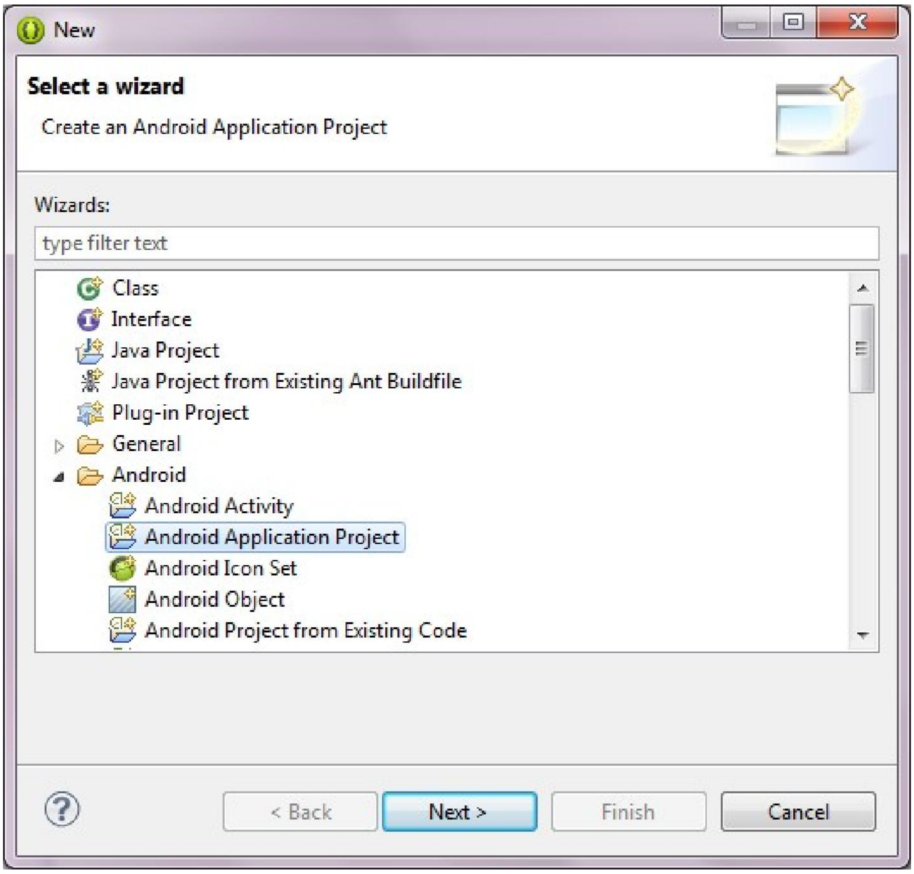
PERTEMUAN 10:
NETWORK

A. TUJUAN PEMBELAJARAN

Adapun tujuan pembelajaran yang akan dicapai sebagai berikut:

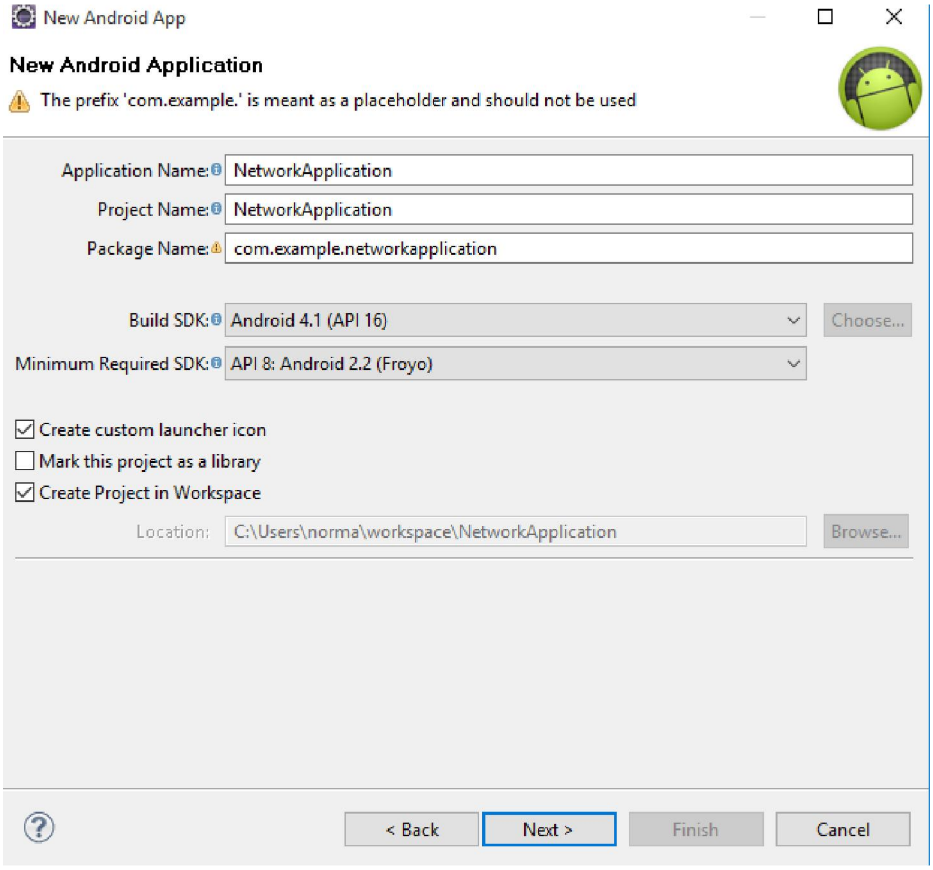
10.1 Membuat Aplikasi Parsing Data dengan Network



Select a wizard

Create an Android Application Project

Gambar 10. 1



New Android App

New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: NetworkApplication

Project Name: NetworkApplication

Package Name: com.example.networkapplication

Build SDK: Android 4.1 (API 16)

Minimum Required SDK: API 8: Android 2.2 (Froyo)

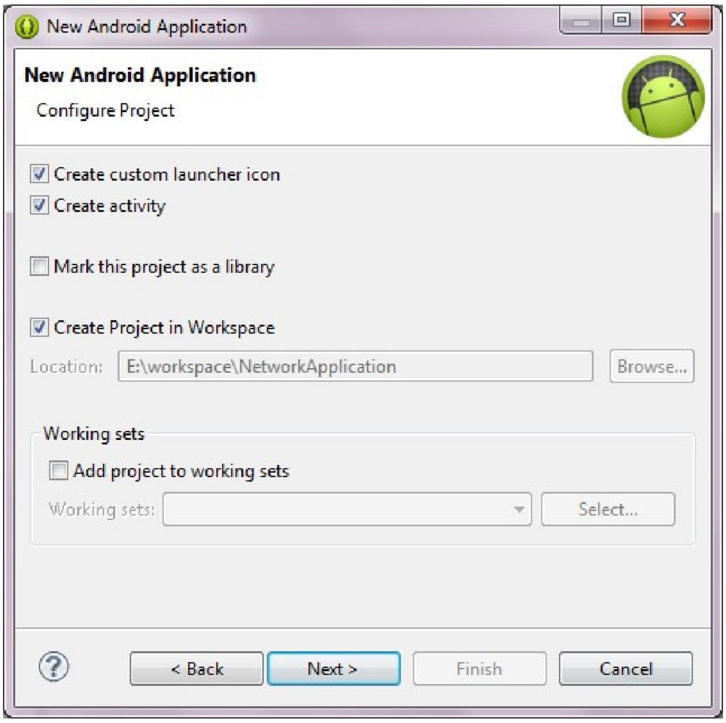
☑ Create custom launcher icon

□ Mark this project as a library

☑ Create Project in Workspace

Location: C:\Users\norma\workspace\NetworkApplication

Gambar 10. 2



New Android Application

New Android Application

Configure Project

☑ Create custom launcher icon

☑ Create activity

□ Mark this project as a library

☑ Create Project in Workspace

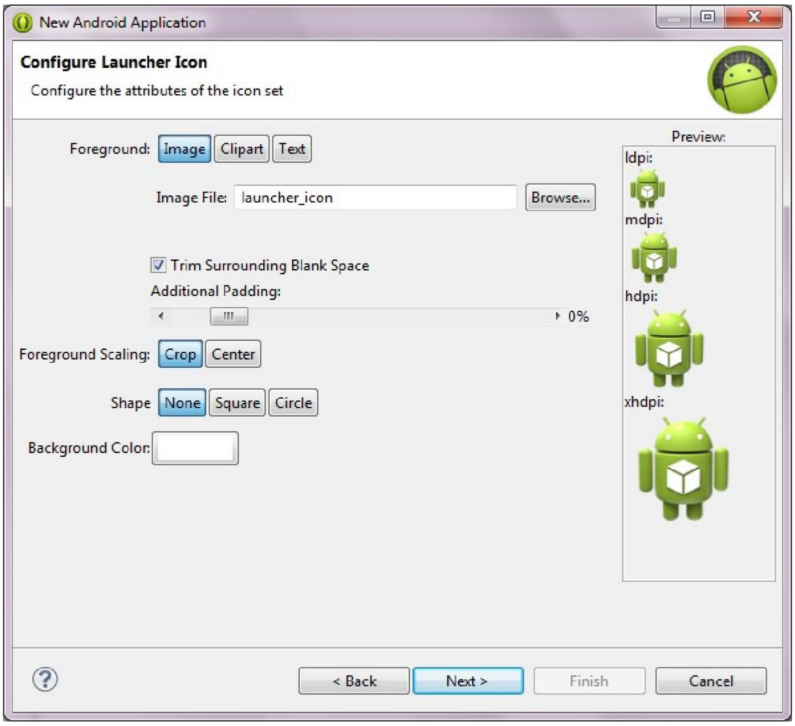
Location: E:\workspace\NetworkApplication

Working sets

□ Add project to working sets

Working sets:

Gambar 10. 3


New Android Application

Configure Launcher Icon

Configure the attributes of the icon set

Foreground: Image | Clipart | Text

Image File: launcher_icon

☑ Trim Surrounding Blank Space

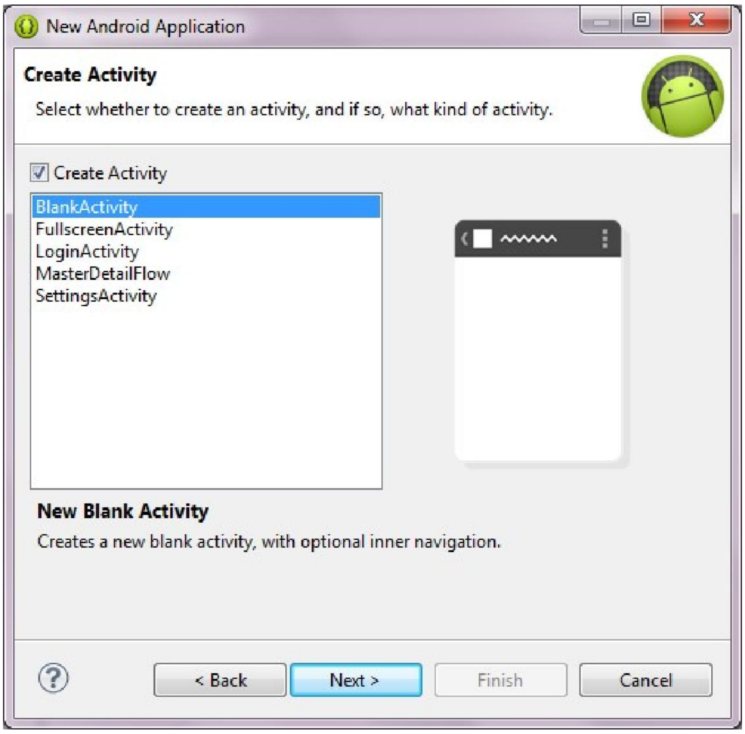
Additional Padding: 0%

Foreground Scaling: Crop | Center

Shape: None | Square | Circle

Background Color:

Gambar 10. 4


New Android Application

Create Activity

Select whether to create an activity, and if so, what kind of activity.

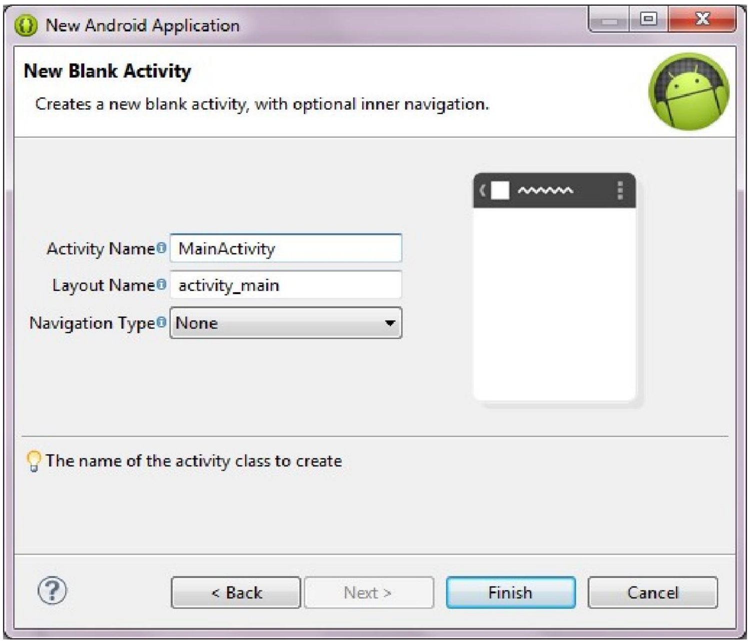
☑ Create Activity

- BlankActivity
- FullscreenActivity
- LoginActivity
- MasterDetailFlow
- SettingsActivity

New Blank Activity

Creates a new blank activity, with optional inner navigation.

Gambar 10. 5



New Android Application

New Blank Activity

Creates a new blank activity, with optional inner navigation.

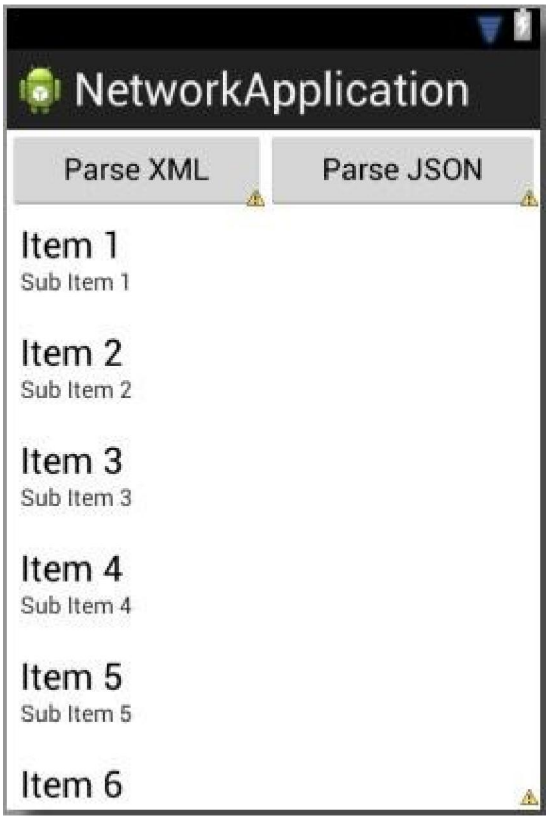
Activity Name: MainActivity

Layout Name: activity_main

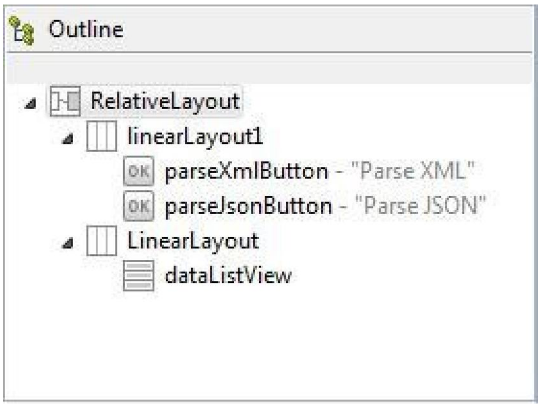
Navigation Type: None

The name of the activity class to create

Gambar 10. 6



Gambar 10. 7 Layout Aplikasi

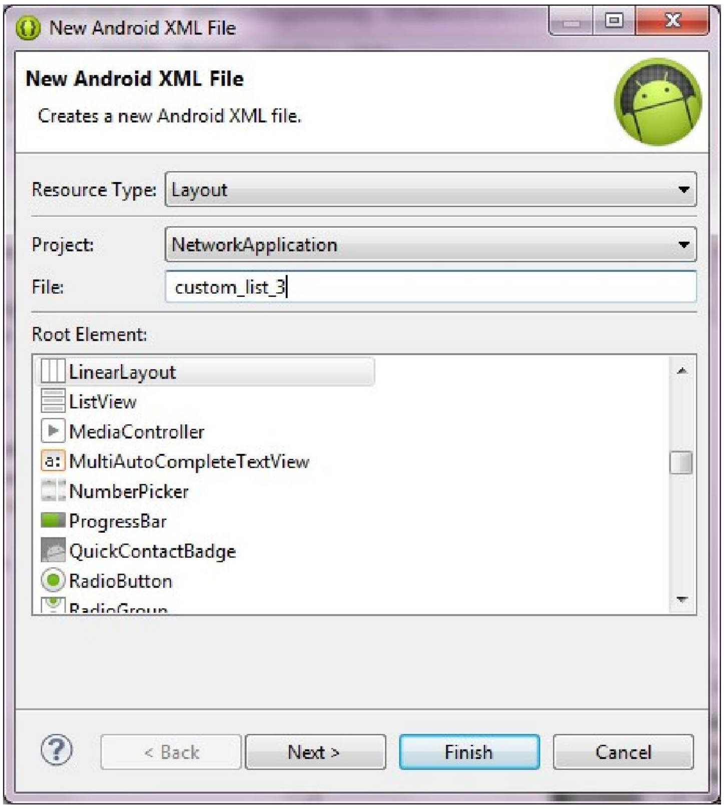


Outline

```
RelativeLayout
    linearLayout1
        parseXmlButton - "Parse XML"
        parseJsonButton - "Parse JSON"
    LinearLayout
        dataListView
```

Gambar 10. 8 Outline project parsing data

Padanama project, klikkanandanpilih New >> Android XML File, kemudianberinama custom_list_3 sepertiGambar 11.9 berikutini:



New Android XML File

New Android XML File

Creates a new Android XML file.

Resource Type: Layout

Project: NetworkApplication

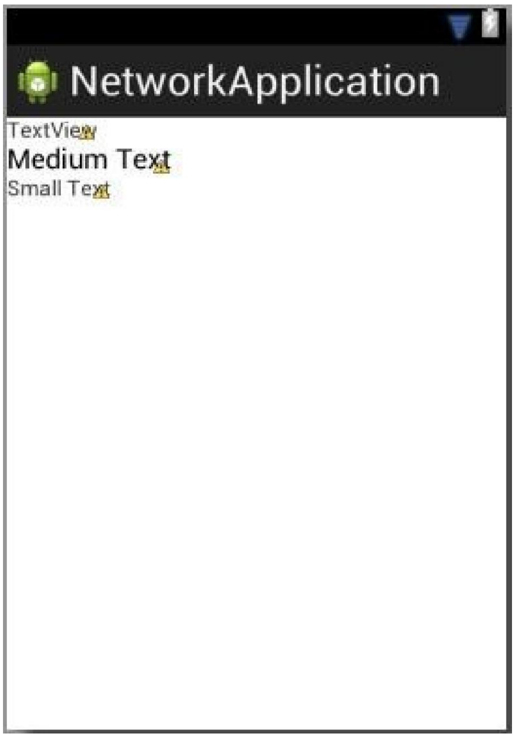
File: custom_list_3

Root Element:

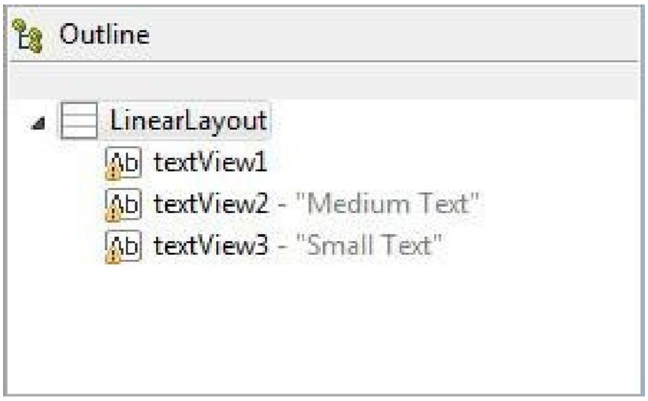
- LinearLayout
- ListView
- MediaController
- MultiAutoCompleteTextView
- NumberPicker
- ProgressBar
- QuickContactBadge
- RadioButton
- RadioGroup

Gambar 10. 9

Aturdesainnyasepertigambar di bawahini:

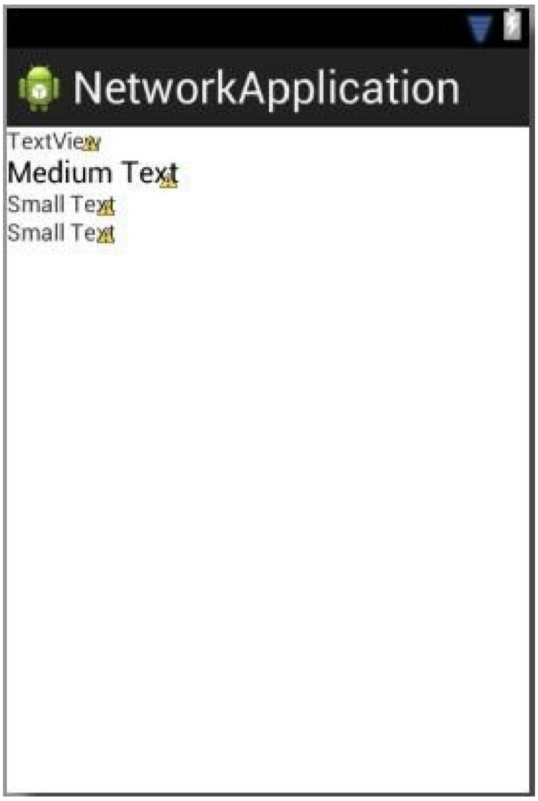


Gambar 10. 10

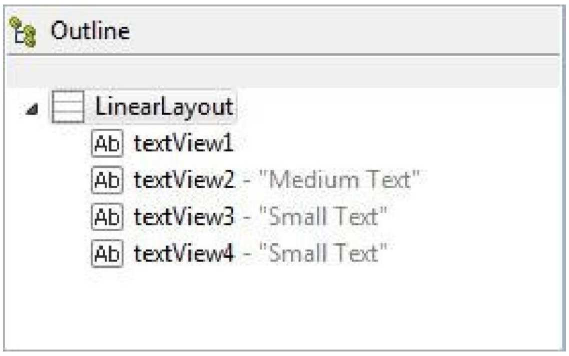


Gambar 10. 11

Dengancara yang sama, tambahkan custom_list_4 dengandesainsebagaiberikut:



Gambar 10. 12



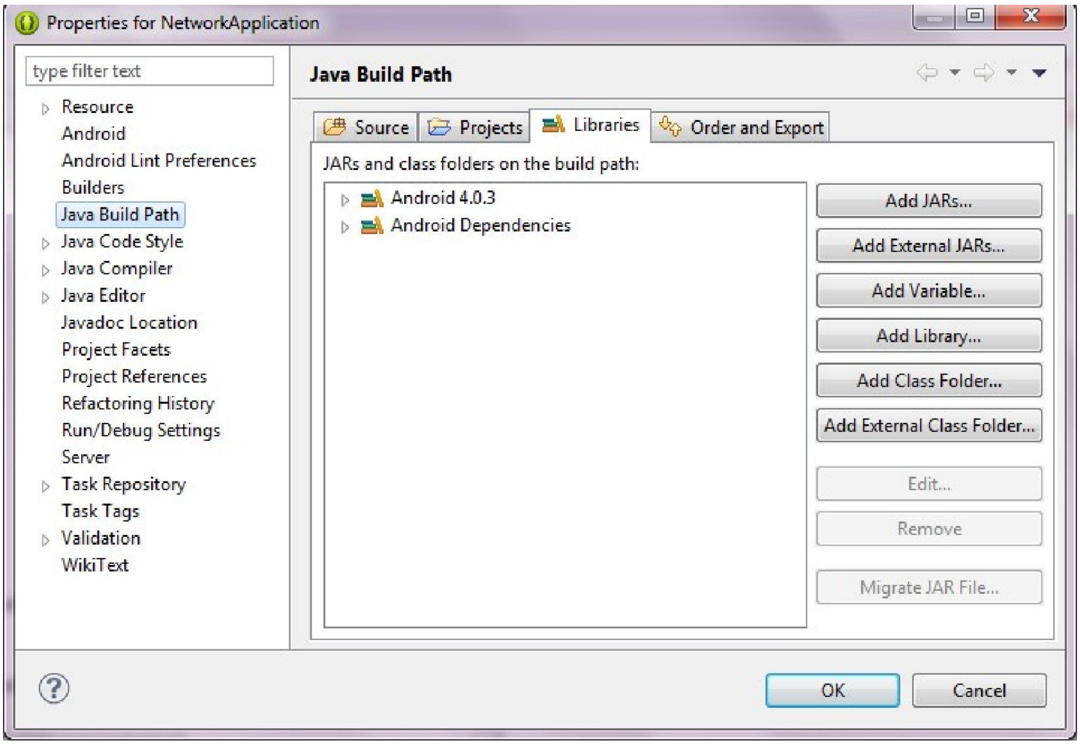
Gambar 11. 13

Untukmelakukanparsing data dengan format XML, download file Simple XML Serialization di http://simple.sourceforge.net/download.php. Sedangkanuntukparsing data JSON, download file embedding java librariesdari http://code.google.com/p/jarjar/downloads/list danjava libraryuntukmengkonvesi JSON keobjek java dari http://code.google.com/p/google-gson/downloads/list. Buat file rules.txt, kemudianisikan “rule com.google.gson.** com.google.myjson.@1” tanpatandakutip.

Bukacommand promptdanjalankanperintahberikutini (sesuaikanversi file yang telah di-download):

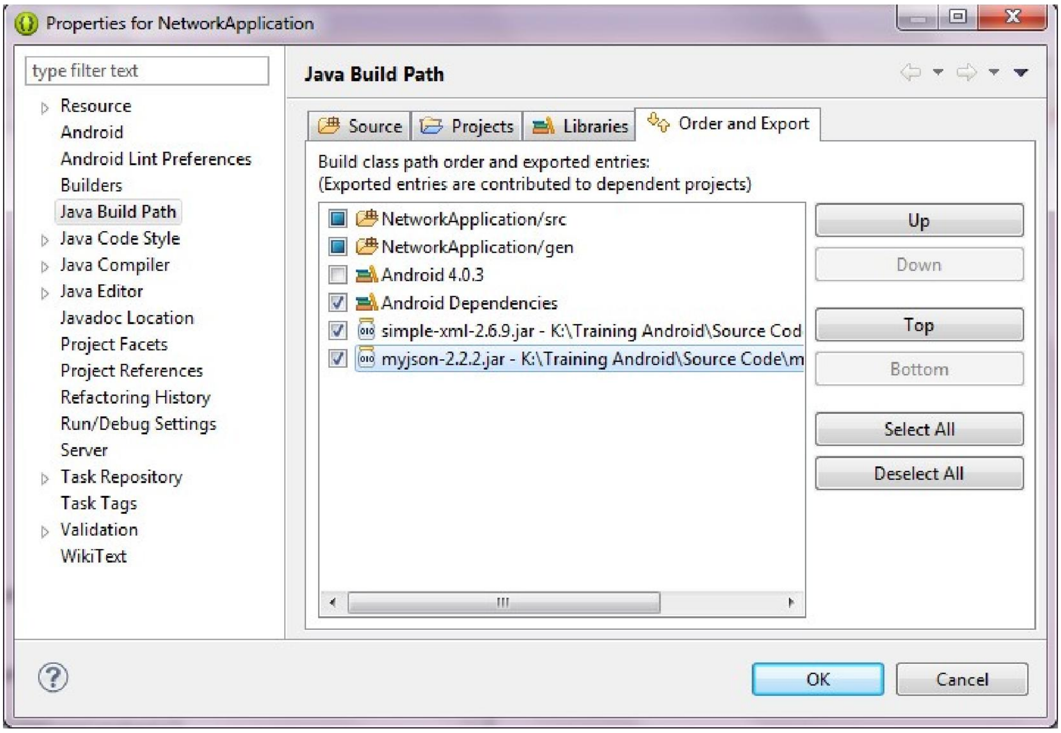
```
java -jar jarjar-1.4.jar process rules.txt gson-2.2.2.jar myjson-
2.2.2.jar
```

Padanama project, klikkanandanpilih Build Path >> Configure Build Path. Pilihkategori Java Build Path danpilih tab Libraries sepertigambar



Gambar 10. 14 kotak dialog import library

Klikpadabutton Add External JARs, tambahkan file simple-xml-2.6.9.jar dan myjson-2.2.2.jar. Kemudianklikpadatab Order and Exportdanberitandacheckpadakedua file tersebutsepertigambar 11.15 berikutini:



Properties for NetworkApplication

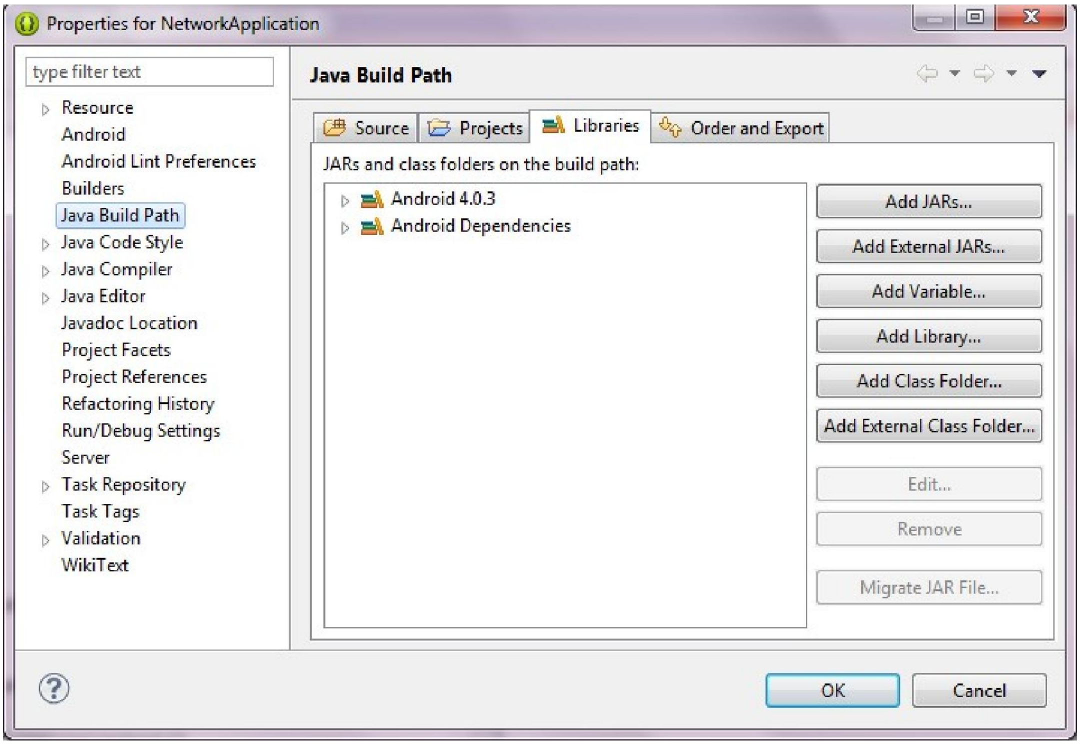
Java Build Path

Build class path order and exported entries:
(Exported entries are contributed to dependent projects)

- ■ NetworkApplication/src
- ■ NetworkApplication/gen
- □ Android 4.0.3
- ☑ Android Dependencies
- ☑ simple-xml-2.6.9.jar - K:\Training Android\Source Cod
- ☑ myjson-2.2.2.jar - K:\Training Android\Source Code\m

Gambar 10. 15

Padanama project, klikkanandanpilih Build Path >> Configure Build Path. Pilihkategori Java Build Path danpilih tab Libraries sepertigambar



Properties for NetworkApplication

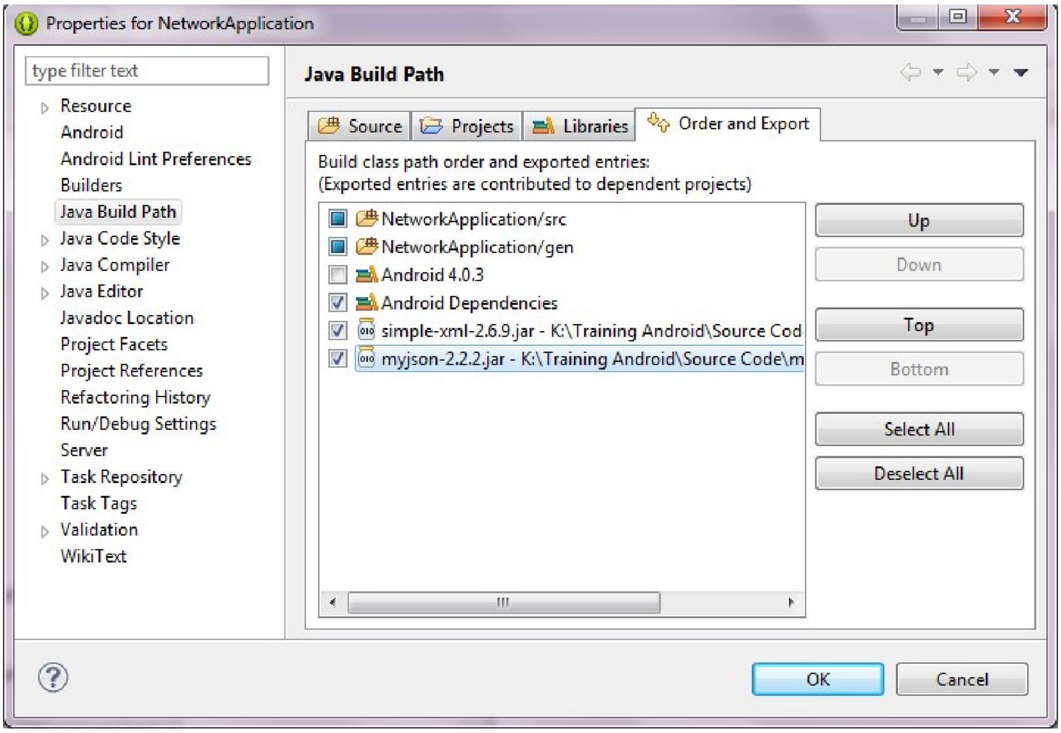
Java Build Path

JARs and class folders on the build path:

- Android 4.0.3
- Android Dependencies

Gambar 10. 14

Klikpadabutton Add External JARs, tambahkan file simple-xml-2.6.9.jar dan myjson-2.2.2.jar. Kemudianklikpadatab Order and Exportdanberitandacheckpadakedua file tersebutsepertigambar 11.15 berikutini:



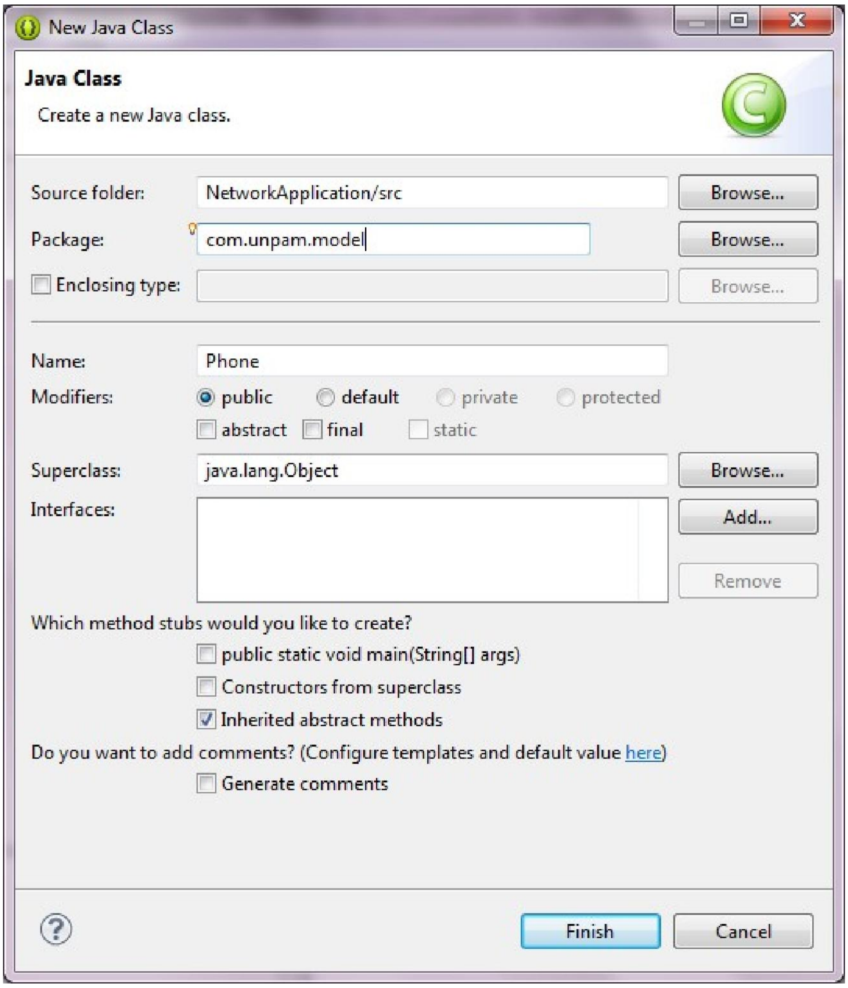
Build class path order and exported entries:
(Exported entries are contributed to dependent projects)

- ■ NetworkApplication/src
- ■ NetworkApplication/gen
- □ Android 4.0.3
- ☑ Android Dependencies
- ☑ simple-xml-2.6.9.jar - K:\Training Android\Source Cod
- ☑ myjson-2.2.2.jar - K:\Training Android\Source Code\m

Gambar 10. 15

Source Code Phone.java

PadanamaprojectklikkanandanpilihNew >> Class, berinamapackage com.unpam.modeldannamaclass Phonesepertigambar 11.16 di bawahini:



Java Class

Create a new Java class.

Source folder: NetworkApplication/src [Browse...]

Package: com.unpam.model [Browse...]

□ Enclosing type: [Browse...]

Name: Phone

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object [Browse...]

Interfaces: [Add...] [Remove]

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

Gambar 10. 16

Ubah source code menjadisepertiberikutini:

```
packagecom.unpam.model;

import org.simpleframework.xml.Attribute;
importorg.simpleframework.xml.Text;

        Attribute
publicclass Phone {
     @
     private String ext;

     @Text
     private String text;
```

```
             public String getExt() {
             returnext;
           }

             public String getText() {
             returntext;
           }









}
```

Source Code Restaurant.java

```
package com.unpam.model;

import org.simpleframework.xml.Element;

publicclass Restaurant {
          @Element
          privateintid;


          @Element    private String name;

        @Element
        private String address;

          @Element    private Phone phone;
         publicint getId() {
                returnid;
        }

          public String getName() {
          returnname;
        }

          public String getAddress() {
          returnaddress;
```

```
    }
     public Phone getPhone()
{          returnphone;
    }

}
```

Source Code Result.java

```
package com.unpam.model;

import java.util.List;

import org.simpleframework.xml.Element;
import org.simpleframework.xml.ElementList;
import org.simpleframework.xml.Root;
@Root
publicclass Result {
     @Element   privateintcount;
@ElementList
private List<Restaurant>data;

publicint getCount() {
     returncount;
}

public List<Restaurant> getData() {
     returndata;
}
}
```

Source Code GRestaurant.java

```
package com.unpam.jsonmodel;

publicclass GRestaurant
{
privateintid;  private
String name;
private String address;

    publicint getId() {
         returnid;
    }

     public String getName() {
          returnname;
    }

     public String getAddress() {
          returnaddress;
     }
}
```

Source Code RestaurantGson.java

```
package com.unpam.jsonmodel;
import java.util.List;
publicclass RestaurantGson {
     privateintcount;
     private
List<GRestaurant>data;

     publicint getCount() {
          returncount;
     }
public List<GRestaurant> getData() {
returndata;
}
}
```

Source Code GResult.java

```
package com.unpam.jsonmodel;

publicclass GResult {
     private RestaurantGson
result;

     public RestaurantGson getResult() {
     returnresult;
    }

}
```

Source Code MainActivity.java

```
package com.unpam.networkapplication;

import java.io.InputStream;
import java.net.URL;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.DefaultHttpClient; import
org.apache.http.protocol.HTTP;
import org.apache.http.util.EntityUtils;
import org.json.JSONException;
import org.simpleframework.xml.Serializer;
importorg.simpleframework.xml.core.Persister;

import android.app.Activity; import
android.app.ProgressDialog; import
android.os.AsyncTask; import
android.os.Bundle;
import android.view.Gravity; import
android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;
import com.google.myjson.Gson;
import com.unpam.jsonmodel.GResult;
import com.unpam.model.Result;

publicclass MainActivity extends Activity implements
OnClickListener {
private String torestoXmlAddress =
"http://toresto.com/restaurant.xml";
private String torestoJsonAddress =
"http://toresto.com/restaurant.json";

privatefinal String ID = "id";
privatefinal String NAMA = "nama";
privatefinal String ALAMAT = "alamat";
privatefinal String PHONE = "phone";

List<Map<String, String>>items = new ArrayList<Map<String,
String>>();
ListView listView;

@Override
protectedvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main;
findViewById(R.id.parseXmlButton).setOnClickListener(this);
findViewById(R.id.parseJsonButton).setOnClickListener(this);
listView = (ListView) findViewById(R.id.dataListView);
}

@Override
publicboolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is
present.
getMenuInflater().inflate(R.menu.activity_main, menu);
      returntrue;
}

@Override
publicvoid onClick(View v) {
// TODO Auto-generated method stub
switch (v.getId()) {             case
R.id.parseXmlButton:
      new Thread() {
      publicvoid run() {
parseSimpleXml();
};
}.start();
break;
case R.id.parseJsonButton:
new MyJsonTask().execute(torestoJsonAddress);
}
}

publicvoid parseSimpleXml() {
try {
URL url = new URL(torestoXmlAddress);
      InputStream is = url.openStream();
      Serializer serializer = new Persister();
final Result restaurant =
serializer.read(Result.class, is);

runOnUiThread(new Runnable() {

@Override
publicvoid run() {
Toast toast =
Toast.makeText(MainActivity.this, "Ada " +
restaurant.getData().size() + " data",Toast.LENGTH_LONG);
                 toast.setGravity(Gravity.TOP |
Gravity.RIGHT, 0, 0);
toast.show();

items.clear();
Map<String, String> map;                         for
(int i = 0; i <
restaurant.getData().size(); i++) {
map = new HashMap<String,
String>();

map.put(ID,Integer.toString(restaurant.getData().get(i).getId
()));
map.put(NAMA,
restaurant.getData().get(i).getName());
map.put(ALAMAT,
restaurant.getData().get(i).getAddress());
map.put(PHONE,
restaurant.getData().get(i).getPhone().getExt()+ " - " +
restaurant.getData().get(i).getPhone().getText());
items.add(map);
}

String[] from = new String[] { ID,
NAMA, ALAMAT, PHONE };
int[] to = newint[] { R.id.textView1,
R.id.textView2,R.id.textView3, R.id.textView4 };
SimpleAdapter adapter = new
SimpleAdapter(MainActivity.this, items,
R.layout.custom_list_4,from, to);
listView.clearChoices();
listView.setAdapter(adapter);
}
});
```

```
} catch (final Exception e) {
runOnUiThread(new Runnable() {

@Override
publicvoid run() {
Toast.makeText(MainActivity.this,
e.getMessage(),Toast.LENGTH_LONG).show();
}
});
}
}

public String getResponseFromURL(String url) {
String response = null;          HttpClient
httpclient = null;
try {
HttpGet httpget = new HttpGet(url);              httpclient =
new DefaultHttpClient();              HttpResponse
httpResponse =
httpclient.execute(httpget);

finalint statusCode =
httpResponse.getStatusLine().getStatusCode();
if (statusCode != HttpStatus.SC_OK) {
      thrownew Exception("Got HTTP " + statusCode
+ " (" + httpResponse.getStatusLine().getReasonPhrase() + ')');
}

response =
EntityUtils.toString(httpResponse.getEntity(),HTTP.UTF_8);
} catch (Exception e) {

} finally {
if (httpclient != null) {

httpclient.getConnectionManager().shutdown();
httpclient = null;
}
}
return response;
}
```

```
public GResult parseGson(String url) throws JSONException {
     String response = getResponseFromURL(url);

Gson gson = new Gson();
GResult gResult = gson.fromJson(response,GResult.class);
gResult.getResult().getData().get(0).getName();
     return gResult;
}

class MyJsonTask extends AsyncTask<String, Void, GResult> {
ProgressDialog dialog;

@Override
protectedvoid onPreExecute() {
dialog = ProgressDialog.show(MainActivity.this, null, "Loading");
}

@Override
protected GResult doInBackground(String... params) {
try {
GResult gResult = parseGson(params[0]);
     return gResult;
} catch (Exception e) {
returnnull;
}
}

@Override
protectedvoid onPostExecute(GResult result) {
dialog.cancel();                  if
(result != null) {
Toast.makeText(MainActivity.this,"Ada "+
result.getResult().getCount() + "
data",Toast.LENGTH_SHORT).show();

items.clear();
Map<String, String> map;                     for
(int i = 0; i <
result.getResult().getData().size(); i++) {
map = new HashMap<String, String>();

map.put(ID,Integer.toString(result.getResult().getData().get(
i).getId()));
map.put(NAMA,
result.getResult().getData().get(i).getName());
map.put(ALAMAT,
result.getResult().getData().get(i).getAddress());

items.add(map);
}

String[] from = new String[] { ID, NAMA,
ALAMAT };
int[] to = newint[] { R.id.textView1,
R.id.textView2, R.id.textView3};
SimpleAdapter adapter = new
SimpleAdapter(MainActivity.this,items, R.layout.custom_list_3,
from, to);
listView.setAdapter(adapter);
} else {
// munculkan pesan error
Toast.makeText(MainActivity.this,"Tidak
dapat membaca data",Toast.LENGTH_SHORT).show();
                }
           }
     }
}
```

Karenaaplikasimembutuhkankoneksike internet, makatambahkan tag usespermission di dalam file AndroidManifest.xml (di antara tag uses-sdkdan application).

```
<uses-permissionandroid:name="android.permission.INTERNET"/>
```

Tampilan

Data yang tersimpanadalah: http://toresto.com/restaurant.xml

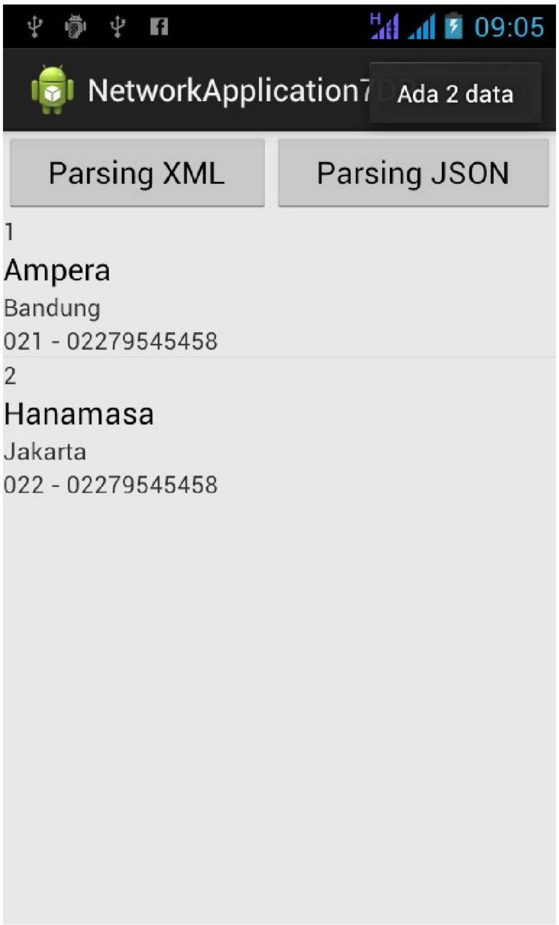
```
<result>
    <count>1</count>
    <data>
        <restaurant>
            <id>1</id>
            <name>Ampera</name>
            <address>Bandung</address>
            <phone ext="021">02279545458</phone>
        </restaurant>
        <restaurant>
            <id>2</id>
```

```
              <name>Hanamasa</name>
              <address>Jakarta</address>
              <phone ext="022">02279545458</phone>
          </restaurant>
     </data>
</result>
```

http://toresto.com/restaurant.json

```
 {"result":{
     "count":1,
     "data":[{
           "id":1,
           "name":"Aaa",
           "address":"Bandung"
     },
     {
           "id":2,
           "name":"BBB",
           "address":"Bandung"
     }
     ]
 }
}
```

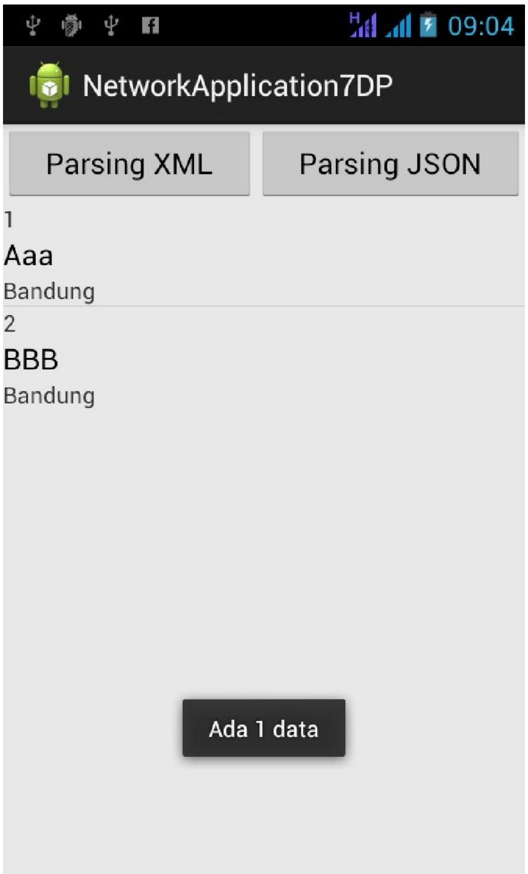
Tampilanaplikasisetelah di klik button Parsing XML:



1
Ampera
Bandung
021 - 02279545458
2
Hanamasa
Jakarta
022 - 02279545458

Gambar 10. 17 Halaman Parsing data

Tampilanaplikasisetelah di klik button Parsing JSON:

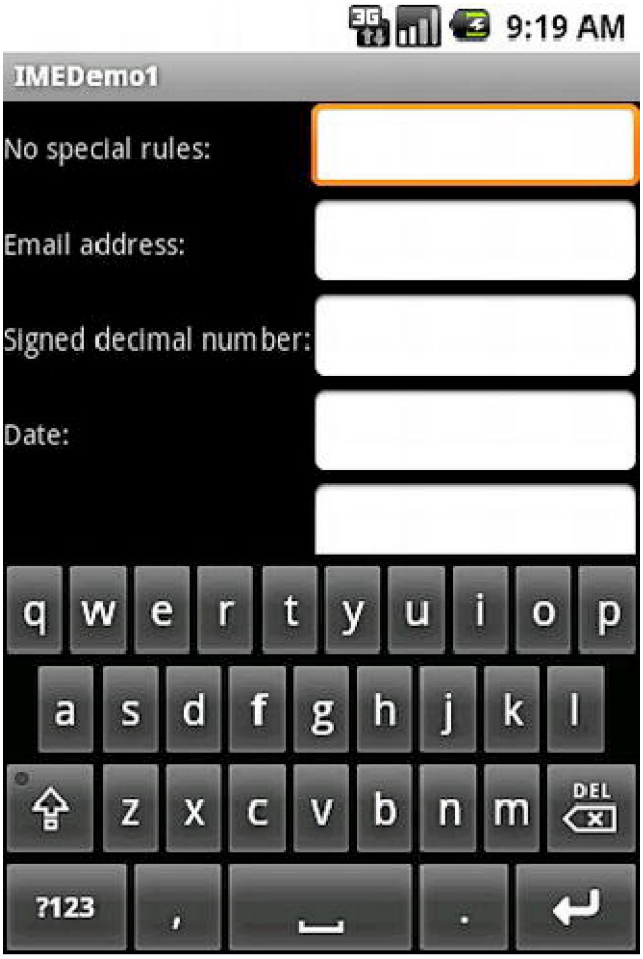


```
1
Aaa
Bandung
2
BBB
Bandung
```

Gambar 10. 18 Halamanhasil parsing data

C. SOAL LATIHAN/TUGAS

Buatlah project berikut!



D. DAFTAR PUSTAKA

Allen, Grant. 2012. Beginning Android 4. New York :Apress.

Safaat, H. Nazruddin. 2015. ANDROID PemrogramanAplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika