**BAB V**

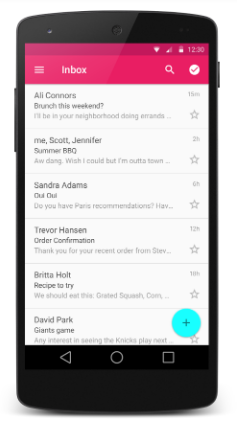
**RecyclerView**

**5.1 Tujuan Pembelajaran**

1. Mahasiswa dapat menggunakan *RecyclerView* ke dalam aplikasi.
2. Mahasiswa mengetahui berbagai macam *RecyclerView* seperti *List*, *Grid*, dan *Card*.
3. Mahasiswa dapat membuat berbagai macam *RecyclerView* seperti *List*, *Grid*, dan *Card*.
   1. ***Software* yang dibutuhkan**
4. *Java JDK*
5. *Android Studio 3.5*
6. *SDK API 29*
   1. **Teori**

Untuk menampilkan data dalam bentuk *list*, biasanya kita menggunakan *ListView* misalnya untuk menampilkan data-data yang berasal dari kontak di *device* pengguna ke layar. Sangat sederhana, tetapi sangat berarti. Mengapa ? Pada dasarnya interaksi umum antara pengguna dengan aplikasi dalam menampilkan data dalam jumlah yang banyak adalah dengan menggunakan *list* yang bisa di *scroll* ke atas dan ke bawah.

*ListView* menjadi komponen pertama yang mengakomodasi hal tersebut, namun semenjak *Google* merilis pendekatan desain bernama *material design*, *RecyclerView* menjadi pilihan pertama yang harus *developer* gunakan. Gambar 5.1 merupakan contoh penggunaan *RecyclerView.* Anda masih bisa menggunakan kedua komponen tersebut secara berdampingan dalam satu aplikasi.



Gambar 5 1 Contoh penggunaan RecyclerView

Secara definisi *RecyclerView* adalah sebuah komponen tampilan *(widget)* yang lebih canggih ketimbang pendahulunya *ListView* dan bersifat lebih fleksibel. Yang paling penting adalah *RecyclerView* memiliki kemampuan untuk menampilkan data secara efisien dalam jumlah yang besar. Terlebih jika Anda memiliki koleksi data yang tiap elemennya mampu berubah-ubah sewaktu dijalankan *(runtime)* karena interaksi pengguna atau karena adanya pengaruh dari jaringan *internet*.

RecyclerView

Adapter

Layout Manager

Dataset

Gambar 5.2 Komponen pembentuk *RecyclerView*

Gambar 5.2 akan menjelaskan beberapa komponen yanga harus Anda tahu sebelum menggunakan *RecyclerView.*

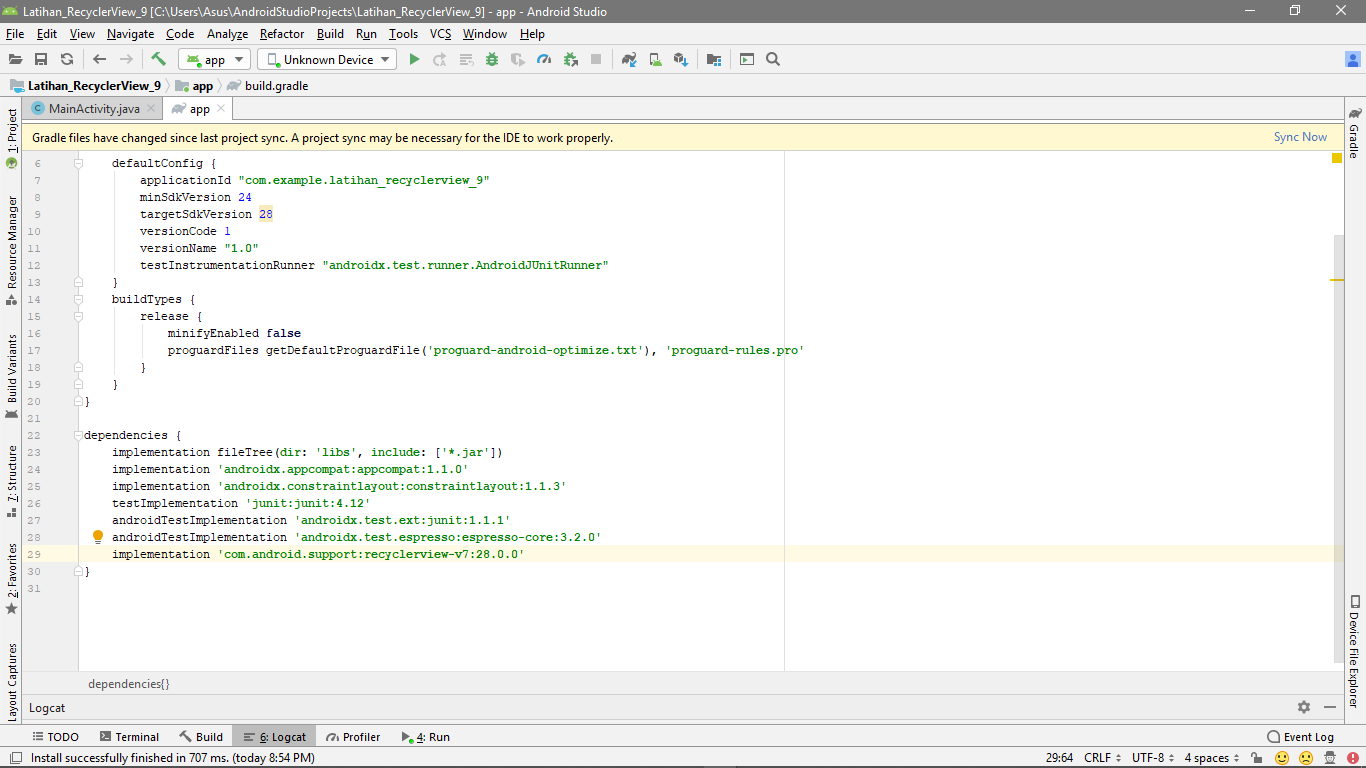
1. *RecyclerView* dan *LayoutManager* : Komponen user interface yang bertugas untuk menampilkan data set yang dimiliki. Di dalamnya terdapat *LayoutManager* yang mana akan mengatur posisi tampilan data baik itu secara *List* (vertikal), *Grid* (baris dan kolom) atau *StaggeredGrid* (Grid yang memiliki susunan tak seragam/tak beraturan) .
2. *Adapter* : Komponen yang akan mengatur bagaimana tampilan data set ke dalam *RecyclerView*. Disinilah terjadi proses pengisian tampilan *(ViewInflate)* dari file layout xml untuk tiap elemen dari set data yang dimiliki sebelum dipasang *(bind)* ke dalam *RecyclerView.*
3. *Dataset* : Kumpulan data yang dimiliki dan ingin ditampilkan, bias berupa *Array, List* maupun objek map.
4. *Item Animator* : Ini yang special, kita bisa set animasi untuk tiap item di dalamnya bahkan bias membuat kustomisasi animasi. Contoh animasi yang umum seperti penambahan *(add)* dan penghapusan *(removal)* item.

Langkah-langkah mengimplementasikan *RecyclerView* sebagai berikut :

1. Tambahkan *dependencies* komponen *RecyclerView* pada file *build.gradle* (modul:app) level modul.
2. Tambhakan objek *RecyclerView* di file layout xml dari *Activity/Fragment*.
3. Definisikan model kelas yang akan digunakan sebagai data *source*.
4. Buat sebuah kelas *adapter* yang inherit ke *RecyclerView.Adapter* dari *ViewHolder* untuk menampilkan tiap elemen data.
5. Definisikan objek *RecyclerView* berikut dengan bentuk yang diinginkan (bisa dalam bentuk *list, grid* atau *staggered*) dan selanjutnya pasang objek *adapter* (*binding*) agar bisa menampilkan koleksi data ke dalam *RecyclerView.*
   1. **Praktikum**
6. Praktikan membuat *project* baru dengan *Empty Activity* dengan nama “Latihan\_RecyclerView”.
7. Tambahkan *dependencies* pada file Gradle Scripts🡪build.gradle(Modul: app) seperti pada

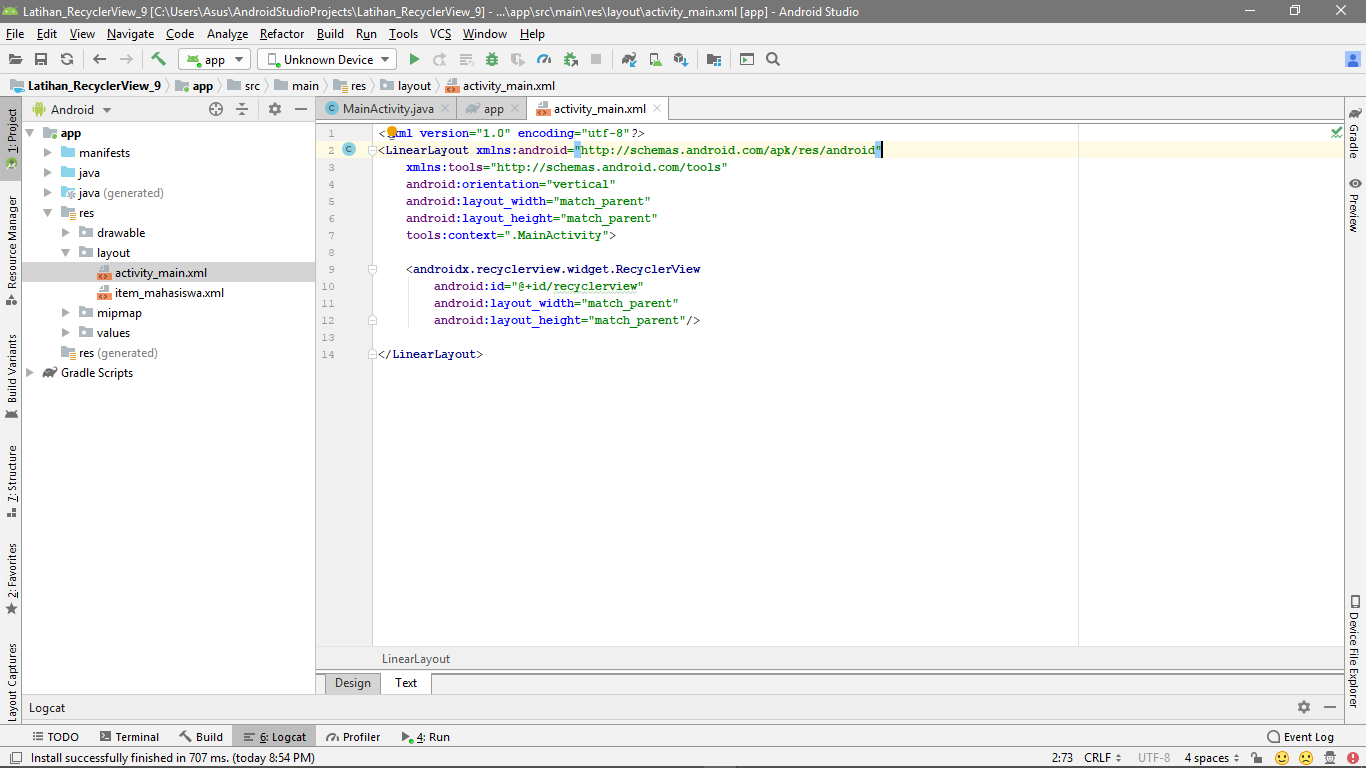
|  |
| --- |
| implementation **'com.android.support:recyclerview-v7:28.0.0'** implementation **'com.android.support:cardview-v7:28.0.0'** |

1. Menambahkan Project Dependency pada widget RecyclerView, kemudian akan mucul tulisan *Sync Now* lalu klik, seperti ditunjukkan pada Gambar 5.4.



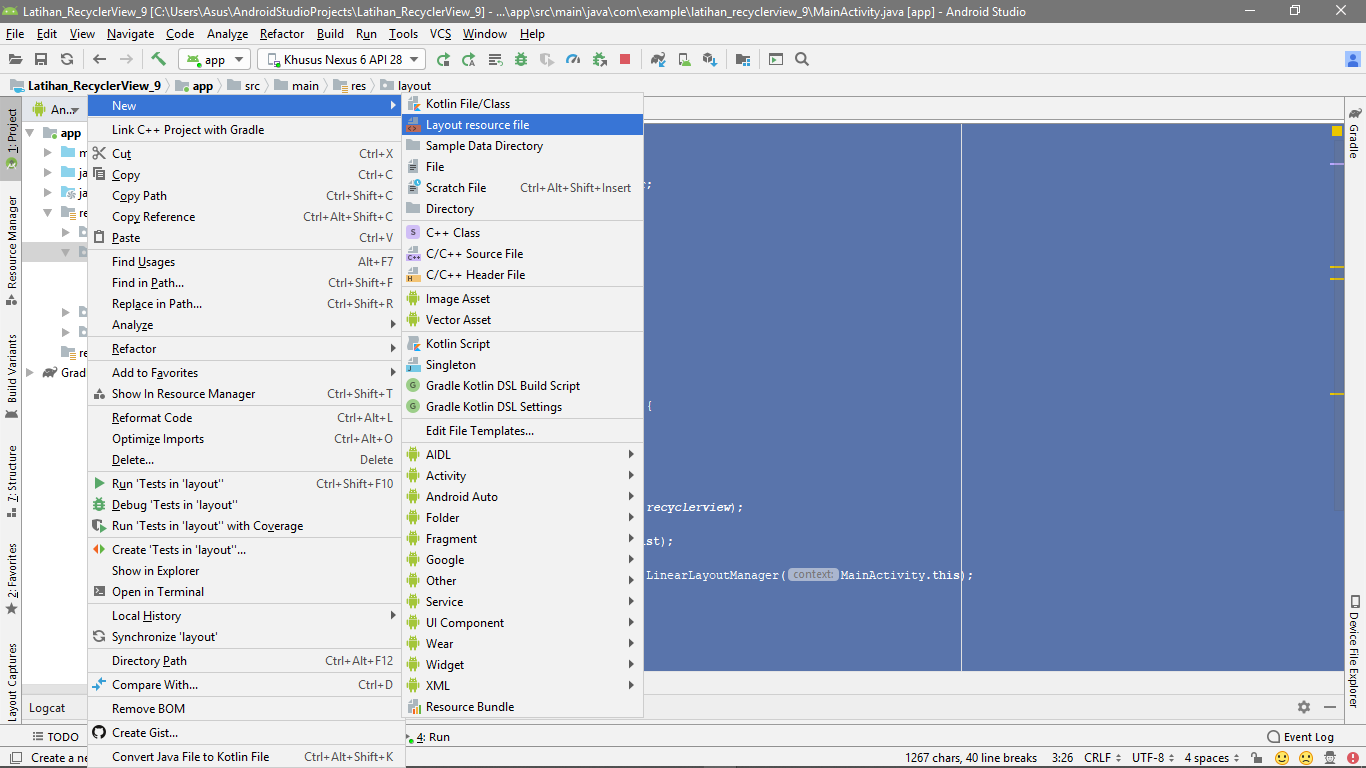
Gambar 5.4 Pop up

1. Tambahkan *widget RecyclerView* pada layout activity\_main.xml, kemudian sesuaikan kode program seperti ditunjukkan Gambar 5.5.

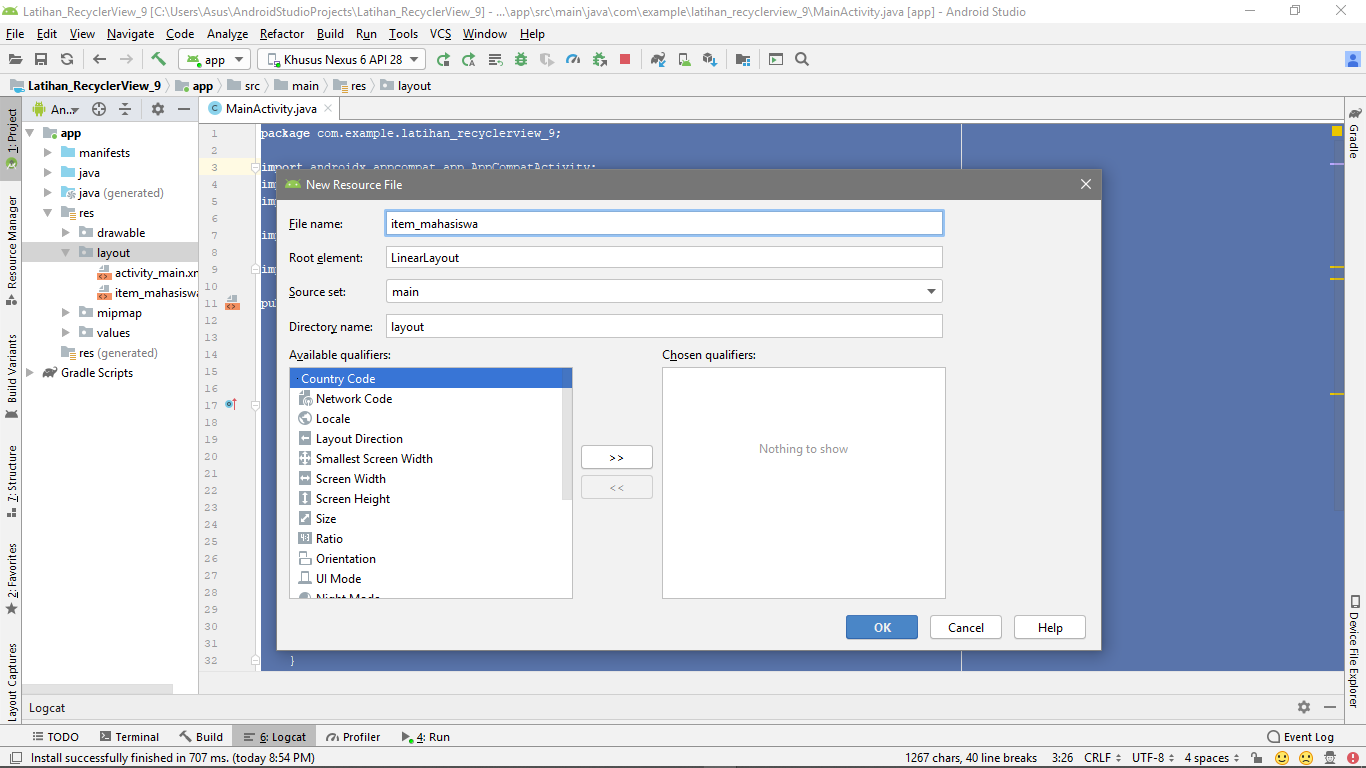


Gambar 5.5 Kode program xml activity\_main.xml

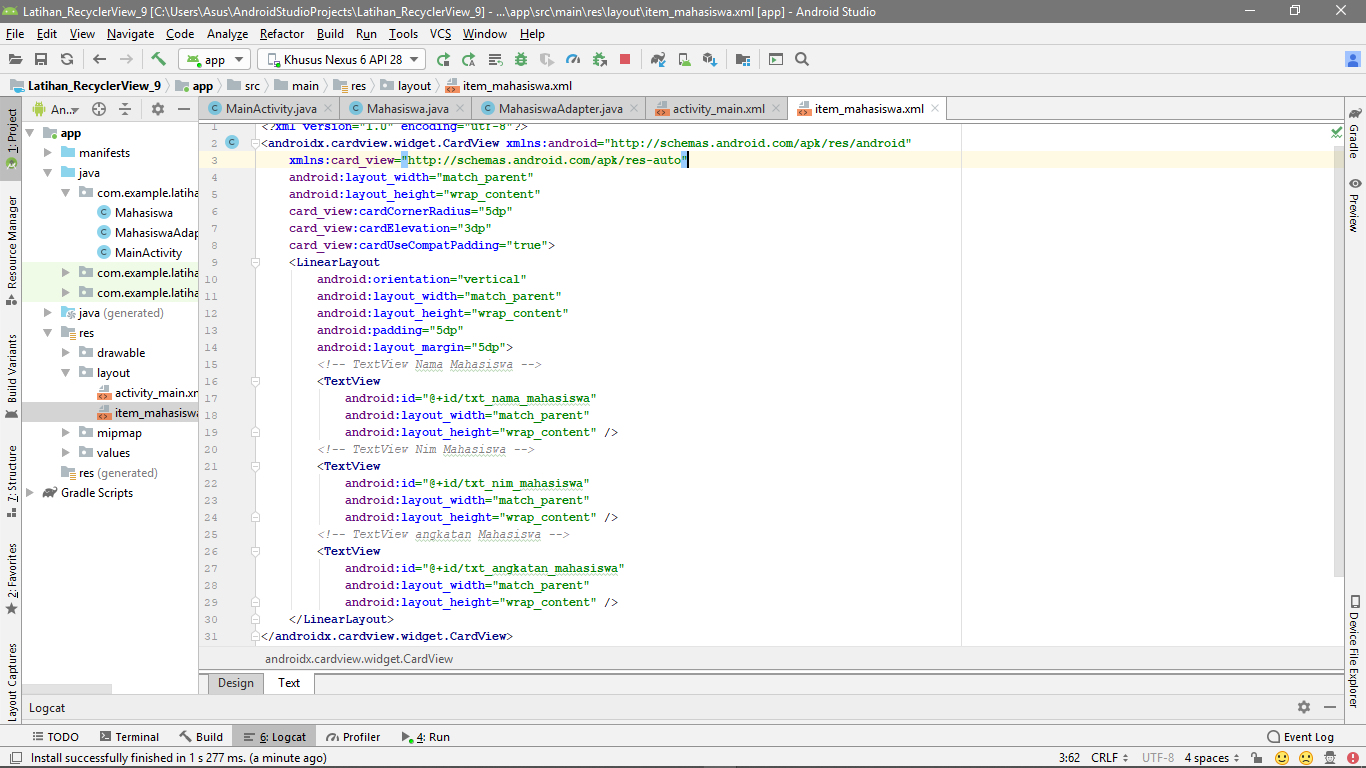
1. Tambahkan loyout baru dengan cara klik kanan pada folder layout > new > Layout resource file.



1. Beri nama layout dengan “item\_mahasiswa” klik OK.



1. Sesuaikan kode program seperti Gambar 5.6.

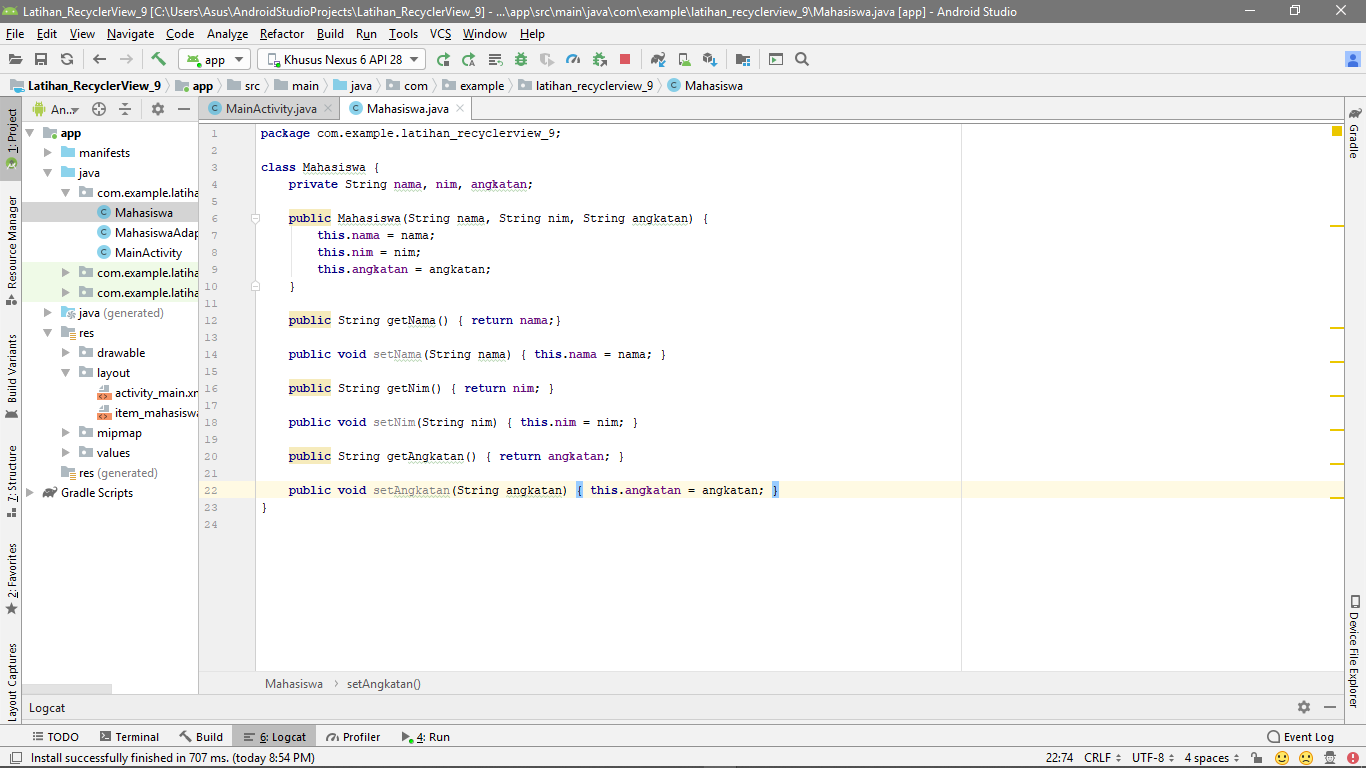


Gambar 5.6 Kode program item\_mahasiswa.xml

1. Sesuaikan kode program untuk MainActivity.java seperti di bawah ini.

|  |
| --- |
| **package** com.example.latihan\_recyclerview\_9;  **import** androidx.appcompat.app.AppCompatActivity; **import** androidx.recyclerview.widget.LinearLayoutManager; **import** androidx.recyclerview.widget.RecyclerView;  **import** android.os.Bundle;  **import** java.util.ArrayList;  **public class** MainActivity **extends** AppCompatActivity {  **private** RecyclerView **recyclerView**;  **private** MahasiswaAdapter **adapter**;  **private** ArrayList<Mahasiswa> **mahasiswaArrayList**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);   addData();   **recyclerView** = (RecyclerView)findViewById(R.id.***recyclerview***);   **adapter** = **new** MahasiswaAdapter(**mahasiswaArrayList**);   RecyclerView.LayoutManager layoutManager = **new** LinearLayoutManager(MainActivity.**this**);   **recyclerView**.setLayoutManager(layoutManager);   **recyclerView**.setAdapter(**adapter**);  }   **private void** addData() {  **mahasiswaArrayList** = **new** ArrayList<>();  **mahasiswaArrayList**.add(**new** Mahasiswa(**"Agung"**, **"1218019"**, **"2012"**));  **mahasiswaArrayList**.add(**new** Mahasiswa(**"Abi"**, **"1118002"**, **"2011"**));  **mahasiswaArrayList**.add(**new** Mahasiswa(**"Yudis"**, **"1118031"**, **"2011"**));  } } |

1. Praktikan membuat class baru dengan nama “Mahasiswa.java”, kemudian sesuaikan kode program seperti Gambar 5.7.

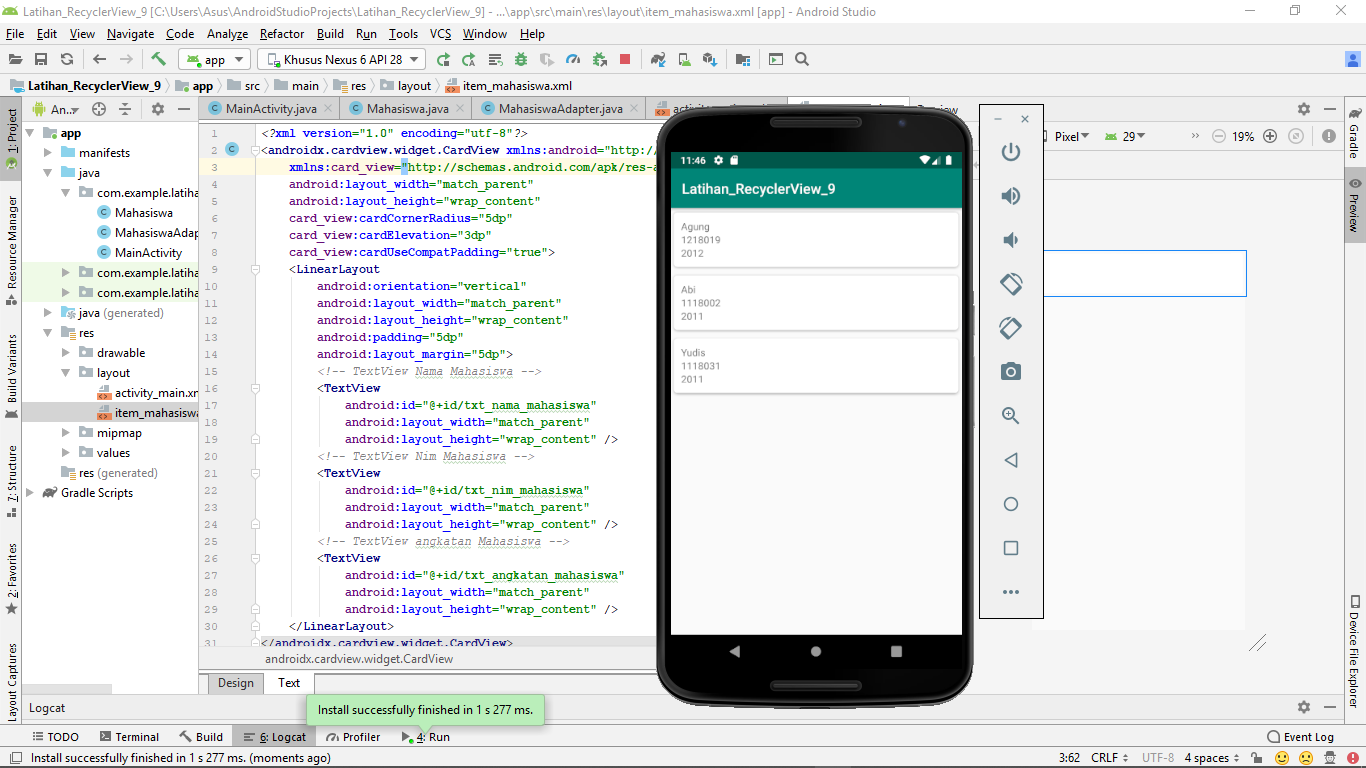


Gambar 5.7 Sintak program Mahasiswa.java

1. Praktikan membuat class baru dengan nama MahasiswaAdapter.java, kemudian sesuaikan kode program di bawah ini.

|  |
| --- |
| **package** com.example.latihan\_recyclerview\_9;  **import** android.view.LayoutInflater; **import** android.view.View; **import** android.view.ViewGroup; **import** android.widget.TextView;  **import** androidx.annotation.NonNull; **import** androidx.recyclerview.widget.RecyclerView;  **import** java.util.ArrayList;  **class** MahasiswaAdapter **extends** RecyclerView.Adapter<MahasiswaAdapter.MahasiswaViewHolder> {   **private** ArrayList<Mahasiswa> **dataList**;   **public** MahasiswaAdapter(ArrayList<Mahasiswa> dataList) {  **this**.**dataList** = dataList;  }   @NonNull  @Override  **public** MahasiswaAdapter.MahasiswaViewHolder onCreateViewHolder(@NonNull ViewGroup parent, **int** viewType) {  LayoutInflater layoutInflater = LayoutInflater.*from*(parent.getContext());  View view = layoutInflater.inflate(R.layout.***item\_mahasiswa***, parent, **false**);  **return new** MahasiswaViewHolder(view);  }   **public void** onBindViewHolder(MahasiswaViewHolder holder, **int** position) {  holder.**txtNama**.setText(**dataList**.get(position).getNama());  holder.**txtNpm**.setText(**dataList**.get(position).getNim());  holder.**txtNoHp**.setText(**dataList**.get(position).getAngkatan());  }   @Override  **public int** getItemCount() {  **return** (**dataList** != **null**) ? **dataList**.size() : 0;  }   **public class** MahasiswaViewHolder **extends** RecyclerView.ViewHolder{  **private** TextView **txtNama**, **txtNpm**, **txtNoHp**;   **public** MahasiswaViewHolder(View itemView) {  **super**(itemView);  **txtNama** = (TextView) itemView.findViewById(R.id.***txt\_nama\_mahasiswa***);  **txtNpm** = (TextView) itemView.findViewById(R.id.***txt\_nim\_mahasiswa***);  **txtNoHp** = (TextView) itemView.findViewById(R.id.***txt\_angkatan\_mahasiswa***);  }  } } |

1. Tampilan aplikasi.



* 1. **Tugas**

Menambahkan gambar pada setiap item yang akan di tampilkan pada RecyclerView.