

Pertemuan 5

Menu

A. Materi

- Options Menu
- Context Menu

B. Kompetensi Dasar

- Mengenal macam menu
- Menerapkan options menu dan context menu

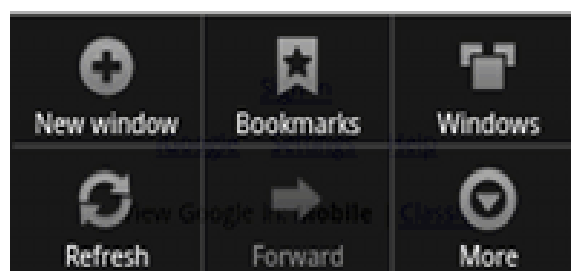
C. Tujuan

- Mahasiswa mampu mengetahui perbedaan options menu dan context menu
- Mahasiswa mampu menerapkan options menu dan context menu

D. Dasar Teori

Menu digunakan untuk menampilkan opsi-opsi lain pada aplikasi. Menu menampilkan fungsi aplikasi tanpa mengkonsumsi banyak ruang view. Pada Android 2.3(Api level 10) dibawah user dapat menemukan Options Menu dengan cara menekan button menu Pada Android 3.0(Api level 11) dan diatasnya item yang ada pada menu disajikan pada action bar.

Pada **Gambar 5.1.** disajikan menu pada Android 2.3 (Api level 10) kebawah sedangkan pada **Gambar 5.2.** disajikan menu pada android 3.0 (Api level 11) keatas. Pada android 3.0 (Api level 11) menu bisa juga disebut dengan ActionBar.



Gambar 5.1. Menu pada Android 2.3 (Api level 10) kebawah



Gambar 5.2. Menu pada Android 3.0 (Api level 11) keatas

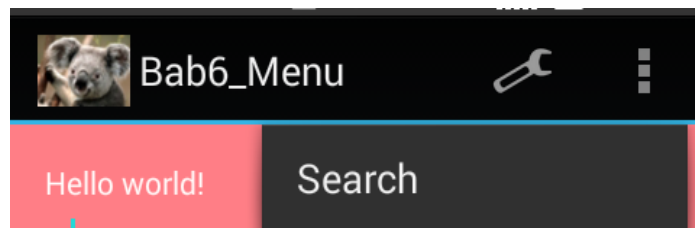
Pada praktikum ini kita langsung mempelajari menu pada android 3.0 (api level 11) keatas. Karena beberapa handphone sudah mulai menghilangkan tombol menu.

Sedangkan Android menyediakan dua tipe menu Android menyediakan yaitu Options menu dan Context menu.

D.1. Options Menu

Options menu atau action bar adalah menu yang disajikan diatas tampilan aplikasi. Untuk membuat Options menu pada activity bisa dengan mengoverride method [onCreateOptionsMenu\(\)](#). Ketika pertama kali project dibuat method ini sudah otomatis tercreate. Bisa dilihat pada file .java sudah terdapat method ini. Begitupula dengan interface untuk menampilkan menu, software otomatis membuat UI menu pada file main_menu.xml yang terletak pada folder app/res/menu/main_menu.xml. Ketika program di running maka akan nampak tiga titik vertical di pojok kanan atas. Sehingga kita hanya perlu menambahkan UI menu yang akan ditampilkan pada main_menu.xml dan menambahkan program pada method [onOptionsItemSelected\(\)](#) untuk memberikan aksi ketika menu dipilih oleh user.

Sebagai latihan kita akan membuat options menu seperti pada **Gambar 5.3.**



Gambar 5.3. Options menu

Langkah untuk membuat Options Menu :

1. Membuat file .xml sebagai UI dari options menu
2. Memilih theme yang support dengan actionBar
3. Menambahkan method [onCreateOptionsMenu\(\)](#) untuk membuat Options menu
4. Menuliskan method [onOptionsItemSelected\(\)](#) untuk menghandle ketika menu dipilih

Membuat file .xml sebagai UI dari options menu

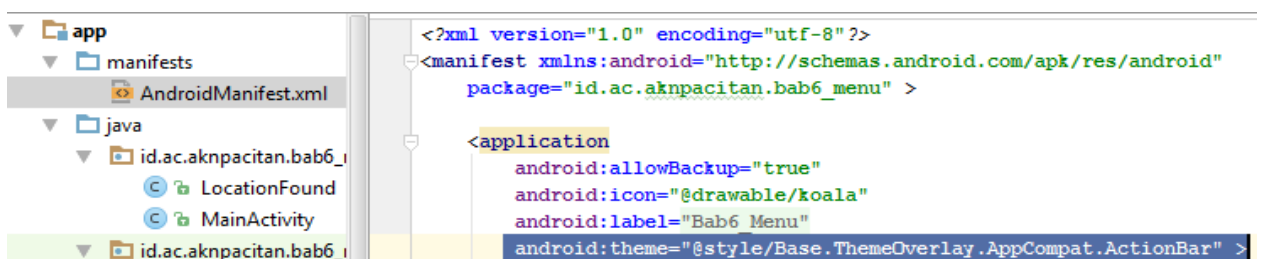
Buat project baru, Cari file main_menu.xml, tambahkan koding sebagai berikut :

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/action_search"
        android:icon="@android:drawable/ic_menu_search"
        android:showAsAction="never"
        android:title="Search">
    </item>
    <item
        android:id="@+id/menu_overflow"
        android:icon="@android:drawable/ic_menu_preferences"
        android:title="Other"
        android:showAsAction="always">
    </item>
</menu>
```

Properties *android:showAsAction="value"* memiliki nilai yang berbeda-beda menyesuaikan kebutuhan. Untuk menu yang ditampilkan diluar menu dengan icon titik tiga set value menjadi *"always"*. Namun apabila ingin menampilkan menu menjadi sub komponen menu dengan icon titik tiga maka set value menjadi *"never"*. Sedangkan properti *android:icon="@drawable/image_name"* digunakan untuk menampilkan menu disertai icon.

Memilih theme yang support dengan actionbar

Theme juga mempengaruhi dalam menampilkan menu. Edit pada file Android Manifest, pada *android:theme* pilih theme yang mendukung action bar. Berikut contohnya pada **Gambar 5.4**.



Gambar 5.4. Setting theme yang mendukung ActionBar

Menambahkan method [onCreateOptionsMenu\(\)](#) untuk membuat Options menu

Pada file .java cari method [onCreateOptionsMenu\(\)](#). Terdapat script inflater.inflate(R.menu.menu_main,menu). menu_main merupakan nama file untuk UI menu. Pastikan nama tersebut sama dengan nama file .xml pada app/res/menu/nama_menu.xml

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_main, menu);
    return super.onCreateOptionsMenu(menu);
}

```

Menuliskan method [onOptionsItemSelected\(\)](#) untuk menghandle ketika menu dipilih

Ketika menu dipilih maka akan diberikan aksi. Misalkan apabila dipilih menu setting akan ditampilkan halaman setting. Kita akan memberikan aksi ketika menu pada **Gambar 5.3**. dipilih dengan menambahkan coding pada `onCreateOptionsMenu()` seperti berikut :

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.menu_overflow) {
        Toast.makeText(this, "Menu Setting", 5).show();
        return true;
    }
    if (id == R.id.action_search) {
        Toast.makeText(this, "Menu Search", 5).show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

Pada coding diatas terdapat conditional if yang berfungsi untuk mengecek menu yang dipilih. Pengecekan ini berdasarkan id dari masing-masing menu. Seperti komponen widget, item menu pada android juga memiliki id. Sehingga akan dicocokkan id yang dipilih dengan id yang terdapat pada `main_menu.xml`, apabila sama maka program akan menjalankan pernyataan di dalam if.

Menampilkan Menu dalam Sub Item

Options menu juga dapat menampilkan sub item dari sebuah menu. Pada **Gambar 5.4**. dapat dilihat bahwa icon preference memiliki sub item diantaranya refresh dan help. Untuk menambahkan sub item edit file `main_menu.xml` tambahkan script berikut :

```

<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/action_search"
        android:icon="@android:drawable/ic_menu_search"
        android:showAsAction="never"
        android:title="Search">
    </item>
    <item
        android:id="@+id/menu_overflow"
        android:icon="@android:drawable/ic_menu_preferences"
        android:showAsAction="always">
        <menu>
            <!-- Refresh -->
            <item
                android:id="@+id/action_refresh"
                android:icon="@android:drawable/ic_menu_revert"
                android:showAsAction="never"
                android:title="Refresh"/>
            <!-- Help -->
            <item
                android:id="@+id/action_help"
                android:icon="@android:drawable/ic_menu_help"
                android:showAsAction="never"
                android:title="Help"/>
        </menu>
    </item>
</menu>

```

Menu dengan id `action_search` dengan property `showAsAction="never"` maka dia akan terletak pada menu didalam icon tiga titik vertical. Sedangkan untuk menambah sub item dengan cara :

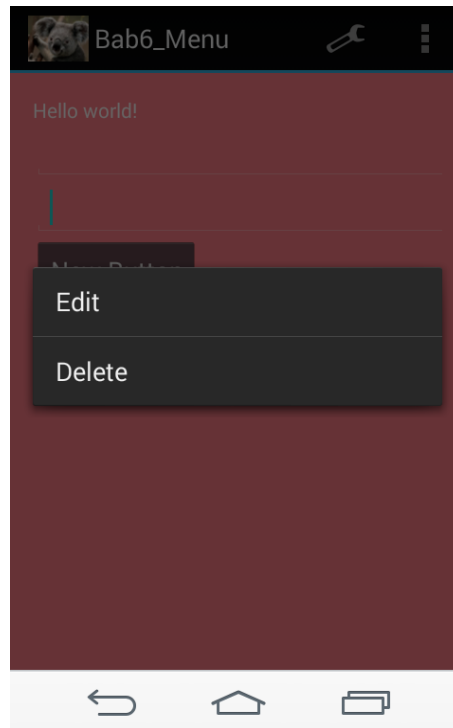
```

<item>
    <menu>
        <item .....property..... />
        <item .....property..... />
        .
        .
        .
        <item .....property..... />
    </menu>
</item>

```

D.2. Context Menu

Context menu adalah menu yang disajikan ketika sebuah komponen ditekan dan ditahan. Untuk membuat Options menu pada activity bisa dengan mengoverride method [onCreateContextMenu\(\)](#) seperti pada **Gambar 5.5**.



Gambar 5.5. Context Menu

Langkah-langkah membuat context menu :

1. Membuat file .xml sebagai UI dari context menu
2. Membuat method [onCreateContextMenu\(\)](#) untuk membuat context menu
3. Menuliskan method [onContextItemSelected\(\)](#) untuk menghandle ketika menu dipilih
4. Menerapkan menu pada suatu komponen widget. Komponen widget tersebut ketika ditekan dan ditahan akan menampilkan context view

Membuat file .xml sebagai UI dari Context View

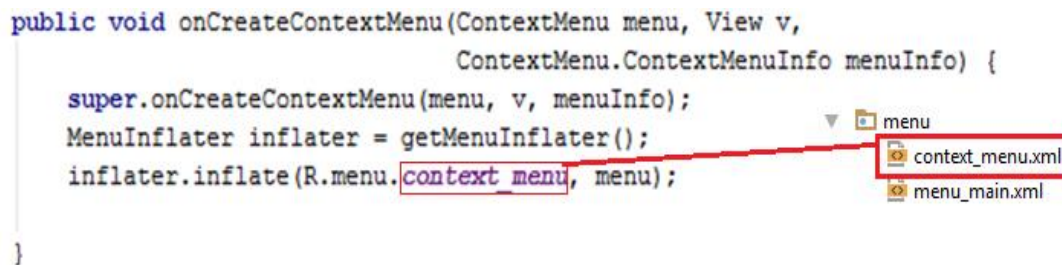
Tambahkan file context_menu.xml dengan berisikan sebagai berikut :

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/edit"
        android:title="Edit" />
  <item android:id="@+id/delete"
        android:title="Delete" />
</menu>
```

Terlihat bahwa menu terdapat dua item yaitu edit dan delete.

Membuat method [onCreateContextMenu\(\)](#) untuk membuat context menu

```
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.context_menu, menu);
}
```



Pastikan nama file .xml sama dengan tulisan yang dikotak merah. Karena menunjuk file menu yang akan ditampilkan.

Menuliskan method [onContextItemSelected\(\)](#) untuk menghandle ketika menu dipilih

Tidak hanya sebatas menampilkan menu. Menu yang dipilih akan memberikan aksi tertentu. Aksi ini ditambahkan pada method onContextItemSelected yang memberikan id menu yang dipilih.

```
public boolean onContextItemSelected(MenuItem item) {
    AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo) item.getMenuInfo();
    switch (item.getItemId()) {
        case R.id.edit:
            // editNote(info.id);
            return true;
        case R.id.delete:
            //deleteNote(info.id);
            return true;
        default:
            return super.onContextItemSelected(item);
    }
}
```

Didalam method tersebut tambahkan coding conditional if untuk melakukan pengecekan id yang dipilih user.

Menerapkan menu pada suatu komponen widget. Komponen widget tersebut ketika ditekan dan ditahan akan menampilkan context view

Ini merupakan langkah terakhir dengan mendaftarkan item yang akan dikenai context menu. Misalkan saja terdapat user interface EditText, widget EditText apabila diklik dan ditahan akan menampilkan context menu. Untuk mendaftarkan EditText tersebut adalah sebagai berikut :

```

<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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity"
    android:orientation="vertical"
    android:background="#ff7e86">

    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView"
        android:textColor="#ffffdc" />
    <EditText
        android:layout_width="353dp"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        />
    <EditText
        android:layout_width="355dp"
        android:layout_height="wrap_content"
        android:id="@+id/editText2"
        />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button"
        android:onClick="proses" />
</LinearLayout>

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    et=(EditText)findViewById(R.id.editText);
    et2=(EditText)findViewById(R.id.editText2);
    registerForContextMenu(et2);
}

```

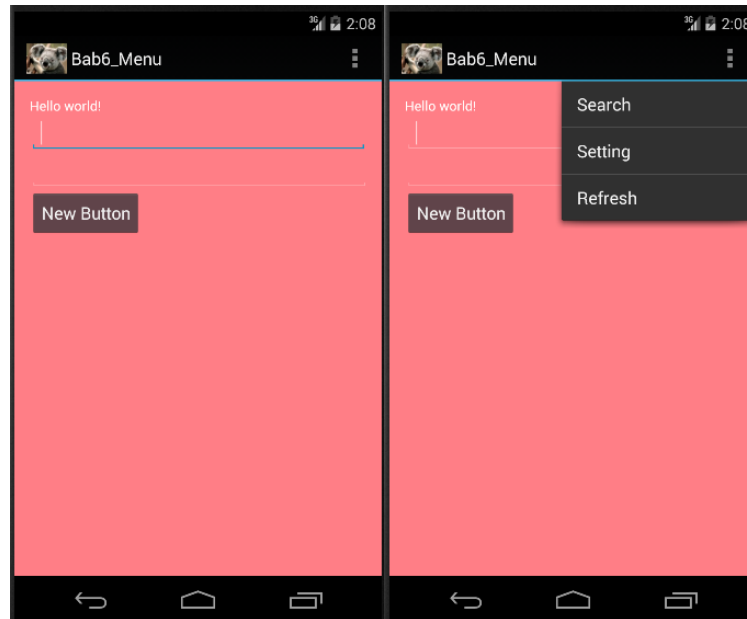
Pada file .java setelah pendeklarasian EditText, daftarkan EditText tersebut dengan method `registerForContextMenu(name_of_variable_object_widget)`

E. Percobaan

Buatlah Options menu dan Context menu dengan mengikuti langkah-langkah pada dasar teori!

F. Praktikum

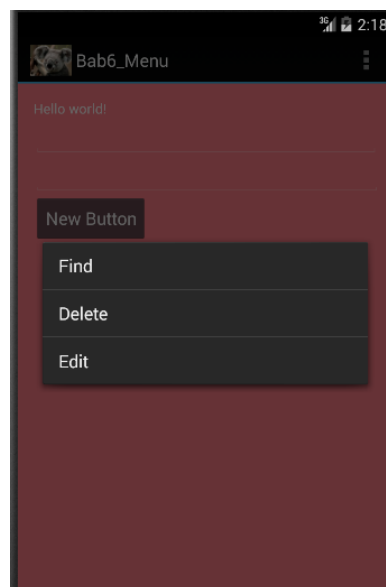
1. Buat Options menu seperti **Gambar 5.6**.



Gambar 5.6. Hasil praktikum 5 no 1

Ketika menu search dipilih akan ditampilkan “*Ini menu search*”, Begitupula dengan menu setting dan refresh.

2. Buat Context menu seperti **Gambar 5.7**.

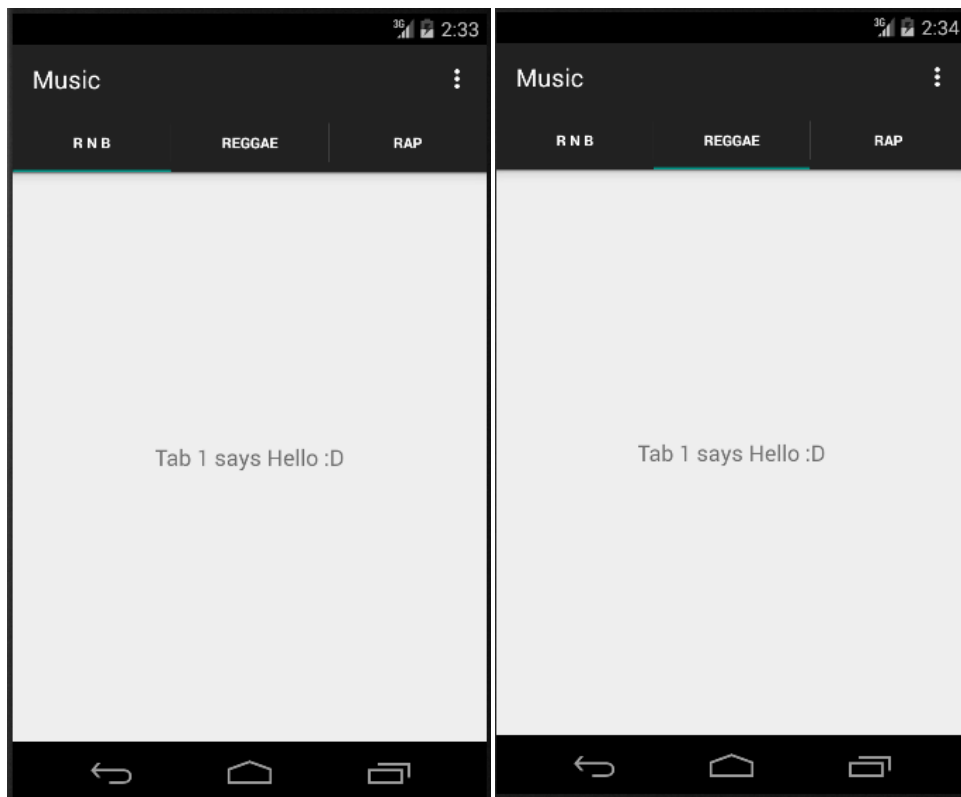


Gambar 5.7. Hasil praktikum 5 no 2

Ketika menu find dipilih akan ditampilkan “*Ini menu find*”, Begitupula dengan menu delete dan Edit

G. Studi Kasus

Membuat Tab menu seperti **Gambar 5.8.**:



Gambar 5.8. Hasil studi kasus praktikum 5