

## **CRUD Firebase Android Studio**

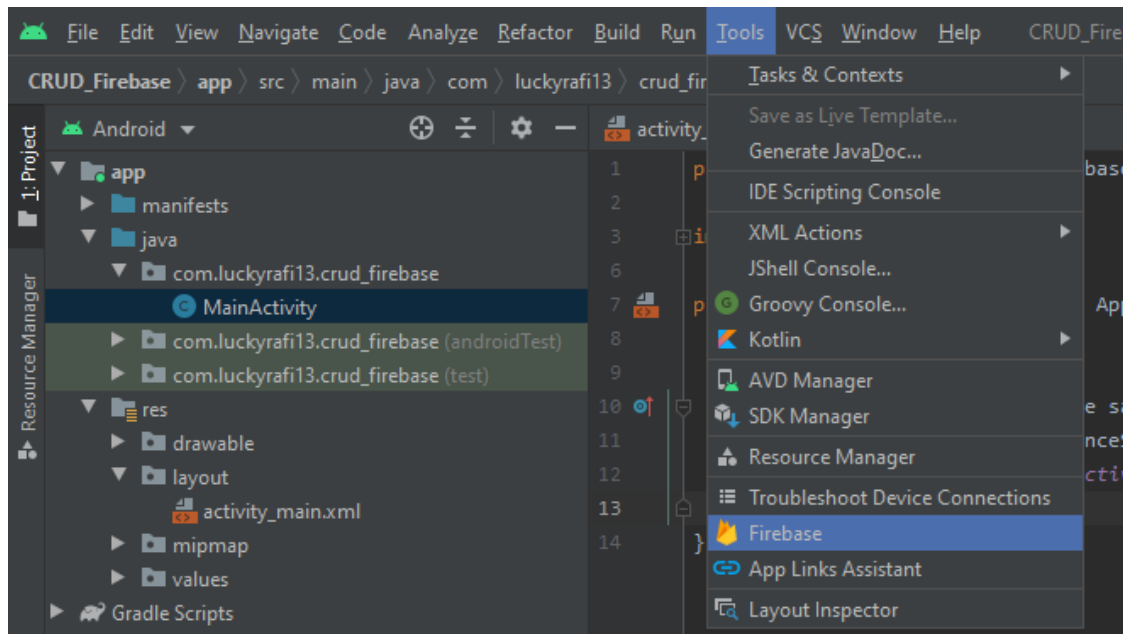
Pada Pembuatan Aplikasi CRUD (Create Read Update Delete) teman teman harus sudah mempersiapkan satu project dan project tersebut sudah terkoneksi ke firebase, untuk penjelasan koneksi ke Firebase silahkan teman teman bisa mengikuti modul yang sebelumnya, pada pembuatan aplikasi ini teman teman akan melakukan beberapa tahapan yaitu:

1. Setting Realtime Database Firebase
2. Membuat Fungsi Create
3. Membuat Fungsi Read
4. Membuat Fungsi Update
5. Membuat Fungsi Delete

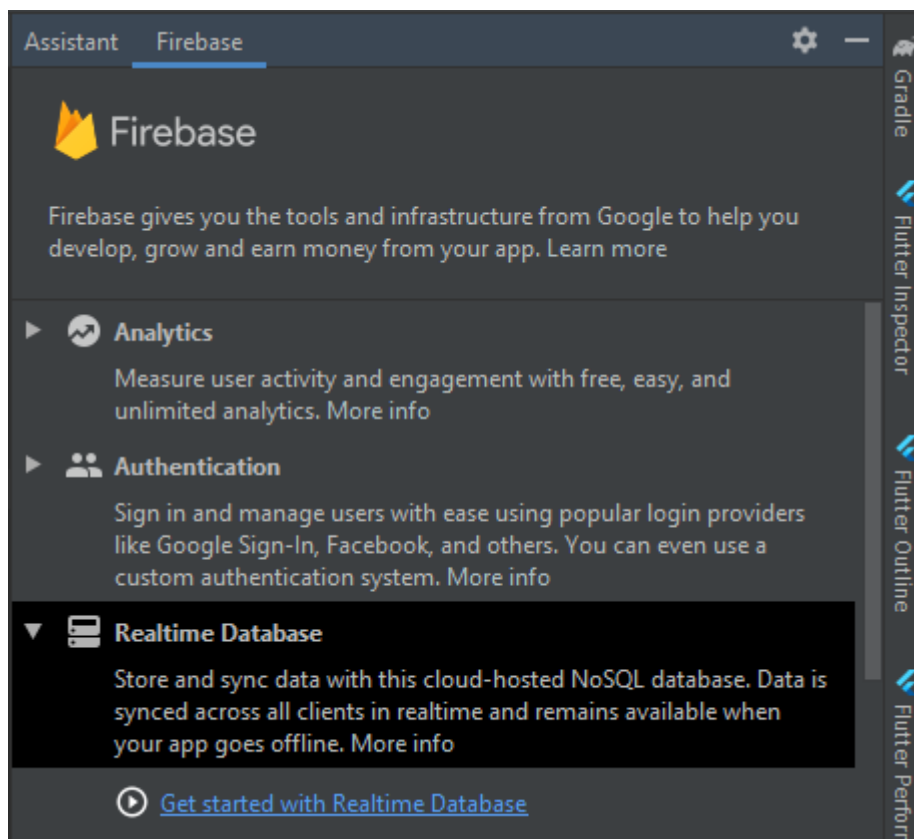
Setelah mempersiapkan semuanya teman teman bisa langsung masuk ke step selanjutnya yaitu membuat Realtime Database.

## Setting Realtime Database

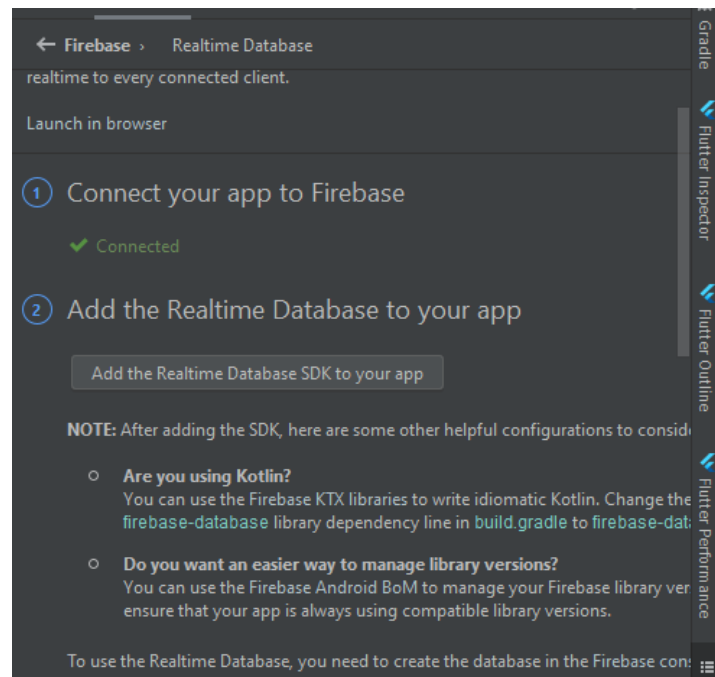
1. Silahkan teman masuk ke bagian **Tools**, lalu pilih **Firebase**



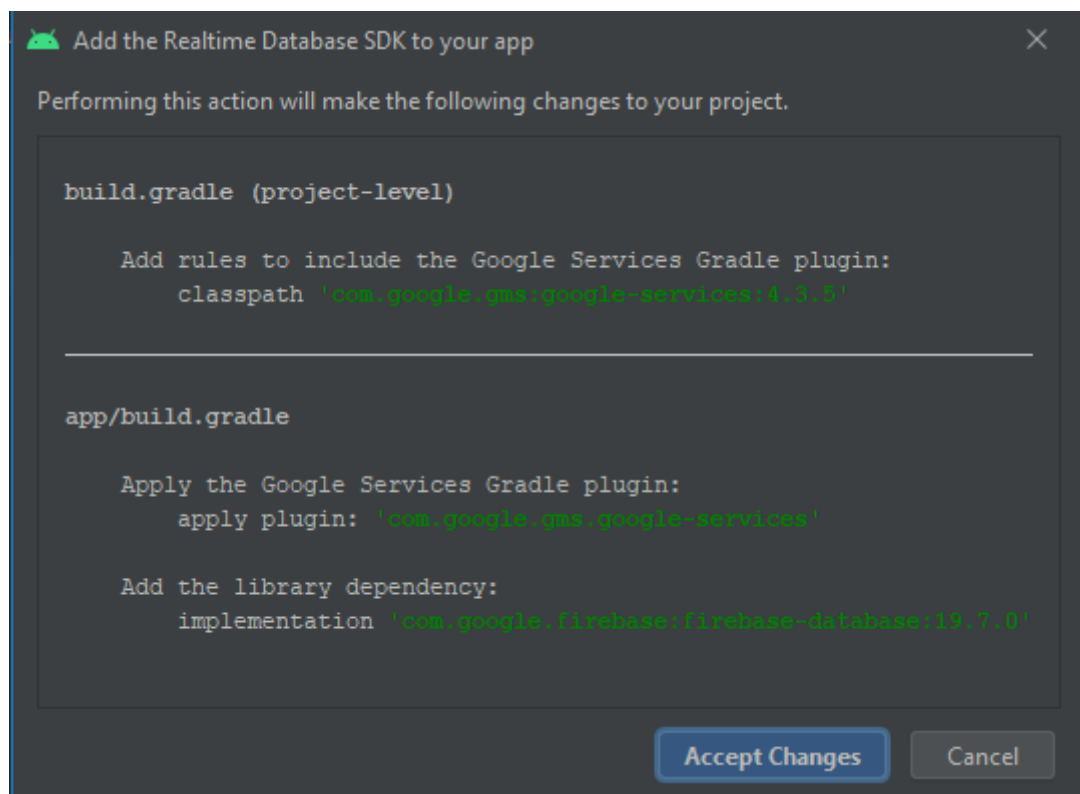
2. Pada jendela disebelah firebase di sebelah kanan silahkan pilih, **Realtime Database** kemudian selanjutnya pilih, **Get Started with Realtime Database**



3. Pastikan bahwa project teman teman sudah terkoneksi ke Firebase dengan ditandai adanya tanda **Connected** berwarna Hijau, kemudian teman teman pilih **Add the Realtime Database SDK to your app**

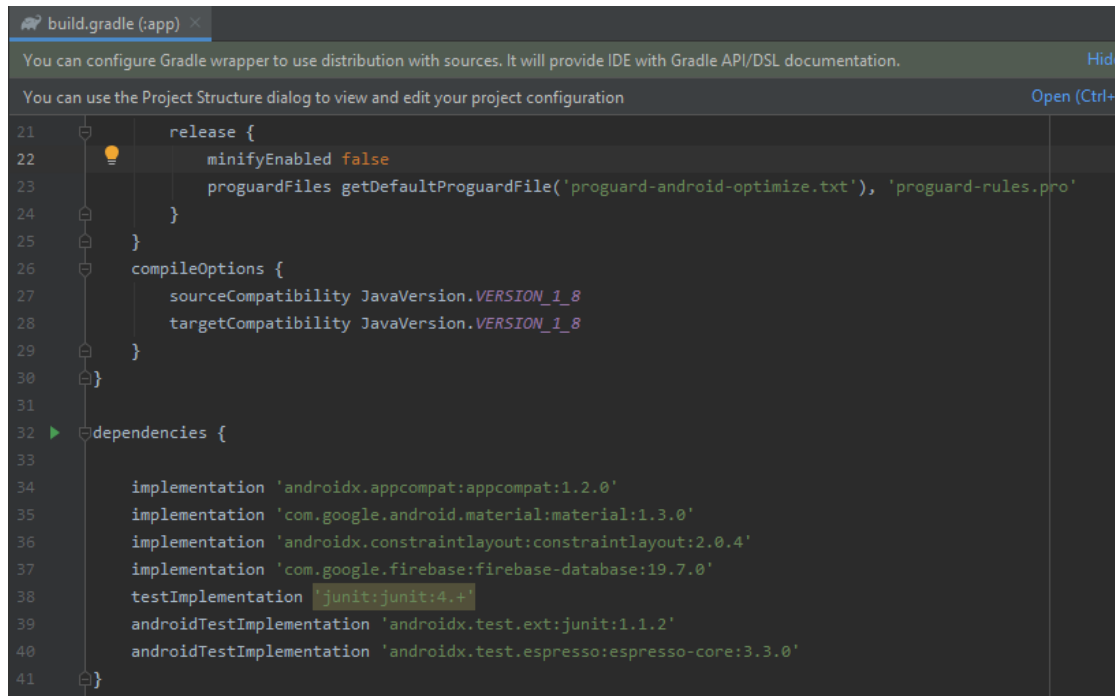


4. Pada Bagian ini silahkan teman teman pilih **Accept Changes**, dan SDK Realtime Database akan otomatis ditambahkan pada project teman teman, dan proses Build Gradle akan berjalan, tunggu sampai proses Build Gradle selesai.

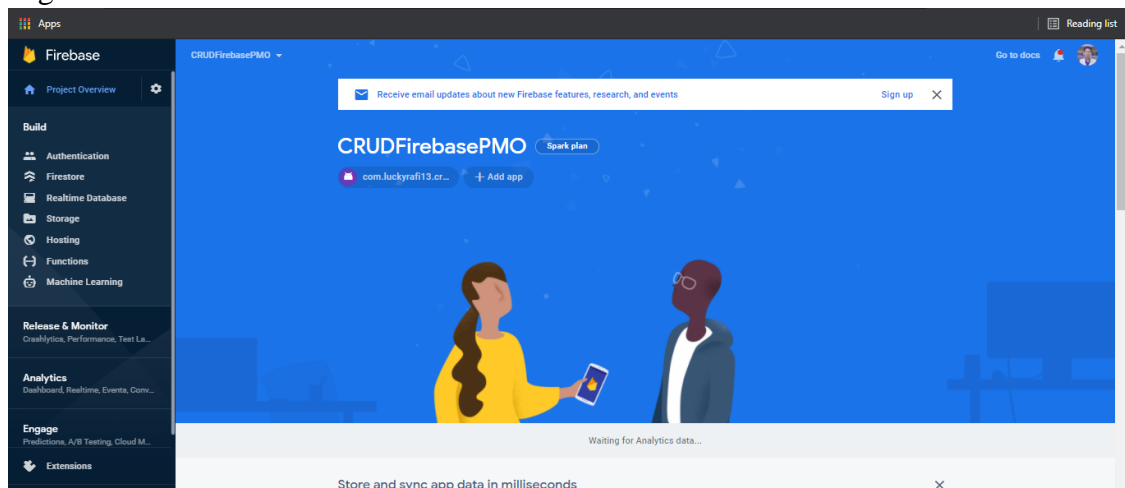


5. Teman teman bisa melakukan pengecekan pada **build.gradle(app)**, pada program dibawah telah ditambahkan

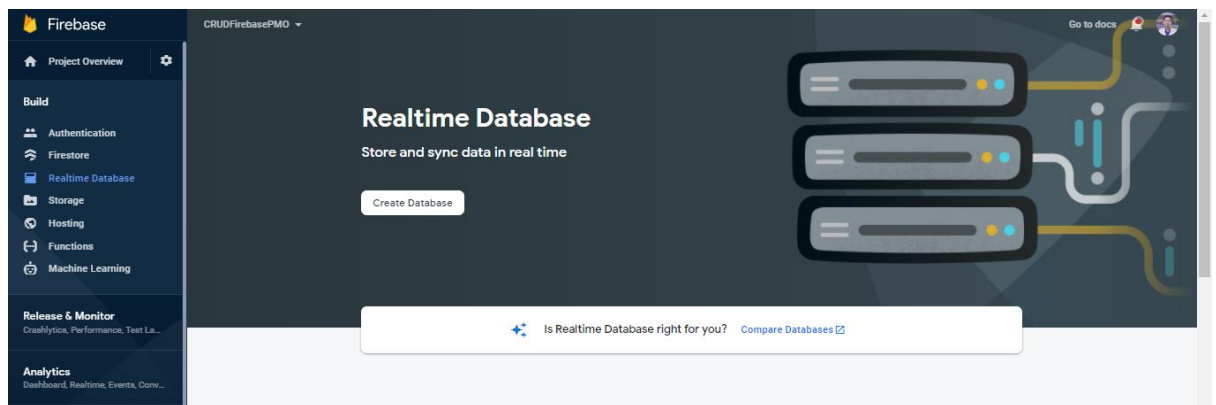
**implementation 'com.google.firebase:firebase-database:19.7.0'**



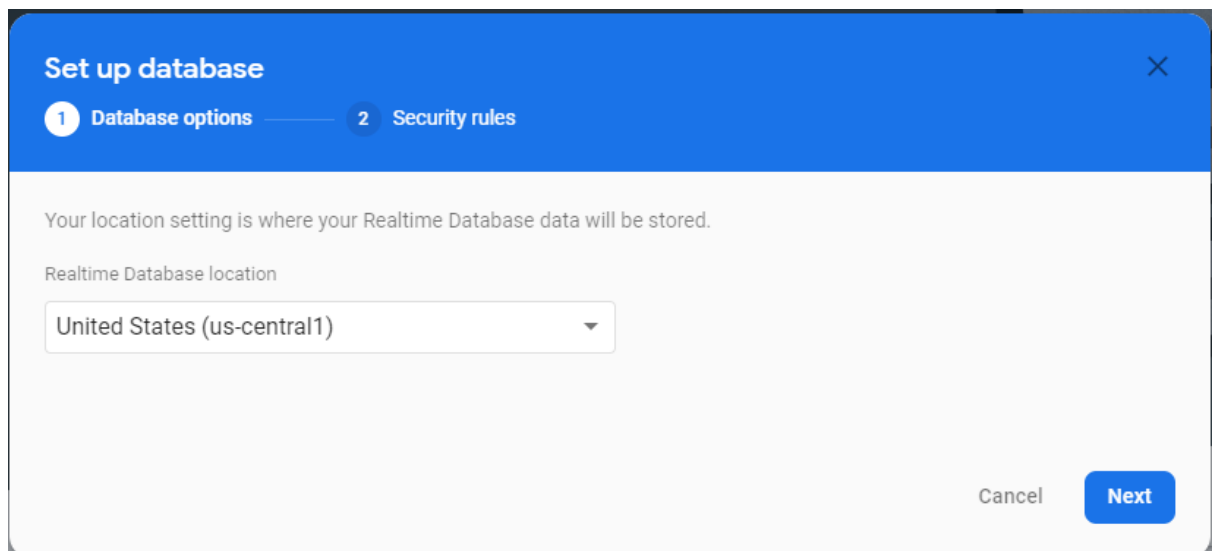
6. Selanjutnya silahkan teman teman buka **console.firebase.google.com**, lalu pilih pada bagian **Realtime Database**



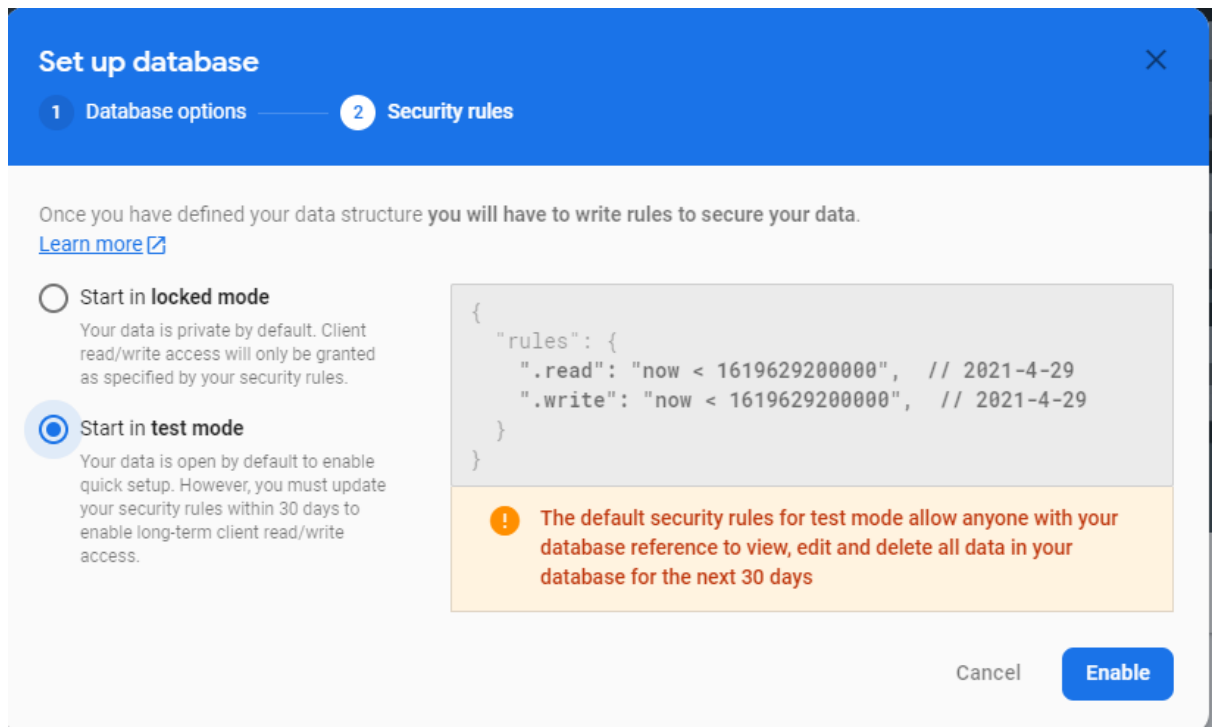
7. Pada halaman Realtime Database klik tombol **Create Database**



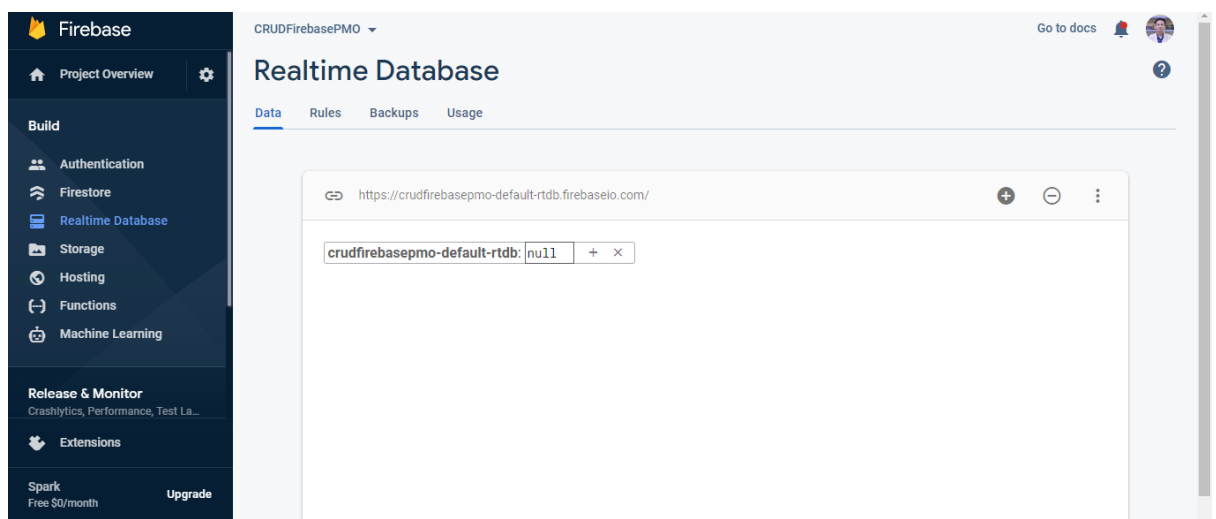
8. Pilih Realtime Database Connection dengan **United States (us-central)** atau teman teman bisa bebas memilih Realtime Database Connection sesuai kebutuhan, lalu klik **Next**



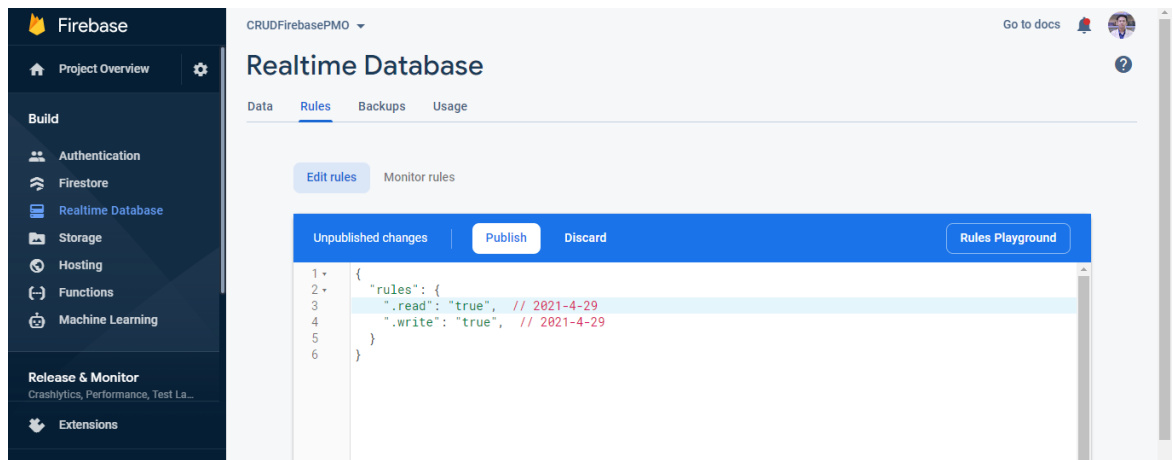
9. Pada Menu **Set up Database**, pilih Radio Button **Start in Test Mode**, pada bagian ini Realtime Database dapat digunakan selama 30 hari kedepan dan setelah itu akan dinonaktifkan, selanjutnya klik tombol **Enable**



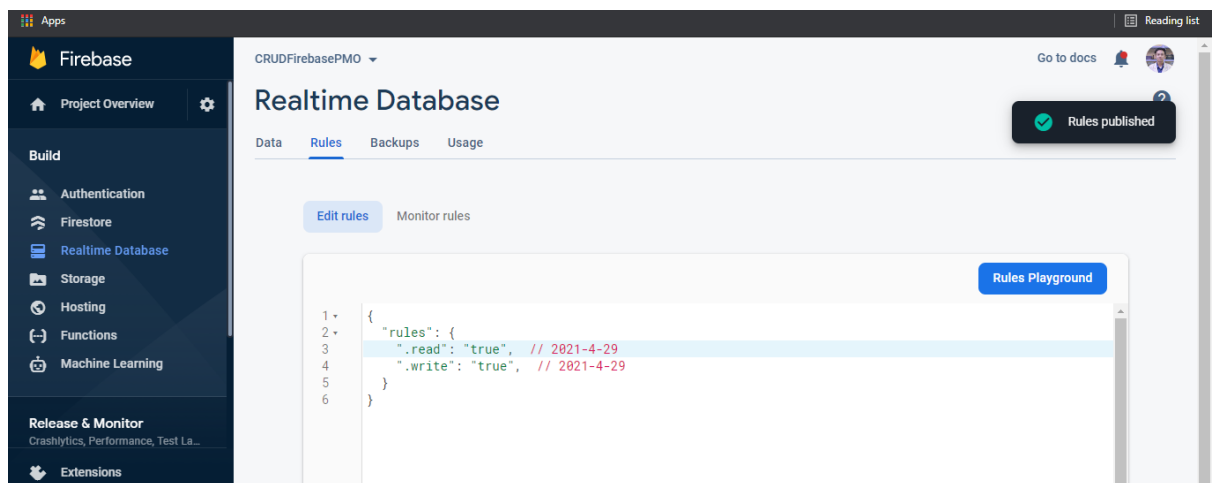
10. Jika sudah muncul tampilan seperti dibawah berarti teman teman sudah berhasil membuat sebuah Realtime Database pada Firebase yang sudah tekoneksi ke project Android teman teman.



11. Masuk pada bagian **Rules**, disini kita akan mengatur supaya Realtime Database dapat digunakan lebih dari 30 hari dengan cara mengubah aturan atau rules yang ada menjadi true untuk Read and Write seperti gambar dibawah.



12. Setelah mengubah rules tersebut silahkan pilih tombol **Publish**, dan akan muncul keterangan **Rules published**, itu tandanya aturan yang sudah diubah tersebut telah diterapkan pada Realtime Database teman teman, dan untuk Setting Realtime Database telah selesai. Sekarang teman teman bisa lanjut ke tahapan yang selanjutnya.

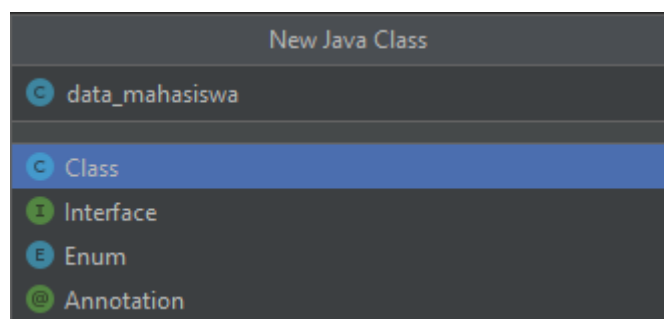
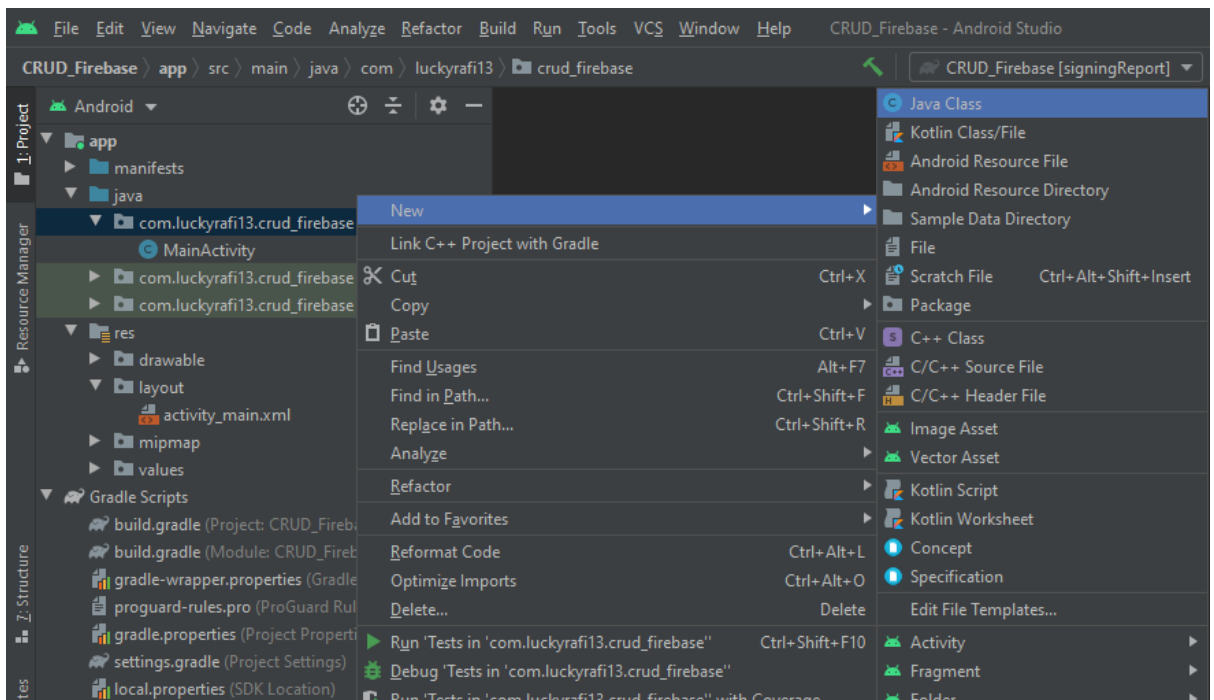


## Membuat Fungsi Create

Aplikasi yang akan kita buat yaitu, aplikasi Data Mahasiswa, jadi didalam aplikasi tersebut kita berperan sebagai Admin yang mengelola pendataan Mahasiswa, fungsi pertama yang akan kita buat yaitu fungsi Create, yang memungkinkan Admin/User dapat meyimpan inputannya pada Database. Data yang akan kita Inputkan berupa NIM, Nama dan Jurusan.

1. Selanjutnya kita akan membuat class model, bernama **data\_mahasiswa**, class ini digunakan untuk menentukan atribut (seperti **NIM, Nama, Jurusan**) untuk masing-masing data mahasiswa yang akan disimpan.

Masing-masing atribut menggunakan method Getter dan Setter, yang nantinya digunakan untuk mengambil dan meyimpan data dari Database. Dan juga terdapat variable key, yang nantinya digunakan untuk proses Edit/Update dan Delete





2. Selanjutnya silahkan teman teman buat deklarasi variabel seperti dibawah ini

```
data_mahasiswa.java x
1 package com.luckyrafi13.crud_firebase;
2
3 public class data_mahasiswa {
4
5     //Deklarasi Variable
6     private String nim;
7     private String nama;
8     private String jurusan;
9     private String key;
10
11 }
12 |
```

3. Setelah membuat deklarasi variabel diatas selanjutnya teman teman membuat getter dan setter untuk masing masing variabel diatas dengan cara, klik kanan lalu pilih **generate** lalu pilih **Getter and Setter**

```
public class data_mahasiswa {

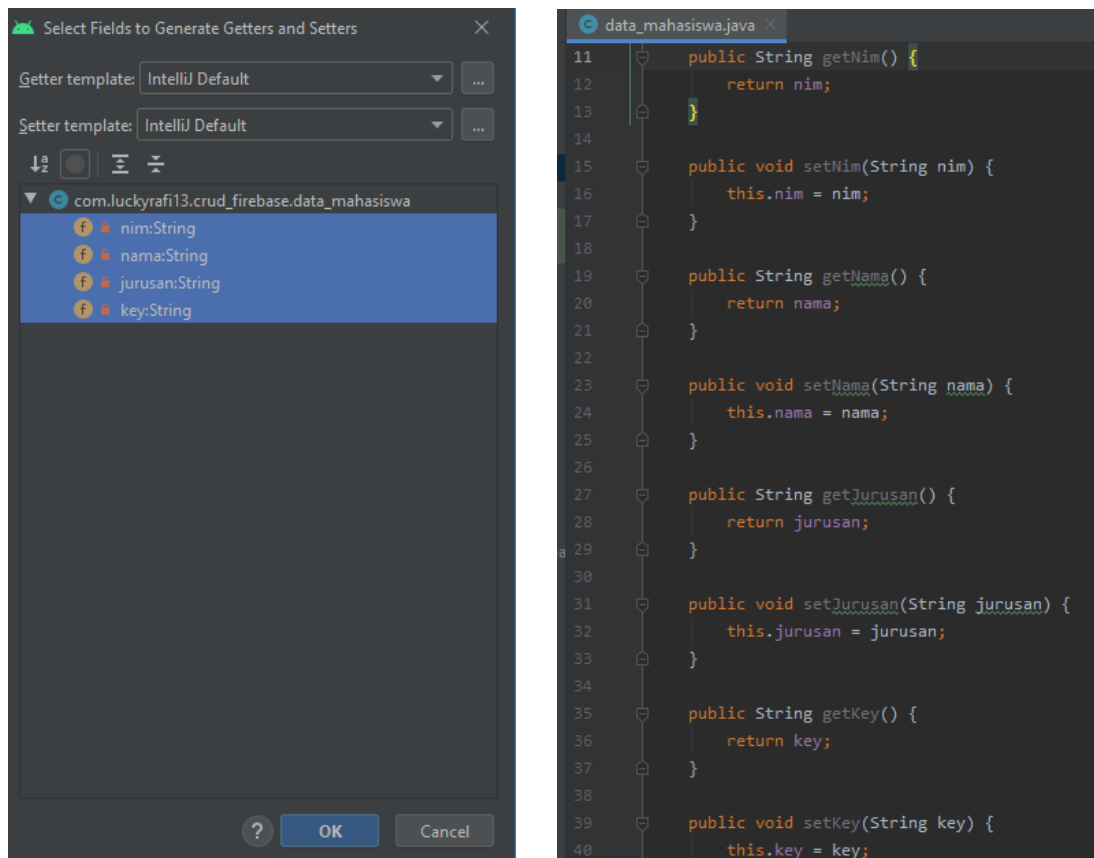
    //Deklarasi Variable
    private String nim;
    private String nama;
    private String jurusan;
    private String key;

}
```

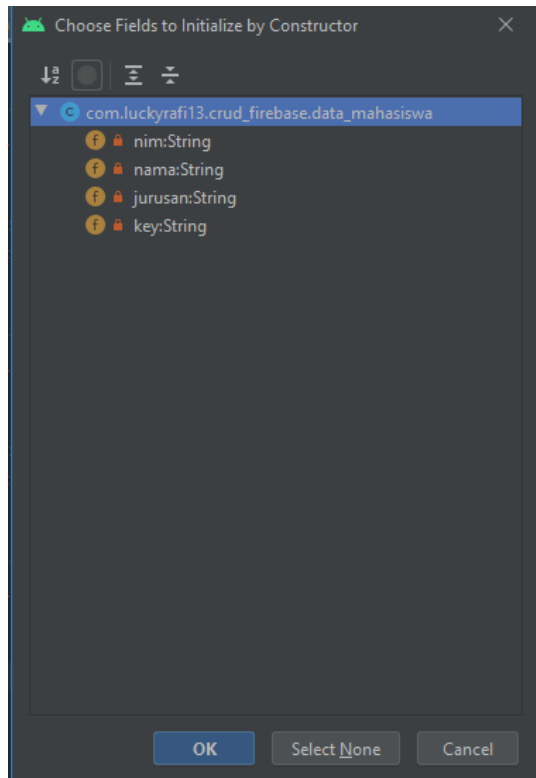
Show Context Actions	Alt+Enter
Copy Reference	Ctrl+Alt+Shift+C
Paste	Ctrl+V
Paste from History...	Ctrl+Shift+V
Paste without Formatting	Ctrl+Alt+Shift+V
Column Selection Mode	Alt+Shift+Insert
Find Usages	Alt+F7
Find Sample Code	Alt+F8
Refactor	
Folding	
Analyze	
Go To	
Generate...	Alt+Insert
Show in Explorer	
File Path	Ctrl+Alt+F12

Generate
Constructor
Getter
Setter
<b>Getter and Setter</b>
equals() and hashCode()
toString()
Override Methods... Ctrl+O
Delegate Methods...
Test...
Copyright

4. Pada menu Select Fields to Generate Getter and Setter, blok semua variabel yang ada seperti gambar dibawah lalu pilih **OK**, maka otomatis akan dibuatkan getter dan setter untuk masing masing variabel yang dipilih

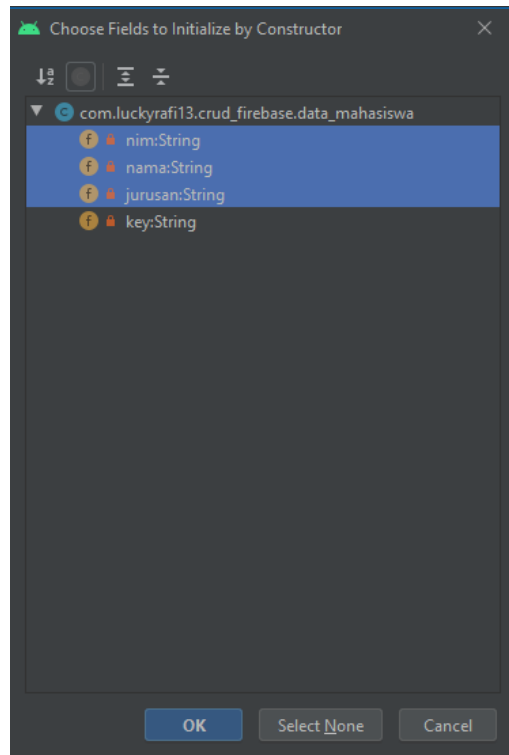


5. Lakukan Hal yang sama untuk membuat Contruktor Kosong yaitu dengan cara **Klik kanan**, pilih **Generate** lalu **Contractor**, pertama buatlah sebuah Constructor kosong dengan memilih package project Android teman teman seperti dibawah lalu pilih OK, nantinya contructor kosong akan dibuatkan secara otomatis



```
public data_mahasiswa() {  
}
```

6. Lakukan Hal yang sama untuk membuat Contruktor berisi ketiga variabel yang sudah dibuat yaitu (**nim, nama jurusan**) yaitu dengan cara **Klik kanan**, pilih **Generate** lalu **Contructor**, pertama buatlah sebuah Constructor dengan isi variabel dengan memilih ketiga variabel nya seperti dibawah lalu pilih OK, nantinya contructor akan dibuatkan secara otomatis



```
public data_mahasiswa(String nim, String nama, String jurusan) {  
    this.nim = nim;  
    this.nama = nama;  
    this.jurusan = jurusan;  
}
```

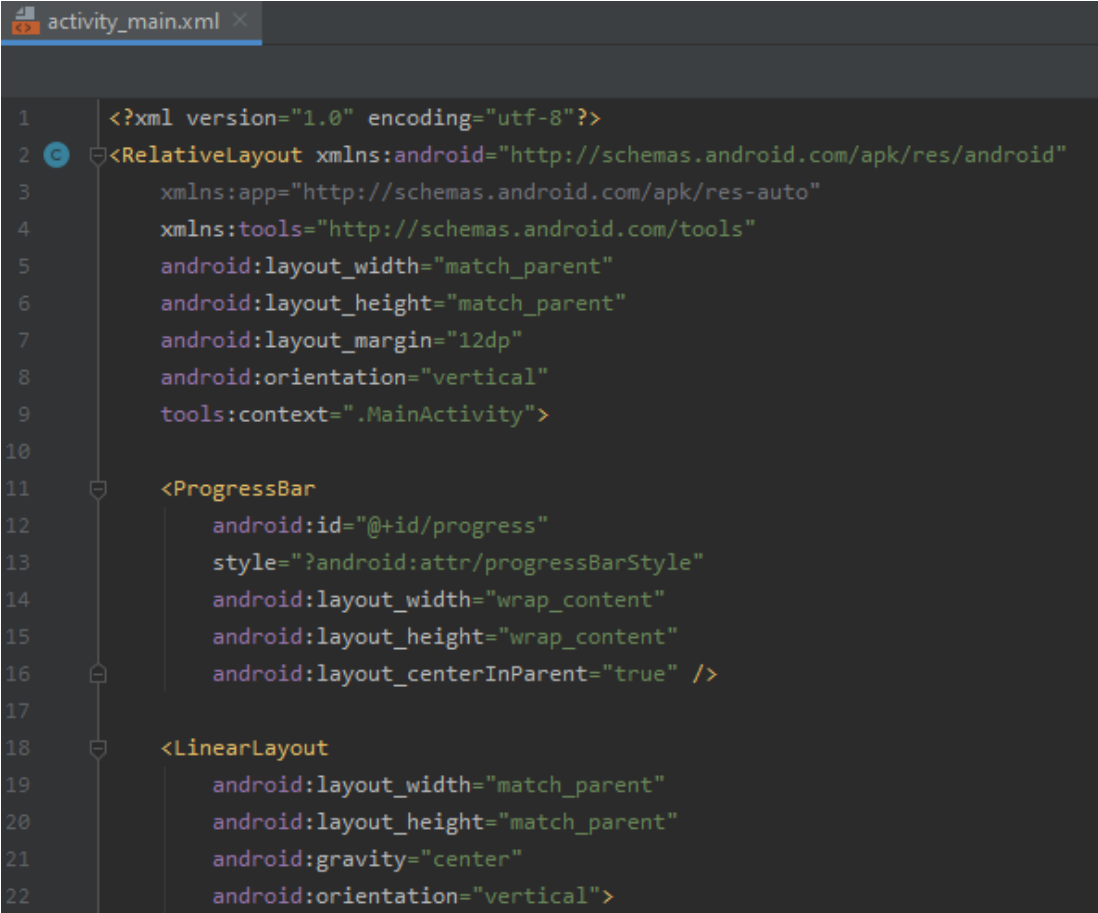
Sehingga Struktur lengkapnya adalah seperti gambar dibawah



```
1 package com.luckyrafi13.crud_firebase;
2
3 public class data_mahasiswa {
4
5     //Deklarasi Variable
6     private String nim;
7     private String nama;
8     private String jurusan;
9     private String key;
10
11     //Setter dan Getter
12     public String getNim() {
13         return nim;
14     }
15
16     public void setNim(String nim) {
17         this.nim = nim;
18     }
19
20     public String getNama() {
21         return nama;
22     }
23
24     public void setNama(String nama) {
25         this.nama = nama;
26     }
27
28     public String getJurusan() {
29         return jurusan;
30     }
31 }
```

```
data_mahasiswa.java x
26     }
27
28     public String getJurusan() {
29         return jurusan;
30     }
31
32     public void setJurusan(String jurusan) {
33         this.jurusan = jurusan;
34     }
35
36     public String getKey() {
37         return key;
38     }
39
40     public void setKey(String key) {
41         this.key = key;
42     }
43
44     // Konstruktur kosong untuk membaca data snapshot
45     public data_mahasiswa() {
46     }
47
48     //Konstruktur dengan beberapa parameter, untuk mendapatkan Input Data dari User
49     public data_mahasiswa(String nim, String nama, String jurusan) {
50         this.nim = nim;
51         this.nama = nama;
52         this.jurusan = jurusan;
53     }
54 }
55
```

7. Pada **activity\_main.xml**, disini kita akan membuat layout untuk tampilan User Interfacenya, komponen yang digunakan seperti **EditText** untuk Input data, dan Beberapa **Button**. Serta **ProgressBar**.

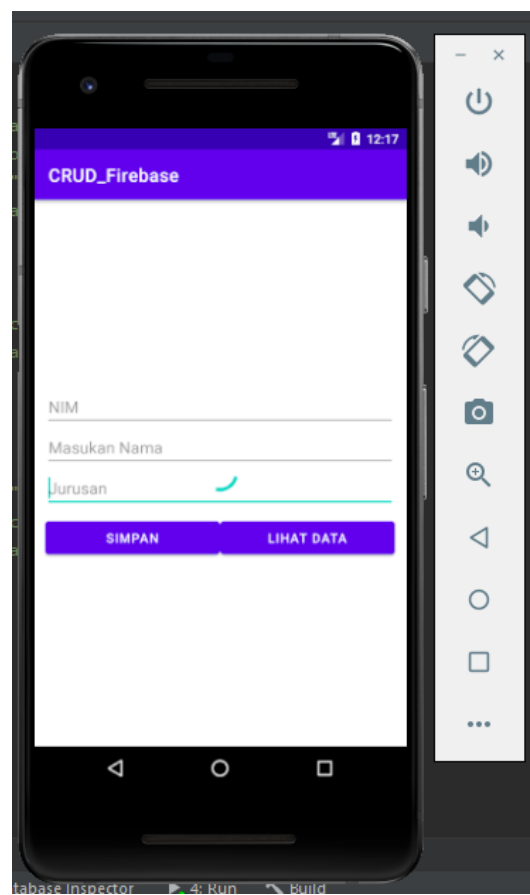
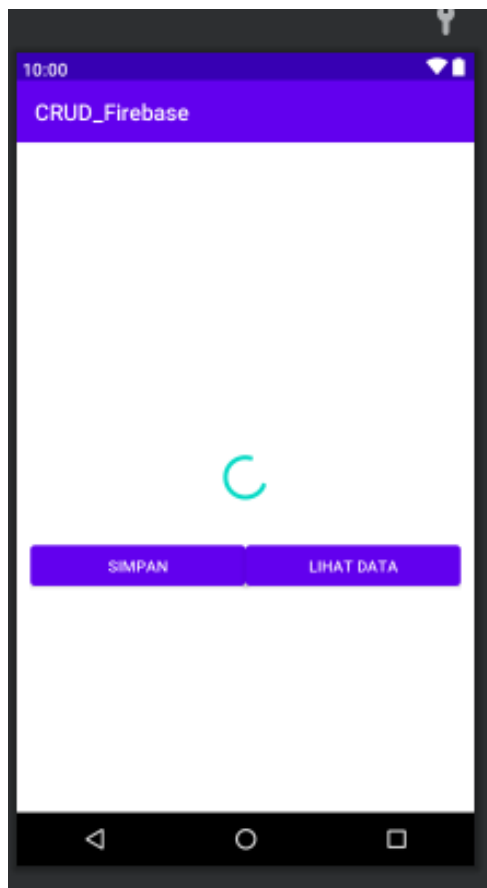


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:layout_margin="12dp"
8      android:orientation="vertical"
9      tools:context=".MainActivity">
10
11      <ProgressBar
12          android:id="@+id/progress"
13          style="?android:attr/progressBarStyle"
14          android:layout_width="wrap_content"
15          android:layout_height="wrap_content"
16          android:layout_centerInParent="true" />
17
18      <LinearLayout
19          android:layout_width="match_parent"
20          android:layout_height="match_parent"
21          android:gravity="center"
22          android:orientation="vertical">
```

```
activity_main.xml x
23
24 <EditText
25     android:id="@+id/nim"
26     android:layout_width="match_parent"
27     android:layout_height="wrap_content"
28     android:ems="10"
29     android:hint="NIM"
30     android:inputType="textPersonName" />
31
32 <EditText
33     android:id="@+id/nama"
34     android:layout_width="match_parent"
35     android:layout_height="wrap_content"
36     android:ems="10"
37     android:hint="Masukan Nama"
38     android:inputType="textPersonName" />
39
40 <EditText
41     android:id="@+id/jurusan"
42     android:layout_width="match_parent"
43     android:layout_height="wrap_content"
44     android:ems="10"
45     android:hint="Jurusan"
46     android:inputType="textPersonName" />
47
```



```
activity_main.xml x
47
48 <LinearLayout
49     android:layout_width="match_parent"
50     android:layout_height="wrap_content"
51     android:layout_marginTop="8dp"
52     android:orientation="horizontal">
53
54     <Button
55         android:id="@+id/save"
56         android:layout_width="match_parent"
57         android:layout_height="wrap_content"
58         android:layout_weight="1"
59         android:text="Simpan" />
60
61     <Button
62         android:id="@+id/showdata"
63         android:layout_width="match_parent"
64         android:layout_height="wrap_content"
65         android:layout_weight="1"
66         android:text="Lihat Data" />
67
68 </LinearLayout>
69
70 </LinearLayout>
71
72 </RelativeLayout>
```



8. Buka file **MainActivity.java**, pada source code tersebut kita akan melakukan penaturan agar dapat melakukan Create Data pada Realtime Database

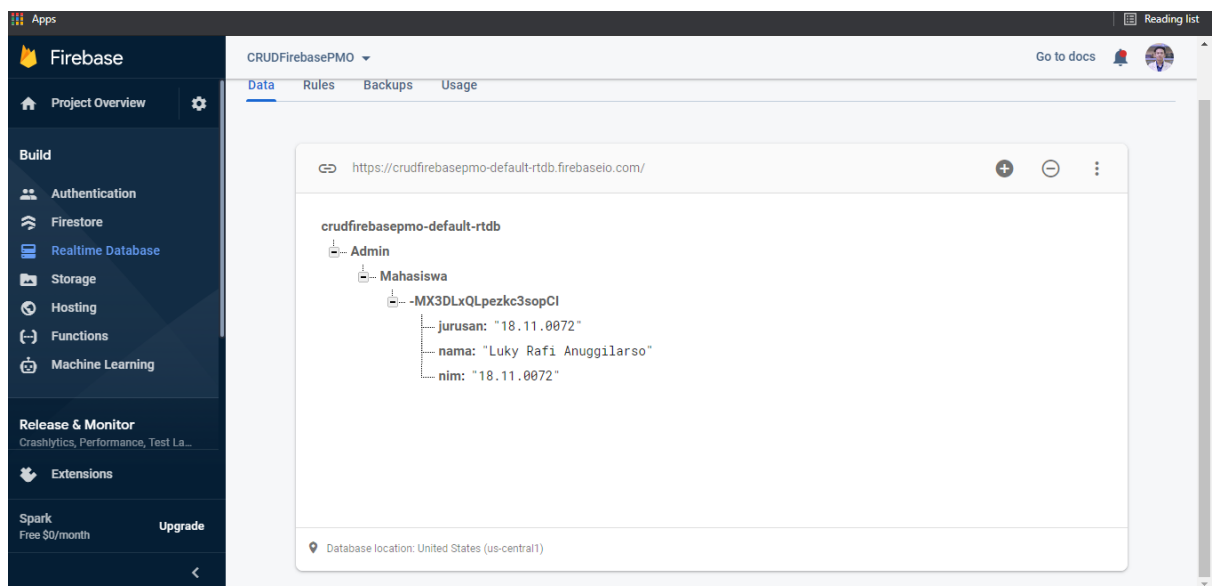
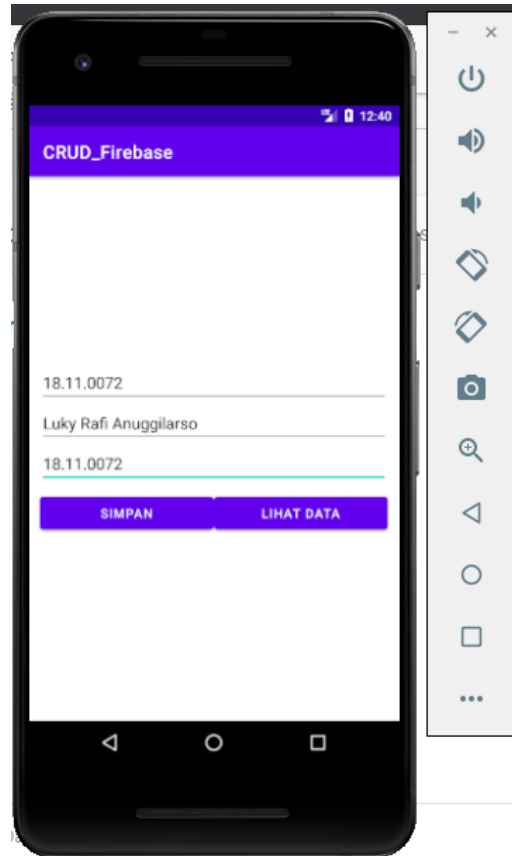
```
MainActivity.java x
1  package com.luckyrafi13.crud_firebase;
2
3  import ...
19
20  public class MainActivity extends AppCompatActivity {
21
22      //Deklarasi Variable
23      private ProgressBar progressBar;
24      private EditText NIM, Nama, Jurusan;
25      private Button Simpan, ShowData;
26      private String getNIM, getName, getJurusan;
27
28      DatabaseReference getReference;
29
30      @Override
31      protected void onCreate(Bundle savedInstanceState) {
32          super.onCreate(savedInstanceState);
33          setContentView(R.layout.activity_main);
34
35          // Inisialisasi ID (Progressbar)
36          progressBar = findViewById(R.id.progress);
37          progressBar.setVisibility(View.GONE);
38
39          //Inisialisasi ID (Button)
40          Simpan = findViewById(R.id.save);
41          ShowData = findViewById(R.id.showdata);
42
```

```
MainActivity.java x
43      //Inisialisasi ID (EditText)
44      NIM = findViewById(R.id.nim);
45      Nama = findViewById(R.id.nama);
46      Jurusan = findViewById(R.id.jurusan);
47
48      //Mendapatkan Instance dari Database
49      FirebaseDatabase database = FirebaseDatabase.getInstance();
50
51      getReference = database.getReference(); // Mendapatkan Referensi dari Database
52
53      // Membuat Fungsi Tombol Simpan
54      Simpan.setOnClickListener(new View.OnClickListener() {
55          @Override
56          public void onClick(View v) {
57
58              //Menyimpan Data yang diinputkan User kedalam Variable
59              getNIM = NIM.getText().toString();
60              getName = Nama.getText().toString();
61              getJurusan = Jurusan.getText().toString();
62
63              checkUser();
64              progressBar.setVisibility(View.VISIBLE);
65
66          }
67      });
```

```
MainActivity.java X
68
69 ShowData.setOnClickListener(new View.OnClickListener() {
70     @Override
71     public void onClick(View v) {
72         //akan diisi pada Step Selanjutnya
73     }
74 });
75
76 }
77
```

```
MainActivity.java X
77
78 private void checkUser() {
79
80     // Mengecek apakah ada data yang kosong
81     if (isEmpty(getNIM) && isEmpty(getNama) && isEmpty(getJurusan)) {
82         //Jika Ada, maka akan menampilkan pesan singkan seperti berikut ini.
83         Toast.makeText(context: MainActivity.this, text: "Data tidak boleh ada yang kosong", Toast.LENGTH_SHORT).show();
84     } else {
85
86         /*
87         Jika Tidak, maka data dapat diproses dan menyimpannya pada Database
88         Menyimpan data referensi pada Database berdasarkan User ID dari masing-masing Akun
89         */
90
91         getReference.child("Admin").child("Mahasiswa").push()
92             .setValue(new data_mahasiswa(getNIM, getNama, getJurusan))
93             .addOnCompleteListener( activity: this, new OnCompleteListener<Void>() {
94                 @Override
95                 public void onComplete(@NonNull Task<Void> task) {
96                     //Peristiwa ini terjadi saat user berhasil menyimpan datanya kedalam Database
97                     NIM.setText("");
98                     Nama.setText("");
99                     Jurusan.setText("");
100                     Toast.makeText(context: MainActivity.this, text: "Data Tersimpan", Toast.LENGTH_SHORT).show();
101                     progressBar.setVisibility(View.GONE);
102                 }
103             });
104     }
105 }
106 }
```

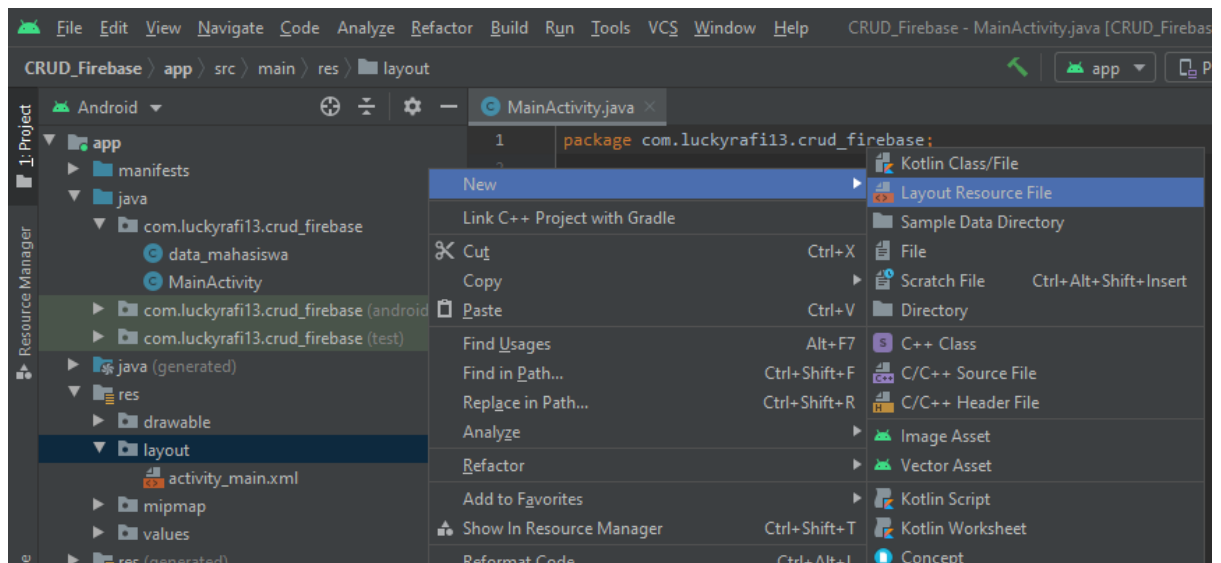
9. Lakukan Pengetesan dengan cara mengisikan data **Edittext** dan tekan tombol Simpan, lalu cek pada bagian **Firestore Console** pada bagian **Realtime Database**, jika sudah sesuai seperti dibawah maka Fungsi Create telah berhasil dan teman teman bisa melanjutkan ke setep selanjutnya yaitu menampilkan data yang sudah kita isikan



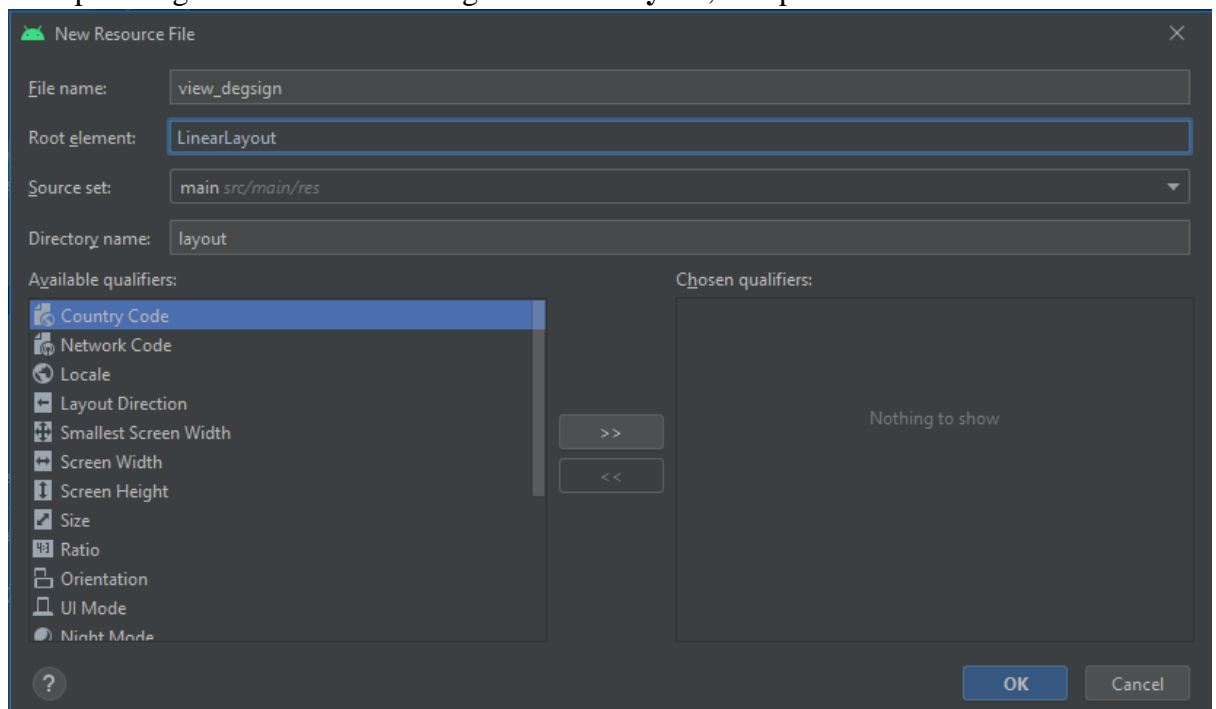
## Membuat Fungsi Read

Pada Fungsi yang akan kita buat ini, kita akan menggunakan RecyclerView untuk menampilkan data data yang ada di dalam Realtime Database. Data yang tersimpan pada Realtime Database Firebase berformat JSON yang meliputi key-value, dan akan kita tampilkan secara Realtime pada RecyclerView, sehingga ketika ada perubahan data pada server akan otomatis ditampilkan pada aplikasi.

1. Buatlah layout yang nantinya akan digunakan untuk membuat tampilan peritem pada RecyclerView, jadi pada RecyclerView ini akan menampilkan berupa list yang berisi item view yang akan kita buat ini. Buat item ini dengan cara klik kanan pada direktori **res/layout**, **new Layout Resource File** dan bernama **view\_design.xml**



2. Atur pada bagian **root element** dengan **LinearLayout**, lalu pilih **OK**



3. Buatlah layout seperti dibawah ini, didalam layout tersebut kita menggunakan 1 buah ImageView dan 3 Buah TextView, untuk asset gambar teman teman bisa menambahkannya sendiri sesuai keinginan. Pada linearLayout yang ber-id "list\_item", nantinya layout tersebut digunakan untuk berinteraksi dengan user, saat klik.

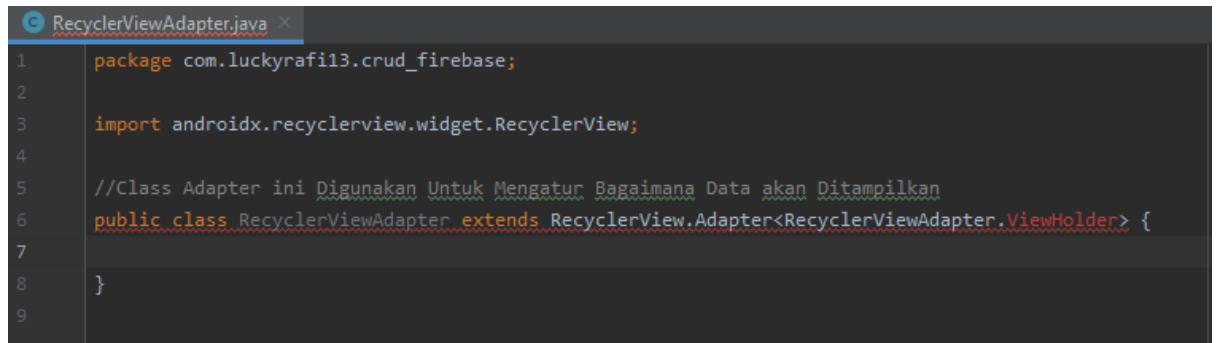


```
view_degsign.xml x
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      android:id="@+id/list_item"
5      android:layout_width="match_parent"
6      android:layout_height="wrap_content"
7      android:background="#FFFFFF"
8      android:gravity="center"
9      android:orientation="horizontal">
10
11      <ImageView
12          android:layout_width="match_parent"
13          android:layout_height="90dp"
14          android:layout_weight="2.5"
15          app:srcCompat="@drawable/graduation" />
16
17      <LinearLayout
18          android:layout_width="match_parent"
19          android:layout_height="wrap_content"
20          android:layout_marginStart="6dp"
21          android:layout_marginLeft="6dp"
22          android:layout_weight="1"
23          android:orientation="vertical">
```

```
view_degsign.xml x
25      <TextView
26          android:id="@+id/nim"
27          android:layout_width="match_parent"
28          android:layout_height="wrap_content"
29          android:layout_marginTop="4dp"
30          android:text="NIM"
31          android:textSize="15sp"
32          android:textStyle="bold" />
33
34      <TextView
35          android:id="@+id/nama"
36          android:layout_width="match_parent"
37          android:layout_height="wrap_content"
38          android:layout_marginTop="4dp"
39          android:text="Nama" />
40
41      <TextView
42          android:id="@+id/jurusan"
43          android:layout_width="match_parent"
44          android:layout_height="wrap_content"
45          android:layout_marginTop="4dp"
46          android:text="Jurusan" />
47
48  </LinearLayout>
49
50 </LinearLayout>
```

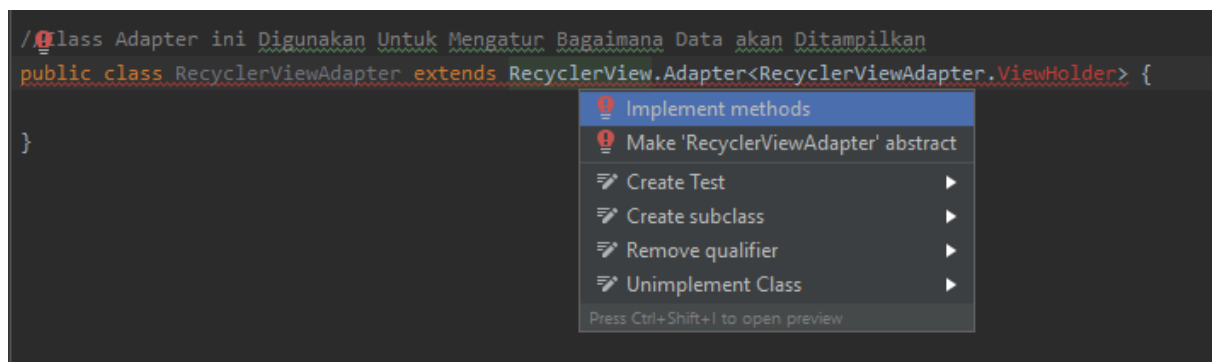


4. Selanjutnya buat class bernama **RecyclerViewAdapter**, class ini gunakan untuk mengatur dan mengolah data yang ingin ditampilkan pada **RecyclerView**. Selanjutnya tambahkan **extends** pada bagian **public class RecyclerViewAdapter** seperti gambar dibawah.



```
1 package com.luckyrafi13.crud_firebase;
2
3 import androidx.recyclerview.widget.RecyclerView;
4
5 //Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
6 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
7
8 }
9
```

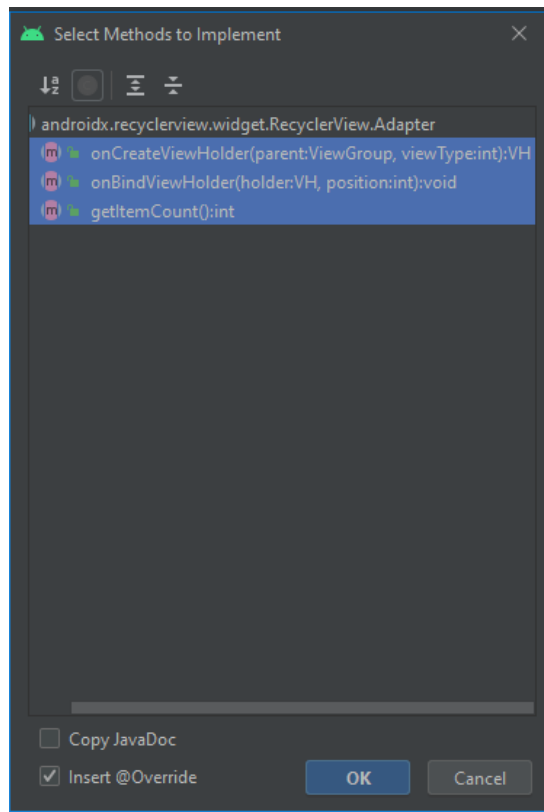
5. Kemudian Tekan **Alt + Enter** pada keyboard untuk menampilkan jendela **Generate**, lalu pilih **Implement Methods**,



```
//Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
}
}
```

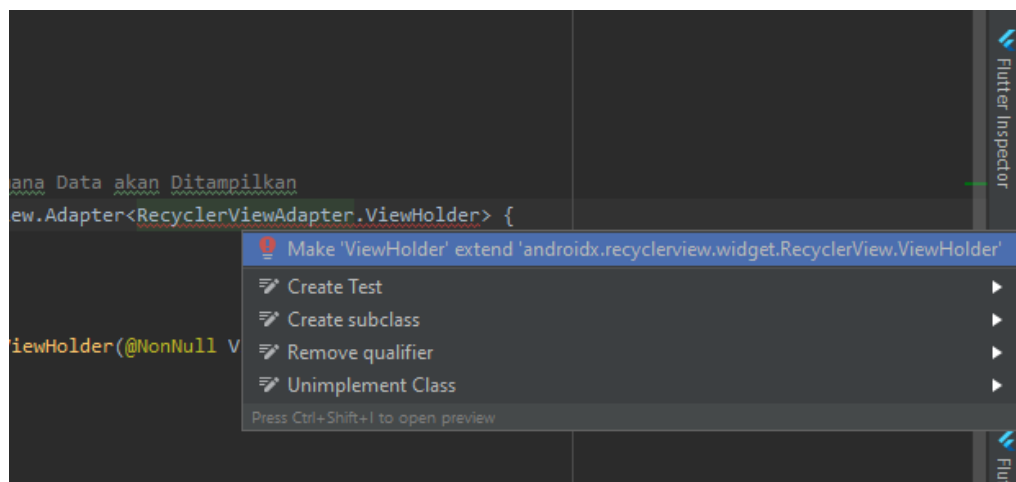
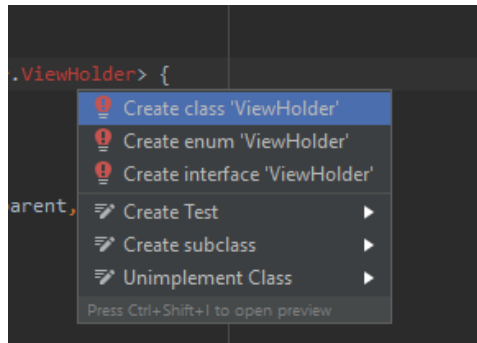
Implement methods  
Make 'RecyclerViewAdapter' abstract  
Create Test  
Create subclass  
Remove qualifier  
Unimplement Class  
Press Ctrl+Shift+I to open preview

6. Pilih ketiga method dibawah yaitu, **Oncreate**, **Onbind**, dan **getItemCount**, lalu pilih **OK**, maka secara otomatis akan dibuatkan method tersebut seperti gambar dibawah



```
RecyclerViewAdapter.java x
1 package com.luckyrafi13.crud_firebase;
2
3 import android.view.ViewGroup;
4
5 import androidx.annotation.NonNull;
6 import androidx.recyclerview.widget.RecyclerView;
7
8 //Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
9 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
10
11     @NonNull
12     @Override
13     public RecyclerViewAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
14         return null;
15     }
16
17     @Override
18     public void onBindViewHolder(@NonNull RecyclerViewAdapter.ViewHolder holder, int position) {
19
20     }
21
22     @Override
23     public int getItemCount() {
24         return 0;
25     }
26 }
```

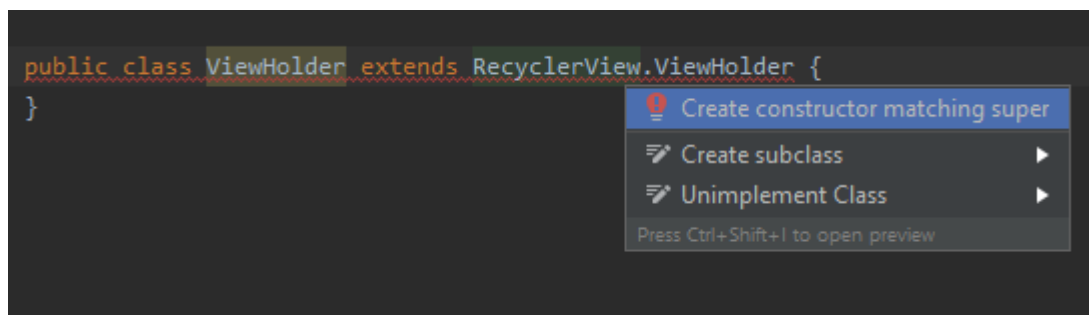
7. Pada Bagian **ViewHolder** tekan **Alt + Enter** untuk membuat class ViewHolder, setelah dibuatkan class baru selanjutnya lakukan hal yang sama dan pilih **Make ViewHolder extends**



8. Maka class ViewHolder akan ditambahkan extends dan akan seperti dibawah

```
public class ViewHolder extends RecyclerView.ViewHolder {  
}
```

9. Pada Class tersebut tekan **Alt + Enter** lalu pilih **Create constructor mathing super**, maka class akan dibuatkan constructor matching super seperti dibawah



```
public class ViewHolder extends RecyclerView.ViewHolder {
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
    }
}
```

10. Selanjutnya buatlah variable **ArrayList** dan juga **Context**

```
RecyclerViewAdapter.java
1 package com.luckyrafi13.crud_firebase;
2
3 import ...
11
12 //Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
13 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
14
15     //Deklarasi Variable
16     private ArrayList<data_mahasiswa> listMahasiswa;
17     private Context context;
18
19
```

11. Buatlah constructor untuk menerima inputan dari database, buatlah dengan melakukan generate dan pilih constructor

```
RecyclerViewAdapter.java
1 package com.luckyrafi13.crud_firebase;
2
3 import ...
11
12 //Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
13 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
14
15     //Deklarasi Variable
16     private ArrayList<data_mahasiswa> listMahasiswa;
17     private Context context;
18
19     public RecyclerViewAdapter(ArrayList<data_mahasiswa> listMahasiswa, Context context) {
20         this.listMahasiswa = listMahasiswa;
21         this.context = context;
22     }
23
```

12. Pada bagian class **ViewHolder** buatlah inisialisasi widget **Textview** dan **LinearLayout** seperti gambar dibawah

```
42 //ViewHolder Digunakan Untuk Menyimpan Referensi Dari View-View
43 public class ViewHolder extends RecyclerView.ViewHolder {
44
45     //Inisialisai Widget
46     private TextView NIM, Nama, Jurusan;
47     private LinearLayout ListItem;
48
49     public ViewHolder(@NonNull View itemView) { super(itemView); }
50 }
51
52 }
53
54 }
```

13. Selanjutnya Inisialisasikan juga View View yang terpasang pada layout design dengan menggunakan id

```
41
42 //ViewHolder Digunakan Untuk Menyimpan Referensi Dari View-View
43 public class ViewHolder extends RecyclerView.ViewHolder {
44
45     //Inisialisai Widget
46     private TextView NIM, Nama, Jurusan;
47     private LinearLayout ListItem;
48
49     public ViewHolder(@NonNull View itemView) {
50         super(itemView);
51
52         //Menginisialisasi View-View yang terpasang pada layout RecyclerView kita
53         NIM = itemView.findViewById(R.id.nim);
54         Nama = itemView.findViewById(R.id.nama);
55         Jurusan = itemView.findViewById(R.id.jurusan);
56         ListItem = itemView.findViewById(R.id.list_item);
57
58     }
59 }
60 }
```

14. Selanjutnya kita akan menyiapkan dan memasang layout pada bagian **oncreate** yang nantinya akan ditampilkan pada recyclerview dengan menggunakan layout yang sebelumnya sudah kita buat

```
@NonNull
@Override
public RecyclerViewAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    //Membuat View untuk Menyiapkan dan Memasang Layout yang Akan digunakan pada RecyclerView
    View V = LayoutInflater.from(parent.getContext()).inflate(R.layout.view_degsign, parent, attachToRoot: false);
    return new ViewHolder(V);
}
```

15. Pada Bagian **Onbind** ambil nilai dari textview dan juga masukan nilai ke dalam view

```
@Override
public void onBindViewHolder(@NonNull RecyclerViewAdapter.ViewHolder holder, int position) {

    //Mengambil Nilai/Value yang terdapat pada RecyclerView berdasarkan Posisi Tertentu
    final String NIM = listMahasiswa.get(position).getNim();
    final String Nama = listMahasiswa.get(position).getNama();
    final String Jurusan = listMahasiswa.get(position).getJurusan();

    //Memasukan Nilai/Value kedalam View (TextView: NIM, Nama, Jurusan)
    holder.NIM.setText("NIM: "+NIM);
    holder>Nama.setText("Nama: "+Nama);
    holder.Jurusan.setText("Jurusan: "+Jurusan);
}

@Override
```

16. Buatlah fungsi juga ketika kita menekan **ListItem** secara lama atau longklik seperti gambar di bawah, untuk fungsi lengkapnya akan kita kerjakan pada step Update dan Delete

```
holder.ListItem.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {
        /*
         *Kodingan untuk membuat fungsi Edit dan Delete,
         *yang akan dibahas pada Tutorial Berikutnya.
         */
        return true;
    }
});
```

17. Selanjutnya tambahkan nilai untuk menentukan data yang akan ditampilkan pada RecyclerView, pada modul ini ditentukan berdasarkan **listmahasiswa** yang ada

```
@Override
public int getItemCount() {
    return listMahasiswa.size();
}
```

18. Sehingga Struktur lengkap dari **RecyclerViewAdapter** adalah seperti ini

```
RecyclerViewAdapter.java
1 package com.luckyrafi13.crud_firebase;
2
3 import ...
4
5 //Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
6 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
7
8     //Deklarasi Variable
9     private ArrayList<data_mahasiswa> listMahasiswa;
10    private Context context;
11
12    public RecyclerViewAdapter(ArrayList<data_mahasiswa> listMahasiswa, Context context) {
13        this.listMahasiswa = listMahasiswa;
14        this.context = context;
15    }
16
17    @NonNull
18    @Override
19    public RecyclerViewAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
20        //Membuat View untuk Menyiapkan dan Memasang Layout yang Akan digunakan pada RecyclerView
21        View V = LayoutInflater.from(parent.getContext()).inflate(R.layout.view_degsign, parent, attachToRoot: false);
22        return new ViewHolder(V);
23    }
24
25 }
```

```

RecyclerViewAdapter.java x
34
35     @Override
36     public void onBindViewHolder(@NonNull RecyclerView.Adapter.ViewHolder holder, int position) {
37
38         //Mengambil Nilai/Value yang terdapat pada RecyclerView berdasarkan Posisi Tertentu
39         final String NIM = listMahasiswa.get(position).getNim();
40         final String Nama = listMahasiswa.get(position).getNama();
41         final String Jurusan = listMahasiswa.get(position).getJurusan();
42
43         //Memasukan Nilai/Value kedalam View (TextView: NIM, Nama, Jurusan)
44         holder.NIM.setText("NIM: "+NIM);
45         holder>Nama.setText("Nama: "+Nama);
46         holder.Jurusan.setText("Jurusan: "+Jurusan);
47
48         holder.ListItem.setOnLongClickListener(new View.OnLongClickListener() {
49             @Override
50             public boolean onLongClick(View v) {
51                 /*
52                  *Kodingan untuk membuat fungsi Edit dan Delete,
53                  *yang akan dibahas pada Tutorial Berikutnya.
54                  */
55                 return true;
56             }
57         });
58
59     }
60

```

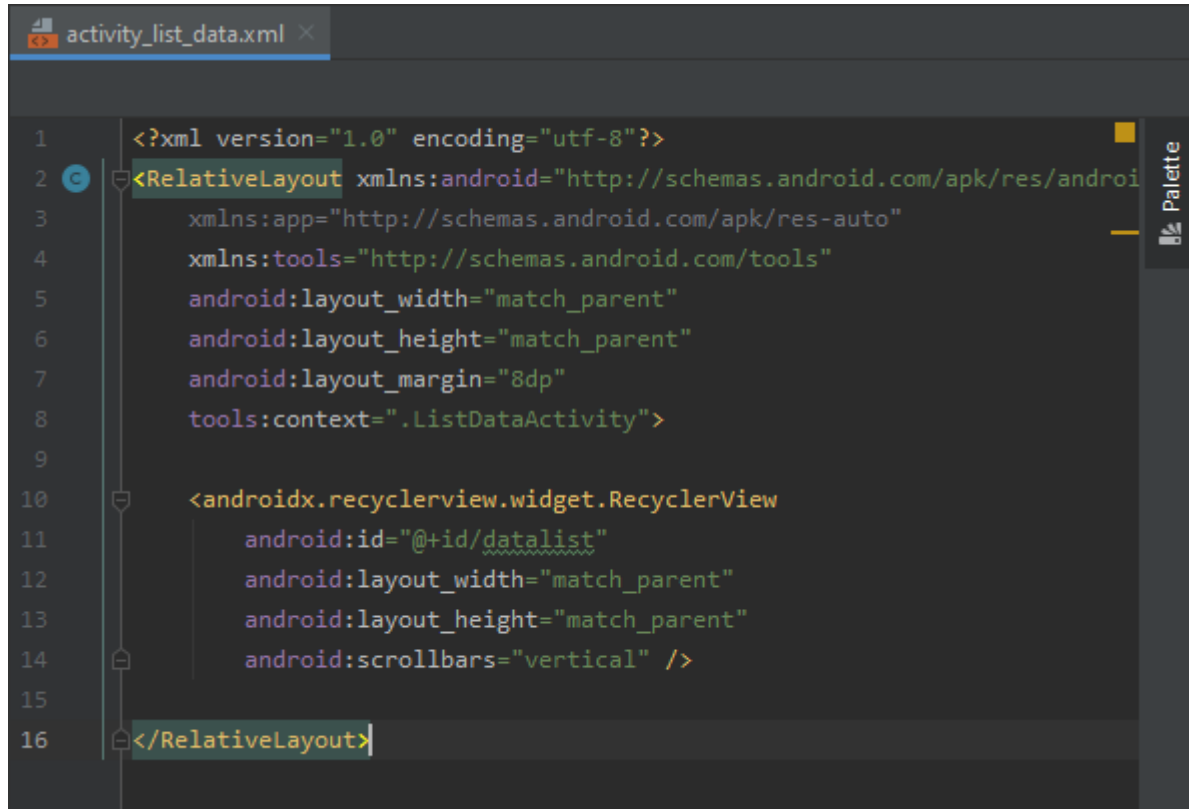
```

RecyclerViewAdapter.java x
60
61     @Override
62     public int getItemCount() {
63         return listMahasiswa.size();
64     }
65
66     //ViewHolder Digunakan Untuk Menyimpan Referensi Dari View-View
67     public class ViewHolder extends RecyclerView.ViewHolder {
68
69         //Inisialisai Widget
70         private TextView NIM, Nama, Jurusan;
71         private LinearLayout ListItem;
72
73         public ViewHolder(@NonNull View itemView) {
74             super(itemView);
75
76             //Menginisialisasi View-View yang terpasang pada layout RecyclerView kita
77             NIM = itemView.findViewById(R.id.nim);
78             Nama = itemView.findViewById(R.id.nama);
79             Jurusan = itemView.findViewById(R.id.jurusan);
80             ListItem = itemView.findViewById(R.id.list_item);
81
82         }
83     }
84
85

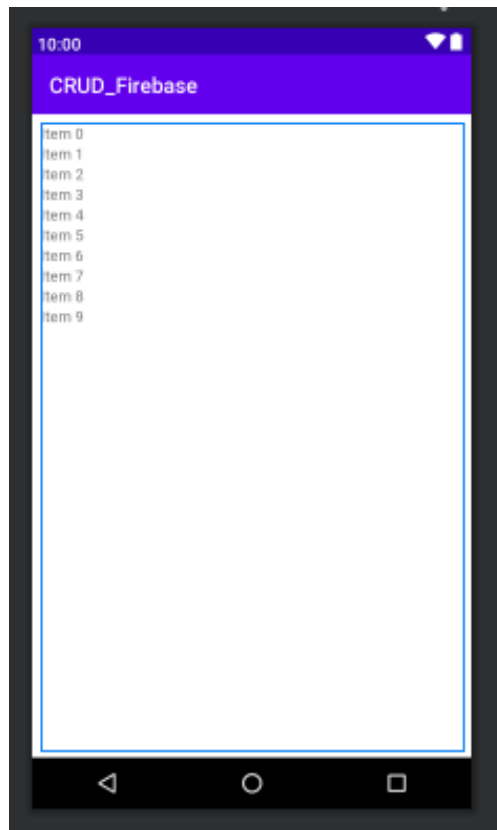
```



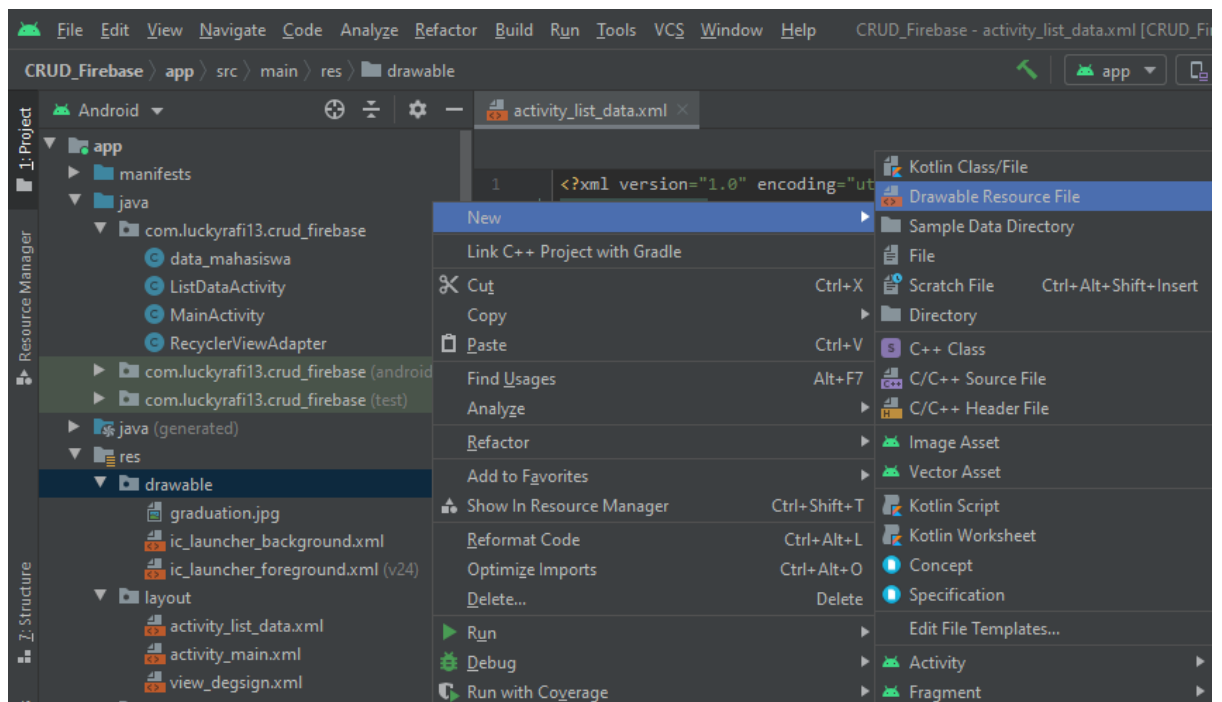
19. Setelah pengaturan RecyclerView selesai, berikutnya kita buat activity baru, bernama **ListData**, lalu kita tambahkan **RecyclerView** kedalam activity tersebut, yang nantinya digunakan untuk menampilkan data yang terdapat pada Database. Masuk ke bagian Activity\_list\_data.xml dan tambahkan recyclerview seperti dibawah ini

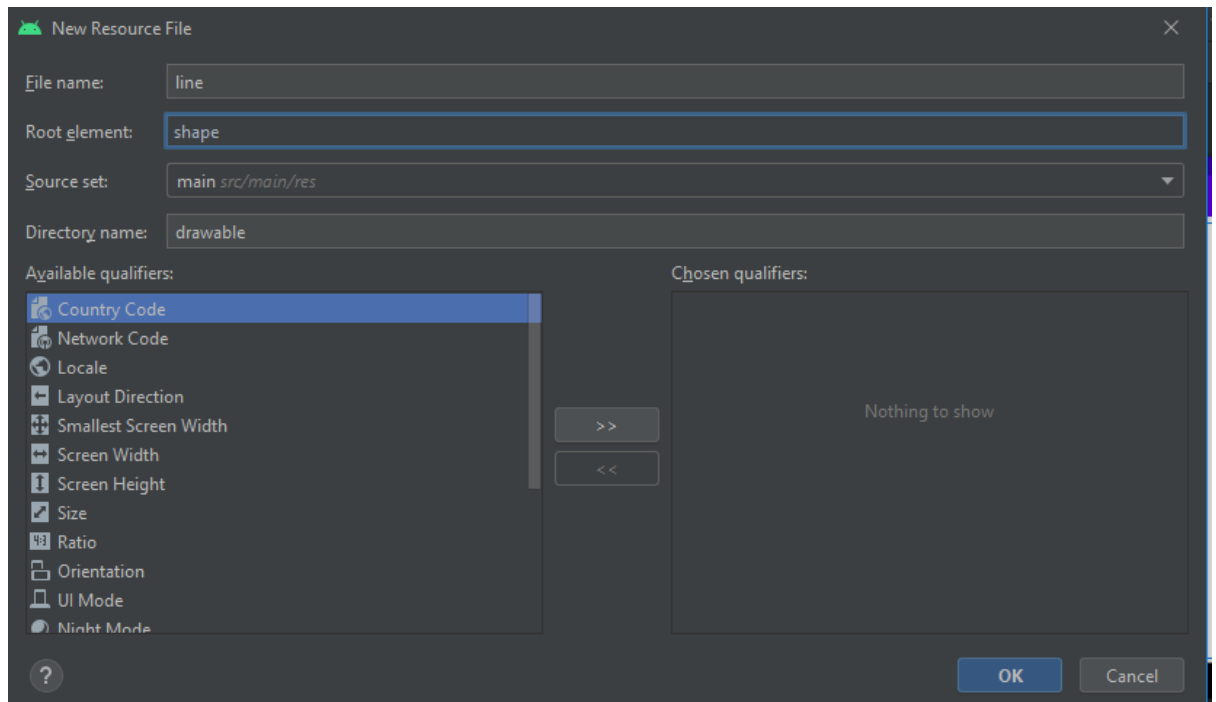


```
1      <?xml version="1.0" encoding="utf-8"?>
2      <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3          xmlns:app="http://schemas.android.com/apk/res-auto"
4          xmlns:tools="http://schemas.android.com/tools"
5          android:layout_width="match_parent"
6          android:layout_height="match_parent"
7          android:layout_margin="8dp"
8          tools:context=".ListDataActivity">
9
10         <androidx.recyclerview.widget.RecyclerView
11             android:id="@+id/datalist"
12             android:layout_width="match_parent"
13             android:layout_height="match_parent"
14             android:scrollbars="vertical" />
15
16     </RelativeLayout>
```



20. Untuk menampilkan **Garis bawah (Underline)** pada setiap item didalam RecyclerView, Buat resource baru bernama **line.xml** pada direktori **res > drawable > New > Drawable resource file**, jika sudah, masukan kode xml berikut ini.





```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    |
    <solid android:color="#FFD8D8D8" />
    <size android:height="0.5dp" />
</shape>
```

21. Sekarang buka file **MainActivity.java**, buatlah fungsi untuk tombol showdata untuk memindahkan activity dari **MainActivity** ke **ActivityListData**

```
ShowData.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent( packageContext: MainActivity.this, ListDataActivity.class);
        startActivity(intent);
    }
});
```

22. Buka class **MyListData.java**, pada class tersebut kita akan memasukkan source code untuk menampilkan Data dari Firebase kedalam **RecyclerView**.  
Deklarasikan variabel untuk RecyclerView dan Database Reference

```
ListDataActivity.java x
1  package com.luckyrafi13.crud_firebase;
2
3  import ...
11
12  public class ListDataActivity extends AppCompatActivity {
13
14      //Deklarasi Variable untuk RecyclerView
15      private RecyclerView recyclerView;
16      private RecyclerView.Adapter adapter;
17      private RecyclerView.LayoutManager layoutManager;
18
19      //Deklarasi Variable Database Reference dan ArrayList dengan Parameter Class Model kita.
20      private DatabaseReference reference;
21      private ArrayList<data_mahasiswa> dataMahasiswa;
22  }
```

23. Koneksikan Wigdet view dengan id dan buat method **MyRecyclerView**, dan **GetData**

```
recyclerView = findViewById(R.id.datalist);
MyRecyclerView();
GetData();
```

24. Tekan **Alt + Enter** untuk membuat method pada ListDataActivity sehingga muncul seperti dibawah ini

```
recyclerView = findViewById(R.id.datalist);
MyRecyclerView();
GetData()
}
```

Create method 'MyRecyclerView' in 'ListDataActivity'  
Rename reference  
Press Ctrl+Shift+I to open preview

```

23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_list_data);
27
28         recyclerView = findViewById(R.id.dataList);
29         MyRecyclerView();
30         GetData();
31     }
32
33
34     private void GetData() {
35     }
36
37     private void MyRecyclerView() {
38     }
39 }

```

25. Isikan Method **GetData** dengan pengecekan atau pengambilan data dari Realtime Database dengan menggunakan datasnapstot.

Dengan menggunakan for-each, akan me-mapping data kedalam objek mahasiswa, lalu pada fungsi **setKey()**, digunakan untuk mengambil semua key, berupa kode unik yang dihasilkan oleh **push()**, nantinya digunakan untuk proses **Update** dan **Delete**.

Semua data yang ter-mapping tersebut dimasukan kedalam ArrayList dataMahasiswa, dan terakhir dimasukan kedalam Adapter, untuk ditampilkan pada RecyclerView. Seperti pada gambar dibawah ini

```

ListDataActivity.java x
39
40 //Berisi baris kode untuk mengambil data dari Database dan menampilkannya kedalam Adapter
41 private void GetData() {
42     Toast.makeText(getApplicationContext(), text: "Mohon Tunggu Sebentar...", Toast.LENGTH_LONG).show();
43     //Mendapatkan Referensi Database
44     reference = FirebaseDatabase.getInstance().getReference();
45     reference.child("Admin").child("Mahasiswa")
46     .addValueEventListener(new ValueEventListener() {
47         @Override
48         public void onDataChange(DataSnapshot dataSnapshot) {
49             //Inisialisasi ArrayList
50             dataMahasiswa = new ArrayList<>();
51
52             for (DataSnapshot snapshot : dataSnapshot.getChildren()){
53                 //Mapping data pada DataSnapshot ke dalam objek mahasiswa
54                 data_mahasiswa mahasiswa = snapshot.getValue(data_mahasiswa.class);
55
56                 //Mengambil Primary Key, digunakan untuk proses Update dan Delete
57                 mahasiswa.setKey(snapshot.getKey());
58                 dataMahasiswa.add(mahasiswa);
59             }
60
61             //Inisialisasi Adapter dan data Mahasiswa dalam bentuk Array
62             adapter = new RecyclerViewAdapter(dataMahasiswa, context: ListDataActivity.this);
63
64             //Memasang Adapter pada RecyclerView
65             recyclerView.setAdapter(adapter);
66 }

```

```

ListDataActivity.java x
66
67 Toast.makeText(getApplicationContext(), text: "Data Berhasil Dimuat", Toast.LENGTH_LONG).show();
68 }
69
70 @Override
71 public void onCancelled(DatabaseError databaseError) {
72     /*
73     Kode ini akan dijalankan ketika ada error dan
74     pengambilan data error tersebut lalu memprint error nya
75     ke LogCat
76     */
77     Toast.makeText(getApplicationContext(), text: "Data Gagal Dimuat", Toast.LENGTH_LONG).show();
78     Log.e( tag: "MyListActivity", msg: databaseError.getDetails()+" "+databaseError.getMessage());
79 }
80 }
81
82 }

```

26. Mengatur RecyclerView dengan **LayoutManager** dan juga membuat Underline untuk setiap Item didalam list

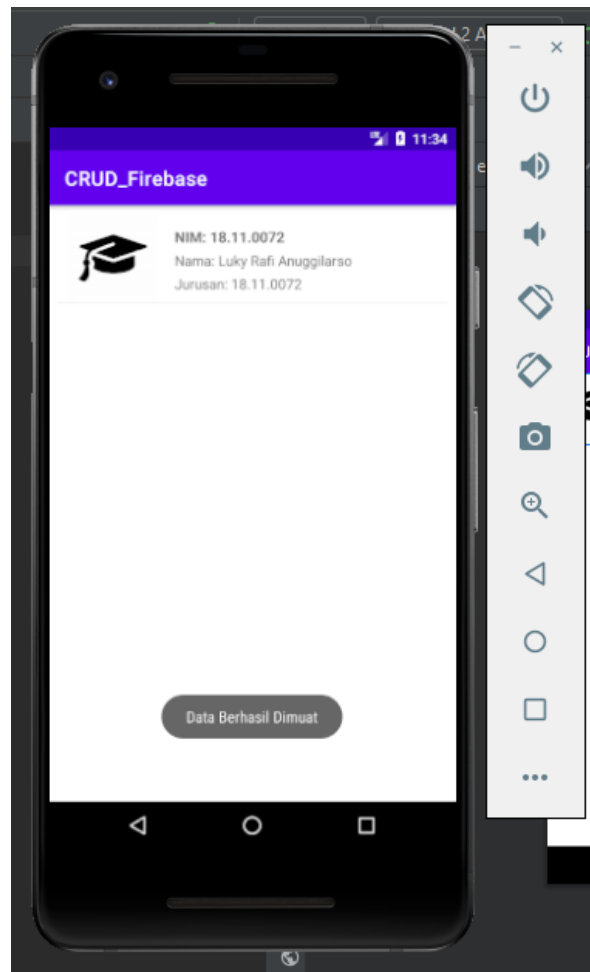
```

//Methode yang berisi kumpulan baris kode untuk mengatur RecyclerView
private void MyRecyclerView() {
    //Menggunakan Layout Manager, Dan Membuat List Secara Vertical
    layoutManager = new LinearLayoutManager( context: this);
    recyclerView.setLayoutManager(layoutManager);
    recyclerView.setHasFixedSize(true);

    //Membuat Underline pada Setiap Item Didalam List
    DividerItemDecoration itemDecoration = new DividerItemDecoration(getApplicationContext(), DividerItemDecoration.VERTICAL);
    itemDecoration.setDrawable(ContextCompat.getDrawable(getApplicationContext(), R.drawable.line));
    recyclerView.addItemDecoration(itemDecoration);
}
}

```

27. Setelah di runing dan memilih button **showdata** makan akan tampil seperti dibawah ini



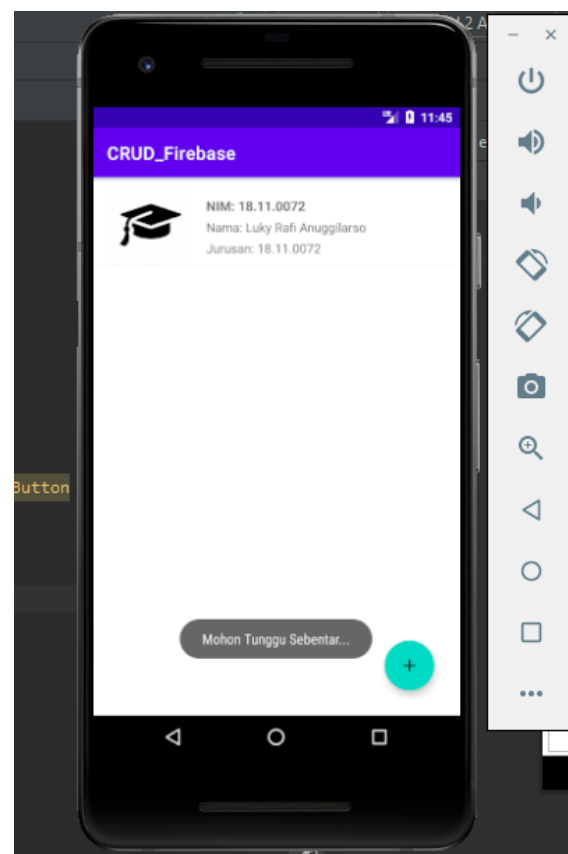
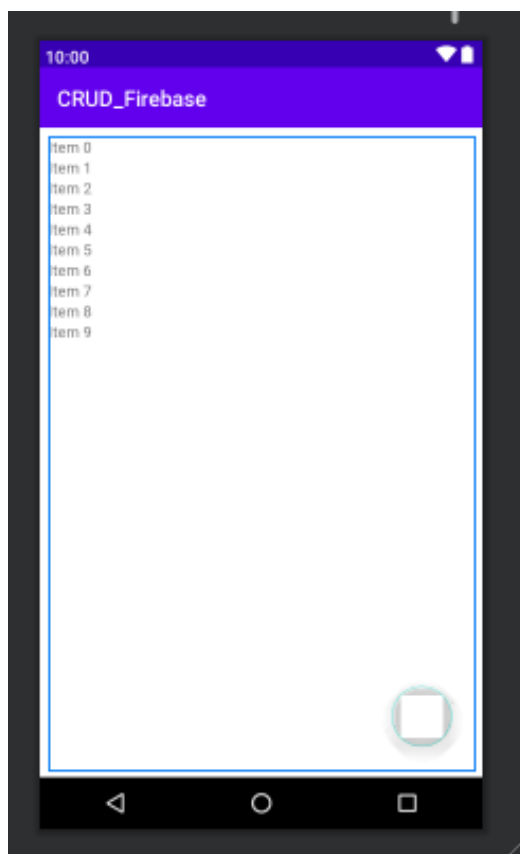
28. Pada **ListDataActivity** tambahkan **floating action button** untuk berpindah ke halaman tambah data, pertama masuk ke bagian `activity_list_data.xml` dan tambahkan view widget **floatingactionbutton**.

```
activity_list_data.xml
<?xml version="1.0" encoding="utf-8"
xmlns:android="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:layout_margin="8dp"
tools:context=".ListDataActivity">

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/datalist"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:scrollbars="vertical" />

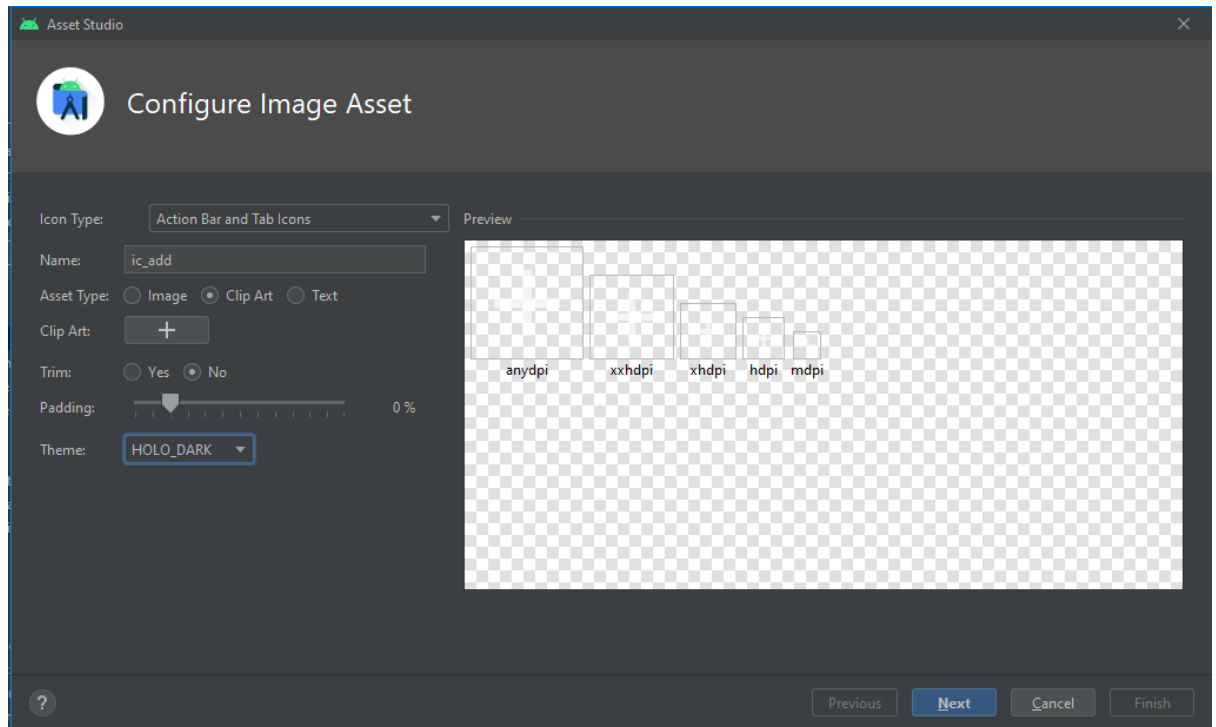
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginRight="20dp"
    android:layout_marginBottom="20dp"
    android:src="@drawable/ic_add" />

</RelativeLayout>
```





29. Untuk menambahkan tanda plus pada bagian floating action button tambahkan resource **Imageasset**, dan pilih **icon type Action Bar and Tab Icons**, lalu beri nama **ic\_add**, dan ubah **clip art** menjadi tanda tamba, kemudian ubah theme menjadi **HOLO\_DARK**. Lalu pilih **Next** dan **Finish**



30. Fungsikan tombol **floatingactionbutton** dengan cara masuk ke **ListDataActivity.java** dan tambahkan Deklarasi Variabel **floatingactionbutton**, koneksikan dengan id dan buat fungsi untuk berpindah dari **ListDataActivity** ke **MainActivity**

```
ListDataActivity.java
1 package com.luckyrafi13.crud_firebase;
2
3 import ...
21
22 public class ListDataActivity extends AppCompatActivity {
23
24     //Deklarasi Variable untuk RecyclerView
25     private RecyclerView recyclerView;
26     private RecyclerView.Adapter adapter;
27     private RecyclerView.LayoutManager layoutManager;
28
29     //Deklarasi Variable Database Reference dan ArrayList dengan Parameter Class Model kita.
30     private DatabaseReference reference;
31     private ArrayList<data_mahasiswa> dataMahasiswa;
32
33     private FloatingActionButton fab;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_list_data);

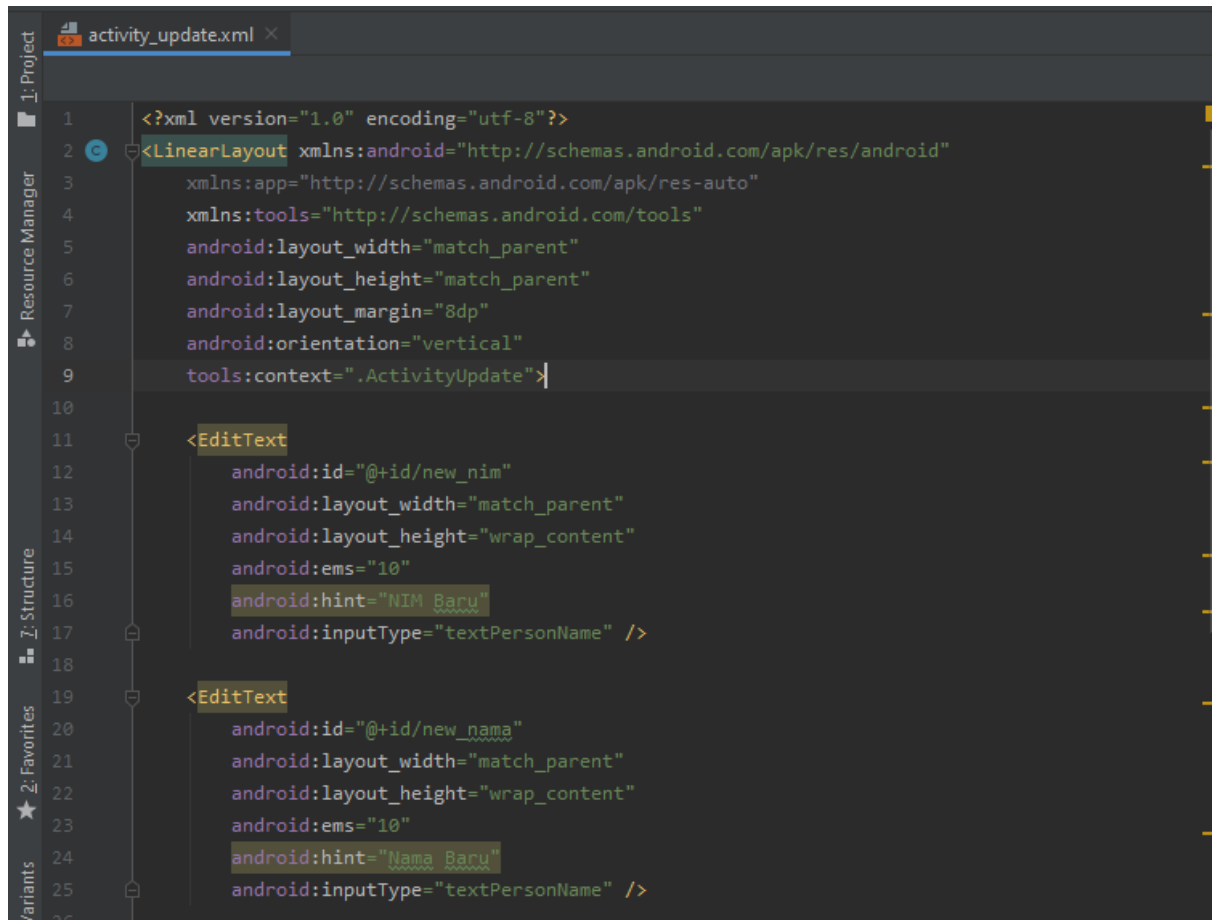
    recyclerView = findViewById(R.id.datalist);
    fab = findViewById(R.id.fab);
    MyRecyclerView();
    GetData();

    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent( packageContext: ListDataActivity.this, MainActivity.class);
            startActivity(intent);
        }
    });
}
```

## Membuat Fungsi Update

pada proses Update yang akan berjalan nanti pertama kita menentukan item/data mana yang akan dipilih berdasarkan posisi tertentu dari RecyclerView, setiap data didalam list tersebut mempunyai kode uniknya masing-masing, kode unik inilah yang digunakan untuk mengambil data yang akan di update oleh user, lalu menyimpannya kembali kedalam database.

1. Pertama kita akan membuat class dan activity baru, bernama **ActivityUpdate** activity ini digunakan untuk memproses data yang ingin kita ubah. Ubah tampilan pada **activity\_update.xml** seperti dibawah ini



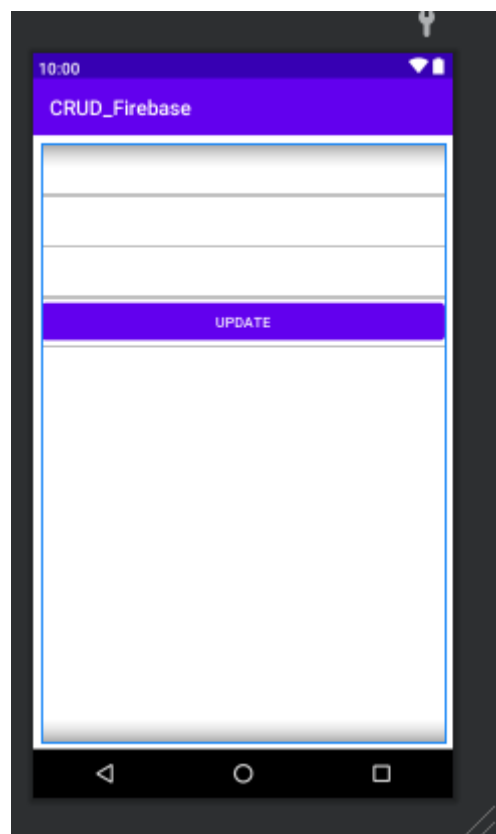
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"
    android:orientation="vertical"
    tools:context=".ActivityUpdate">

    <EditText
        android:id="@+id/new_nim"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="NIM Baru"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/new_nama"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Nama Baru"
        android:inputType="textPersonName" />

</LinearLayout>
```

```
activity_update.xml
26
27 <EditText
28     android:id="@+id/new_jurusan"
29     android:layout_width="match_parent"
30     android:layout_height="wrap_content"
31     android:ems="10"
32     android:hint="Jurusan Baru"
33     android:inputType="textPersonName" />
34
35 <Button
36     android:id="@+id/update"
37     android:layout_width="match_parent"
38     android:layout_height="wrap_content"
39     android:text="Update" />
40
41 </LinearLayout>
42
43
44
```



- Setelah kita membuat activity tersebut, selanjutnya kita beralih pada class **RacyclerViewAdapter**, didalam method **onLongClickListener()**, kita tambahkan baris kode berikut ini.

```
RecyclerViewAdapter.java
52 holder.ListItem.setOnLongClickListener(new View.OnLongClickListener() {
53     @Override
54     public boolean onLongClick(View v) {
55
56         final String[] action = {"Update", "Delete"};
57         AlertDialog.Builder alert = new AlertDialog.Builder(v.getContext());
58         alert.setItems(action, new DialogInterface.OnClickListener() {
59             @Override
60             public void onClick(DialogInterface dialog, int i) {
61                 switch (i){
62                     case 0:
63                         /*
64                          Berpindah Activity pada halaman layout updateData
65                          dan mengambil data pada listMahasiswa, berdasarkan posisinya
66                          untuk dikirim pada activity selanjutnya
67                         */
68                         Bundle bundle = new Bundle();
69                         bundle.putString("dataNIM", listMahasiswa.get(position).getNim());
70                         bundle.putString("dataNama", listMahasiswa.get(position).getNama());
71                         bundle.putString("dataJurusan", listMahasiswa.get(position).getJurusan());
72                         bundle.putString("getPrimaryKey", listMahasiswa.get(position).getKey());
73                         Intent intent = new Intent(v.getContext(), ActivityUpdate.class);
74                         intent.putExtras(bundle);
75                         context.startActivity(intent);
76                         break;
```

```
RecyclerViewAdapter.java
78
79                     case 1:
80                         //Pembahasan selanjutnya mengenai fungsi Delete
81                         break;
82                 }
83             }
84         });
85         alert.create();
86         alert.show();
87         return true;
88     }
89 }
```

Saat user menekan salah satu item dari list, maka akan muncul menu **Update** dan **Delete**. Saat user memilih menu **Update**, maka disana kita menggunakan **Class Bundle** untuk mengambil dan menyimpan data dari mahasiswa yang kita pilih.

Data yang disimpan pada **Bundle** berupa **NIM**, **Nama**, **Jurusan** dan **Key** (kode unik yang dihasilkan **Push**), key ini digunakan untuk menentukan data mana yang akan diubah didalam database, karena setiap data yang tampil didalam **ReyclerView**,

mempunyai **PrimaryKeynya** masing-masing, yang dihasilkan oleh **push()** pada saat user **mengCreate data**.

3. Buka class **updateData.java**, disini kita akan menambahkan source code untuk memproses update, berdasarkan data yang telah ditentukan.

```
ActivityUpdate.java x
1 package com.luckyrafi13.crud_firebase;
2
3 import ...
17
18 public class ActivityUpdate extends AppCompatActivity {
19
20     //Deklarasi Variable
21     private EditText nimBaru, namaBaru, jurusanBaru;
22     private Button update;
23     private DatabaseReference database;
24     private String cekNIM, cekNama, cekJurusan;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_update);
30
31         getSupportActionBar().setTitle("Update Data");
32         nimBaru = findViewById(R.id.new_nim);
33         namaBaru = findViewById(R.id.new_nama);
34         jurusanBaru = findViewById(R.id.new_jurusan);
35         update = findViewById(R.id.update);
36
37         database = FirebaseDatabase.getInstance().getReference();
38         getData();
```

```
ActivityUpdate.java x
39
40     update.setOnClickListener(new View.OnClickListener() {
41         @Override
42         public void onClick(View v) {
43
44             //Mendapatkan Data Mahasiswa yang akan dicek
45             cekNIM = nimBaru.getText().toString();
46             cekNama = namaBaru.getText().toString();
47             cekJurusan = jurusanBaru.getText().toString();
48
49             //Mengecek agar tidak ada data yang kosong, saat proses update
50             if(isEmpty(cekNIM) || isEmpty(cekNama) || isEmpty(cekJurusan)){
51                 Toast.makeText( context: ActivityUpdate.this, text: "Data tidak boleh ada yang kosong", Toast.LENGTH_SHORT).show();
52             }else {
53                 /*
54                  Menjalankan proses update data.
55                  Method Setter digunakan untuk mendapatkan data baru yang diinputkan User.
56                  */
57                 data_mahasiswa setMahasiswa = new data_mahasiswa();
58                 setMahasiswa.setNim(nimBaru.getText().toString());
59                 setMahasiswa.setNama(namaBaru.getText().toString());
60                 setMahasiswa.setJurusan(jurusanBaru.getText().toString());
61                 updateMahasiswa(setMahasiswa);
62             }
63         }
64     });
65
66 }
```

```

ActivityUpdate.java x
67
68 // Mengecek apakah ada data yang kosong, sebelum diupdate
69 private boolean isEmpty(String s){
70     return TextUtils.isEmpty(s);
71 }
72
73 private void getData() {
74     //Menampilkan data dari item yang dipilih sebelumnya
75     final String getNIM = getIntent().getExtras().getString( key: "dataNIM");
76     final String getNama = getIntent().getExtras().getString( key: "dataNama");
77     final String getJurusan = getIntent().getExtras().getString( key: "dataJurusan");
78     nimBaru.setText(getNIM);
79     namaBaru.setText(getNama);
80     jurusanBaru.setText(getJurusan);
81 }

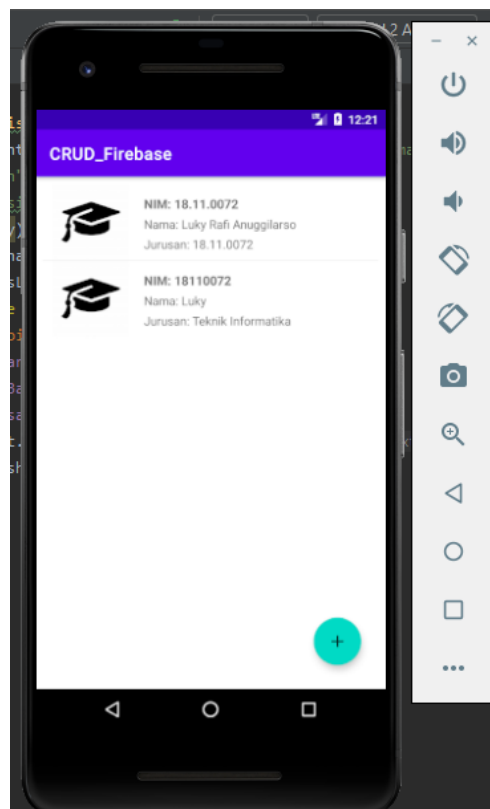
```

```

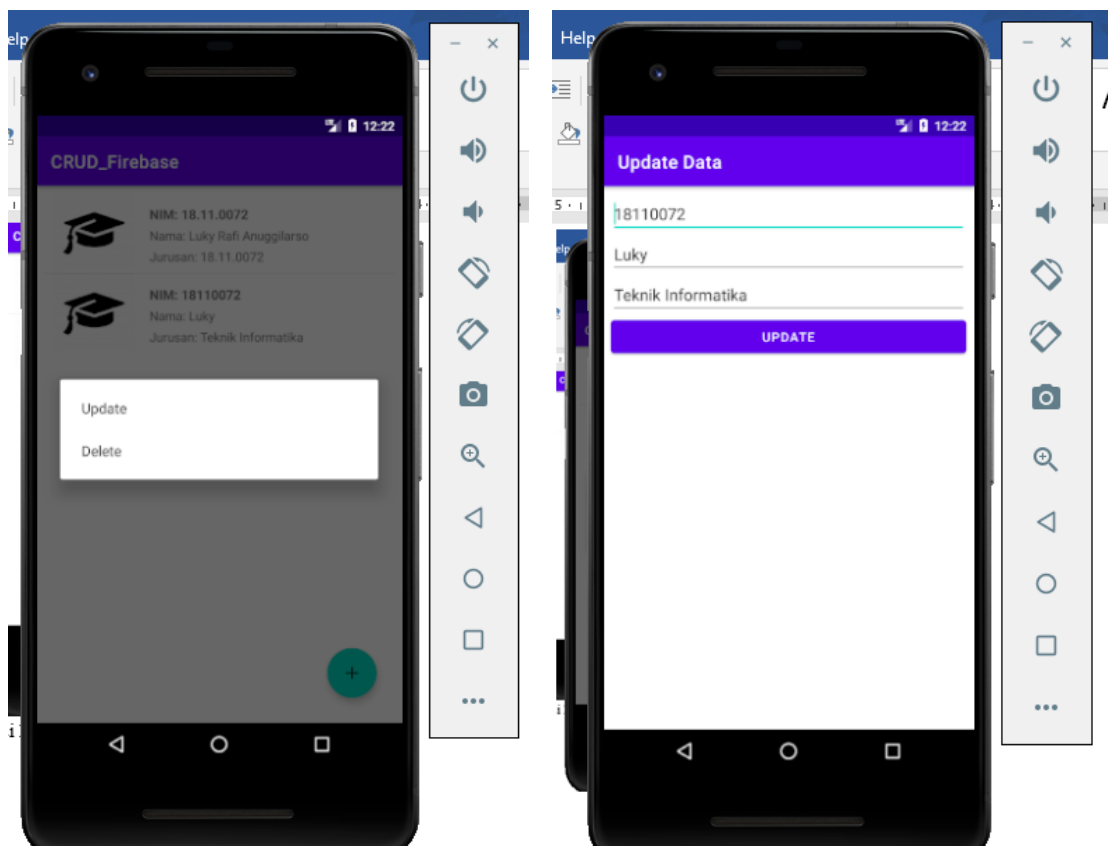
ActivityUpdate.java x
82
83 private void updateMahasiswa(data_mahasiswa mahasiswa) {
84     String getKey = getIntent().getExtras().getString( key: "getPrimaryKey");
85     database.child("Admin")
86         .child("Mahasiswa")
87         .child(getKey)
88         .setValue(mahasiswa)
89         .addOnSuccessListener(new OnSuccessListener<Void>() {
90             @Override
91             public void onSuccess(Void aVoid) {
92                 nimBaru.setText("");
93                 namaBaru.setText("");
94                 jurusanBaru.setText("");
95                 Toast.makeText( context: ActivityUpdate.this, text: "Data Berhasil diubah", Toast.LENGTH_SHORT).show();
96                 finish();
97             }
98         });
99 }
100 }
101

```

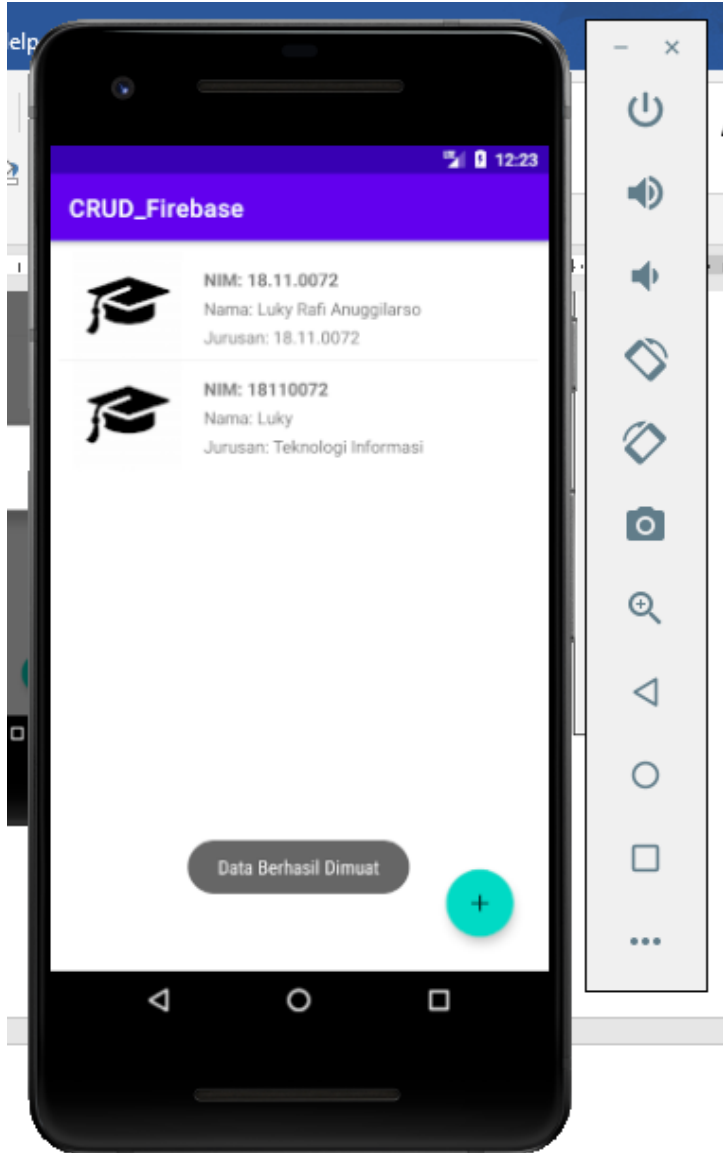
4. Hasil nya adalah seperti dibawah ini



5. Ketika salah satu item di klik lama atau longclick







## CRUD\_Firebase



NIM: 18.11.0072  
Nama: Luky Rafi Anuggilarso  
Jurusan: 18.11.0072



NIM: 18110072  
Nama: Luky  
Jurusan: Teknologi Informasi

Data Berhasil Dimuat



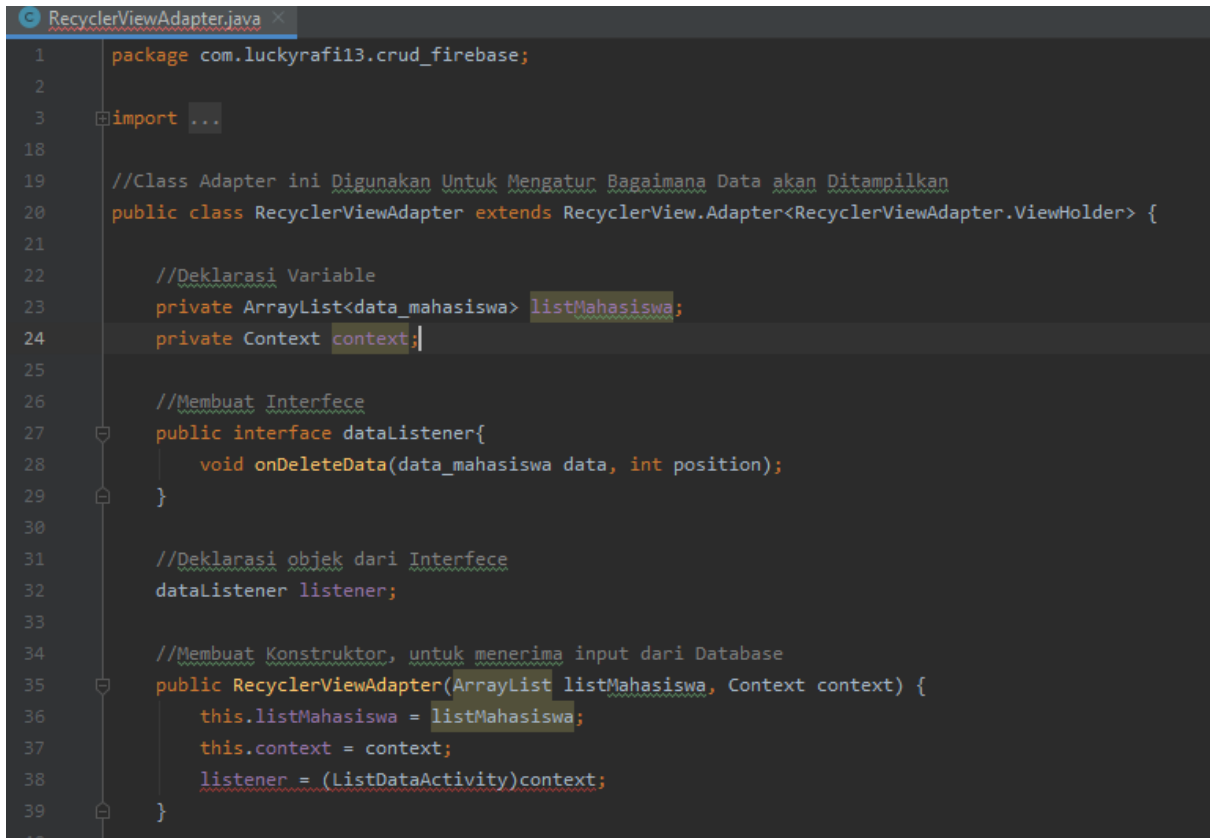
## Membuat Fungsi Delete

Konsep pada fitur **Delete** hampir sama dengan fitur **Update**, yaitu dengan menggunakan fungsi `longClick`, untuk menampilkan menu delete pada item didalam list.

Cara kerja dari fitur delete tersebut, pertama kita membuat sebuah interface untuk menghubungkan adapter dengan class **MyListData.java**, karena pada adapter tersebut tidak mempunyai referensi database, jadi kita membuat sebuah interface untuk menghubungkannya.

1. Pertama buka class **RecyclerViewAdapter**, pada class tersebut kita tambahkan interface, yang digunakan untuk menjembatani data yang akan dikirim dari Adapter pada class **MyListData**.

Pada class **RecyclerViewAdaper**, kita menambahkan **interface** dan menambahkan inisialisasi variabelnya pada constructor, untuk mendapatkan context dari class **MyListData**.



```
1 package com.luckyrafi13.crud_firebase;
2
3 import ...
4
5 //Class Adapter ini Digunakan Untuk Mengatur Bagaimana Data akan Ditampilkan
6 public class RecyclerViewAdapter extends RecyclerView.Adapter<RecyclerViewAdapter.ViewHolder> {
7
8     //Deklarasi Variable
9     private ArrayList<data_mahasiswa> listMahasiswa;
10    private Context context;
11
12    //Membuat Interface
13    public interface dataListener{
14        void onDeleteData(data_mahasiswa data, int position);
15    }
16
17    //Deklarasi objek dari Interface
18    dataListener listener;
19
20    //Membuat Konstruktor, untuk menerima input dari Database
21    public RecyclerViewAdapter(ArrayList listMahasiswa, Context context) {
22        this.listMahasiswa = listMahasiswa;
23        this.context = context;
24        listener = (ListDataActivity)context;
25    }
26 }
```

2. Jika terjadi **error**, itu dikarenakan kita belum mengimplementasikan method **onDeleteData()** pada class **MyListData**, yang menjadi di contextnya, Solusinya, tekan (**alt + enter**) pada bagian yang error, lalu implementasikan method tersebut, seperti pada gambar berikut ini.

```
//Membuat Konstruktor, untuk menerima input dari Database
public RecyclerViewAdapter(ArrayList listMahasiswa, Context context) {
    this.listMahasiswa = listMahasiswa;
    this.context = context;
    listener = (ListDataActivity)context;
}

@NonNull
@Override
public RecyclerViewAdapter...
```

Cast to 'com.luckyrafi13.crud\_firebase.RecyclerViewAdapter.dataListener'  
Change field 'listener' type to 'ListDataActivity'  
Make 'ListDataActivity' implement 'com.luckyrafi13.crud\_firebase.RecyclerViewAdapter.dataListener'  
Migrate listener type to 'ListDataActivity'

3. Setelah kalian selesai mengimplementasi metnod **onDeleteData()**, selanjutnya buka kembali class **RecyclerViewAdapter**, tambahkan baris kode berikut ini didalam **onBindViewHolder()**, pada method **onLongClickListener()**, seperti berikut ini.

```
case 1:
    //Menggunakan interface untuk mengirim data mahasiswa, yang akan dihapus
    listener.onDeleteData(listMahasiswa.get(position), position);
    break;
```

4. Buka class **ListDataActivity**, didalam method **onDeleteData()** yang sudah kalian implementasi tadi, tambahkan baris kode berikut ini.

```
ListDataActivity.java
115 @Override
116 public void onDeleteData(data_mahasiswa data, int position) {
117     /*
118      * Kode ini akan dipanggil ketika method onDeleteData
119      * dipanggil dari adapter pada RecyclerView melalui interface.
120      * kemudian akan menghapus data berdasarkan primary key dari data tersebut
121      * Jika berhasil, maka akan memunculkan Toast
122      */
123     if (reference != null) {
124         reference.child("Admin")
125             .child("Mahasiswa")
126             .child(data.getKey())
127             .removeValue()
128             .addOnSuccessListener(new OnSuccessListener() {
129                 @Override
130                 public void onSuccess(Object o) {
131                     Toast.makeText(context, ListDataActivity.this, "Data Berhasil Dihapus", Toast.LENGTH_SHORT).show();
132                 }
133             });
134     }
135 }
```

5. Hasilnya akan seperti dibawah ini

