LAPORAN PEMROGRAMAN MOBILE

JOB SHEET MINGGU 5

ANDROID INTENT

Oleh:

DION MAULANA WILLIAWARMA NIM. 1731710111



PROGRAM STUDI MANAJEMEN INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

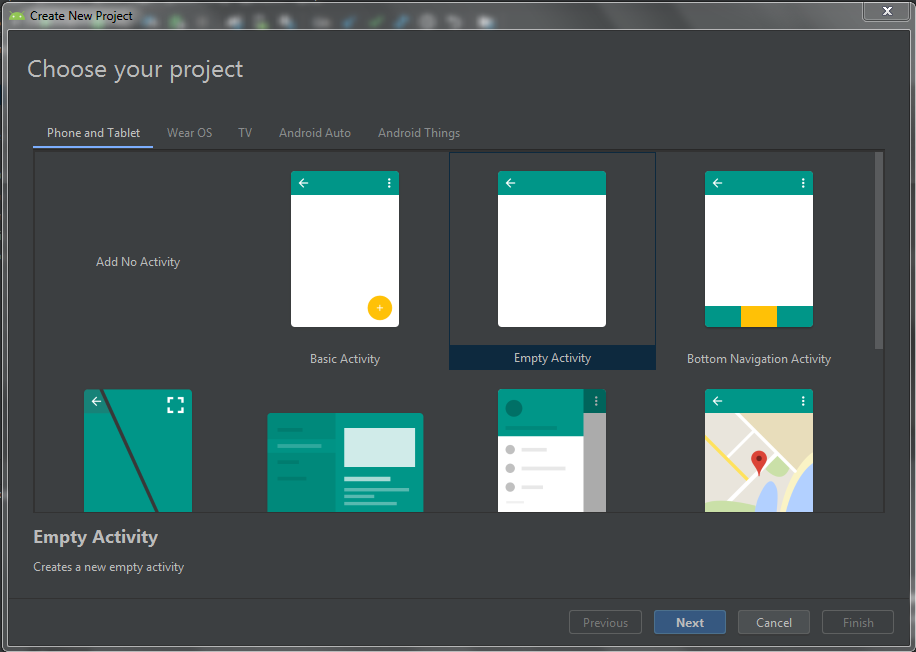
POLITEKNIK NEGERI MALANG

MARET 2019

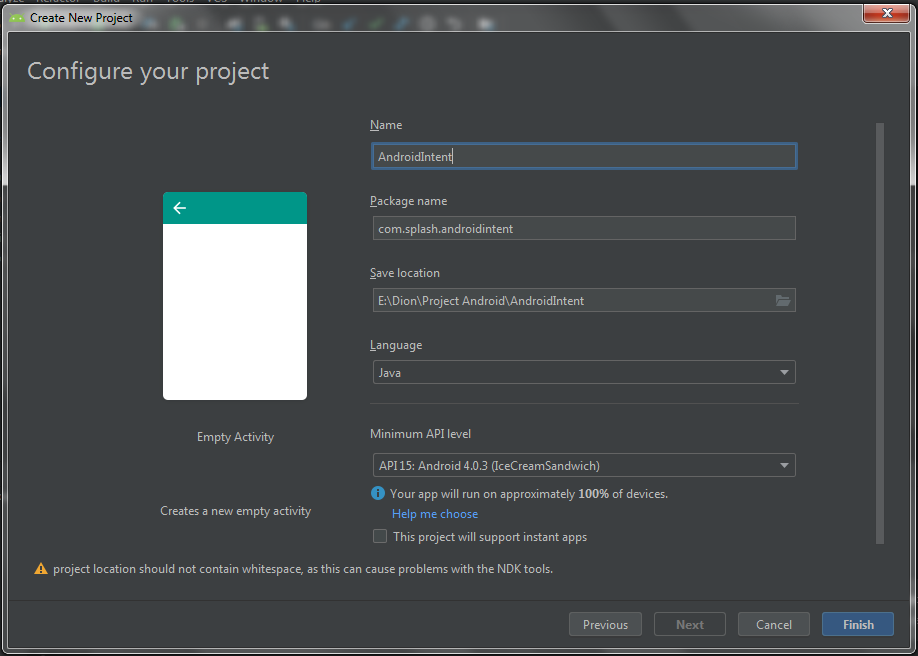
**Praktikum 1 - Mengirim Data antar Activity (Explicit Intent)**

**Membuat Activity Pertama**

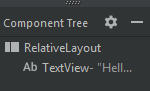
1. Buka aplikasi Android Studio dan buat sebuah project baru. Pilih Empty Activity untuk template project



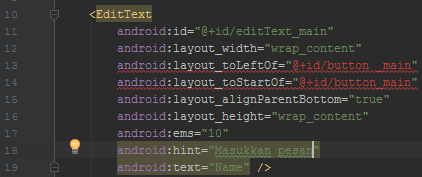
1. Nama aplikasi bisa diisikan sesuai dengan keinginan, Pilihlah minimum sdk sesuai kebutuhan, misalnya pada praktikum ini kita gunakan API 15: Android 4.0.3



1. Buka res/layout/activity\_main.xml. Pada Layout Editor, klik tab Design di bagian bawah layar. Ubah layout yang digunakan menjadi **RelativeLayout**

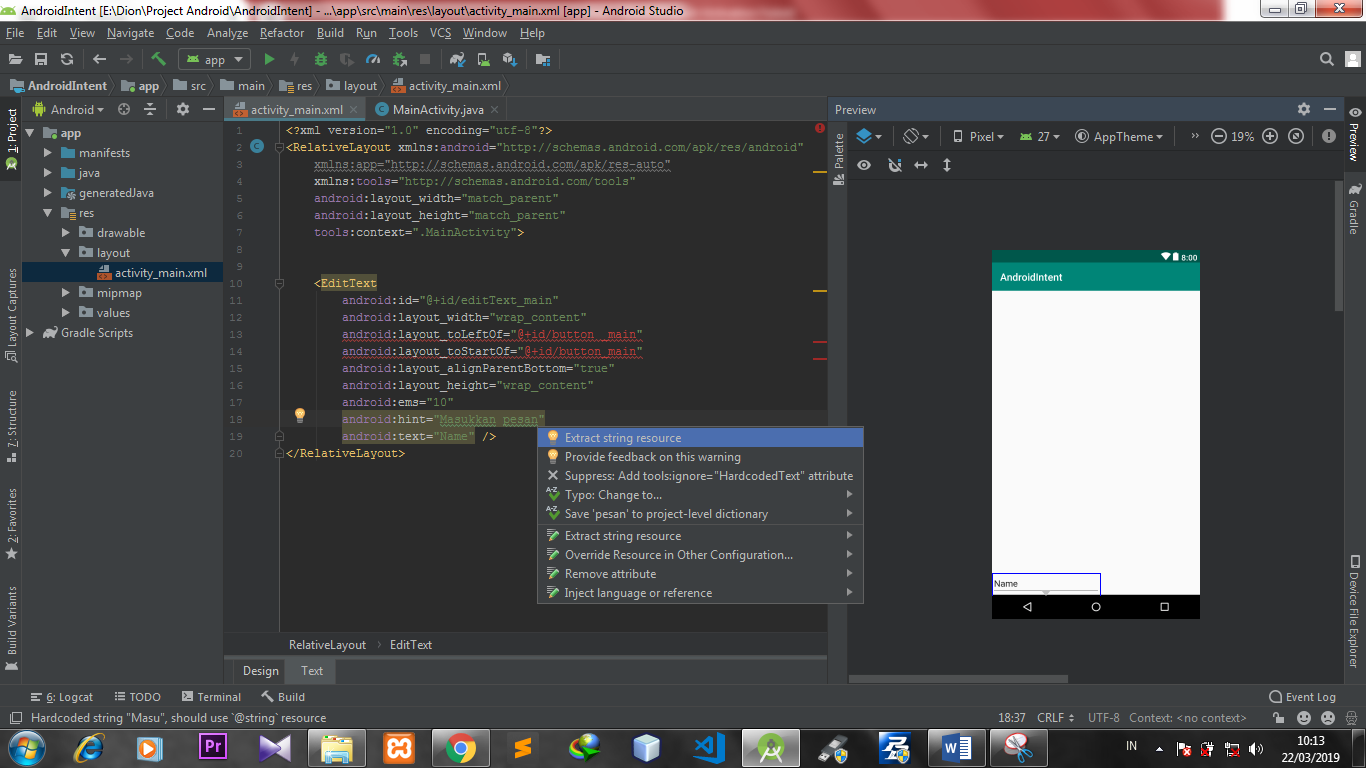


1. Hapus TextView yang berbunyi *"Hello World"*
2. Tambahkan sebuah Plain Text. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut Plain Text sebagai berikut:

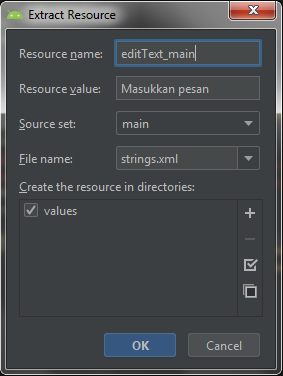


Jika terdapat kesalahan, abaikan sementara kesalahan yang muncul tersebut. Masalah ini akan kita atasi di langkah selanjutnya.

1. Letakkan kursor pada kata "Ketikkan pesan**". Tekan Alt+Enter (Option+Enter di Mac),** dan pilih **Extract string resources**.

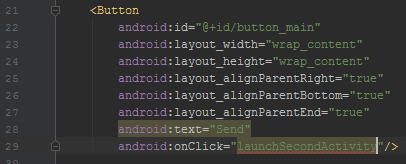


1. Ubah nama Sumber Daya menjadi editText\_main dan klik OK.

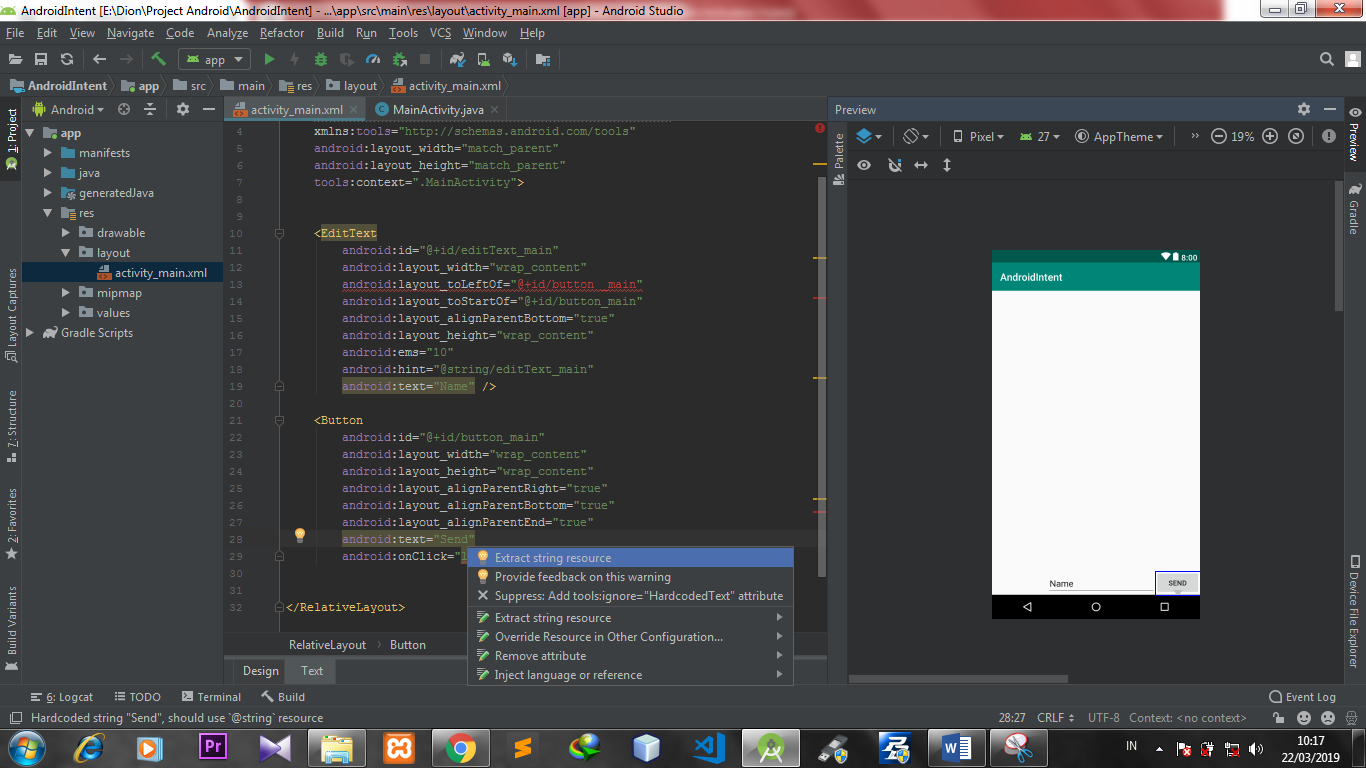


Proses ini membuat sumber daya string dalam file values/res/string.xml, dan string dalam kode Anda digantikan dengan referensi ke sumber daya string tersebut.

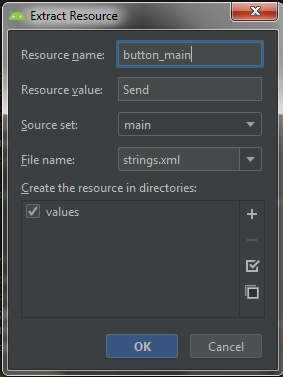
1. Tambahkan sebuah Button ke layout dalam posisi apa pun. Beralih ke XML Editor (klik tab Text) dan modifikasi atribut Button sebagai berikut:



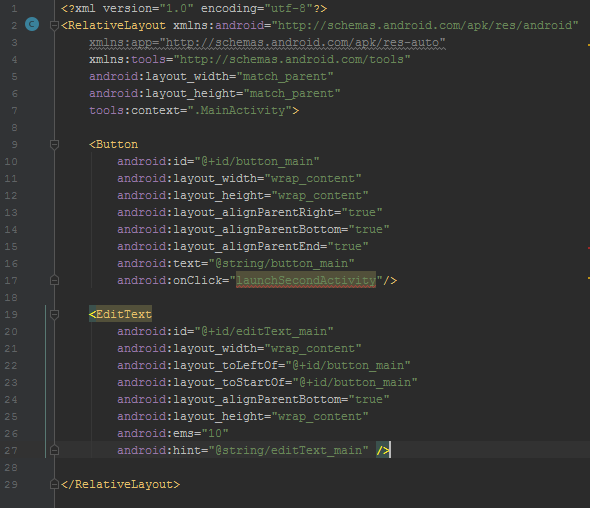
1. Letakkan kursor pada kata "Send". Tekan **Alt+Enter (Option+Enter di Mac**), dan pilih **Extract string resources**.



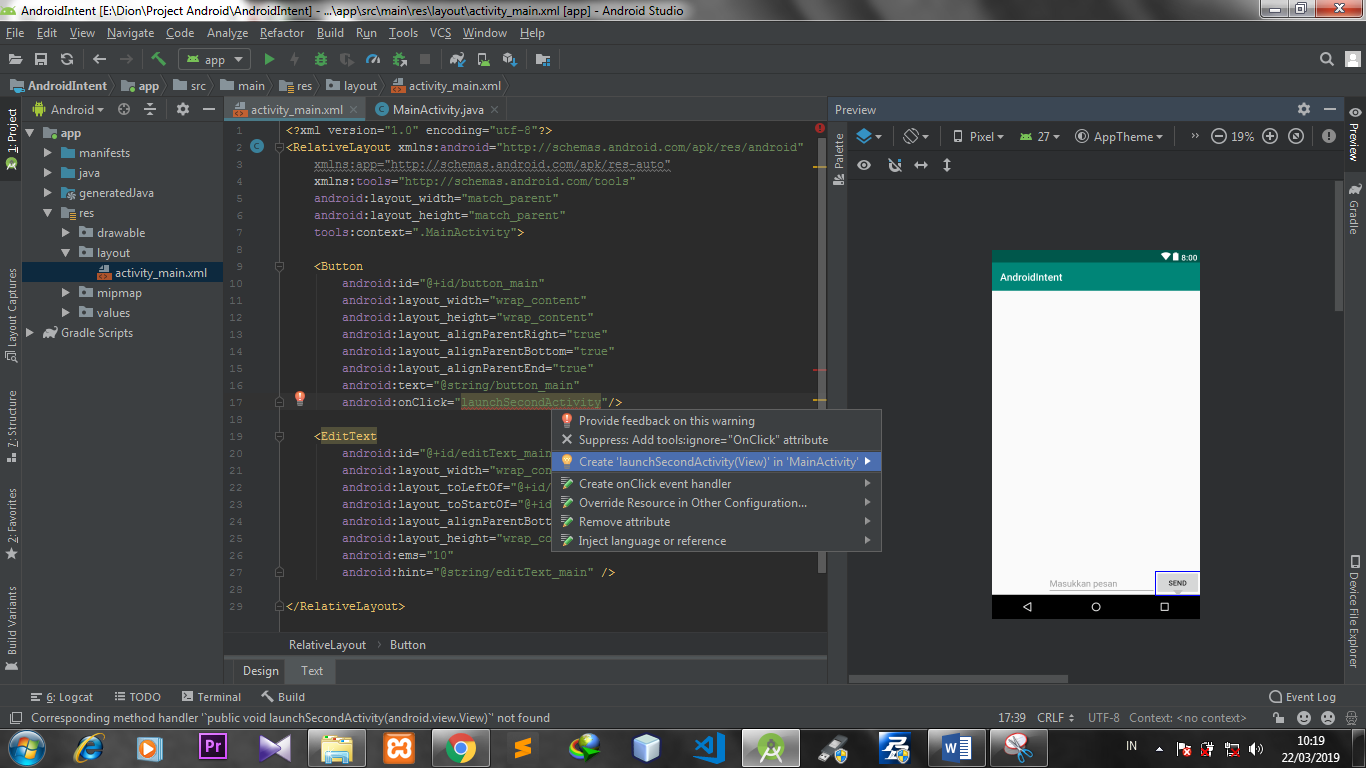
1. Ubah nama Sumber Daya menjadi button\_main dan klik OK.



1. Kode Program secara keseluruhan untuk activity\_main.xml seperti gambar di bawah ini.



1. Dalam Editor XML, letakkan kursor pada kata "launchSecondActivity". Tekan **Alt+Enter(Option+Enter di Mac)** dan pilih **Create 'launchSecondActivity(View)' in 'MainActivity**.



1. Dalam launchSecondActivity, tambahkan pernyataan log yang bernama *"Button diklik!"*

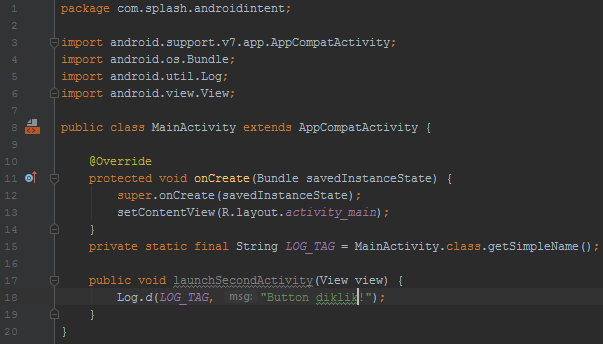


Log dan LOG\_TAG akan terlihat berwarna merah. Definisi untuk variabel tersebut akan ditambahkan dalam langkah selanjutnya.

1. Tempatkan kursor pada kata "Log" dan tekan **Alt+Enter (Option+Enter di Mac).** Android Studio menambahkan pernyataan impor untuk **android.util.Log**.

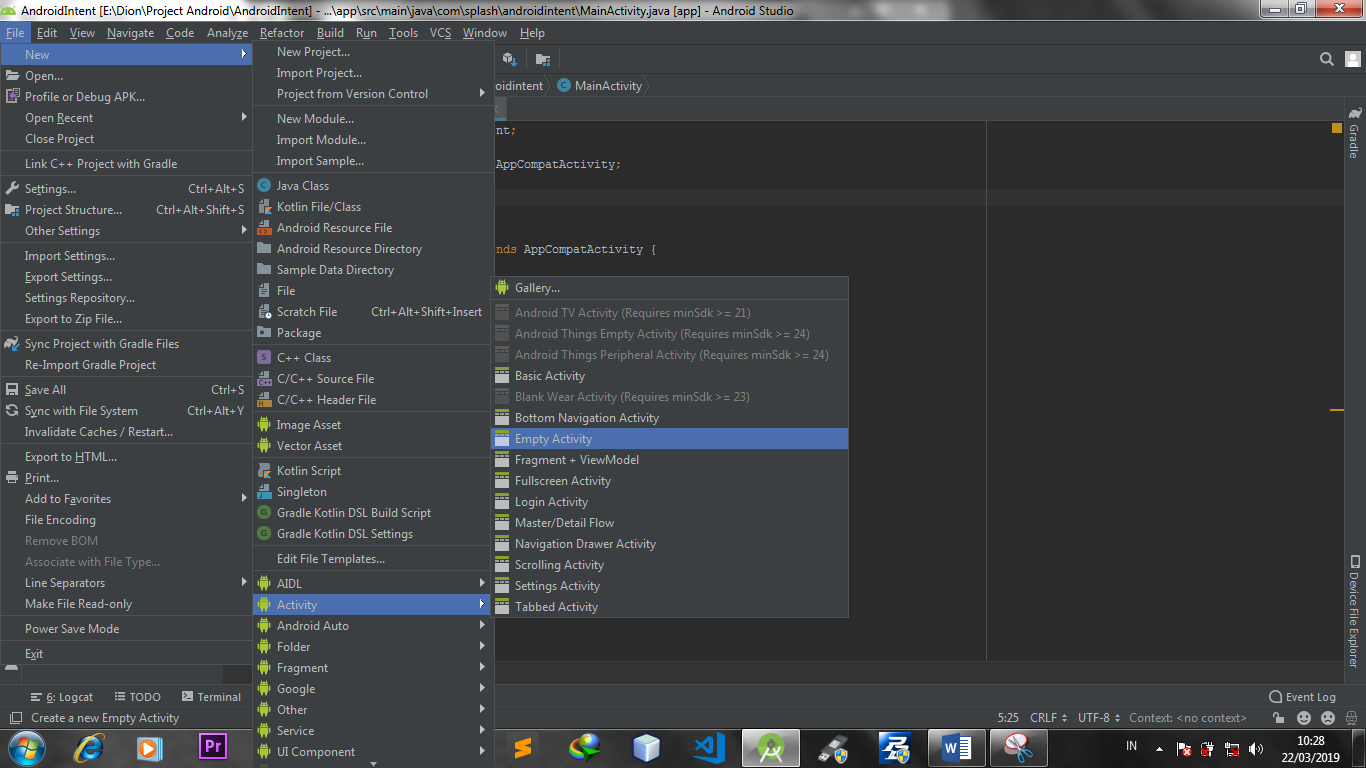
Pada bagian atas kelas, tambahkan konstanta untuk variabel LOG\_TAG:

*private static final String LOG\_TAG = MainActivity.class.getSimpleName();*

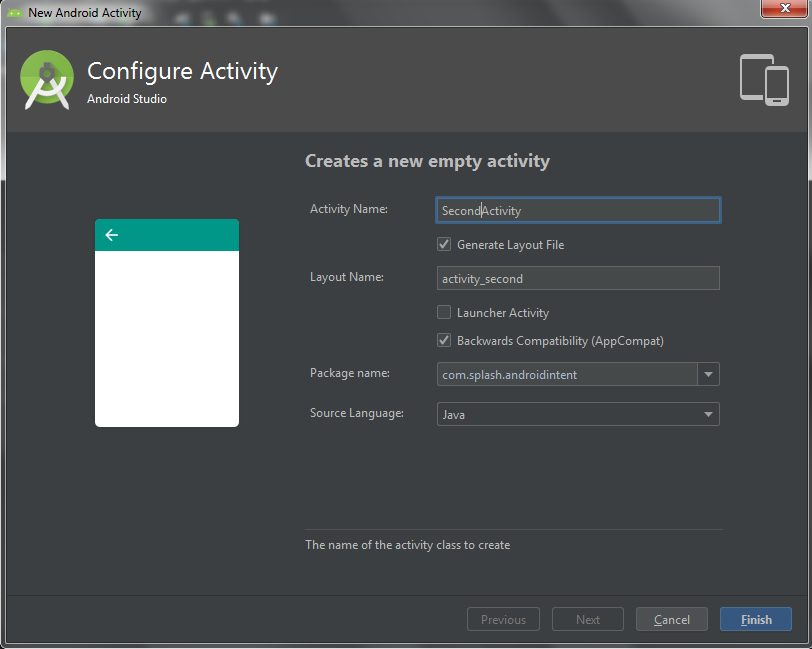


**Membuat Activity Kedua**

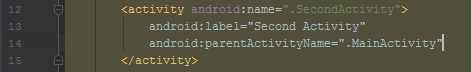
1. Klik folder app untuk proyek Anda dan pilih **File > New > Activity > Empty Activity.**



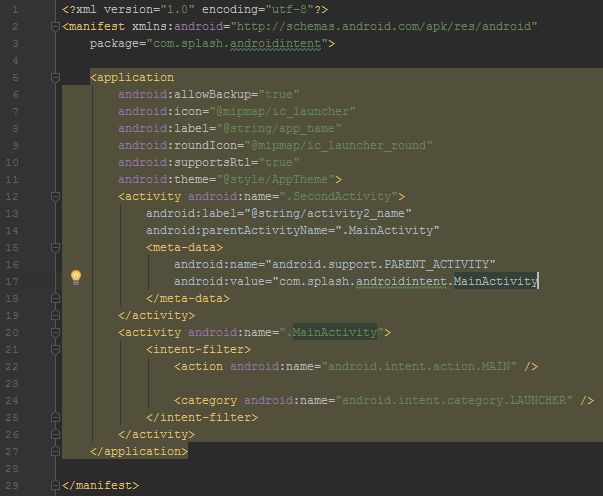
1. Namakan aktivitas baru "SecondActivity." Pastikan Generate Layout File dicentang, dan nama layout akan diisi dengan activity\_second.



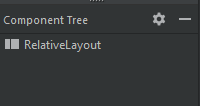
1. Klik Finish. Android Studio menambahkan layout aktivitas baru (activity\_second) dan file Java baru (SecondActivity) ke proyek Anda untuk aktivitas baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan aktivitas baru.
2. Buka manifests/AndroidManifest.xml.
3. Temukan elemen <activity> yang dibuat Android Studio untuk aktivitas kedua.
4. Tambahkan atribut ini ke elemen <activity>:



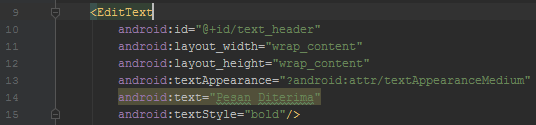
1. Tambahkan elemen <meta-data> element inside the <activity> di dalam elemen untuk aktivitas kedua. Gunakan atribut berikut ini:



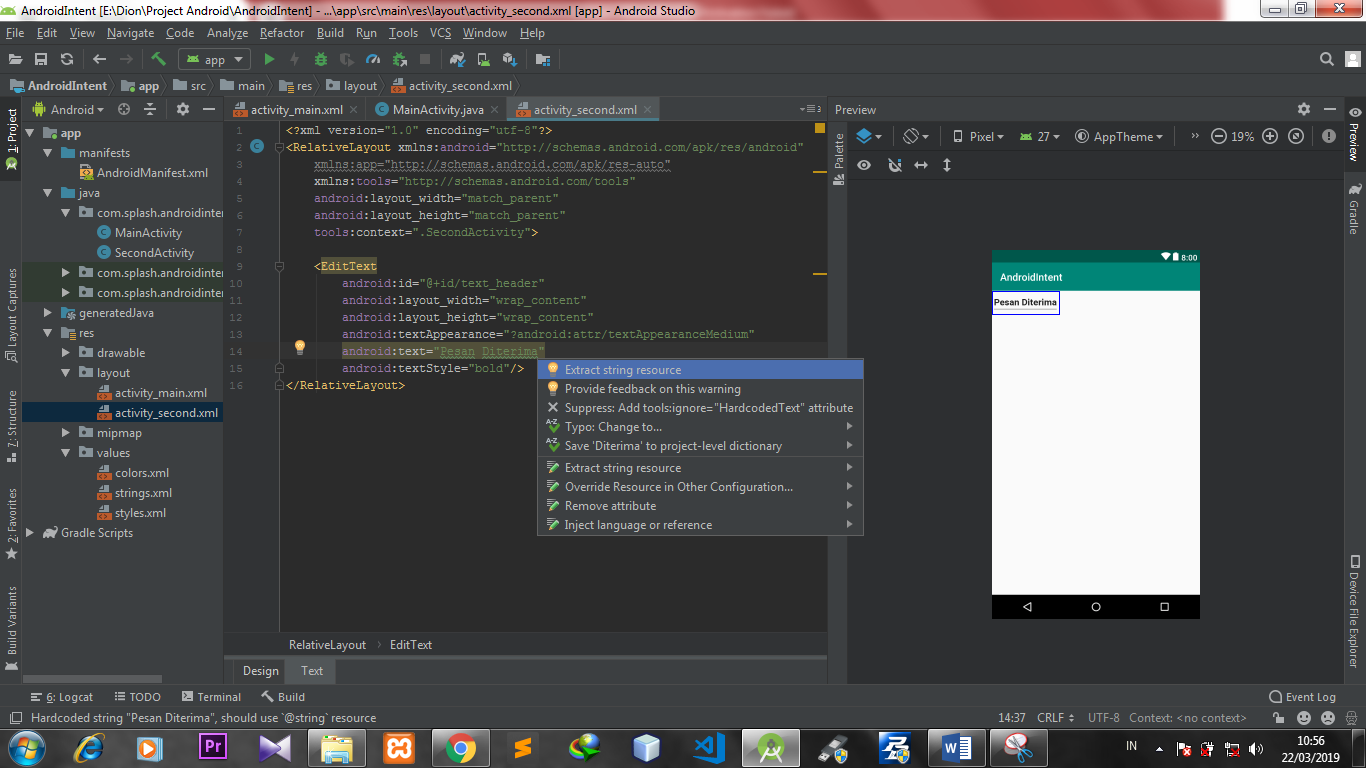
1. Buka res/layout/activity\_second.xml. Ubah layout yang digunakan menjadi **RelativeLayout**

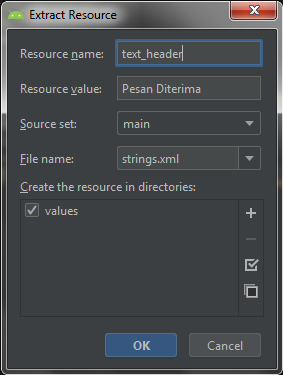


1. Tambahkan TextView ("Plain Textview" dalam Layout Editor). Berikan atribut berikut pada TextView:

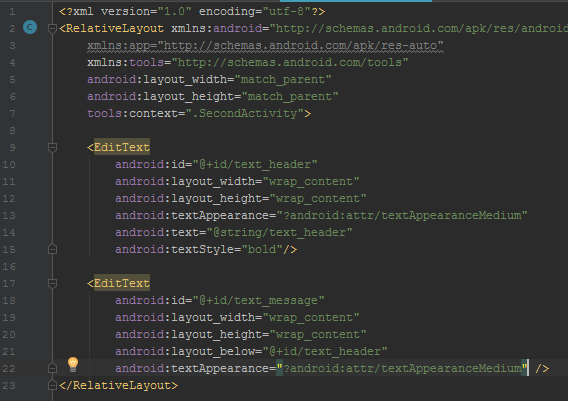


1. Ekstrak string "Pesan Diterima" ke sumber daya bernama text\_header.



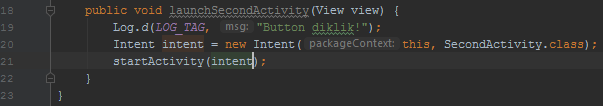


1. Tambahkan TextView yang kedua (Plain Textview dalam Layout Editor). Berikan atribut berikut pada TextView:



**Menambahkan Intent**

1. Buat intent baru dalam metode launchSecondActivity()
2. Panggil metode startActivity() dengan intent baru sebagai argumennya.



1. Tambahkan konstanta publik di bagian atas kelas untuk mendefinisikan kunci untuk ekstra intent:

*public static final String EXTRA\_MESSAGE = "com.splash.androidintent.extra.MESSAGE";*

1. Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText. Mengimpor kelas EditText.

*private EditText mMessageEditText;*

1. Dalam metode **onCreate(),** gunakan findViewByID untuk mendapatkan referensi tentang contoh EditText dan menetapkannya ke variabel privat tersebut:

*mMessageEditText = (EditText) findViewById(R.id.editText\_main);*

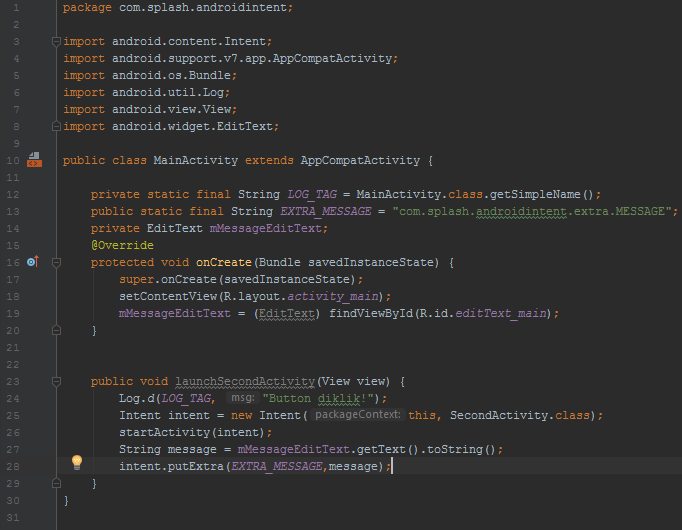
1. Dalam metode launchSecondActivity(), di bawah intent baru, dapatkan teks dari EditText sebagai string:

*String message = mMessageEditText.getText().toString();*

1. Tambahkan string tersebut ke intent sebagai sebuah ekstra dengan konstanta EXTRA\_MESSAGE sebagai kunci dan string-nya sebagai nilai:

*intent.putExtra(EXTRA\_MESSAGE, message);*

1. Kode Program secara keseluruhan untuk MainActivity.java seperti gambar di bawah ini.



1. Buka file SecondActivity.java, Dalam metode onCreate(), dapatkan intent yang mengaktifkan aktivitas berikut ini:

*Intent intent = getIntent();*

1. Dapatkan string berisi pesan dari ekstra intent menggunakan variabel statis MainActivity.EXTRA\_MESSAGE sebagai kuncinya

*String message = intent.getStringExtra(MainActivity.EXTRA\_MESSAGE);*

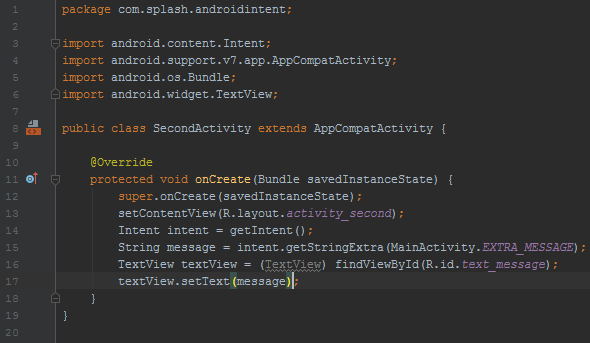
1. Gunakan findViewByID untuk mendapatkan referensi ke TextView untuk pesan dari layout (Anda mungkin perlu mengimpor kelas TextView):

*TextView textView = (TextView) findViewById(R.id.text\_message);*

1. Atur teks TextView tersebut ke string dari ekstra intent:

*textView.setText(message);*

1. Kode Program secara keseluruhan untuk SecondActivity.java seperti gambar di bawah ini.



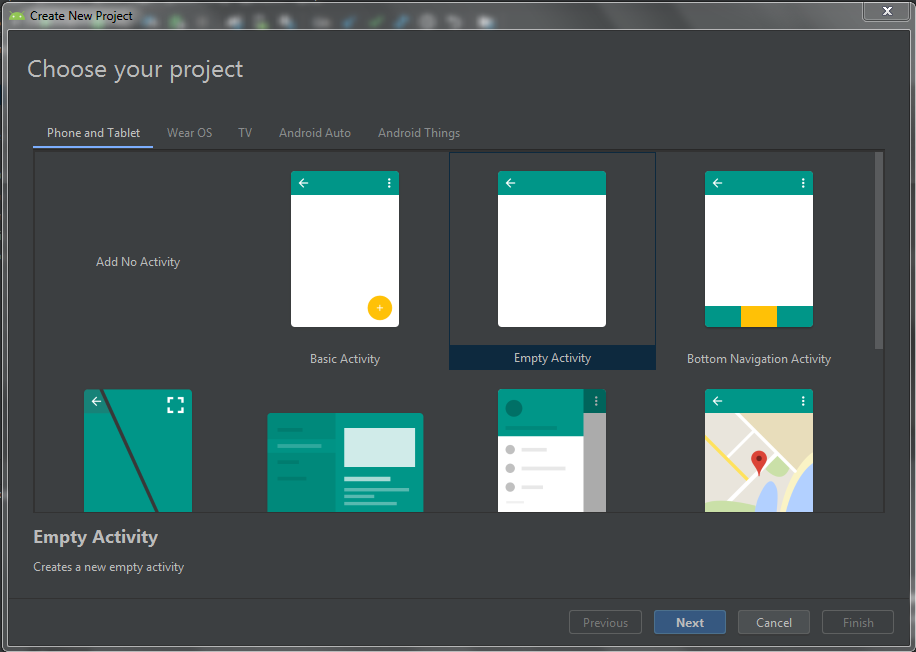
***Kesimpulan :***

Ketika di MainActivity kita memasukkan sebuah pesan pada atribut EditText dan dikirim melalui Button Send, maka dari Button tersebut mempunyai proses ketika di klik yaitu melalui metode launchSecondActivity. Selanjutnya data dari EditText akan dikirim ke SecondActivity dan ketika di klik langsung menuju ke SecondActivity lebih tepatnya di activity\_second.xml, dan pada metode onCreate() pada SecondActivity akan menerima intent dan data dari MainActivity. Untuk mengakses data yang dikirim yaitu dengan perintah **getStringExtra(MainActivity.EXTRA\_MESSAGE);** dan selanjutnya meng-set TextView yang ada dengan nilai sama dengan nilai yang telah dikirim dari MainActivity.

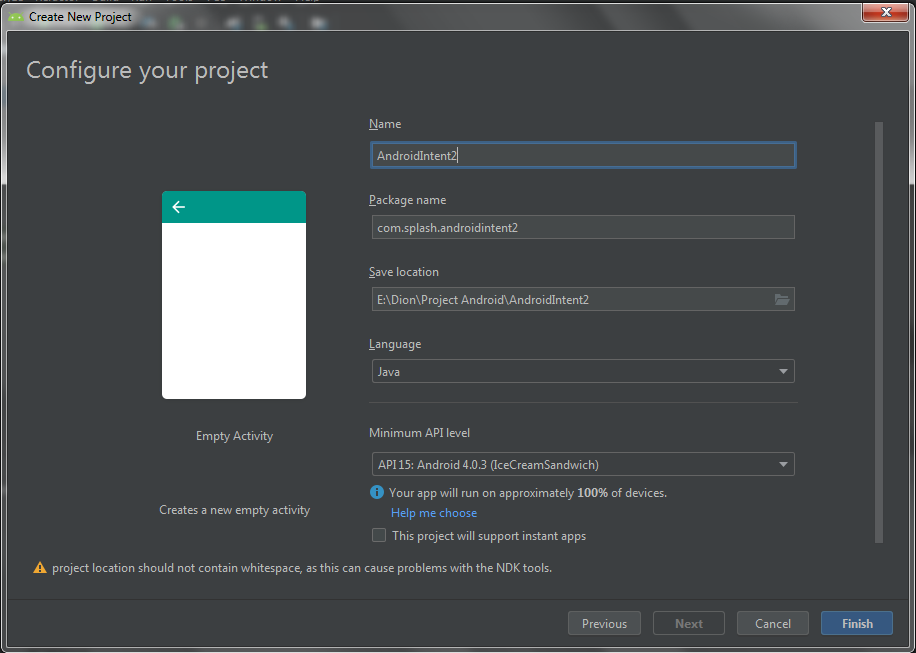
**Praktikum 2 - Memanfaatkan Defined Action (Implicit Intent)**

**Membuat Layout**

1. Buka aplikasi Android Studio dan buat sebuah project baru. Pilih Empty Activity untuk template project

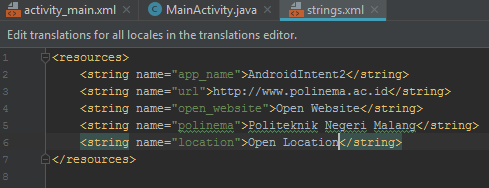


1. Nama aplikasi bisa diisikan sesuai dengan keinginan, Pilihlah minimum sdk sesuai kebutuhan, misalnya pada praktikum ini kita gunakan API 15: Android 4.0.3

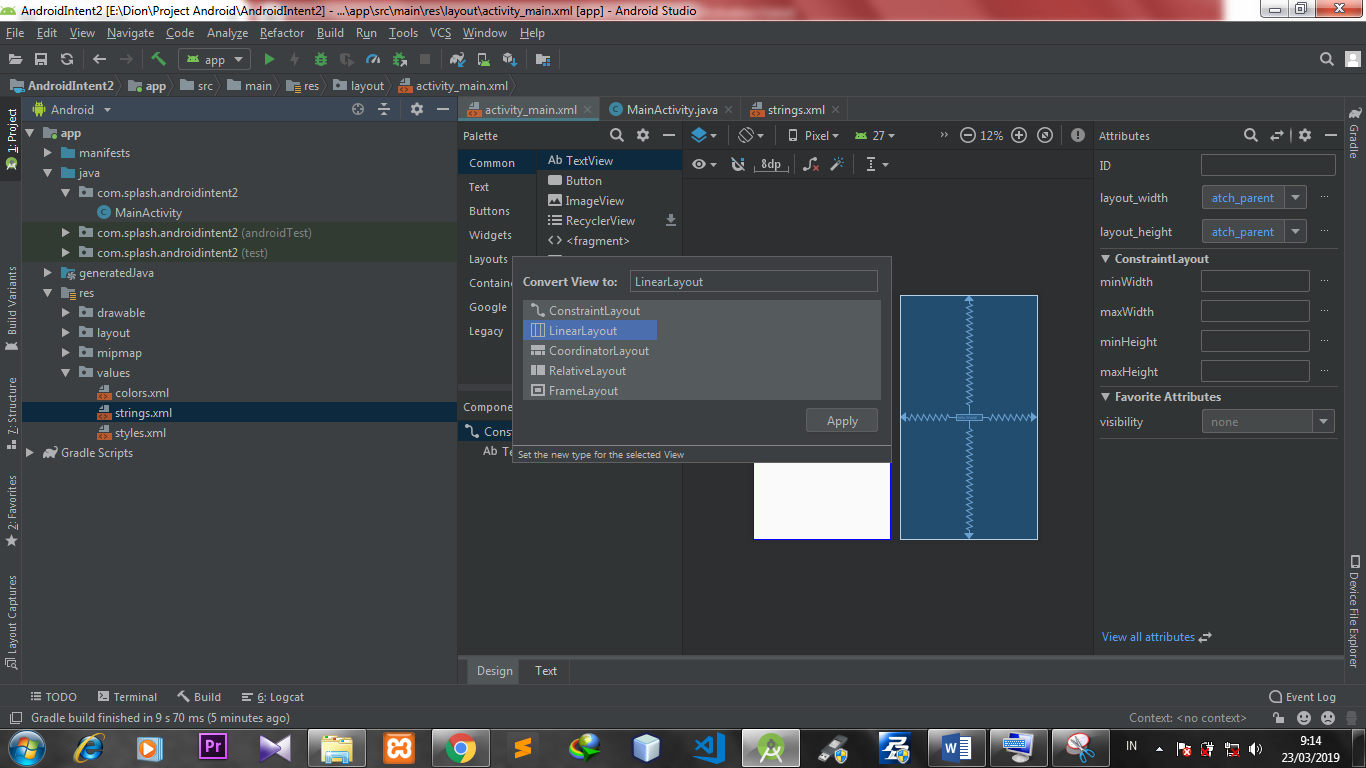


1. Buka file res/values/strings.xml dan tambahkan sumber daya string berikut ini:

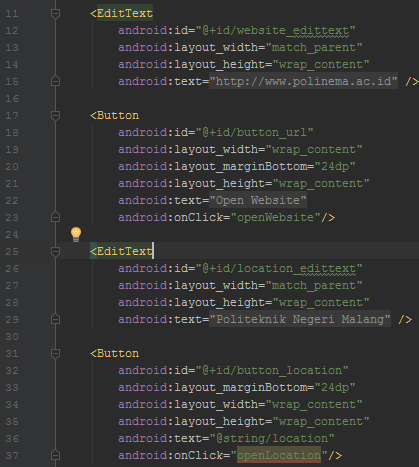
http://www.polinema.ac.id Open Website Politeknik Negeri Malang Open Location



1. Ubah RelativeLayout default ke LinearLayout. Tambahkan atribut android:orientation dan berikan nilai "vertical."
2. Buka res/layout/activity\_main.xml. Pada Layout Editor, klik tab Design di bagian bawah layar. Ubah layout yang digunakan menjadi **LinearLayout**



1. Tambahkan 2 Plain Text (Edit Text) dan 2 Button dengan tampilan seperti gambar berikut.



**Mengimplementasikan Buka Situs Website**

Kita akan mengimplementasikan metode handler on-click untuk tombol pertama dalam layout ("Open Website.") Tindakan ini menggunakan intent implisit untuk mengirimkan URI yang diberikan ke aktivitas yang dapat menangani intent implisit tersebut (seperti browser web).

1. Buka file MainActivity.java., Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText untuk URI situs web :

*private EditText mWebsiteEditText;*

1. Dalam metode onCreate(), gunakan findViewByID() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut :

*mWebsiteEditText = (EditText) findViewById(R.id.website\_edittext);*

1. Buat metode baru bernama openWebsite()

*public void openWebsite(View view) { }*

1. Dapatkan nilai string EditText menggunakan kode berikut.

*String url = mWebsiteEditText.getText().toString();*

1. Enkode dan parse string ke dalam objek Uri

*Uri webpage = Uri.parse(url);*

1. Buat intent baru dengan Intent.ACTION\_VIEW sebagai tindakan dan URI sebagai data

*Intent intent = new Intent(Intent.ACTION\_VIEW, webpage);*

1. Gunakan resolveActivity() dan pengelola paket Android untuk menemukan aktivitas yang dapat menangani intent implisit. Periksa untuk memastikan permintaan berhasil diatasi.

*if (intent.resolveActivity(getPackageManager()) != null) {*

*}*

1. Dalam pernyataan if, panggil startActivity() untuk mengirimkan intent.

*startActivity(intent);*

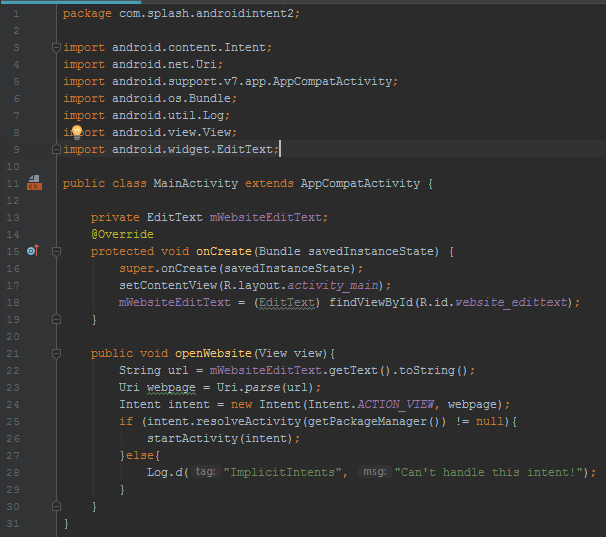
1. Tambahkan blok else untuk mencetak pesan log jika intent tidak bisa diatasi.

*} else {*

*Log.d("ImplicitIntents", "Can't handle this!");*

*}*

Kode Program secara keseluruhan untuk MainActivity.java seperti gambar di bawah ini.



**Kesimpulan**

Untuk melakukan penelusuran website diperlukan suatu inputan yang didapat dari EditText yang selanjutnya di klik button yang akan memproses method openWebsite() yang di dalam nya ada proses pengaksesan isi dari EditText nya, dan pembuatan objek uri yang gunanya untuk melakukan browsing. Selanjutnya dilakukan inisialisasi Objek Intent dengan parameter action ACTION\_VIEW dan objek Uri. Lalu ada proses pengecekan jika activity paket manager nya tidak bernilai null maka akan memulai activity browser yang akan dipilih, jika bernilai null maka keluar pesan error

**Mengimplementasikan Buka Lokasi**

1. Buka file MainActivity.java., Tambahkan variabel privat di bagian atas kelas untuk menampung objek EditText bagi URI lokasi.

*private EditText mLocationEditText;*

1. Dalam metode **onCreate()**, gunakan findViewByID() untuk mendapatkan referensi tentang instance EditText dan menetapkannya ke variabel privat tersebut

*mLocationEditText = (EditText) findViewById(R.id.location\_edittext);*

1. Buat metode baru bernama openLocation untuk digunakan sebagai metode onClick untuk tombol Open Location.
2. Dapatkan nilai string EditText mLocationEditText.

*String loc = mLocationEditText.getText().toString();*

1. Parse string itu ke dalam objek Uri dengan kueri penelusuran geo:

*Uri addressUri = Uri.parse("geo:0,0?q=" + loc);*

1. Buat intent baru dengan Intent.ACTION\_VIEW sebagai tindakan dan loc sebagai datanya.

*Intent intent = new Intent(Intent.ACTION\_VIEW, addressUri);*

1. Pecahkan intent dan periksa untuk memastikan intent berhasil diatasi. Jika demikian, startActivity(), jika tidak catat log pesan kesalahan.

*if (intent.resolveActivity(getPackageManager()) != null) {*

*startActivity(intent);*

*} else {*

*Log.d("ImplicitIntents", "Can't handle this intent!");*

*}*

1. Kode Program secara keseluruhan untuk MainActivity.java seperti gambar di bawah ini.



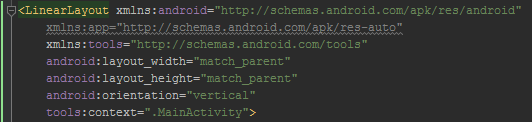
**Kesimpulan**

Untuk melakukan penelusuran lokasi (GPS) diperlukan suatu inputan yang didapat dari EditText yang selanjutnya di klik button yang akan memproses method openLocation() yang di dalam nya ada proses pengaksesan isi dari EditText nya, dan pembuatan objek uri yang gunanya untuk melakukan pengaksesan Google Maps. Selanjutnya dilakukan inisialisasi Objek Intent dengan parameter action ACTION\_VIEW dan objek Uri. Lalu ada proses pengecekan jika activity paket manager nya tidak bernilai null maka akan memulai activity browser yang akan dipilih, jika bernilai null maka keluar pesan error

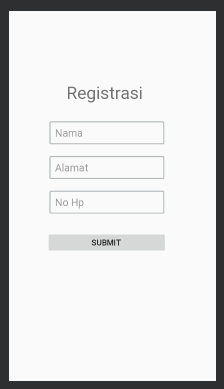
**TUGAS**

**Membuat Activity Pertama**

1. Buka aplikasi Android Studio dan buat sebuah project baru. Pilih Empty Activity untuk template project
2. Nama aplikasi bisa diisikan sesuai dengan keinginan, Pilihlah minimum sdk sesuai kebutuhan, misalnya pada praktikum ini kita gunakan API 15: Android 4.0.3
3. Buka res/layout/activity\_main.xml. Pada Layout Editor, klik tab Design di bagian bawah layar. Ubah layout yang digunakan menjadi **LinearLayout**
4. Gantilah orientation menjadi vertical



1. Buatlah desain seperti gambar dibawah ini



Kode program seperti ini :

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginLeft="100dp"  
 android:layout\_marginTop="120dp"  
 android:layout\_marginBottom="30dp">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Registrasi"  
 android:textSize="30dp" />  
 </LinearLayout>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <EditText  
 android:id="@+id/edittext\_nama"  
 android:layout\_width="200dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="70dp"  
 android:layout\_marginBottom="20dp"  
 android:background="@drawable/edittext"  
 android:hint="Nama"  
 android:paddingLeft="10dp" />  
  
 <EditText  
 android:id="@+id/edittext\_alamat"  
 android:layout\_width="200dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="70dp"  
 android:layout\_marginBottom="20dp"  
 android:background="@drawable/edittext"  
 android:hint="Alamat"  
 android:paddingLeft="10dp" />  
  
 <EditText  
 android:id="@+id/edittext\_noh"  
 android:layout\_width="200dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="70dp"  
 android:layout\_marginBottom="20dp"  
 android:background="@drawable/edittext"  
 android:hint="No Hp"  
 android:paddingLeft="10dp" />  
  
 <Button  
 android:id="@+id/button\_submit"  
 android:layout\_width="210dp"  
 android:layout\_height="40dp"  
 android:layout\_marginLeft="65dp"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginBottom="20dp"  
 android:onClick="launchResult"  
 android:paddingLeft="10dp"  
 android:text="Submit" />  
  
  
 </LinearLayout>  
  
  
</LinearLayout>

1. Lalu buatlah method onclick launchResult() untuk proses button nya



1. Lalu diatas kelas lakukan pembuatan variabel untuk mengakses isi dari EditText

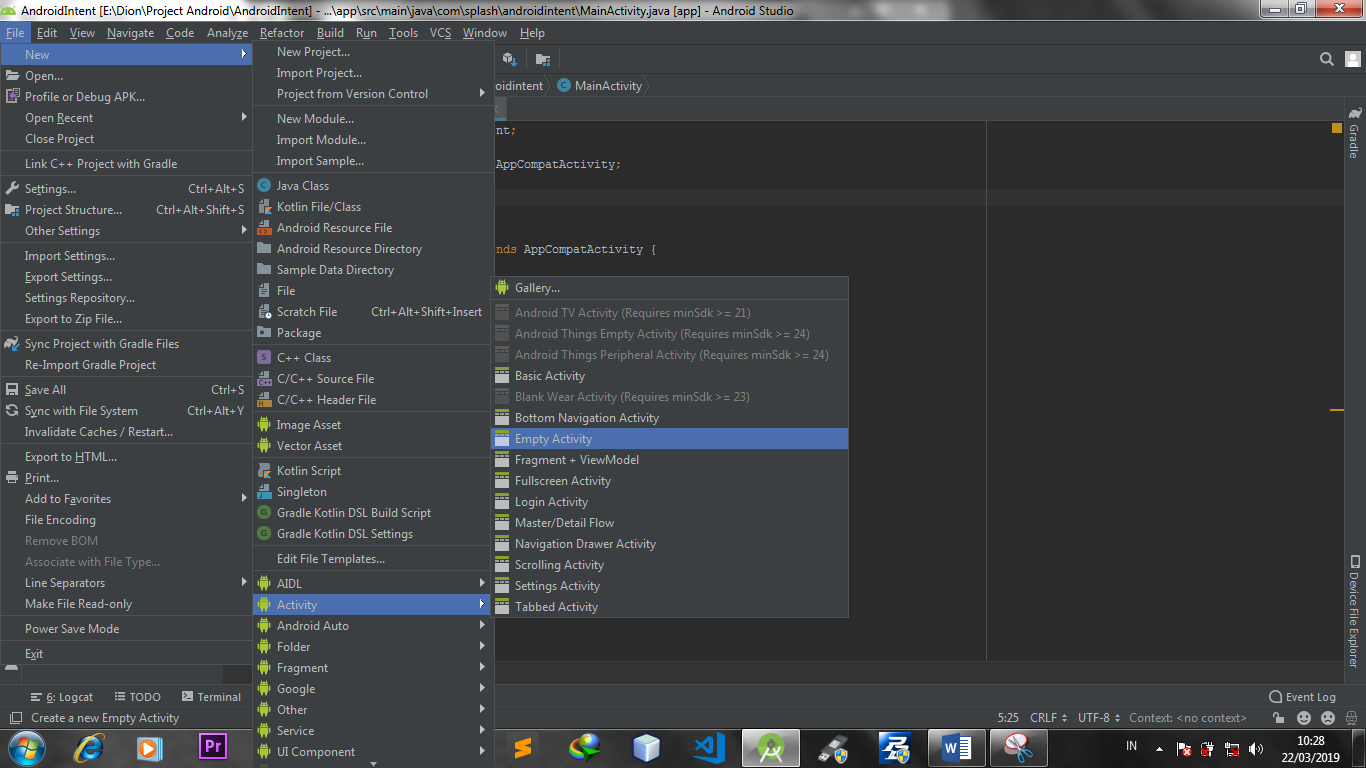
private EditText nama;  
private EditText alamat;  
private EditText nohp;

1. Pada method onCreate() tambahkan inisialisasi dari tiap variabel diatas untuk mengakses tiap EditText nya

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 nama = (EditText) findViewById(R.id.*edittext\_nama*);  
 alamat = (EditText) findViewById(R.id.*edittext\_alamat*);  
 nohp = (EditText) findViewById(R.id.*edittext\_noh*);  
  
}

**Membuat Activity Kedua**

1. Klik folder app untuk proyek Anda dan pilih **File > New > Activity > Empty Activity.**



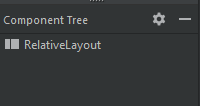
1. Namakan aktivitas baru "ResultActivity." Pastikan Generate Layout File dicentang, dan nama layout akan diisi dengan activity\_result.
2. Klik Finish. Android Studio menambahkan layout aktivitas baru (activity\_result) dan file Java baru (ResultActivity) ke proyek Anda untuk aktivitas baru tersebut. Ini juga akan memperbarui manifes Android untuk menyertakan aktivitas baru.
3. Buka manifests/AndroidManifest.xml.
4. Temukan elemen <activity> yang dibuat Android Studio untuk aktivitas kedua.
5. Tambahkan atribut ini ke elemen <activity>:

<activity android:name=".ResultActivity"  
 android:label="Result Activity"  
 android:parentActivityName=".MainActivity"  
 >

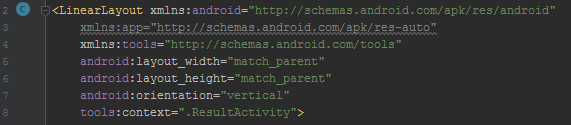
1. Tambahkan elemen <meta-data> element inside the <activity> di dalam elemen untuk aktivitas kedua. Gunakan atribut berikut ini:

<meta-data  
 android:name="android.support.PARENT\_ACTIVITY"  
 android:value="com.splash.tugasintent.MainActivity" />

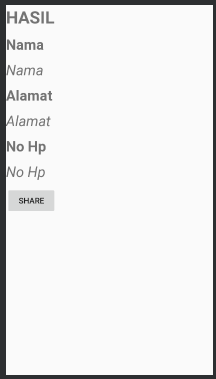
1. Buka res/layout/activity\_second.xml. Ubah layout yang digunakan menjadi **RelativeLayout**



1. Gantilah orientation menjadi vertical



1. Lalu buatlah desain activity\_result.xml seperti ini



Kode program keseluruhan seperti ini:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".ResultActivity">  
  
 <TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="HASIL"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="bold"  
 android:textSize="30dp"/>  
 <TextView  
 android:id="@+id/textView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Nama"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="bold"  
 android:textSize="25dp"/>  
 <TextView  
 android:id="@+id/textViewNama"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Nama"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="italic"  
 android:textSize="25dp"/>  
 <TextView  
 android:id="@+id/textView3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Alamat"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="bold"  
 android:textSize="25dp"/>  
 <TextView  
 android:id="@+id/textViewAlamat"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Alamat"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="italic"  
 android:textSize="25dp"/>  
 <TextView  
 android:id="@+id/textView4"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="No Hp"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="bold"  
 android:textSize="25dp"/>  
 <TextView  
 android:id="@+id/textViewNohp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="No Hp"  
 android:layout\_marginBottom="10dp"  
 android:textStyle="italic"  
 android:textSize="25dp"/>  
 <Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Share"  
 android:onClick="launchShare"/>  
</LinearLayout>

**Membuat Intent**

1. Pada MainActivity.java di dalam method tambahkan proses intent dengan fungsi extras dimana lebih dari satu data akan dikirim ke ResultActivity

Kode program seperti ini :

public void launchResult(View view) {  
 Intent intent = new Intent(this,ResultActivity.class);  
 String isi\_nama = nama.getText().toString();  
 String isi\_alamat = alamat.getText().toString();  
 String isi\_nohp = nohp.getText().toString();  
  
 Bundle extras = new Bundle();  
 extras.putString("NAME",isi\_nama);  
 extras.putString("ALAMAT",isi\_alamat);  
 extras.putString("NOH",isi\_nohp);  
 intent.putExtras(extras);  
 startActivity(intent);  
}

Dengan inisialisasi Intent baru menuju ke ResultActivity, Lalu membuat variabel untuk menyimpan data isi\_nama, isi\_alamat, isi\_nohp dari EditText.

Pada pembuatan Class Bundle ada pembuatan objek baru yang bernama extras yang nantinya dikirim ke ResultActivity. Selanjutnya ada proses memasukkan data ke objek extras yaitu nama, alamat, dan no hp. Selanjutnya memulai activity baru dengan perintah startActivity(intent).

1. Pada ResultActivity.java diatas kelas lakukan deklarasi variabel untuk menyimpan data dari TextView

private TextView textNama;  
private TextView textAlamat;  
private TextView textNoh;

1. Di dalam method onCreate() lakukan inisialisasi Intent dengan mengambil data dari intent MainActivity, yaitu dengan cara GetIntent(), untuk akses data nya yaitu dengan membuat objek bundle baru yang bernama extras untuk mengambil data yaitu dengan cara Intent.getExtras().

Lalu simpanlah data nya pada variabel baru.

Lakukan inialisasi dari variabel textView diatas untuk melakukan manipulasi pada textView tersebut.

Terakhir lakukan set text di dalam textView dengan data yang sesuai dengan data yang dari MainActivity

Kode program seperti berikut :

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_result*);  
  
 Intent intent = getIntent();  
 Bundle extras = intent.getExtras();  
  
 String nama = extras.getString("NAME");  
 String alamat = extras.getString("ALAMAT");  
 String noh = extras.getString("NOH");  
  
 textNama = (TextView) findViewById(R.id.*textViewNama*);  
 textAlamat = (TextView) findViewById(R.id.*textViewAlamat*);  
 textNoh = (TextView) findViewById(R.id.*textViewNohp*);  
  
 textNama.setText(nama);  
 textAlamat.setText(alamat);  
 textNoh.setText(noh);  
}

1. Buatlah method launchShare untuk melakukan share data ke platform lain

Kode program seperti ini :

public void launchShare(View view) {  
 String hasil = "Nama : "+textNama.getText().toString() + "\n" +  
 "Alamat : "+textAlamat.getText().toString();  
 Intent sendIntent = new Intent();  
 sendIntent.setAction(Intent.*ACTION\_SEND*);  
 sendIntent.putExtra(Intent.*EXTRA\_TEXT*, hasil);  
 sendIntent.setType("text/plain");  
 startActivity(sendIntent);  
}

Pertama lakukan inialisasi variabel hasil dengan isi nama yang diambil dari textView dan Alamat yang diambil dari textView. Lalu lakukan inisialisasi objek Intent dengan action nya Intent.ACTION\_SEND, lalu lakukan put data dengan parameter Intent.EXTRA\_TEXT dan variabel yang menyimpan data nama dan alamat, Lalu set Intent dengan type text/plain yang berarti akan meng-outputkan sebuah text.

Terakhir lakukan startActivity(Intent).