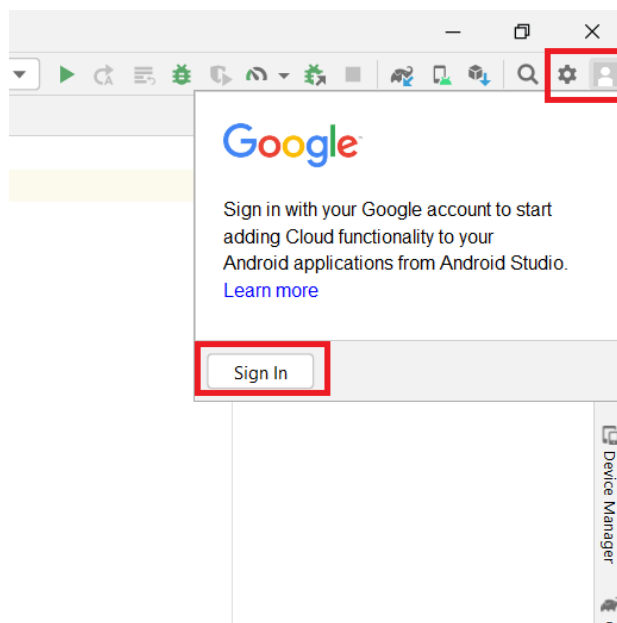


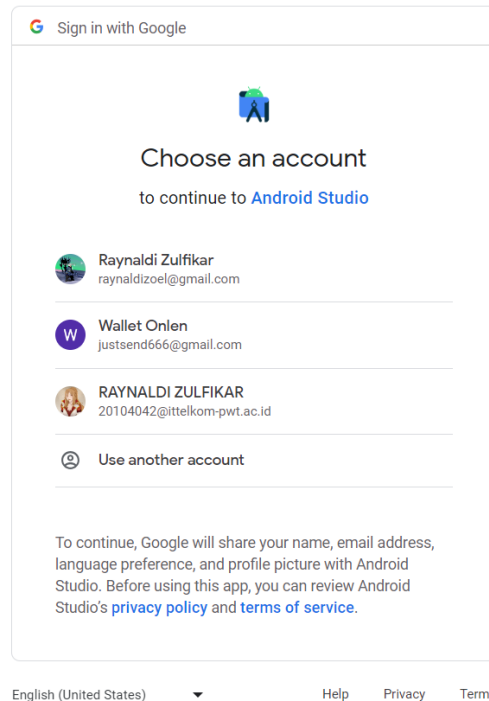
## 1. Sinkronisasi Project Android Dengan Akun Google

Sebelum melakukan instalasi firebase sebagai penyimpanan data pada Android Studio diperlukan sinkronisasi antara Android Studio dengan akun Google. Google sebagai penyedia layanan Firebase dan pengembang aplikasi Android Studio untuk melakukan akses Firebase Android Studio maka pada aplikasi Android Studio anda harus disinkronisasi dengan akun Google. Sinkronisasi dilakukan dengan cara klik pada sudut kanan atas dimana terdapat icon orang dengan warna abu abu kemudian klik maka akan muncul tampilan seperti pada gambar 1. Apabila Android Studio belum tersinkronisasi dengan akun Google maka akan tampak seperti gambar 1. Langkah berikutnya kemudian klik sign in untuk melakukan proses Masuk pada akun Google anda.



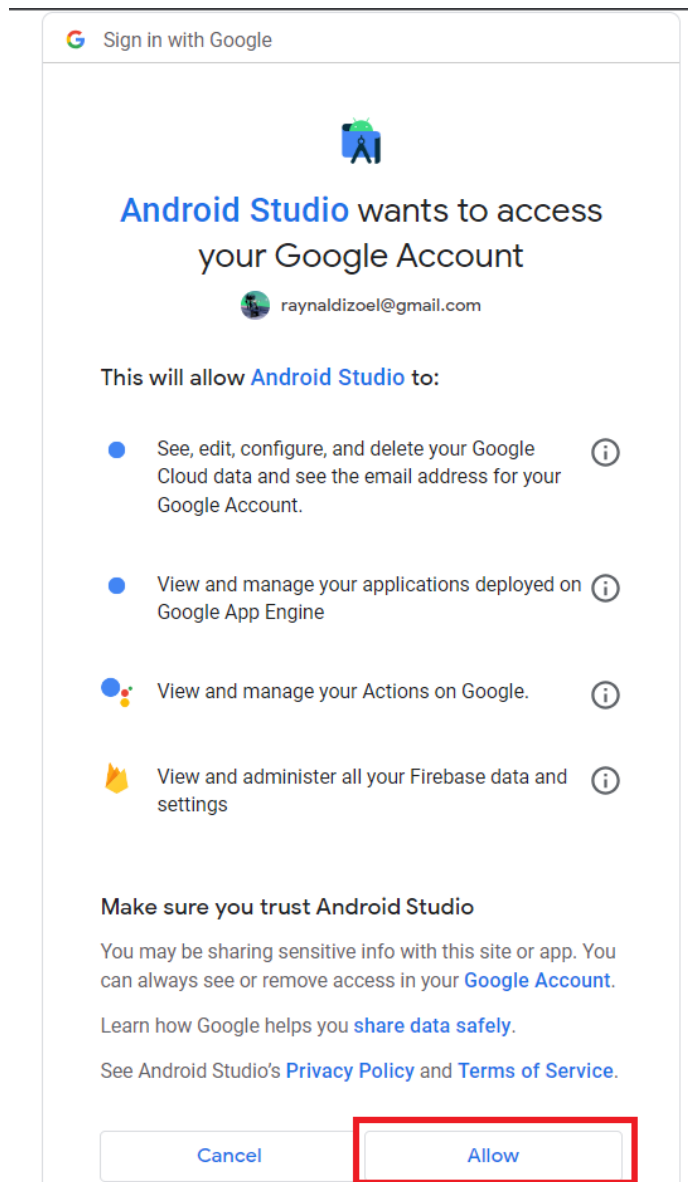
Gambar 1

Setelah sign akan muncul seperti pada gambar 2 dimana pengguna diminta memilih akun mana yang ingin digunakan untuk dilakukan proses sinkronisasi.



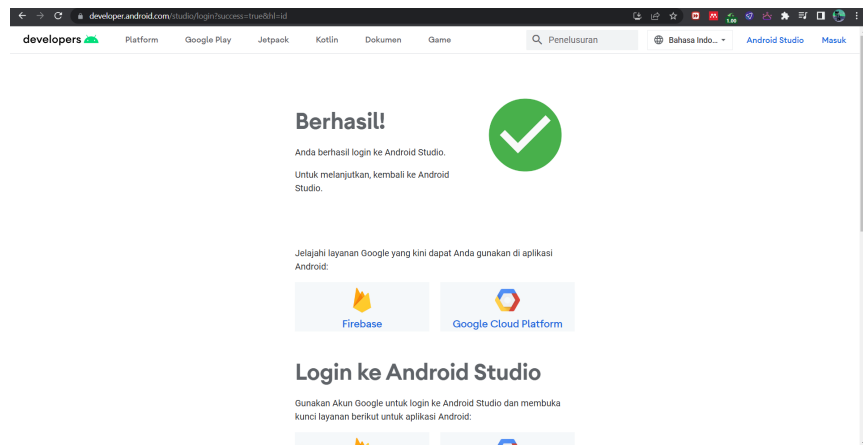
Gambar 2

Setelah memilih akun yang digunakan kemudian akan muncul tampilan seperti pada gambar 3 dimana anda diminta persetujuan tentang paket atau fitur yang akan dapat diakses pada akun Google anda pada aplikasi apabila sudah kemudian pilih izinkan

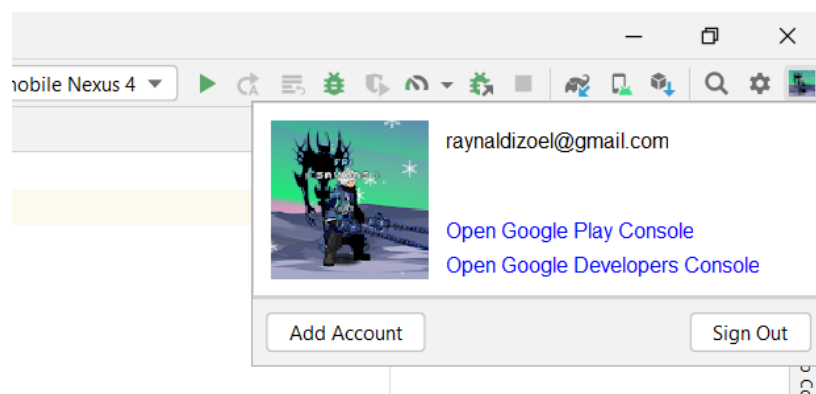


Gambar 3

Apabila sudah diklik maka akan muncul tampilan seperti pada gambar 4 yang menunjukkan bahwa akun Google anda sudah tersinkronisasi dengan aplikasi Android Studio yang anda gunakan. Pada gambar 5 menunjukkan akun google anda di aplikasi Android Studio



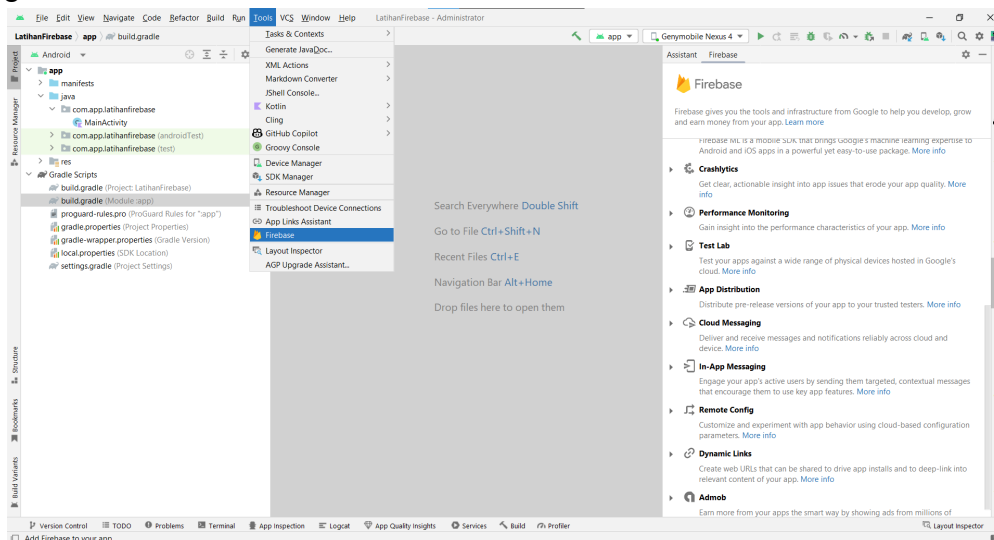
Gambar 4



Gambar 5

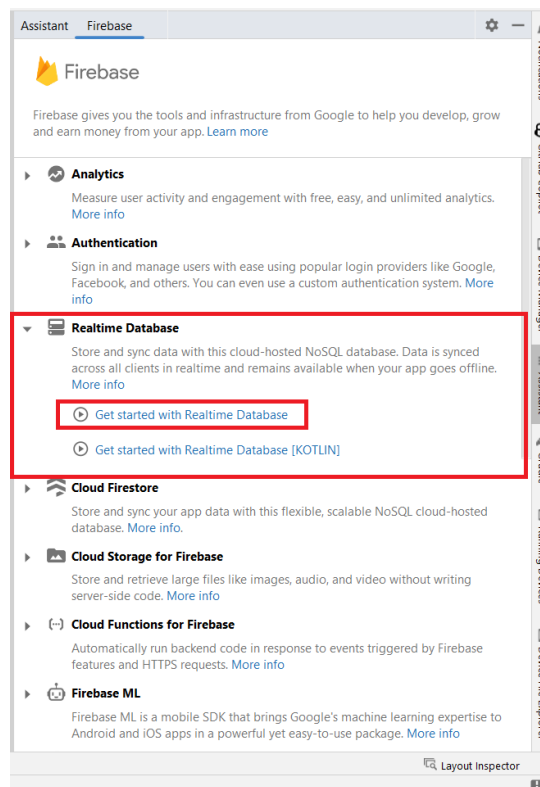
## 2. Koneksi Project Android Dengan Project Firebase

Setelah melakukan koneksi dengan akun Google kemudian selanjutnya melakukan koneksi Project Android dengan Project Firebase. Untuk mengakses Firebase pada Android Studio sudah tersedia plugin yang dapat langsung diakses pada Android Studio, dengan cara akses adalah pilih menu **Tools->Firebase** kemudian akan muncul **Assistant Firebase** pada bagian kanan seperti pada gambar 6.

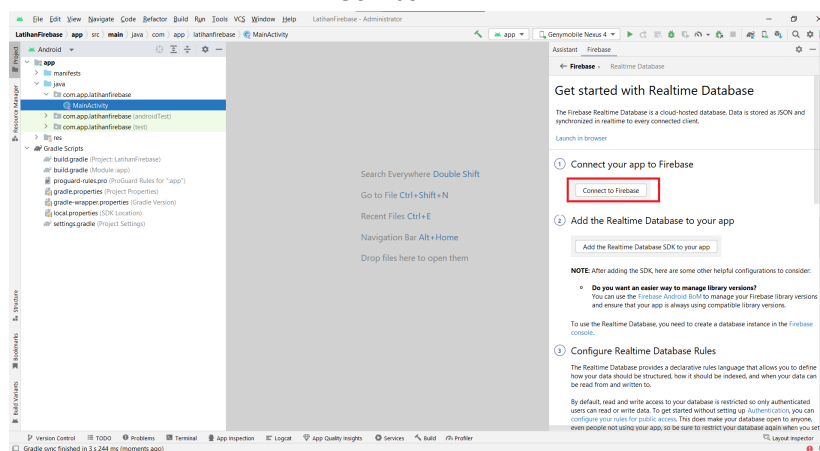


Gambar 6

Selanjutnya untuk melakukan sinkronisasi Project Android Studio dengan Layanan Realtime Database pada Project Firebase yang anda buat klik pada **“Realtime Database”** kemudian klik **“Get started with Realtime Database”** seperti pada gambar 7. Selanjutnya tampilan akan muncul dua buah pilihan seperti pada gambar 8 yaitu **“Connect your app to Firebase”** dan **“Add the Realtime Database SDK to your app”**. **“Connect your app to Firebase”** berfungsi untuk melakukan koneksi aplikasi dengan Firebase yang sudah dibuat. **“Add the Realtime Database SDK to your app”** berfungsi untuk menambahkan packet SDK untuk Realtime Database di project Android Studio.

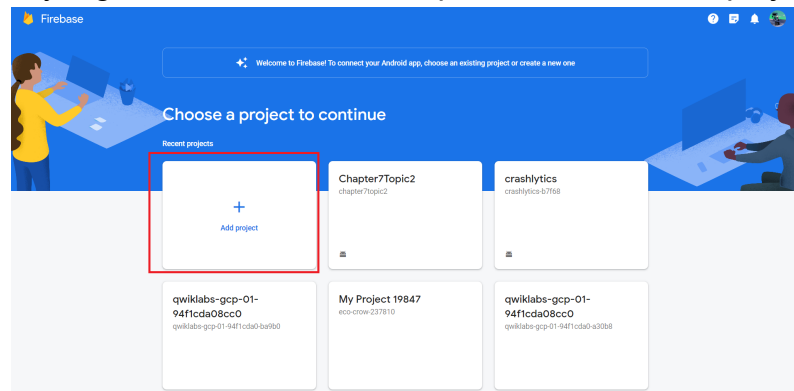


Gambar 7



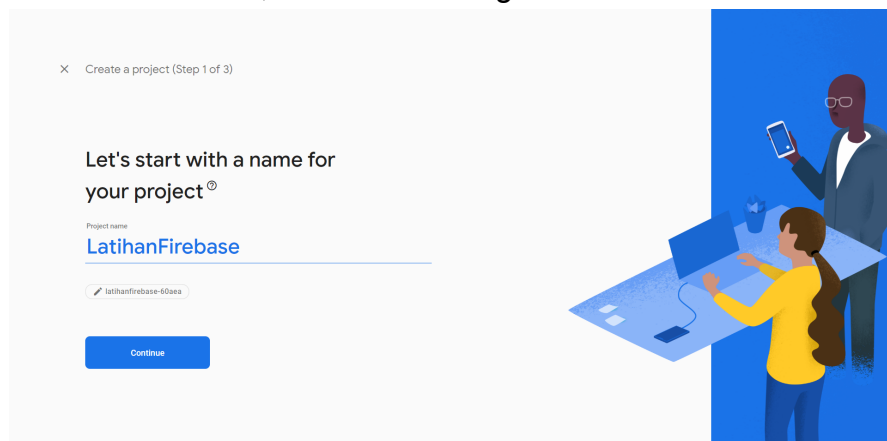
Gambar 8

Langkah awal klik **“Connect your app to Firebase”** untuk mengkoneksikan dengan project Firebase, kemudian akan muncul tampilan pada gambar 9 dimana kita diminta untuk memilih project Firebase yang dikoneksikan, dimana pada contoh ini add project.

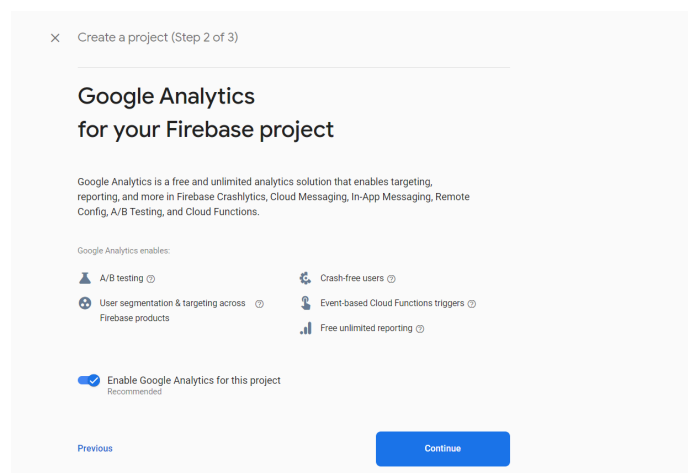


Gambar 9

Setelah itu akan muncul perintah seperti ini, masukan nama project. Setelah itu continue, dan continue lagi

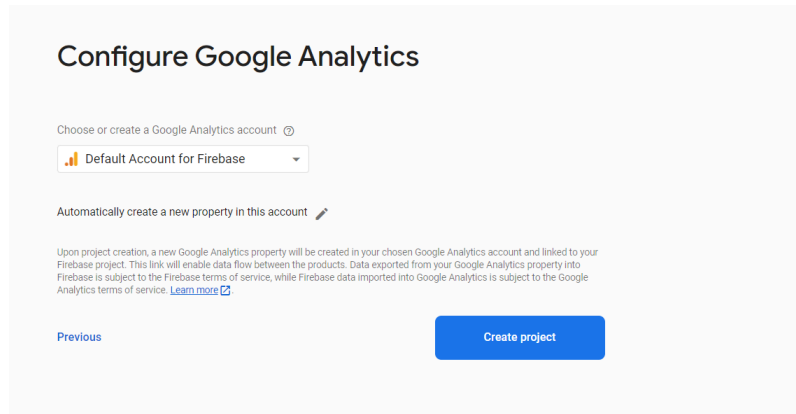


Gambar 10



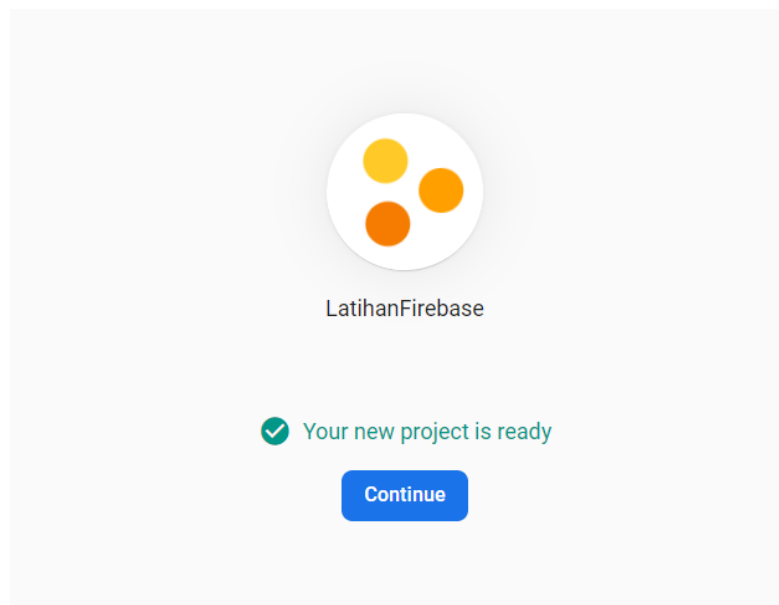
Gambar 11

Kemudian pilih default account for firebase, dan create project.



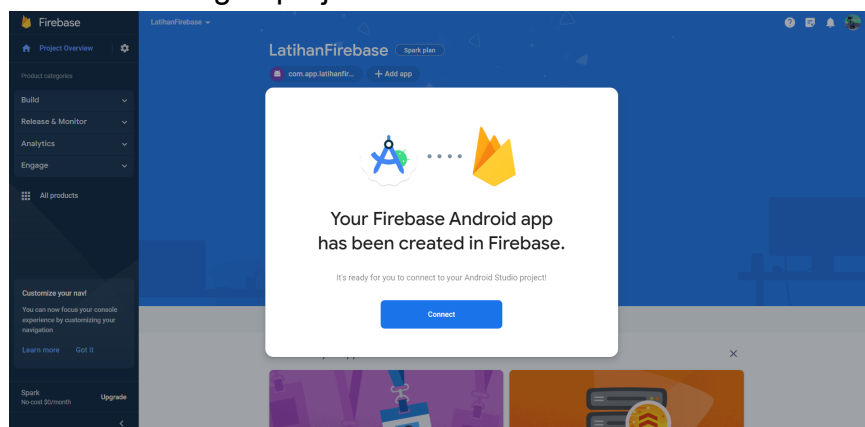
Gambar 12

Setelah tampilan seperti ini maka tekan continue.



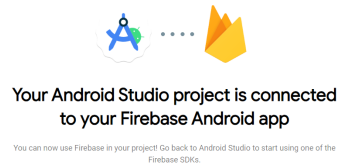
Gambar 13

Setelah di klik akan muncul tampilan seperti pada gambar 14 kemudian pilih hubungkan untuk melanjutkan proses koneksi project Android Studio dengan project Firebase.



Gambar 14

Setelah proses selesai maka akan muncul tampilan seperti Gambar 15 pada web browser. Kemudian pada Android Studio tombol **“Connect your app to Firebase”** akan terdisable dan tidak bisa di klik



Gambar 15

Setelah berhasil login, anda bisa langsung tekan tombol Firebase. Setelah itu akan muncul halaman baru dan klik Mulai.

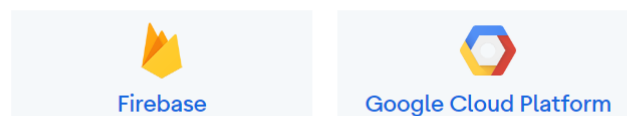
## Berhasil!

Anda berhasil login ke Android Studio.

Untuk melanjutkan, kembali ke Android Studio.



Jelajahi layanan Google yang kini dapat Anda gunakan di aplikasi Android:

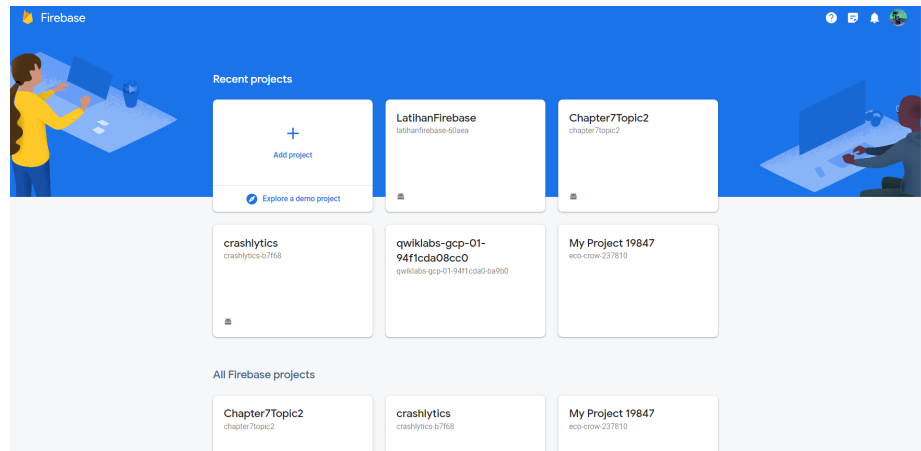


## Login ke Android Studio

Gambar 16

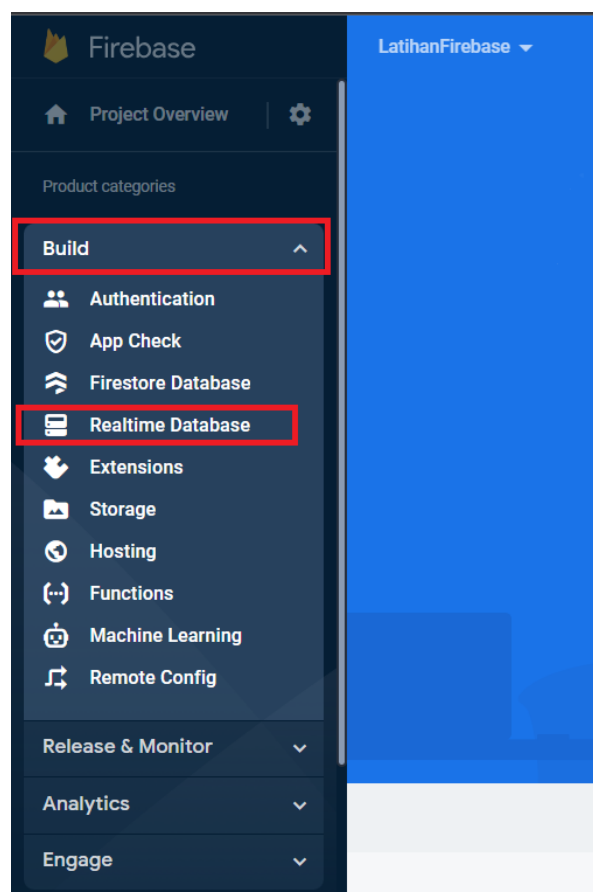


Jika sudah muncul tampilan seperti Gambar 17 maka tekan project yang baru saja dibuat.



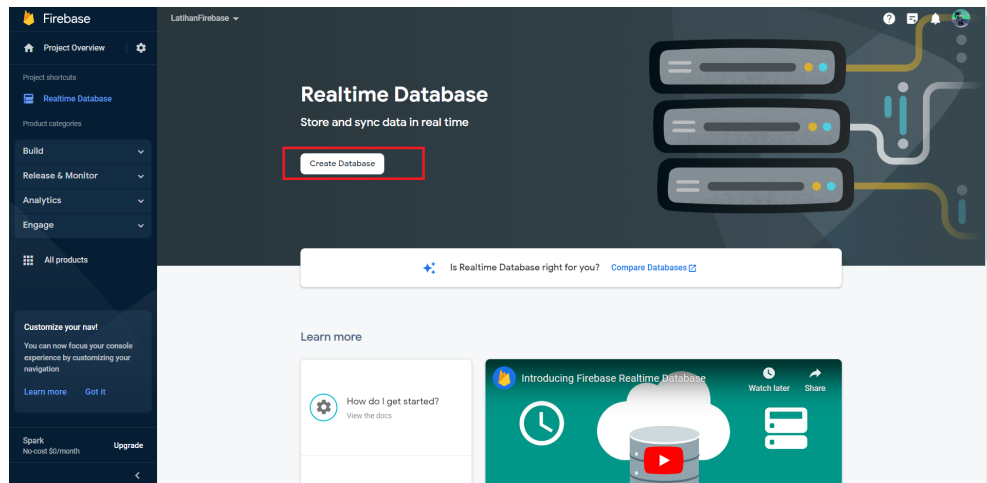
Gambar 17

Selanjutnya pada sidebar, tekan Build dan pilih Realtime Database.



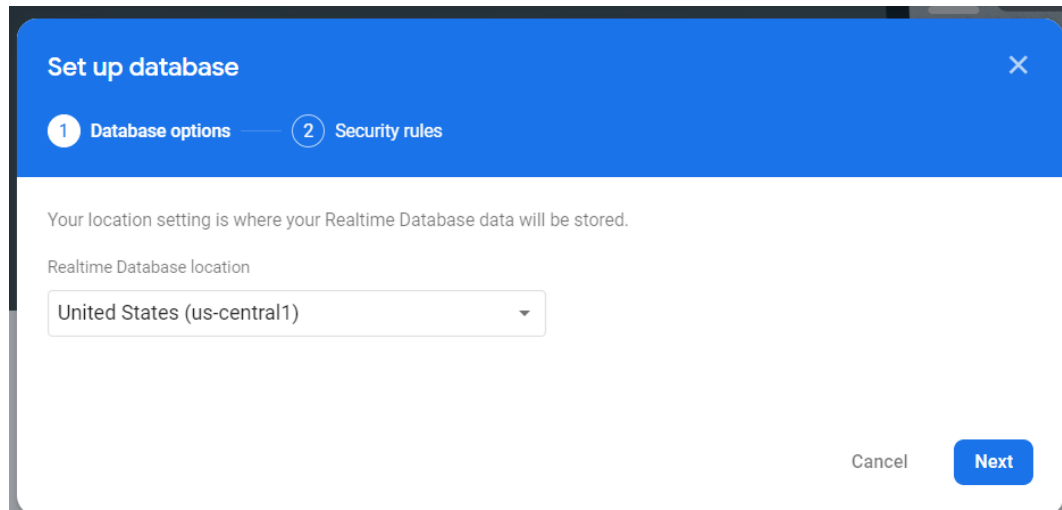
Gambar 18

Setelah itu tekan create database seperti Gambar 19.



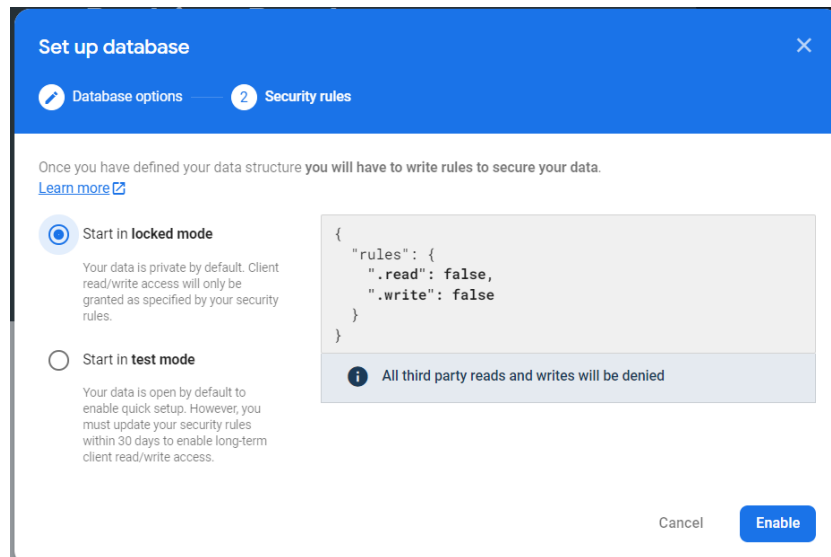
Gambar 19

Pada tahap ini tekan next saja



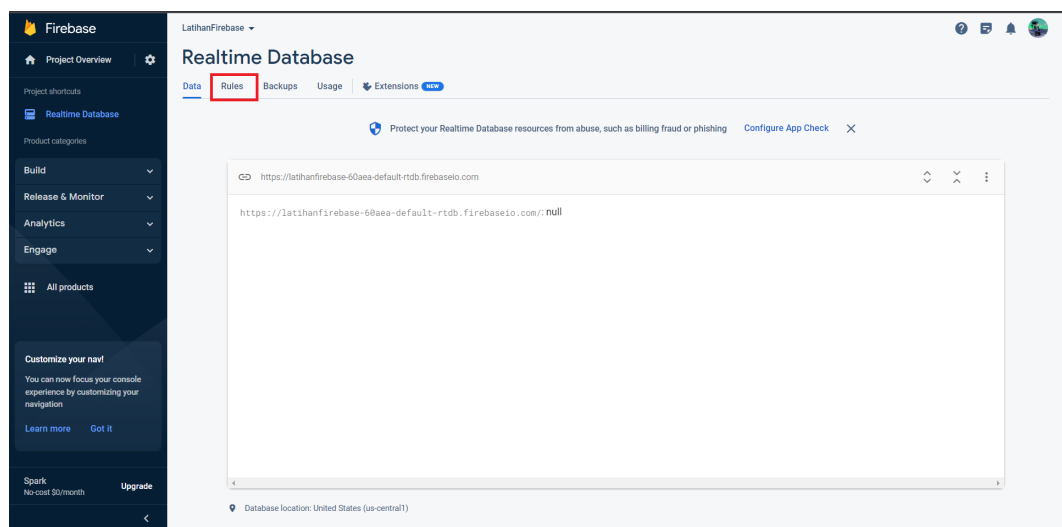
Gambar 20

Pada tahap ini tekan tombol enable, dan tunggu sampai proses pembuatan database selesai.



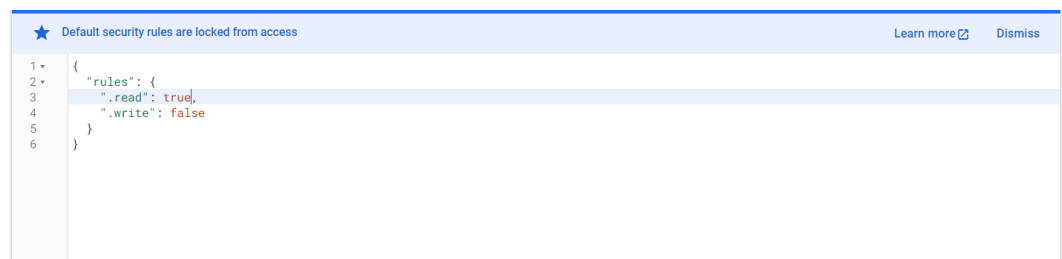
Gambar 21

Setelah proses pembuatan database selesai, selanjutnya pergi ke tab Rules dan buat value read menjadi true seperti Gambar 23.



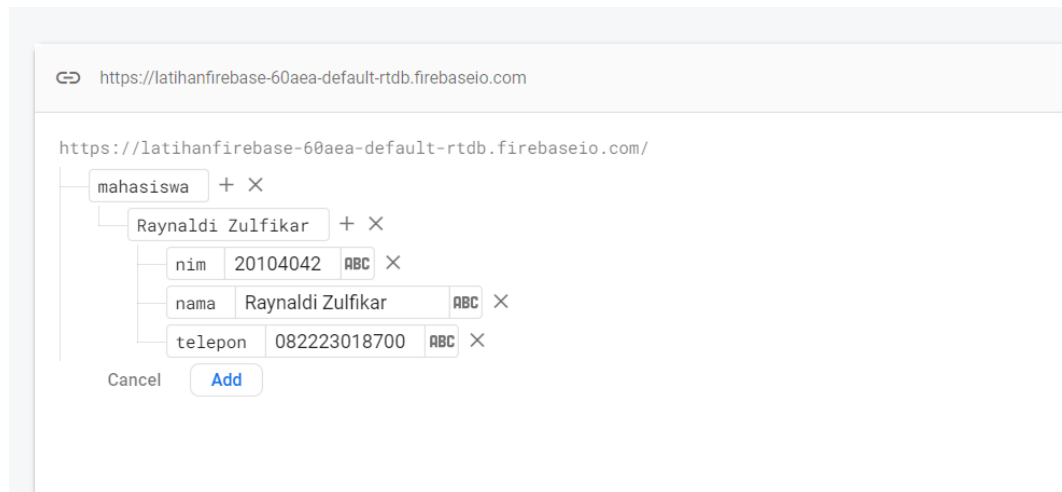
Gambar 22

Tekan Publish.



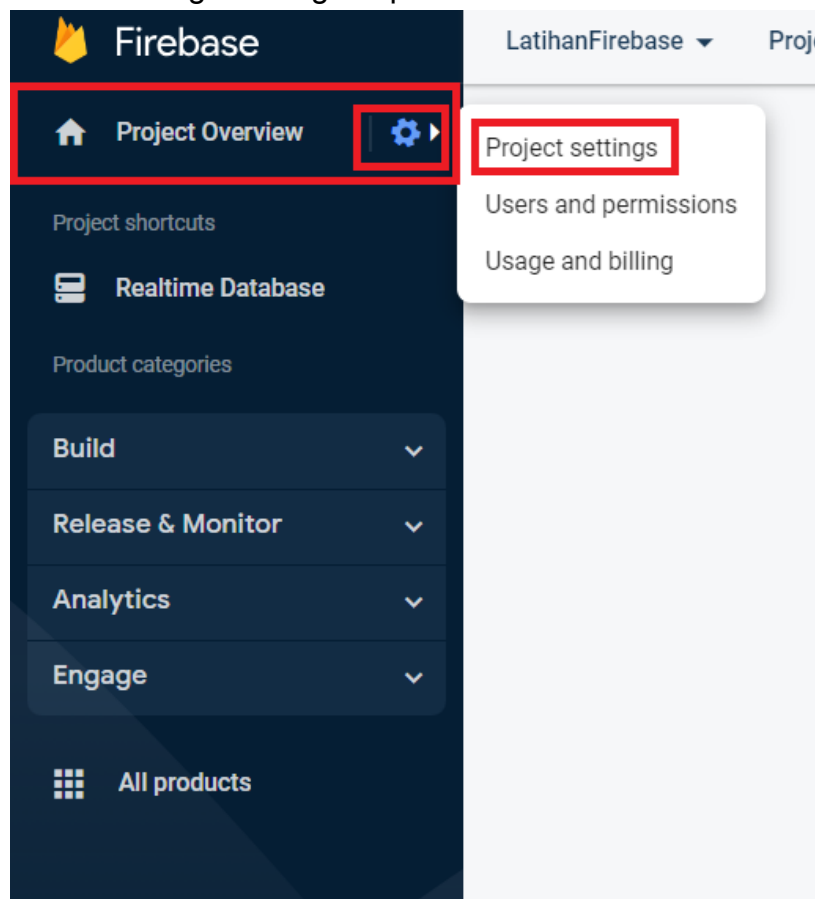
Gambar 23

Setelah itu kita kembali ke tab data, dan buat data menjadi seperti ini. Jika sudah maka tekan Add.



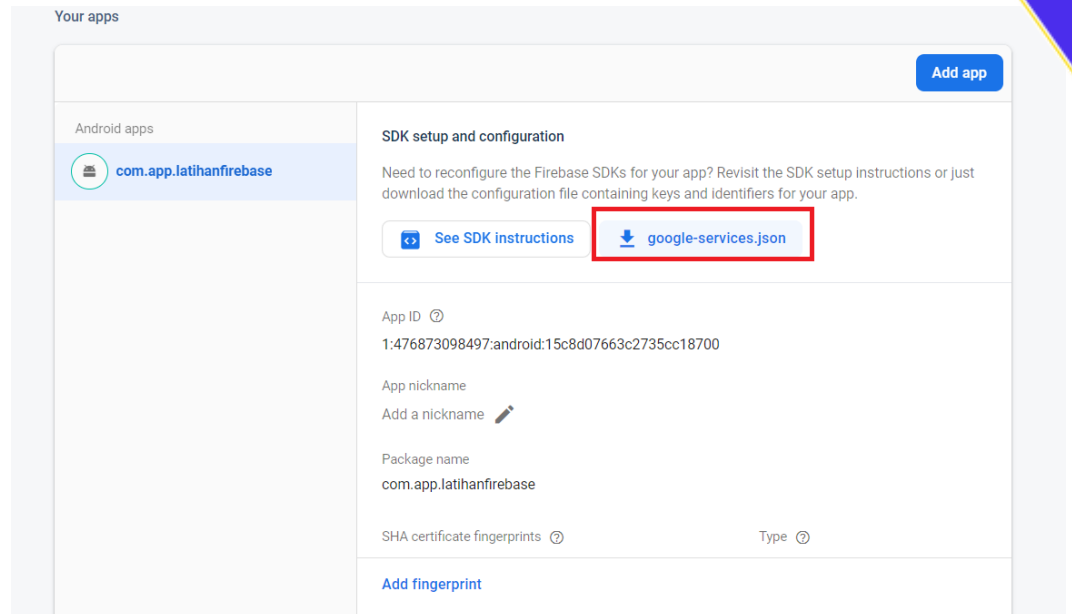
Gambar 24

Kemudian untuk integrasi ke aplikasi kita, kita membutuhkan sebuah file dengan nama `google-services.json` yang dapat kita download dengan melakukan langkah-langkah pada Gambar 25



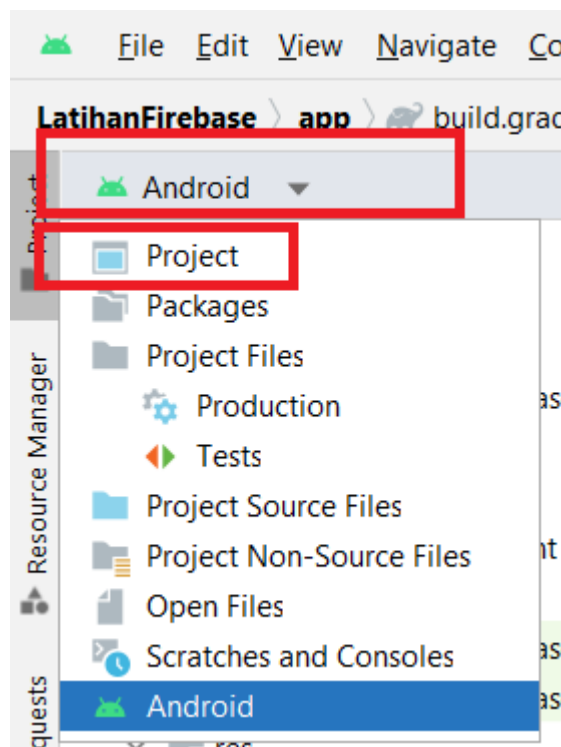
Gambar 25

Tekan tombol google-services.json. Jika sudah di download maka pindahkan file tersebut ke Project folder dengan mengikuti langkah-langkah pada Gambar 27.

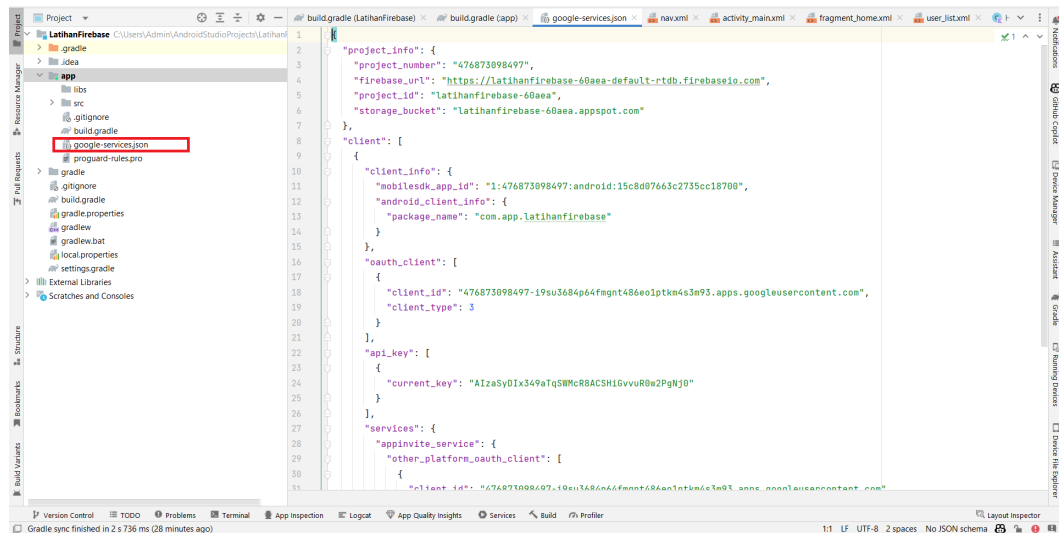


Gambar 26

Tekan dropdown Android dan pilih project. Setelah itu pindahkan Project > App. Jika sudah maka akan seperti Gambar 28.



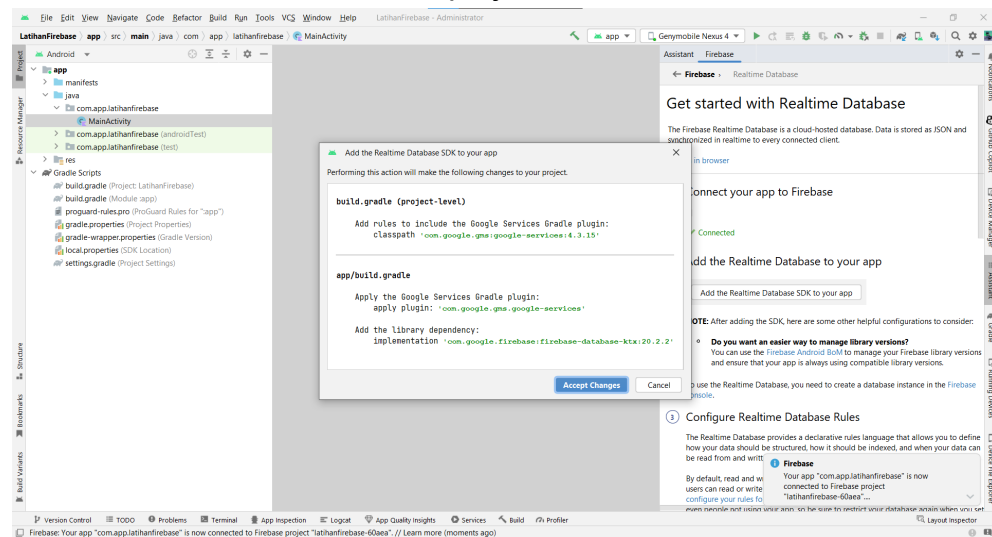
Gambar 27



Gambar 28

### 3. Menambahkan Firebase SDK di project Android Studio

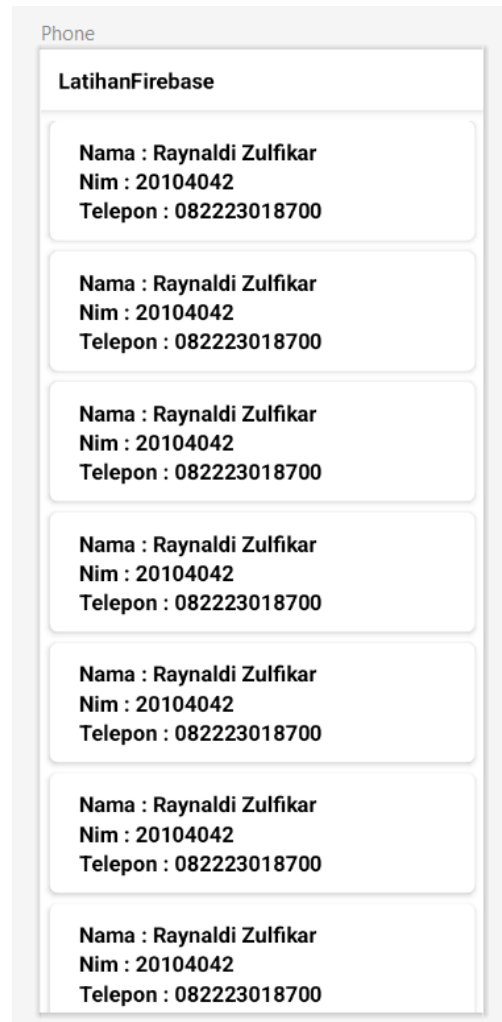
Untuk menggunakan Firebase kita harus menambahkan Firebase SDK pada Android Studio atau menambahkan implementasi pada Gradle Script. Menggunakan cara kedua dengan klik **“Add the Realtime Database SDK to your app”**. **“Connect your app to Firebase”**, kemudian akan muncul tampilan seperti pada gambar 29 lalu klik **“Accept Changes”**. Setelah proses selesai maka Firebase SDK sudah berhasil ditambahkan di project Android Studio.



Gambar 29

#### 4. Menambahkan Firebase SDK di project Android Studio

Aplikasi yang akan dibuat pada tutorial ini akan seperti pada gambar 30 dimana pada User Interface Aplikasi adalah menggunakan **Recyclerview** dengan **Cardview**.



Gambar 30

##### 4.1. Setting Gradle

Pada langkah ini kita harus menambahkan tools tambahan pada **build.gradle(Module)** di **project Android Studio**. Penambahan pertama adalah menambahkan dengan menambahkan tools untuk akses firebase database dimana pada langkah 3 tiga sudah dilakukan maka akan muncul syntax bawah ini pada gradle

```
implementation 'com.google.firebase:firebase-database:20.2.2'
```

Apabila syntax diatas belum ada kita dapat menambahkan secara manual dengan cara mengetikan. Setelah itu implementasi **viewBinding**.

```
buildFeatures{  
    viewBinding true  
}
```

## 4.2. Setting Layout

### 4.2.1. Layout

Pada syntax layout yang digunakan terdapat dua buah file .xml dimana pada file tersebut dinamai dengan **activity\_main.xml** dan **data\_item.xml**.

#### 4.2.1.1. activity\_main.xml

**activity\_main.xml** merupakan layout yang digunakan untuk **Activity Main** atau **Activity utama** yang akan digunakan. Untuk syntax dari **activity\_main.xml** dapat dilihat dibawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <androidx.fragment.app.FragmentContainerView
        android:id="@+id/fragmentContainerView"

android:name="androidx.navigation.fragment.NavHostFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:defaultNavHost="true"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:navGraph="@navigation/nav" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

#### 4.2.1.2. user\_list.xml

Sedangkan **user\_list.xml** digunakan untuk layout Item yang akan dimunculkan pada **Home Fragment**. Untuk syntax dari **user\_list.xml** dapat dilihat pada syntax dibawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    android:id="@+id/cardView"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:elevation="6dp"
    app:cardCornerRadius="8dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="10dp"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <RelativeLayout
        android:layout_width="match_parent"
```



```

        android:layout_margin="15dp"
        android:layout_height="wrap_content">

        <TextView
            android:id="@+id/txtNama"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:layout_marginStart="12dp"
            android:text="Nama : Raynaldi Zulfikar"
        />

        <TextView
            android:id="@+id/txtNim"
            android:layout_below="@id/txtNama"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:layout_marginStart="12dp"
            android:text="Nim : 20104042"
        />

        <TextView
            android:id="@+id/txtTelepon"
            android:layout_below="@id/txtNim"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:layout_marginStart="12dp"
            android:text="Telepon : 082223018700"
        />

    </RelativeLayout>

</androidx.cardview.widget.CardView>

```

#### 4.2.1.3. fragment\_home.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".view.HomeFragment">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"

```

```

app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent">

<com.google.android.material.appbar.AppBarLayout
    android:id="@+id/toolBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white">
    <androidx.appcompat.widget.Toolbar
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/app_name"
            android:textColor="@color/black"
            android:textSize="20sp"
            android:textStyle="bold"
        />
    </androidx.appcompat.widget.Toolbar>
</com.google.android.material.appbar.AppBarLayout>

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvUser"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/toolBar"
    android:layout_marginTop="10dp"
    tools:listitem="@layout/user_list">

</androidx.recyclerview.widget.RecyclerView>

</RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

### 4.3. Kotlin File .kt

Pada file Kotlin terdapat 4 file yang digunakan yaitu **HomeAdapter.kt**, **Mahasiswa.kt**, **MainActivity.kt**, dan **HomeFragment.kt**. **Mahasiswa.kt** digunakan untuk menyimpan data yang didownload dari Firebase sebelum ditampilkan pada Adapter.

#### 4.3.1. HomeAdapter.kt

File HomeAdapter.kt berfungsi sebagai adapter untuk RecyclerView dalam tampilan HomeFragment. Adapter digunakan untuk menghubungkan data dengan tampilan item dalam RecyclerView. Dalam kasus ini, HomeAdapter.kt menghubungkan data Mahasiswa dengan tampilan item daftar pengguna di RecyclerView.

```
package com.app.latihanfirebase.adapter
```

```

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.app.latihanfirebase.model.Mahasiswa
import com.app.latihanfirebase.databinding.UserListBinding

class HomeAdapter (private val dataMhs :
ArrayList<Mahasiswa>) :
RecyclerView.Adapter<HomeAdapter.ViewHolder>() {

    class ViewHolder (val binding : UserListBinding) :
RecyclerView.ViewHolder(binding.root)

        override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): ViewHolder {

            val view =
UserListBinding.inflate(LayoutInflater.from(parent.context),
parent, false)

            return ViewHolder(view)

        }

        override fun getItemCount(): Int {

            return dataMhs.size

        }

        override fun onBindViewHolder(holder: ViewHolder,
position: Int) {

            holder.binding.txtNama.text = dataMhs[position].nama

            holder.binding.txtNim.text = dataMhs[position].nim

            holder.binding.txtTelepon.text =
dataMhs[position].telepon

        }

        fun setData(newData: ArrayList<Mahasiswa>) {

            dataMhs.clear()

            dataMhs.addAll(newData)

            notifyDataSetChanged()

        }

    }

```

#### 4.3.2. Mahasiswa.kt

Data class Mahasiswa digunakan untuk merepresentasikan objek mahasiswa dengan atribut-atribut seperti nim, nama, dan telepon. Data class digunakan untuk menyimpan data dan biasanya tidak memiliki logika bisnis yang kompleks.

```
package com.app.latihanfirebase.model

data class Mahasiswa(val nim : String = "", val nama : String = "", val telepon : String = "")
```

Kemudian **HomeFragment.kt** berfungsi sebagai fragmen yang menampilkan halaman utama aplikasi. Fragmen ini berisi tampilan dan logika yang berkaitan dengan tampilan halaman utama, seperti tampilan daftar mahasiswa.

```
package com.app.latihanfirebase.view

import android.os.Bundle
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import androidx.fragment.app.Fragment
import androidx.recyclerview.widget.LinearLayoutManager
import com.app.latihanfirebase.adapter.HomeAdapter
import com.app.latihanfirebase.databinding.FragmentHomeBinding
import com.app.latihanfirebase.model.Mahasiswa
import com.google.firebase.database.*

class HomeFragment : Fragment() {
    private lateinit var binding: FragmentHomeBinding
    private lateinit var adapter: HomeAdapter
    private lateinit var databaseRef: DatabaseReference

    override fun onCreateView(
        inflater: LayoutInflater, container: ViewGroup?,
        savedInstanceState: Bundle?
    ): View? {
        binding = FragmentHomeBinding.inflate(inflater, container, false)
        return binding.root
    }

    override fun onViewCreated(view: View, savedInstanceState: Bundle?) {
        super.onViewCreated(view, savedInstanceState)

        adapter = HomeAdapter(ArrayList())
        binding.rvUser.adapter = adapter
        binding.rvUser.layoutManager =
            LinearLayoutManager(requireContext())
    }
}
```

```

        databaseRef =
        FirebaseDatabase.getInstance().getReference("mahasiswa")
        databaseRef.addValueEventListener(object :
        ValueEventListener {
            override fun onDataChange(dataSnapshot: DataSnapshot)
            {
                val mahasiswaList = ArrayList<Mahasiswa>()
                for (snapshot in dataSnapshot.children) {
                    val mahasiswa =
                    snapshot.getValue(Mahasiswa::class.java)
                    mahasiswa?.let { mahasiswaList.add(it) }
                }
                adapter.setData(mahasiswaList)
            }

            override fun onCancelled(databaseError:
            DatabaseError) {
                // Handle database error
            }
        })
    }
}

```

Video Youtube : <https://youtu.be/caPkNt4vFj4>

Link Github : [https://github.com/raynaldizf/LatihanFirebase\\_GET](https://github.com/raynaldizf/LatihanFirebase_GET)