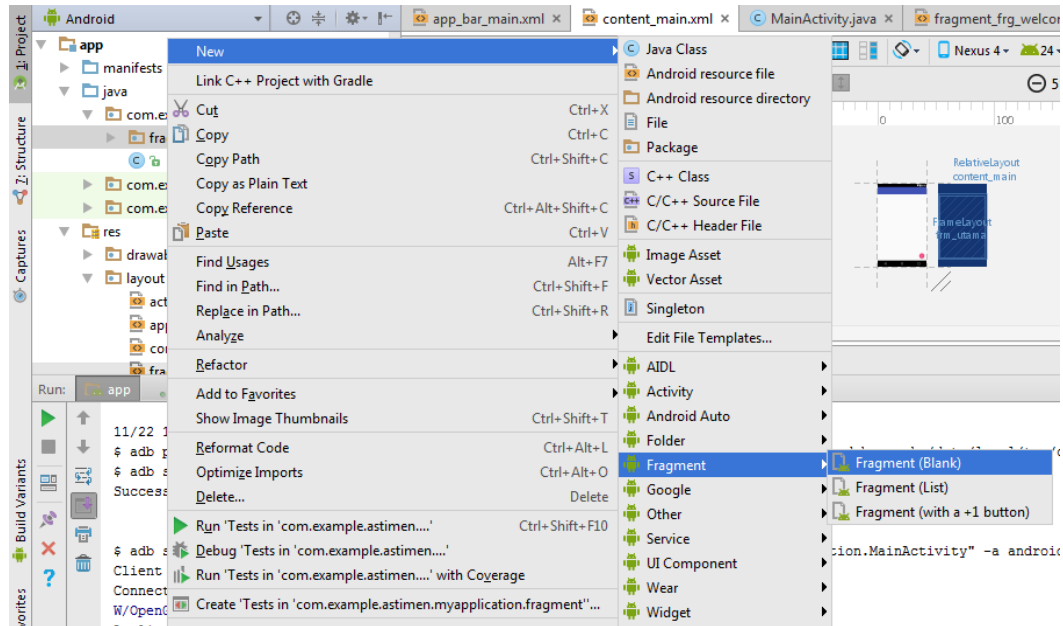
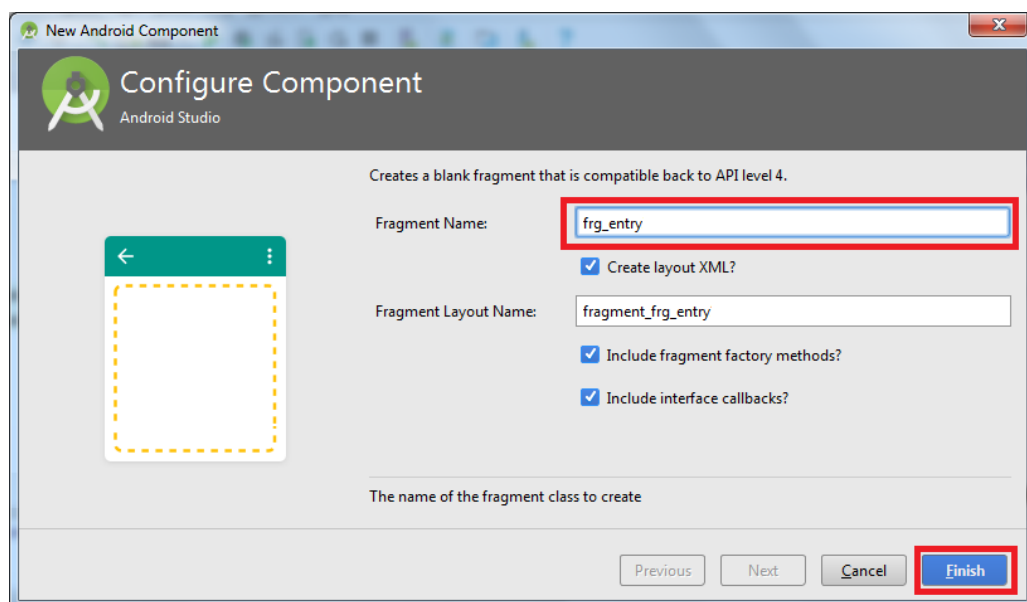


BUTTON CLICK PADA FRAGMENT DAN RELATIVE LAYOUT

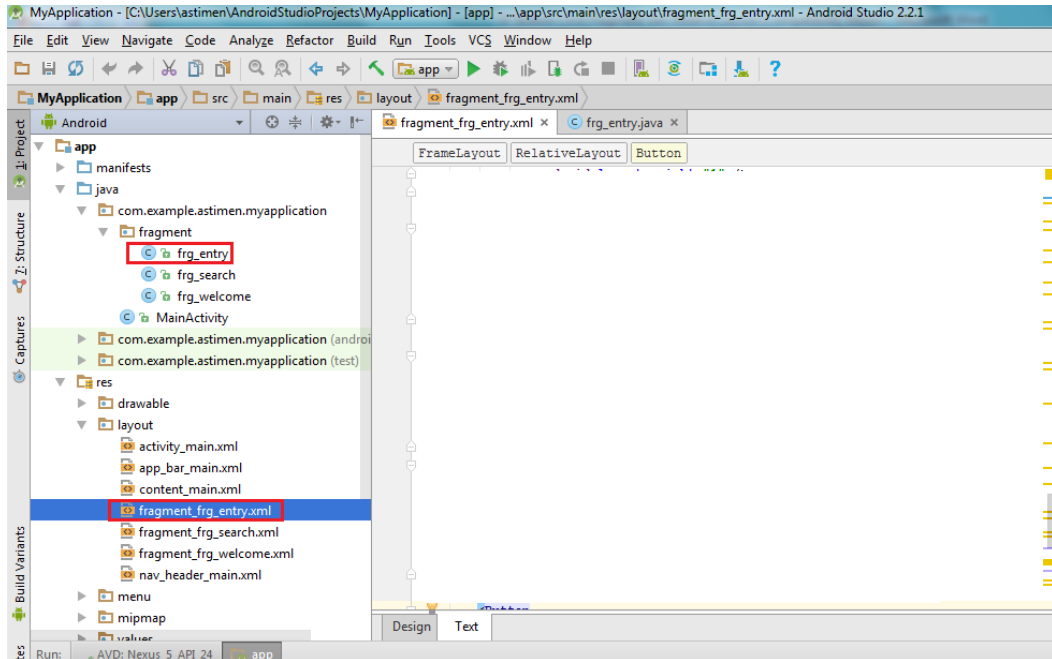
1. Pada modul latihan ini kita akan membuat salah satu layout: RelativeLayout form entri untuk tabel user pada sebuah fragment frg_entry. Pada layout tersebut terdapat dua buah Button NEW dan SAVE. Jika diklik Button tersebut akan ditampilkan informasi pada TOAST. Pendefinisian Toast pada main activity akan berbeda syntax nya jika dieksekusi di fragment.
2. Klik kanan package fragment. Klik menu **New→Fragment→Fragment(Blank)**



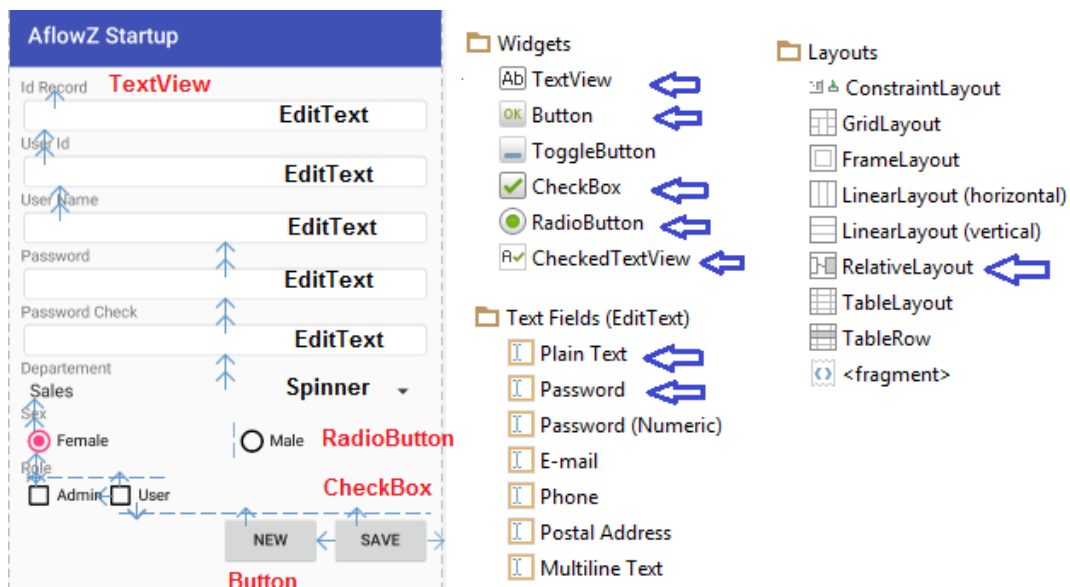
3. Akan ditampilkan form dialog Configure Component. Isi Fragment Name dengan frg_entry.



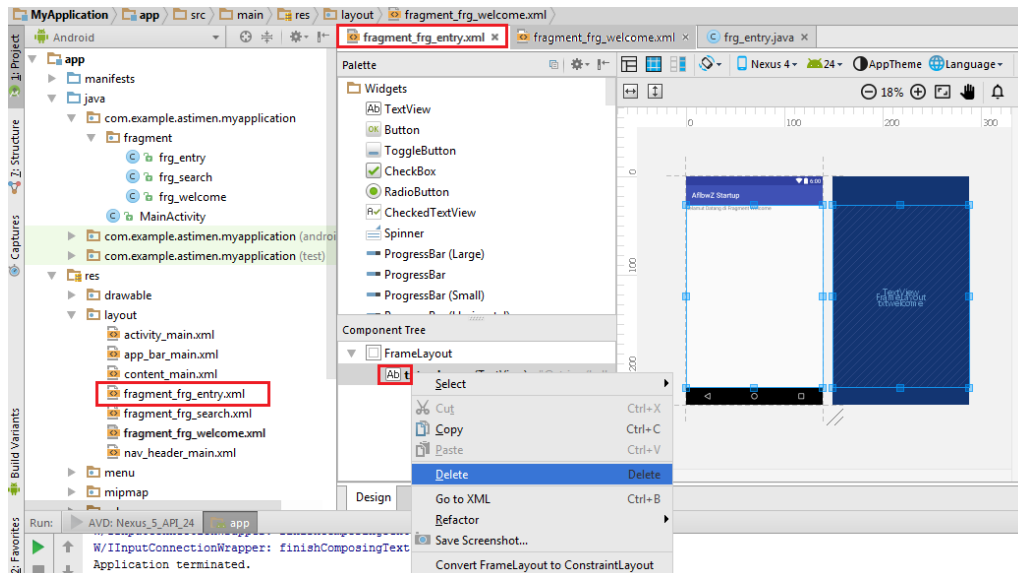
- Setelah klik button FINISH maka pada folder **app→java→com→fragment** akan ditambahkan file java fragment **frg_entry.java**. Pada folder **app→res→layout** akan ditambahkan file layout **fragment_frg_entry.xml**



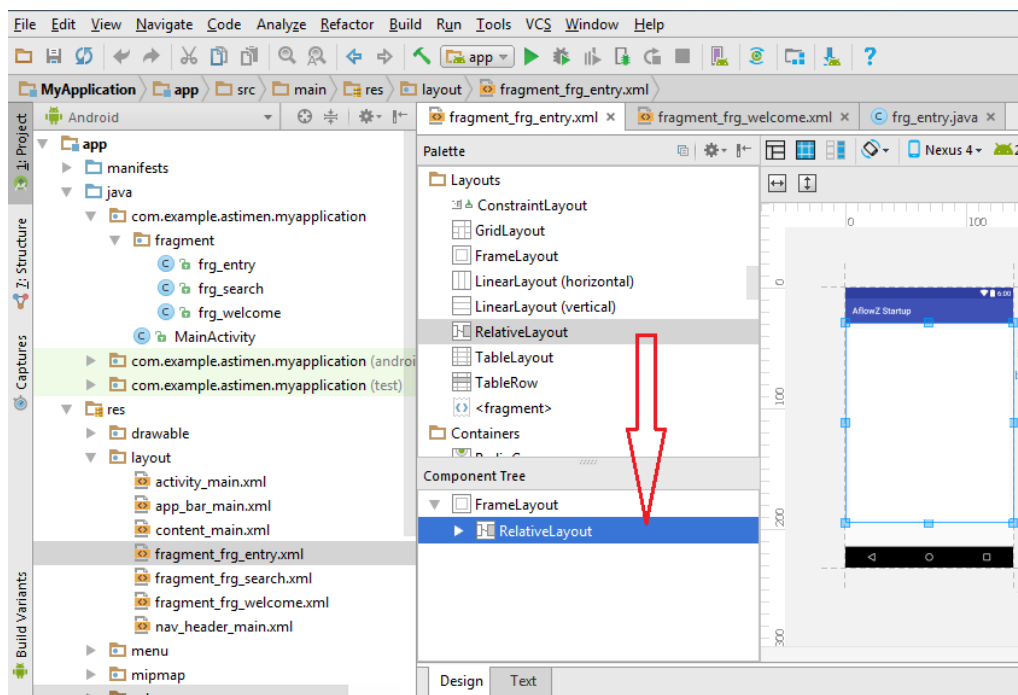
- Kita akan membuat RelativeLayout untuk sebuah form entry dengan field id_record, user_id, user_nama, password, password2, department, female dan male, admin dan user, button NEW dan Button SAVE



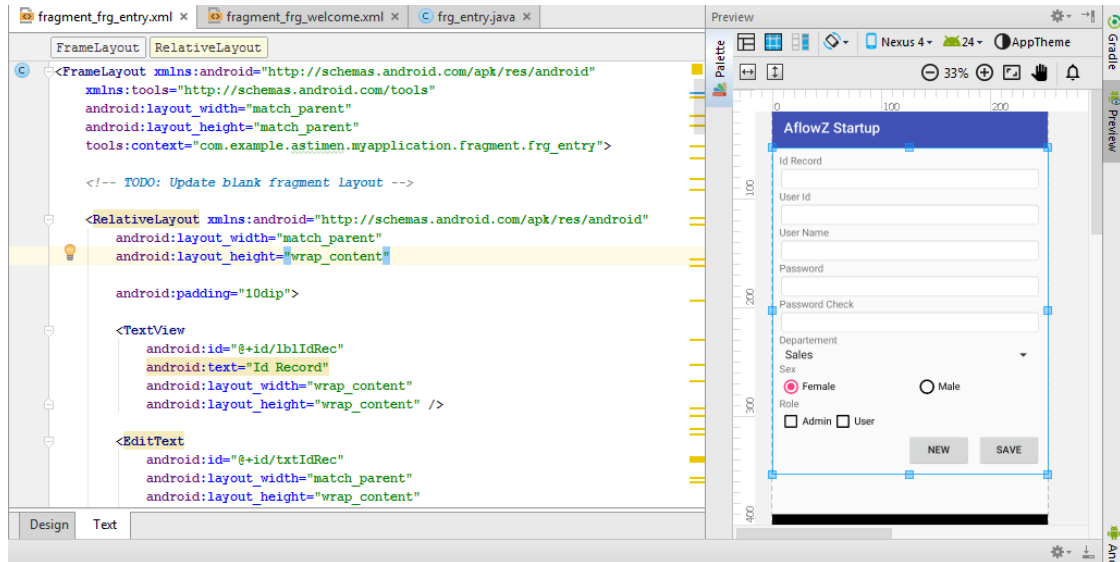
6. RelativeLayout adalah sebuah model layout berbasis xml yang posisi suatu komponen relative terhadap komponen lain. Misalnya EditText txtIdRec berada dibawah TextView lblIdRec. Dengan mengacu pada komponen paling atas sebagai referensi awal kita bisa menyusun letak secara berkelanjutan dibawahnya.
7. Double klik untuk membuka file xml layout **fragment_frg_entry.xml** dalam mode DESIGN secara visual. Delete TextView dalam FrameLayout agar bersih.



8. Drag and drop palette layouts RelativeLayout ke Component Tree didalam FrameLayout.



9. Drag and drop dari palette kelompok Widget **TextView** dan kelompok TextField **EditText** untuk label di layout dan kotak isian sesuai rancangan. Dalam bentuk mode TEXT akan seperti dibawah ini.



```
<FrameLayout 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"
    tools:context="com.example.astimen.myapplication.fragment.frg_entry">

    <!-- TODO: Update blank fragment layout -->

    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:padding="10dip">

        <TextView
            android:id="@+id/lblIdRec"
            android:text="Id Record"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />

        <EditText
            android:id="@+id/txtIdRec"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/lblIdRec"
            android:background="@android:drawable/editbox_background" />

        <TextView
            android:id="@+id/lblUserId"
            android:text="User Id"
            android:layout_below="@id/txtIdRec"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
```

```
<EditText
    android:id="@+id/txtUserId"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/lblUserId"
    android:background="@android:drawable/editbox_background"/>

<TextView
    android:id="@+id/lblUserName"
    android:text="User Name"
    android:layout_below="@id/txtUserId"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<EditText
    android:id="@+id/txtUserName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/lblUserName"
    android:background="@android:drawable/editbox_background"/>

<TextView
    android:id="@+id/lblPassword"
    android:layout_below="@id/txtUserName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Password"/>

<EditText
    android:id="@+id/txtPassword"
    android:inputType="textPassword"
    android:layout_below="@id/lblPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background" />

<TextView
    android:id="@+id/lblPassword2"
    android:layout_below="@id/txtPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Password Check"/>

<EditText
    android:id="@+id/txtPassword2"
    android:inputType="textPassword"
    android:layout_below="@id/lblPassword2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background" />

<TextView
    android:id="@+id/lblDepartment"
    android:layout_below="@id/txtPassword2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Departement"/>
```

```
<Spinner
    android:layout_below="@id/lblDepartment"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:prompt="@string/department_prompt"
    android:entries="@array/department_array"

    android:id="@+id/spnDepartment" />

<TextView
    android:id="@+id/lblSex"
    android:layout_below="@id/spnDepartment"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Sex"/>

<RadioGroup
    android:id="@+id/rgSex"
    android:layout_below="@id/lblSex"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:checkedButton="@+id/rbFemale">

    <RadioButton
        android:text="Female"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/rbFemale"
        android:layout_weight="1" />

    <RadioButton
        android:text="Male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/rbMale"
        android:layout_weight="1" />
</RadioGroup>

<TextView
    android:id="@+id/lblRole"
    android:layout_below="@id/rgSex"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Role"/>

<CheckBox
    android:text="Admin"
    android:layout_below="@id/lblRole"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/cbxAdmin" />

<CheckBox
    android:text="User"
    android:layout_below="@id/lblRole"
    android:layout_toRightOf="@id/cbxAdmin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/cbxUser" />
```

```
<Button
    android:id="@+id/btnSave"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/cbxUser"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10dip"
    android:text="SAVE" />

<Button
    android:id="@+id/btnNew"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/btnSave"
    android:layout_below="@id/cbxUser"
    android:text="NEW" />

</RelativeLayout>

</FrameLayout>
```

10. Beberapa properties yang penting dari komponen android adalah sebagai berikut:

- Setiap komponen punya properties id, layout_width, layout_height dan layout_bellow.
- android:id="@+id/txtUserName"** adalah id komponen layout sebagai pengenalan untuk diisikan kepada komponen lain yang memiliki relative posisi dibawahnya seperti properties TextView lblPassword yang berada dibawah txtUserName sehingga dituliskan propertie **android:layout_below="@id/txtUserName"**
- android:layout_height="wrap_content"** artinya tingginya sesuai dengan tinggi default (30 dp) atau sesuai tinggi yang diinginkan misalnya **android:layout_height="35dp"**
- android:layout_width="match_parent"** artinya lebarnya sesuai dengan lebar layout parentnya yaitu RelativeLayout, jadi lebar maksimal

11. Untuk TextView Password dan Password2 tipe text nya adalah password sehingga ditambahkan properties **android:inputType="textPassword"**.

12. Untuk spinner adroid atau combo box terdapat tambahan properties tersendiri yaitu **android:entries="@array/department_array"** untuk mengisikan item yang akan muncul dan **android:prompt="@string/department_prompt"**. Item array ini didefinisikan di file xml **app→res→values→string.xml** dengan menambahkan text xml item sebagai berikut:

```
<resources>
    <string name="app_name">AflowZ Startup</string>

    <string name="navigation_drawer_open">Open navigation drawer</string>
    <string name="navigation_drawer_close">Close navigation drawer</string>

    <string name="action_settings">Settings</string>

    <string name="action_aplikasi">Aplikasi</string>
    <string name="action_bantuan">Bantuan</string>
    <string name="action_keluar">Keluar</string>

    <!-- TODO: Remove or change this placeholder text -->
```

```

<string name="hello_frg_welcome">Selamat Datang di Fragment Welcome</string>
<string name="hello_frg_search">Selamat Datang di Fragment Search</string>

<!-- TODO: Remove or change this placeholder text -->
<string name="hello_blank_fragment">Hello blank fragment</string>

<string name="department_prompt">Choose Department</string>
<string-array name="department_array">
    <item>Sales</item>
    <item>Finance</item>
    <item>Delivery</item>
    <item>Assurance</item>
    <item>Marketing</item>
    <item>Accounting</item>
</string-array>
</resources>

```

13. Untuk komponen RadioButton karena bersifat hanya salah satu RadioButon yang boleh aktif maka harus diletakan di dalam komponen RadioGroup. RadioGroup memiliki properties orientation (**android:orientation="horizontal"**) yang dapate disetting vertical atau horizontal untuk menyusun semua RadioButton yang berada dalam RadioGroup tersebut.
14. Untuk kedua Button NEW dan SAVE karena berada dalam satu row yang sama maka ada properties tambahan bahwa Button NEW berada di sebelah kiri Button SAVE maka ditambahkan properties **android:layout_toLeftOf="@id/btnSAVE"**)
15. Untuk dapat diakses dari menu drawer maka kita perlu menambahkan fragment frg_entry ke MainActivity. Open file MainActivity.java di app→java→com→MainActivity dengan double click. Tambahkan kode **frg_entry.OnFragmentInteractionListener** pada class MainActivity sbb:

```

package com.example.astimen.myapplication;

import ...

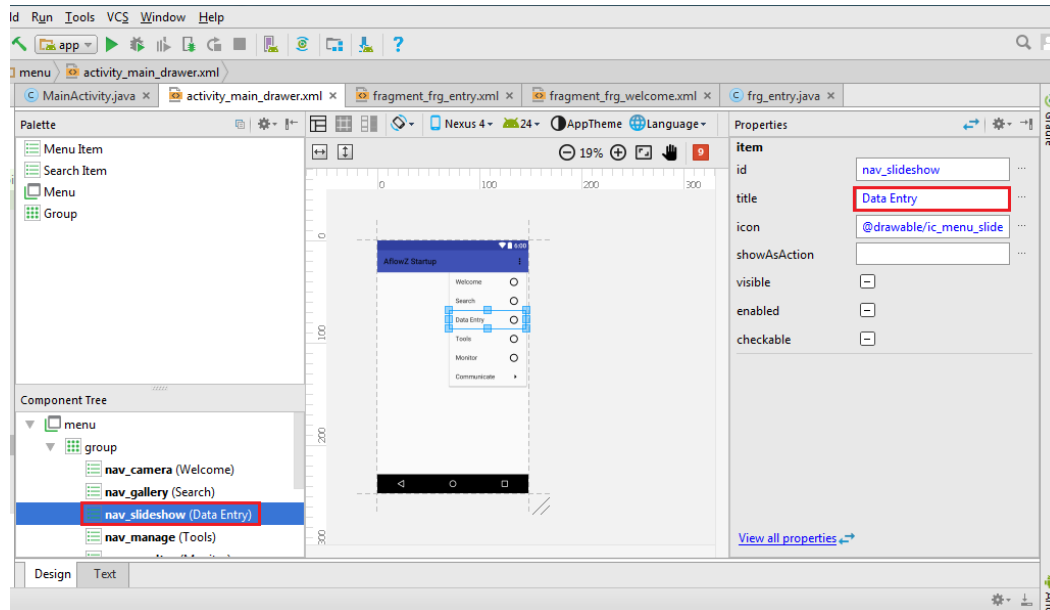
public class MainActivity extends AppCompatActivity
    implements frg_entry.OnFragmentInteractionListener,
    frg_welcome.OnFragmentInteractionListener,
    frg_search.OnFragmentInteractionListener
    ,NavigationView.OnNavigationItemSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
    }
}

```

16. Setiap menambahkan sebuah fragment baru maka harus ditambahkan OnFragmentInteractionListener nya ke dalam public class MainActivity pada posisi implements agar dapat diakses dari menudrawer yang diinisiasi dalam Activity MainActivity. Dalam point 10 terlihat sudah tiga fragment yang didaftarkan frg_welcome, frg_search dan frg_entry.

17. Open file menu activity_main_drawer.xml di app→res→menu→ activity_main_drawer.xml pada mode DESIGN dan ubah title untu nav_slideshow menjadi Data Entry untuk menu akses fragment frg_entry.



18. Tambahkan kode untuk mengakses frg_entry pada class OnNavigationItemSelectedListener pada id menu nav_slideshow yang title nya sudah kita ganti menjadi “Entry Data” pada point 13 seperti yang telah kita pelajari pada modul sebelumnya dengan menambahkan title “Entry Data” pada bar utama dan pesan TOAST “fragment frg_entry ditampilkan”.

```
@Override
public boolean onNavigationItemSelectedListener(MenuItem item) {
    // Handle navigation view item clicks here.
    int id = item.getItemId();
    if (id == R.id.nav_camera) {
        ...
    } else if (id == R.id.nav_gallery) {
        ...
    } else if (id == R.id.nav_slideshow) {

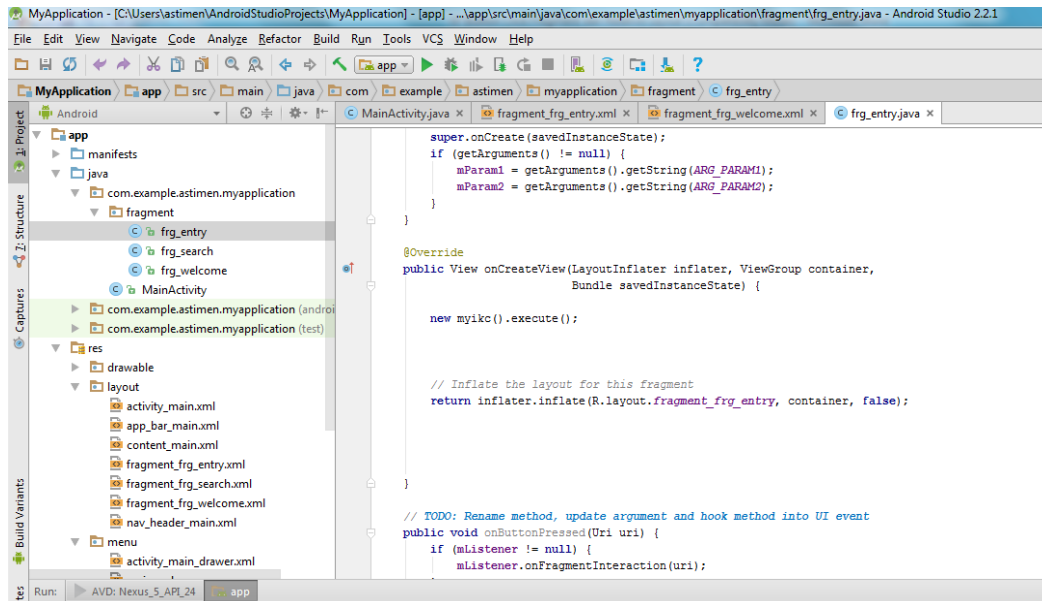
        pesan_toast("Fragment frg_entry ditampilkan ...");

        getSupportFragmentManager()
            .beginTransaction()
            .replace(R.id.frm_utama, frg_entry.newInstance("0","0"))
            .commit();

        getSupportActionBar().setTitle("Entry Data");

    } else if (id == R.id.nav_manage) {
        pesan_toast("Hasil akses menu manage");
    }
    ...
}
```

19. Sekarang kita buka `frg_entry.java` pada `app`→`com`→`fragment`→`frg_entry.java` untuk menambahkan kode pada event click button NEW dan SAVE.



20. Tambahkan kode untuk membuat sebuah class AsyncTask bagi penanganan event klik di fragment (karena berbeda proses dengan activity drawer MainActivity) di public View onCreateView. Tempat layout `fragment_frg_entry` didefinisikan pada return inflater-nya. Beri nama mykic dengan code : `new mykic().execute();`

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {

    new mykic().execute();

    // Inflate the layout for this fragment
    return inflater.inflate(R.layout.fragment_frg_entry, container, false);
}
```

21. Dibawah badan program buat sebuah class pembantu dengan nama mykic AsyncTask seperti dibawah ini.

```
class mykic extends AsyncTask<String, String, String> {
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }
    protected String doInBackground(String... args) {
        return null;
    }
    protected void onPostExecute(String file_url)
    {
    }
}
```

22. Kita akan membuat code handling event click pada method `onPostExecute` sbb :

```
class myikc extends AsyncTask<String, String, String> {

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
    }

    protected String doInBackground(String... args) {

        return null;
    }

    protected void onPostExecute(String file_url) {

        Button btn_new = (Button) getView().findViewById(R.id.btnNew);
        btn_new.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // do something

                Toast.makeText(getActivity(), "Button New Clicked",
                Toast.LENGTH_LONG).show();

            }
        });

        Button btn_save = (Button) getView().findViewById(R.id.btnSave);
        btn_save.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // do something

                Toast.makeText(getActivity(), "Button Save Clicked",
                Toast.LENGTH_LONG).show();

                EditText etIdRec = (EditText) getView().findViewById(R.id.txtIdRec);
                String dtxtIdrec=etIdRec.getText().toString();

                EditText etUserId = (EditText) getView().findViewById(R.id.txtUserId);
                String dtxtUserId=etUserId.getText().toString();

                //String dtxtIdrec = ((EditText)
                getView().findViewById(R.id.txtIdRec)).getText().toString();
                //String dtxtUserId = ((EditText)
                getView().findViewById(R.id.txtUserId)).getText().toString();
                String dtxtUserName = ((EditText)
                getView().findViewById(R.id.txtUserName)).getText().toString();
                String dtxtPassword = ((EditText)
                getView().findViewById(R.id.txtPassword)).getText().toString();
                String dtxtPassword2 = ((EditText)
                getView().findViewById(R.id.txtPassword2)).getText().toString();
                String dspnDepartment = ((Spinner)
                getView().findViewById(R.id.spnDepartment)).getSelectedItem().toString();
                int drbSex = ((RadioGroup)
                getView().findViewById(R.id.rgSex)).getCheckedRadioButtonId();
```

```
        if(dtxtIdrec.trim().length()==0)
        {
            pesandialog("Entrian Id Record tidak boleh kosong ...!");
            etIdRec.requestFocus();
        }

        else if(dtxtUserId.trim().length()==0)
        {
            pesandialog("Entrian User Id tidak boleh kosong ...!");
            etUserId.requestFocus();
        }

        else if(dtxtUserName.trim().length()==0)
        {
            pesandialog("Entrian User Name tidak boleh kosong ...!");
            ((EditText)
            getView().findViewById(R.id.txtUserName)).requestFocus();
        }

        else if(dtxtPassword.trim().length()==0)
        {
            pesandialog("Entrian Password tidak boleh kosong ...!");
            ((EditText)
            getView().findViewById(R.id.txtPassword)).requestFocus();
        }

    }
}

void pesandialog(String pesan){
    DialogInterface.OnClickListener dialogClickListener = new
    DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which){
            switch (which){
                case DialogInterface.BUTTON_POSITIVE:
                    Toast.makeText(getActivity(), "Isi kembali entrian yang kosong
                    ...", Toast.LENGTH_LONG).show();
                    break;
                case DialogInterface.BUTTON_NEGATIVE:
                    break;
            }
        }
    }
};

AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
builder
    .setTitle("AflowZ")
    .setMessage(pesan)
    .setCancelable(false)
    .setPositiveButton("OK", dialogClickListener)
    .show();
}
```

23. Melakukan inisiasi komponen Button yang telah didefinisikan pada layout dilakukan dengan syntax sebagai berikut :

- a. Buat sebuah variable **btn_new** untuk komponen button dengan id **btnNew**. Karena berada di fragment maka perlu ditambahkan **getView()**, sebagai berikut

```
Button btn_new = (Button) getView().findViewById(R.id.btnNew);
```

- b. Definisikan event **OnClick** pada variable button **btn_new** sbb

```
btn_new.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // do something  
  
        Toast.makeText(getActivity(), "Button New Clicked",  
            Toast.LENGTH_LONG).show();  
  
    }  
});
```

- c. Untuk dapat dilakukan testing pada virtual device kita tambahkan code **TOAST** seperti pada point c. Perhatikan definisi **TOAST** pada **MainActivity** dan **Fragment** berbeda. Pada **Fragment** menggunakan **getActivity()** dan pada **MainActivity** menggunakan **getApplicationContext()**.

```
Toast.makeText(getActivity(), pesan, Toast.LENGTH_LONG).show();
```

Pada **MainActivity** :

```
Toast.makeText(getApplicationContext(), pesan, Toast.LENGTH_LONG).show();
```

24. Buat kembali event **onClick** untuk button **SAVE** sebagai berikut :

```
Button btn_save = (Button) getView().findViewById(R.id.btnSave);  
btn_save.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // do something  
  
        // akan menampilkan hasil pemeriksaan apakah isian text kosoong  
        atau tidak dan diinformasikan dalam kotak dialog  
  
    }  
});
```

25. Buat fungsi tersendiri dengan nama **pesandialog** dengan variable **String** **pesan** untuk menampilkan pesan alert bahwa isian tertentu tidak boleh ketika button **SAVE** diclick. Buatlah code fungsi tersebut pada bagian kosong paling bawah code **frg_entry** sbb :

```

void pesandialog(String pesan){
    DialogInterface.OnClickListener dialogClickListener =
        new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which){
            switch (which){
                case DialogInterface.BUTTON_POSITIVE:
                    Toast.makeText(getActivity(), "Isilah ", Toast.LENGTH_LONG).show();
                    break;
                case DialogInterface.BUTTON_NEGATIVE:
                    break;
            }
        }
    };

    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    builder
        .setTitle("AflowZ")
        .setMessage(pesan)
        .setCancelable(false)
        .setPositiveButton("OK", dialogClickListener)
        .show();
}

```

26. Untuk mendapatkan nilai sebuah EditText misalnya id txtIdRec buatlah sebuah variable etIdRec dan variable String dtxtIdRec dengan method getText() sebagai berikut :

```

EditText etIdRec = (EditText) getView().findViewById(R.id.txtIdRec);
String dtxtIdrec=etIdRec.getText().toString();

```

27. Periksa apakah panjang text itu nol yang berarti kosong atau spasi. Jika demikian maka tampilkan pesan melalui pemanggilan fungsi pesandialog. Dan kembalikan cursor android ke isian text tersebut melalui metode requestFocus(), sebagai berikut :

```

if(dtxtIdrec.trim().length()==0)
{
    pesandialog("Entrian Id Record tidak boleh kosong ...!");
    etIdRec.requestFocus();
}

```

28. Lakukan langkah yang sama untuk EditText selanjutnya dengan menambahkan else if.

29. Untuk melakukan pengujian apakah password satu sama atau equals dengan password2 gunakan properties .equals() dengan sintax dibawah ini. Untuk menyatakan jika tidak equals maka gunakan tanda seru (!):

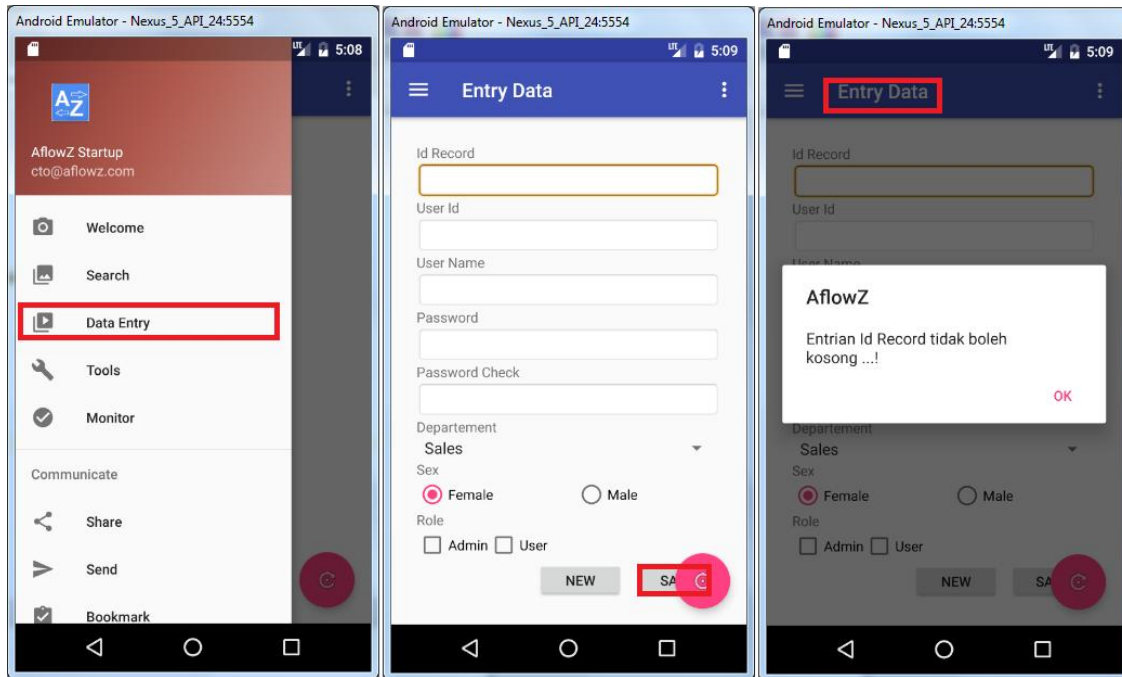
```

if(!dtxtPassword.equals(dtxtPassword2))
{
    pesandialog("Password dan Pasword2 tidak sama ...!");
    ((EditText) getView().findViewById(R.id.txtPassword2)).requestFocus();
}

```

30. Hal yang membedakan coding di activity dan fragment adalah pada fragment tidak menggunakan (this) tapi menggunakan (getView().getContext()).

31. Klik Button Run App akan kita coba di virtual device yang telah didefinisikan.



32. Code Lengkap frg_entry.java adalah sbb :

```
package com.example.astimen.myapplication.fragment;
```

```
import android.content.Context;
import android.content.DialogInterface;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v7.app.AlertDialog;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.Toast;
```

```
import com.example.astimen.myapplication.R;
```

```
/**
```

```
* A simple {@link Fragment} subclass.
* Activities that contain this fragment must implement the
* {@link frg_entry.OnFragmentInteractionListener} interface
* to handle interaction events.
* Use the {@link frg_entry#newInstance} factory method to
* create an instance of this fragment.
*/
public class frg_entry extends Fragment {
    // TODO: Rename parameter arguments, choose names that match
    // the fragment initialization parameters, e.g. ARG_ITEM_NUMBER
    private static final String ARG_PARAM1 = "param1";
    private static final String ARG_PARAM2 = "param2";

    // TODO: Rename and change types of parameters
    private String mParam1;
    private String mParam2;

    private OnFragmentInteractionListener mListener;

    public frg_entry() {
        // Required empty public constructor
    }

    /**
     * Use this factory method to create a new instance of
     * this fragment using the provided parameters.
     *
     * @param param1 Parameter 1.
     * @param param2 Parameter 2.
     * @return A new instance of fragment frg_entry.
     */
    // TODO: Rename and change types and number of parameters
    public static frg_entry newInstance(String param1, String param2) {
        frg_entry fragment = new frg_entry();
        Bundle args = new Bundle();
        args.putString(ARG_PARAM1, param1);
        args.putString(ARG_PARAM2, param2);
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        if (getArguments() != null) {
            mParam1 = getArguments().getString(ARG_PARAM1);
            mParam2 = getArguments().getString(ARG_PARAM2);
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {

        new myikc().execute();
    }
}
```



```
// Inflate the layout for this fragment
return inflater.inflate(R.layout.fragment_frg_entry, container,
false);

}

// TODO: Rename method, update argument and hook method into UI event
public void onPressed(Uri uri) {
    if (mListener != null) {
        mListener.onFragmentInteraction(uri);
    }
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    if (context instanceof OnFragmentInteractionListener) {
        mListener = (OnFragmentInteractionListener) context;
    } else {
        throw new RuntimeException(context.toString()
            + " must implement OnFragmentInteractionListener");
    }
}

@Override
public void onDetach() {
    super.onDetach();
    mListener = null;
}

/**
 * This interface must be implemented by activities that contain this
 * fragment to allow an interaction in this fragment to be
communicated
 * to the activity and potentially other fragments contained in that
 * activity.
 * <p>
 * See the Android Training lesson <a href=
"http://developer.android.com/training/basics/fragments/communicating.html
"
 * >Communicating with Other Fragments</a> for more information.
 */
public interface OnFragmentInteractionListener {
    // TODO: Update argument type and name
    void onFragmentInteraction(Uri uri);
}

class myikc extends AsyncTask<String, String, String> {
```

```
@Override
protected void onPreExecute() {
    super.onPreExecute();
}

protected String doInBackground(String... args) {

    return null;
}

protected void onPostExecute(String file_url) {

    Button btn_new = (Button) getView().findViewById(R.id.btnNew);
    btn_new.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // do something

            Toast.makeText(getActivity(), "Button New Clicked",
Toast.LENGTH_LONG).show();

        }
    });

    Button btn_save = (Button)
getView().findViewById(R.id.btnSave);
    btn_save.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // do something

            Toast.makeText(getActivity(), "Button Save Clicked",
Toast.LENGTH_LONG).show();

            EditText etIdRec = (EditText)
getView().findViewById(R.id.txtIdRec);
            String dtxtIdrec=etIdRec.getText().toString();

            EditText etUserId = (EditText)
getView().findViewById(R.id.txtUserId);
            String dtxtUserId=etUserId.getText().toString();

            //String dtxtIdrec = ((EditText)
getView().findViewById(R.id.txtIdRec)).getText().toString();
```

```

        //String dtxtUserId = ((EditText)
getView().findViewById(R.id.txtUserId)).getText().toString();
        String dtxtUserName = ((EditText)
getView().findViewById(R.id.txtUserName)).getText().toString();
        String dtxtPassword = ((EditText)
getView().findViewById(R.id.txtPassword)).getText().toString();
        String dtxtPassword2 = ((EditText)
getView().findViewById(R.id.txtPassword2)).getText().toString();
        String dspnDepartment = ((Spinner)
getView().findViewById(R.id.spnDepartment)).getSelectedItem().toString();
        int drbSex = ((RadioGroup)
getView().findViewById(R.id.rgSex)).getCheckedRadioButtonId();

        if(dtxtIdrec.trim().length()==0)
        {
            pesandialog("Entrian Id Record tidak boleh kosong
...!");
            etIdRec.requestFocus();
        }
        else if(dtxtUserId.trim().length()==0)
        {
            pesandialog("Entrian User Id tidak boleh kosong
...!");
            etUserId.requestFocus();
        }
        else if(dtxtUserName.trim().length()==0)
        {
            pesandialog("Entrian User Name tidak boleh kosong
...!");
            ((EditText)
getView().findViewById(R.id.txtUserName)).requestFocus();
        }
        else if(dtxtPassword.trim().length()==0)
        {
            pesandialog("Entrian Password tidak boleh kosong
...!");
            ((EditText)
getView().findViewById(R.id.txtPassword)).requestFocus();
        }
    }
});
}
}

```

```
void pesandialog(String pesan){
    DialogInterface.OnClickListener dialogClickListener = new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which){
            switch (which){
                case DialogInterface.BUTTON_POSITIVE:
                    Toast.makeText(getActivity(), "Isi kembali entrian
yang kosong ...", Toast.LENGTH_LONG).show();
                    break;
                case DialogInterface.BUTTON_NEGATIVE:
                    break;
            }
        }
    };

    AlertDialog.Builder builder = new
AlertDialog.Builder(getActivity().getContext());
    builder
        .setTitle("AflowZ")
        .setMessage(pesan)
        .setCancelable(false)
        .setPositiveButton("OK", dialogClickListener)
        .show();
}
```