Nama : Tika Bunga Cintia Manurung

NPM : 1941411

Kelas : SI-B

**" JAWABAN UTS APPLICATION MOBILE DEVELOPMENT "**

Aplikasi Todo

**1. New Project**

Baik kita mulai saja, buat Project baru dengan nama ToDoList, dengan cara File>New Project>Congfigure your new project, isikan nama project, dsb lalu finish. Kita akan menemukan layout Hello World untuk layout default new project Android. Didalam project terdapat file/komponen AndroidManifest.xml, dimana akan terlihat seperti berikut:

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

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

package="com.logbook.wira.todolist" >

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:supportsRtl="true"

android:theme="@style/Theme.AppCompat.Light.NoActionBar" >

<activity

android:name=".MainActivity"

android:label="@string/app\_name"

android:theme="@style/AppTheme.NoActionBar" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

</application>

</manifest>

Dan juga file MainActivity.java seperti berikut :

package com.logbook.wira.todolist;

import android.app.Activity;

import android.os.Bundle;

public class MainActivity extends Activity {

/\*\*

\* Called when the activity is first created.

\*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

}

}

Selain itu kita juga akan menemukan file content\_main.xml yang memiliki struktur seperti berikut :

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

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

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

xmlns:app="http://schemas.android.com/apk/res-auto" android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:paddingBottom="@dimen/activity\_vertical\_margin"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

tools:showIn="@layout/activity\_main" tools:context=".MainActivity">

<TextView

android:text="Hello World!"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

</RelativeLayout>

**2. Membuat View-nya**

Oke mari kita mulai dengan membuat daftar list kegiatan dengan cara menimpa code dibawah ini yang berada di file content\_main.xml:

<TextView

android:text="Hello World!"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

menjadi seperti ini :

<ListView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@android:id/list"

android:layout\_weight="1" />

Next layout baru didalam folder res/layout dengan nama file task\_view.xml, lalu isikan code berikut kedalamnya:

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

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

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<RelativeLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

xmlns:android="http://schemas.android.com/apk/res/android">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="New Text"

android:id="@+id/list"

android:layout\_alignParentTop="true"

android:layout\_alignParentLeft="true"

android:layout\_alignParentStart="true"

android:textSize="20dp"

android:layout\_toLeftOf="@+id/delete"

android:layout\_alignBottom="@+id/delete"

android:gravity="center\_vertical" />

<Button

android:onClick="onDeleteButtonClick"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Delete"

android:id="@+id/delete"

android:layout\_alignParentTop="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentEnd="true" />

<Button

android:onClick="update"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Update"

android:id="@+id/update"

android:layout\_alignParentTop="true"

android:layout\_toLeftOf="@+id/delete"

android:layout\_toStartOf="@+id/delete" />

</RelativeLayout>

</LinearLayout>

**3. Membuat Fungsi-nya**

Selanjutnya buka file MainActivity.java, lalu buat method baru dengan nama addtask(). Isikan code berikut pada method tersebut.

public void addtask () {

Log.d("MainActivity", "Add a new task");

}

Lalu timpa method onCreate yang sebelumnya sudah ada pada file MainActivity.java dengan code berikut :

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu,menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_add\_task:

Log.d("MainActivity","Add a new task");

return true;

default:

return false;

}

}

Hal ini bertujuan untuk menampilkan dialog add new task, dimana nantinya pada baris code ini kita akan membuat fungsi tambah task baru. Jalankan aplikasi, dan lihat LogCat nya di Android Monitor. Harusnya akan tampil seperti ini :

**com.logbook.wira.todolist D/MainActivity﹕ Add a new task**

Kalau udah berhasil menanpilkan LogCat nya, mari kita buat fitur tambah task, dengan cara mengupdate method addtask() sebelumnya dengan code berikut :

public void addtask () {

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Add a task");

builder.setMessage("What do you want to do?");

final EditText inputField = new EditText(this);

builder.setView(inputField);

builder.setPositiveButton("Add", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

Log.d("MainActivity",inputField.getText().toString());

}

});

builder.setNegativeButton("Cancel", null);

builder.create().show();

}

Sebelum menjalankan aplikasi, mari kita buat tombol untuk menjalankan method diatas dengan cara memanfaatkan FloatingActionButton yang sudah ada saat pertama kali kita membuat project. Cari code yang hampir sama seperti code dibawah ini

**FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);**

Lalu jadikan seperti code ini.

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);

fab.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

// Snackbar.make(view, "Replace with your own action", Snackbar.LENGTH\_LONG)

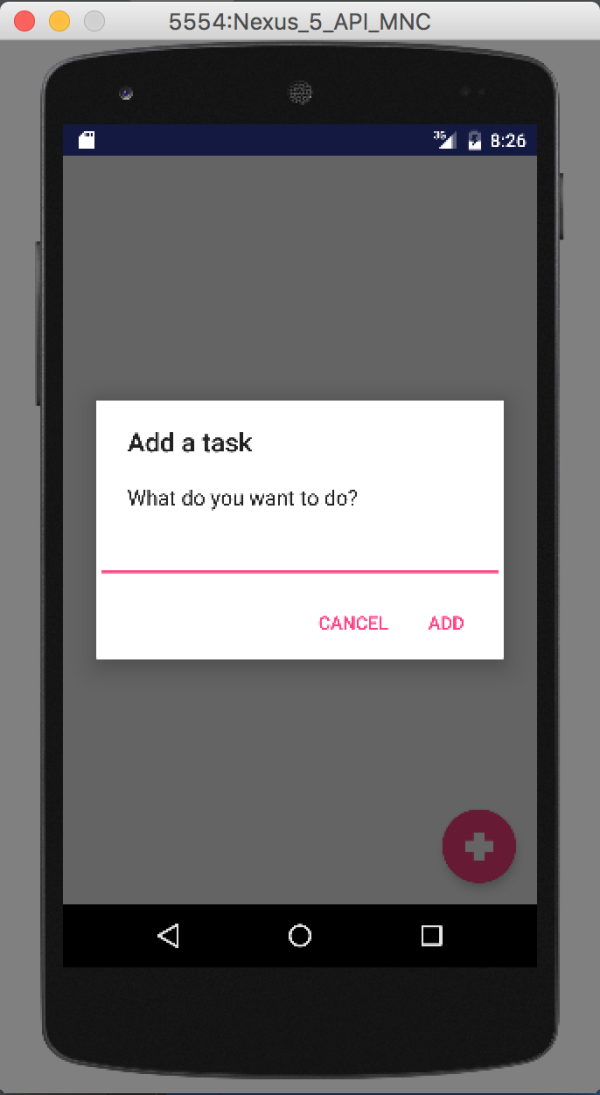
// .setAction("Action", null).show();

addtask();

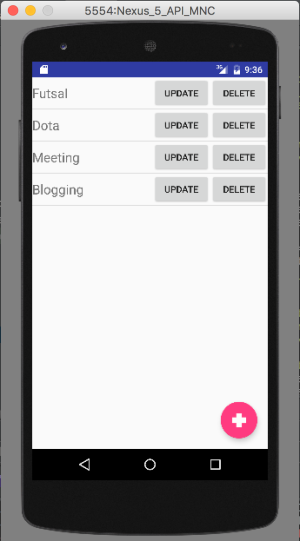
}

});

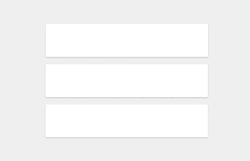
Sekarang kita sudah bisa jalankan aplikasi :



Penambahan Tulisan Di Aplikasi :



**Membuat Fitur Delete.**



Jika kita telah berhasil menambahkan task baru, yang perlu kita lakukan selanjutnya adalah bagaimana cara menghapusnya atau menandakan bahwa task ini sudah selesai dikerjakan. Pada tutorial kali ini akan dibahas dua metode untuk menghapus, pertama cara konvensional (tombol delete), dan yang kedua swipe to dismiss. Tapi alangkah baiknya kita mulai bertahap dari cara konvesional.

Jika sebelumnya kita telah membuat button delete dengan id seperti ini:

**android:onClick=&amp;quot;onDoneButtonClick&amp;quot;**

Maka selanjutnya kita membuat fungsinya, yaitu dengan menambahkan method onDoneButtonClick. Yang artinya method ini akan dipanggil ketika tombol diatas di sentuh. Buat method onDoneButtonClick seperti dibawah ini, dan tambahkan pada MainActivity.java.

public void onDeleteButtonClick(View view) {

View v = (View) view.getParent();

TextView taskTextView = (TextView) v.findViewById(R.id.list);

String task = taskTextView.getText().toString();

String sql = String.format("DELETE FROM %s WHERE %s = '%s'",

TaskContract.TABLE, TaskContract.Columns.TASK, task);

helper = new TaskDBHelper(MainActivity.this);

SQLiteDatabase sqlDB = helper.getWritableDatabase();

sqlDB.execSQL(sql);

updateUI();

}

Jalankan Aplikasi Tekan Tombol Delete.

Ini Tampilan Aplikasi Setelah Di Tekan Tombol Delete.

