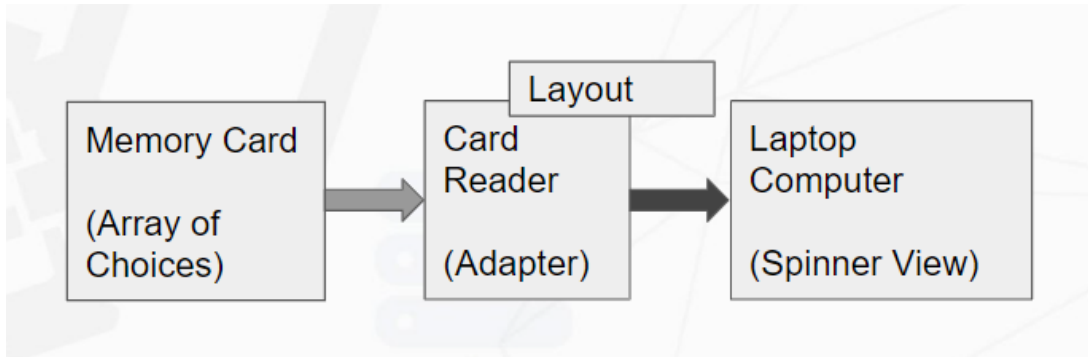
```
// In onCreate()  
Spinner spinner = (Spinner) findViewById(R.id.label_spinner);  
if (spinner != null) {  
    spinner.setOnItemSelectedListener(this);  
}
```



Adapter?

An adapter is like a bridge, or intermediary, between two incompatible interfaces

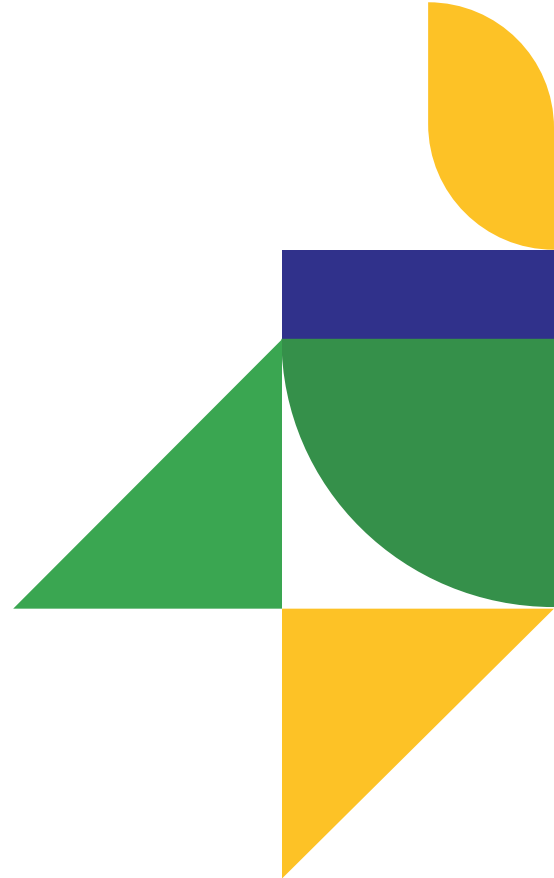
For example, a memory card reader acts as an adapter between the memory card and a laptop



Adapter?

Create ArrayAdapter using string array
and default spinner layout

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



Me;ampirkan adapter ke spinner

Specify the layout for the drop down menu

```
adapter.setDropDownViewResource(  
    android.R.layout.simple_spinner_dropdown_item);
```

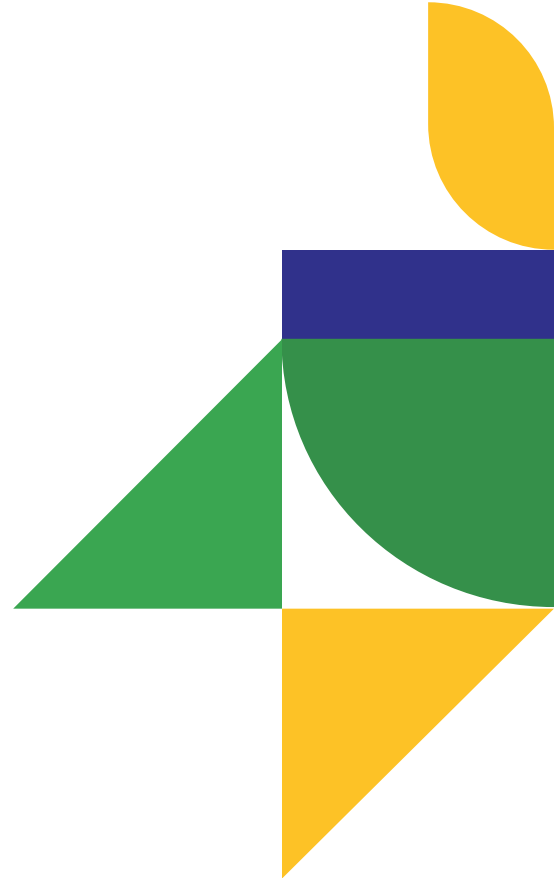
Attach the adapter to the spinner
`spinner.setAdapter(adapter);`



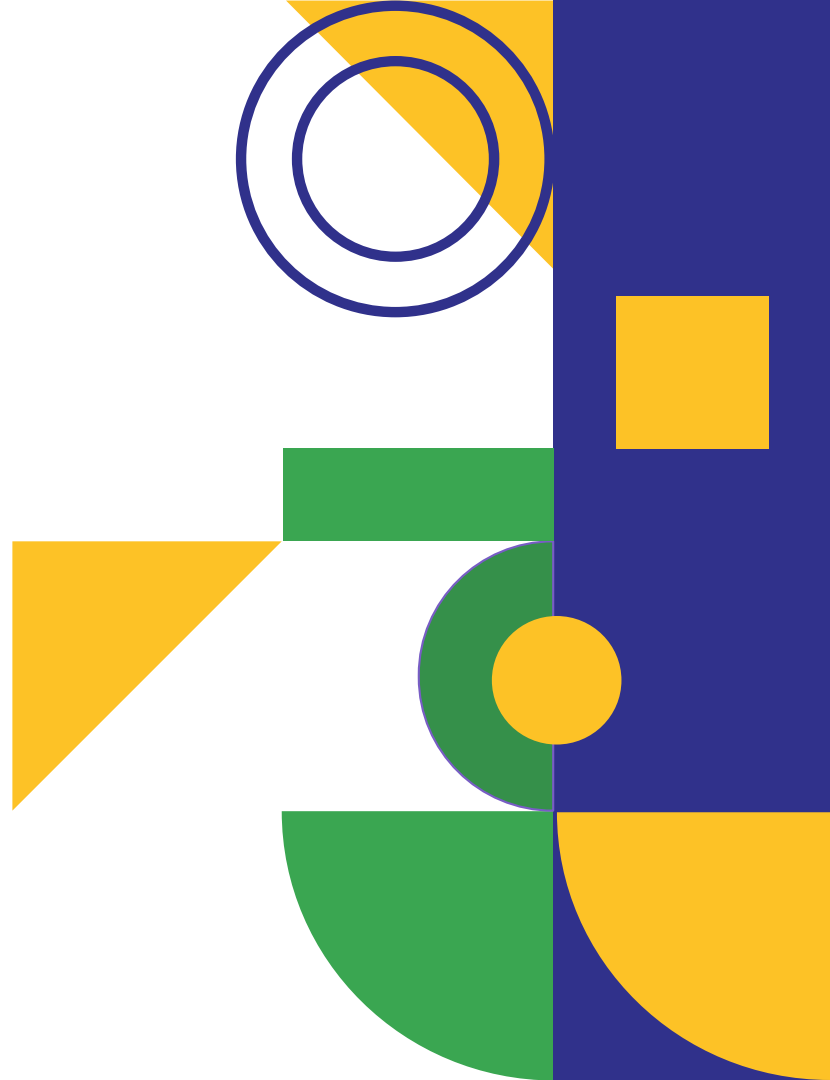
Implementasi onItemSelectedListener

```
public class MainActivity extends AppCompatActivity implements  
AdapterView.OnItemSelectedListener
```

```
public void onItemSelected(AdapterView<?> adapterView,  
    View view, int pos, long id) {  
    String spinner_item =  
        adapterView.getItemAtPosition(pos).toString();  
    // Do something here with the item  
}  
public void onNothingSelected(AdapterView<?> adapterView) {  
    // Do something  
}
```



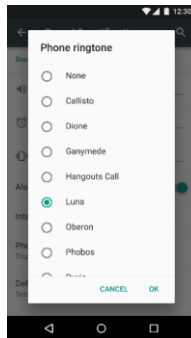
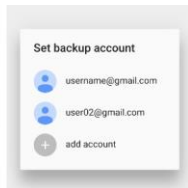
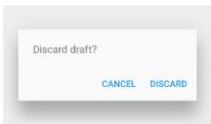
Dialogs



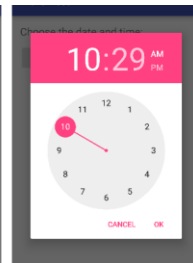
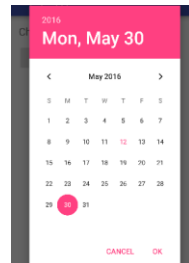
Dialogs

Dialog appears on top, interrupting the flow of activity

Require user action to dismiss



Alert Dialog



Date & Time Picker

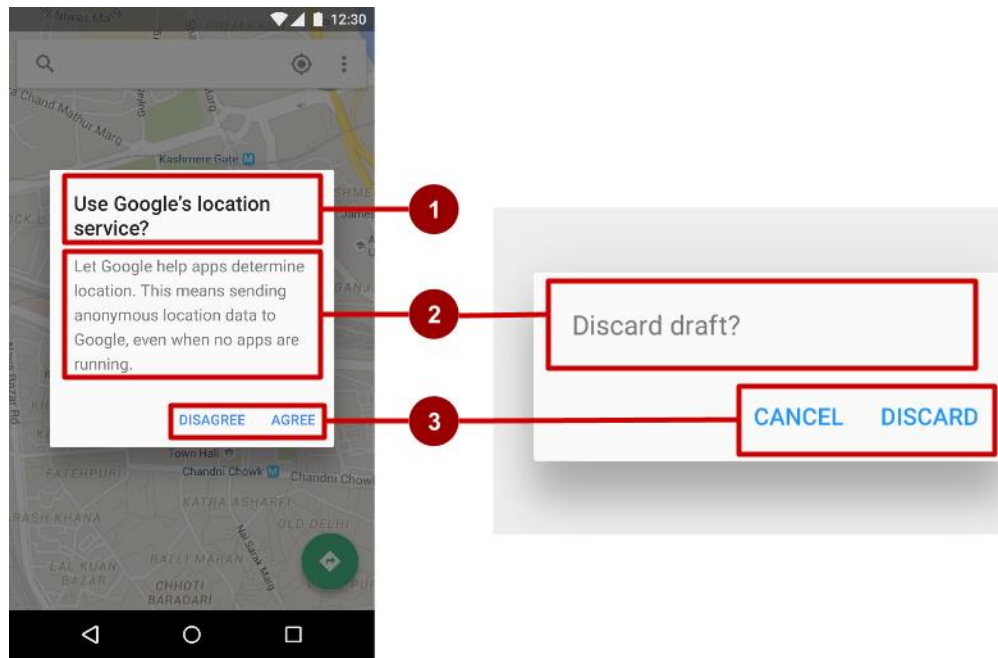
Alert Dialog

AlertDialog can show:

Title (optional)

Content area

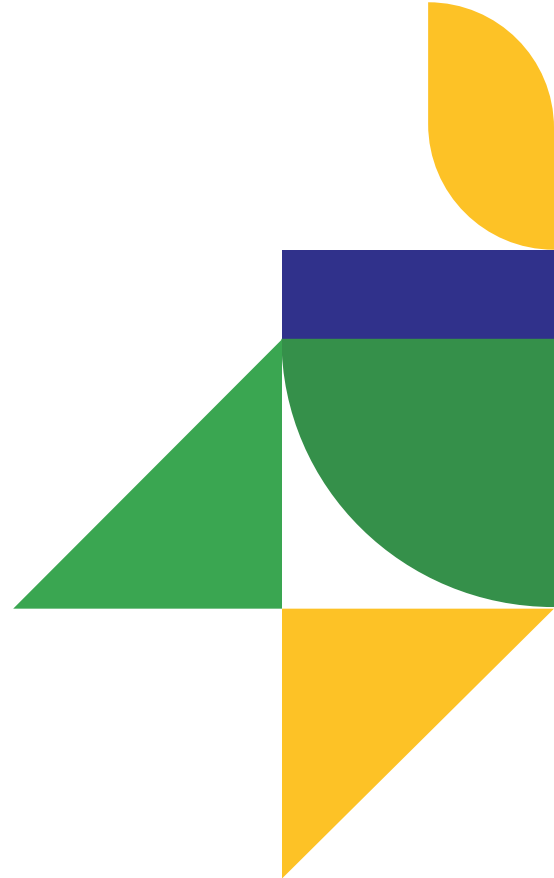
Action buttons



Membuat Alert Dialog

Use AlertDialog.Builder to build a standard alert dialog and set attributes:

```
public void onClickShowAlert(View view) {  
    AlertDialog.Builder alertDialog = new  
        AlertDialog.Builder(MainActivity.this);  
    alertDialog.setTitle("Connect to Provider");  
    alertDialog.setMessage(R.string.alert_message);  
    ...  
}
```



Alert Dialog

```
alertDialog.setPositiveButton()  
alertDialog.setNeutralButton()  
alertDialog.setNegativeButton()
```

Contoh

```
alertDialog.setPositiveButton(  
    "OK", new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog, int which) {  
            // User clicked OK button.  
        }  
    });
```

Same pattern for `setNegativeButton()` and `setNeutralButton()`



Picker use fragment

Use DialogFragment to show a picker

DialogFragment is a window that floats on top of activity's window



Fragment?

A fragment is like a mini-activity within an activity

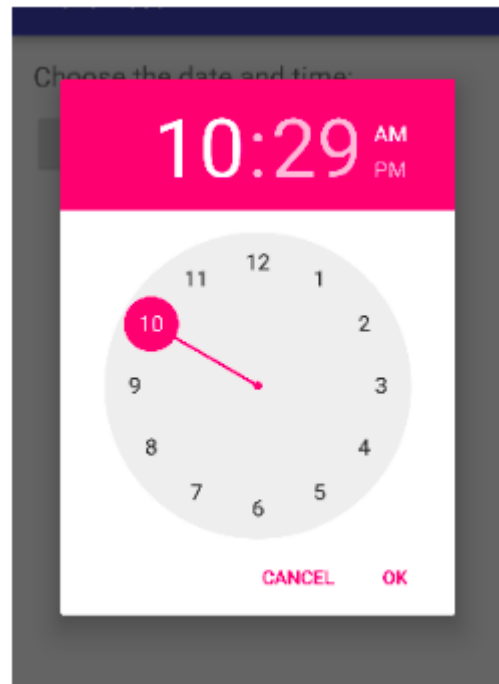
Manages its own own lifecycle.

Receives its own input events.

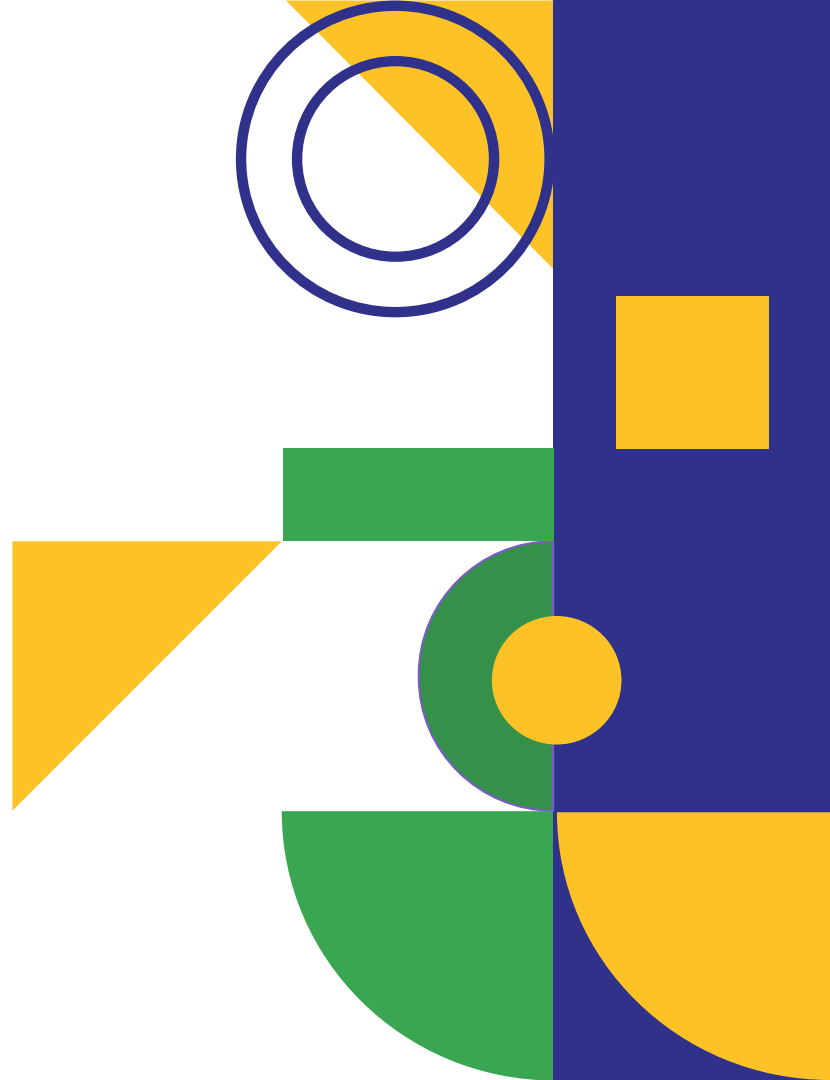
Can be added or removed while parent activity is running

Multiple fragments can be combined in a single activity

Can be reused in multiple activities



Gestures



Gestures

Touch gestures include:

- long touch
- double-tap
- fling
- drag
- scroll
- pinch



Gestures

Classes and methods are available to help you handle gestures.

GestureDetectorCompat class for common gestures

MotionEvent class for motion events



Thanks!

Do you have any questions?

niken@sttbandund.ac.id

089606684746

