









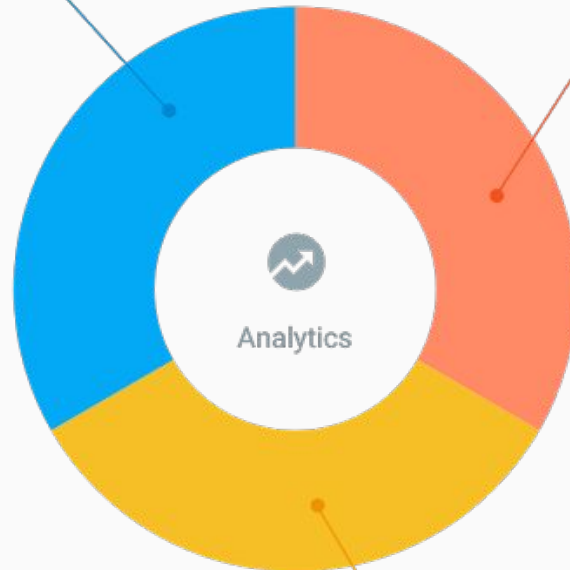
Praktikum Firebase

Zulfakar Muhammad








DEVELOP

-  Realtime Database
-  Authentication
-  Cloud Messaging
-  Storage
-  Hosting
-  Remote Config
-  Test Lab
-  Crash Reporting



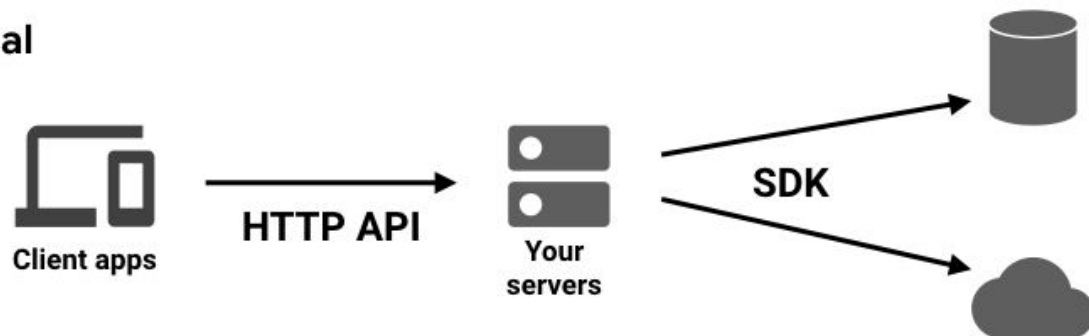
GROW

-  Notifications
-  App Indexing
-  Dynamic Links
-  Invites
-  AdWords

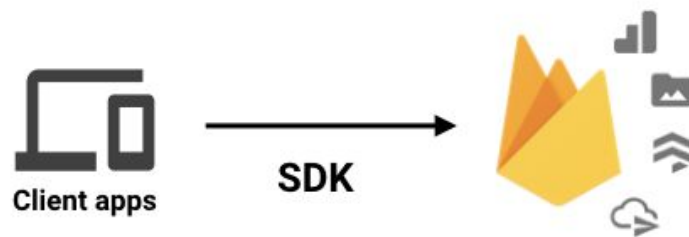
EARN

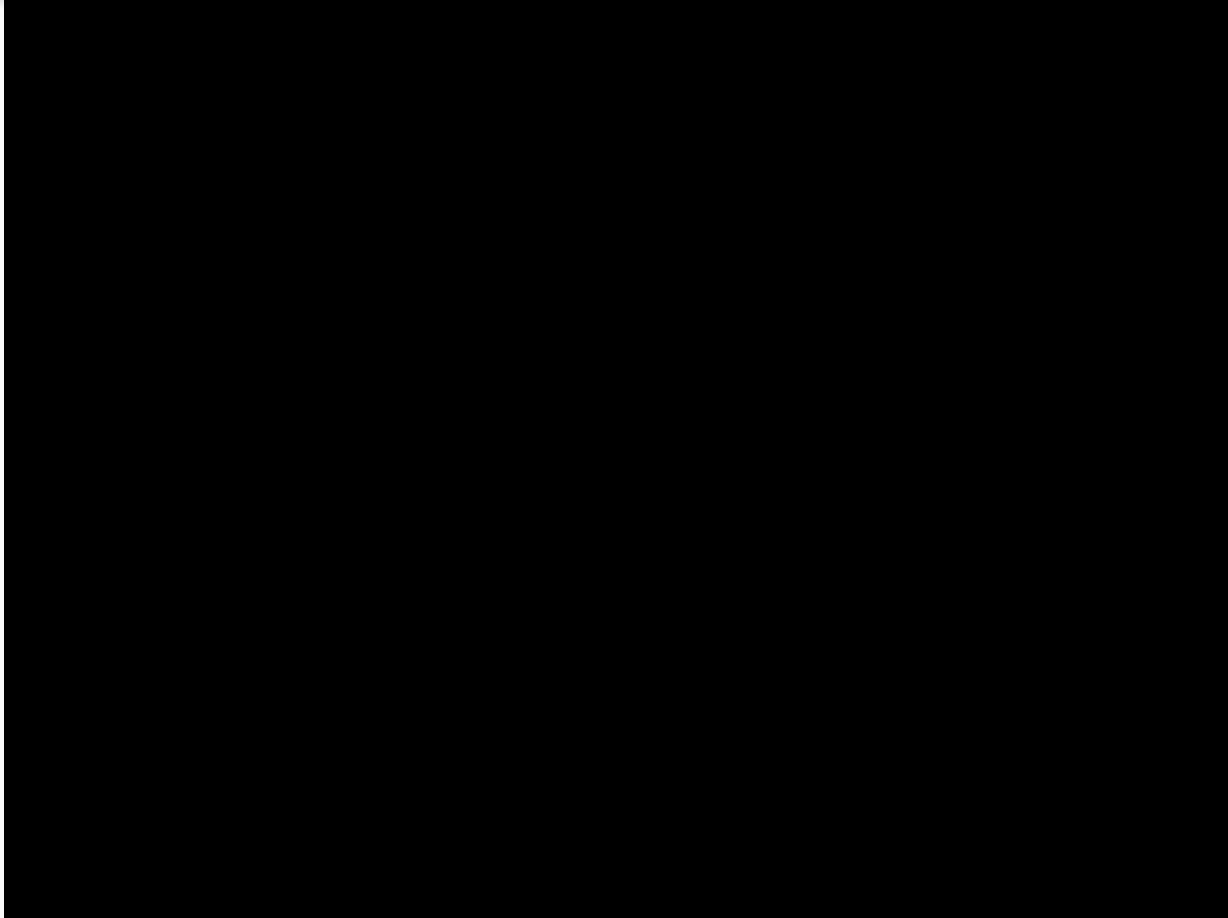


Traditional



Firebase





So, let's start...

- ❖ Buat proyek baru seperti biasanya, perhatikan bahasa yang digunakan (JAVA)
- ❖ Perlu diperhatikan untuk package name

Package name
`com.namaKamu.namaApps`

- ❖ Cek email student teman-teman



Zulfakar Muhammad
zulfakar.ytb@gmail.com

amikom.ac.id,

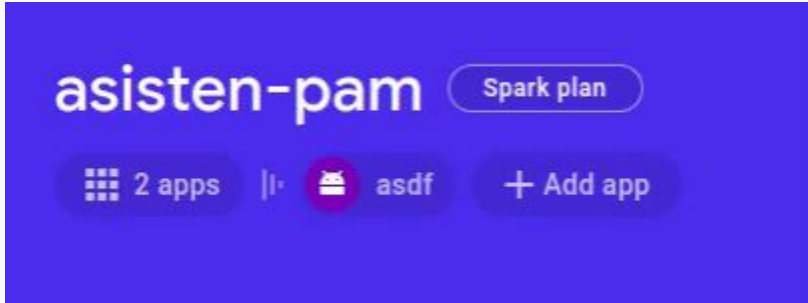
I invited you to be an owner of the Firebase project **asisten-pam**. Click the button below to to accept my invitation.

[Learn more about Firebase](#)

See pending invitation

Adding your apps to existing firebase project

- ❖ Tambahkan aplikasi baru ke dalam projek firebase



- ❖ Masukan *Package name* yang sama dengan langkah ke 2 tadi

1 Register app

Android package name ?

Adding your apps to existing firebase project

❖ Download config filenya dan ikuti instruksi

2 Download config file

Instructions for Android Studio below | [C++](#)

 Download google-services.json

Switch to the **Project** view in Android Studio to see your project root directory.

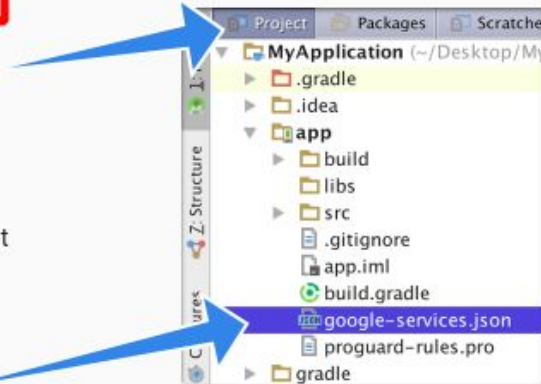
Move the google-services.json file you just downloaded into your Android app module root directory.



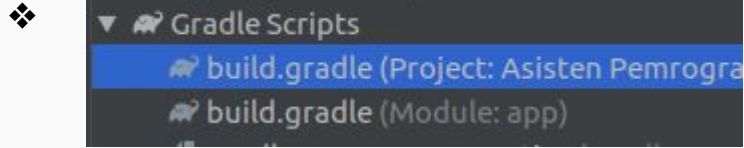
google-services.json

Previous

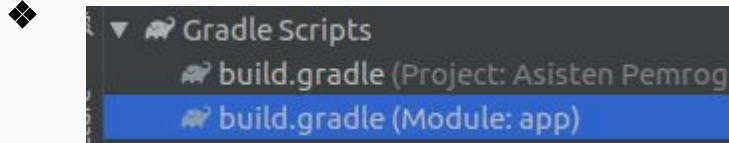
Next



Setting up our apps dependency



Pada file build.gradle yang (Project: ...) tambahkan baris berikut ke dalam block dependency:
classpath 'com.google.gms:google-services:4.3.2'



Kemudian pada file build.gradle (Module: app) tambahkan baris berikut ke dalam block dependency:
implementation 'com.google.firebase:firebase-database:19.2.0'
implementation 'androidx.recyclerview:recyclerview:1.1.0'

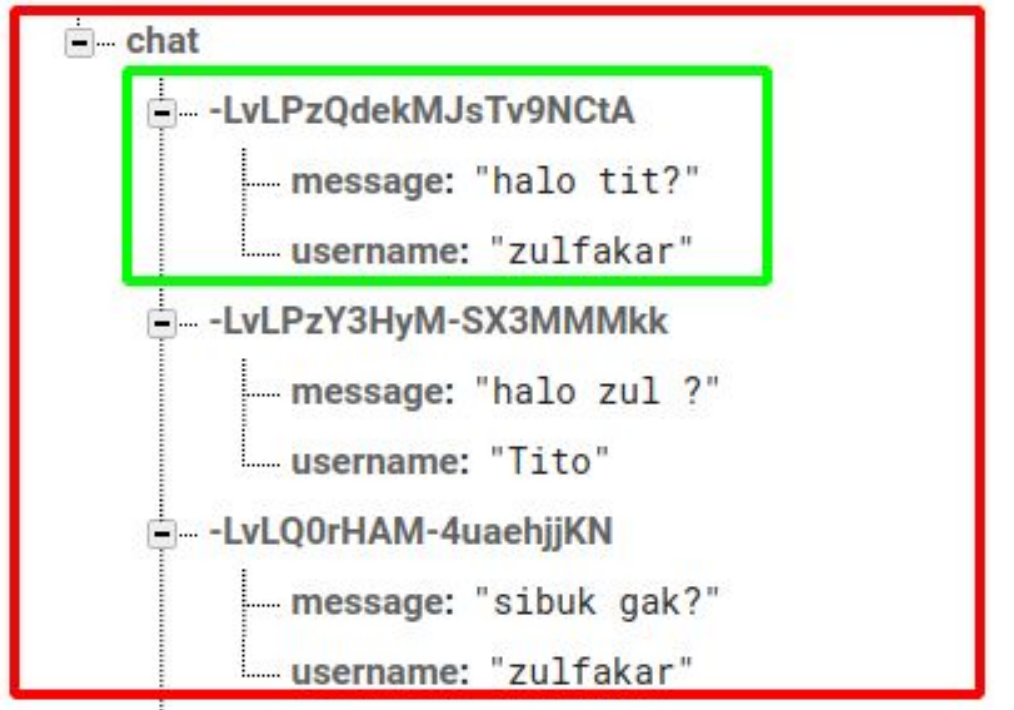
Dan tambahkan baris berikut di bagian paling bawah:
apply plugin: 'com.google.gms.google-services'

Flashback

Minggu kemarin kita belajar tentang `httpRequest` dengan menggunakan `OkHttp` dan response yang kita dapat dari server adalah berupa `JSON`. Selain itu kita juga melakukan parsing terhadap `JSON` tersebut agar bisa ditampilkan ke dalam User Interface aplikasi kita.

Firestore Realtime Database pun juga demikian, jika diperhatikan gambar disamping, warna merah merupakan `JSONArray` dan warna hijau merupakan `JSONObject`.

Maka dari itu kita juga harus melakukan parsing seperti minggu kemarin.



Creating the activity_main layout.

Untuk skenario kali ini kita akan membutuhkan 3 View di dalam layout kita. Masing-masing adalah RecyclerView (untuk menampilkan data yang berupa list), EditText (untuk memasukan pesan yang akan dikirim), dan Button (untuk mengirim pesan).

Untuk ViewGroup yang digunakan adalah LinearLayout (namun jika temen-temen sudah faham dengan ConstraintLayout maka menggunakannya lebih disarankan). Here's the xml code:

<https://gist.github.com/muzafakar/73aeb4401022aa03fa5d5a8f89a4cd96>

*Jangan lupa ritual (inisialisasi) di MainActivity.java



Apakah setiap kali
berurusan dengan
JSON harus
parsing manual?



“Pakai firebase tidak sesulit itu”



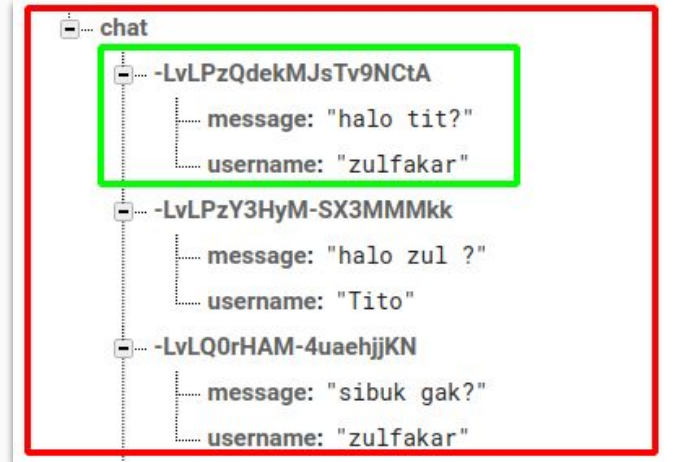
- Fernando

Cukup buatlah sebuah class dengan property yang sama dengan JSON nya

Class Chat.java

```
public class Chat {  
    private String username;  
    private String message;  
  
    // Constructor  
    // setter dan getter  
}
```

JSON



Buat file java baru dengan nama Chat.java

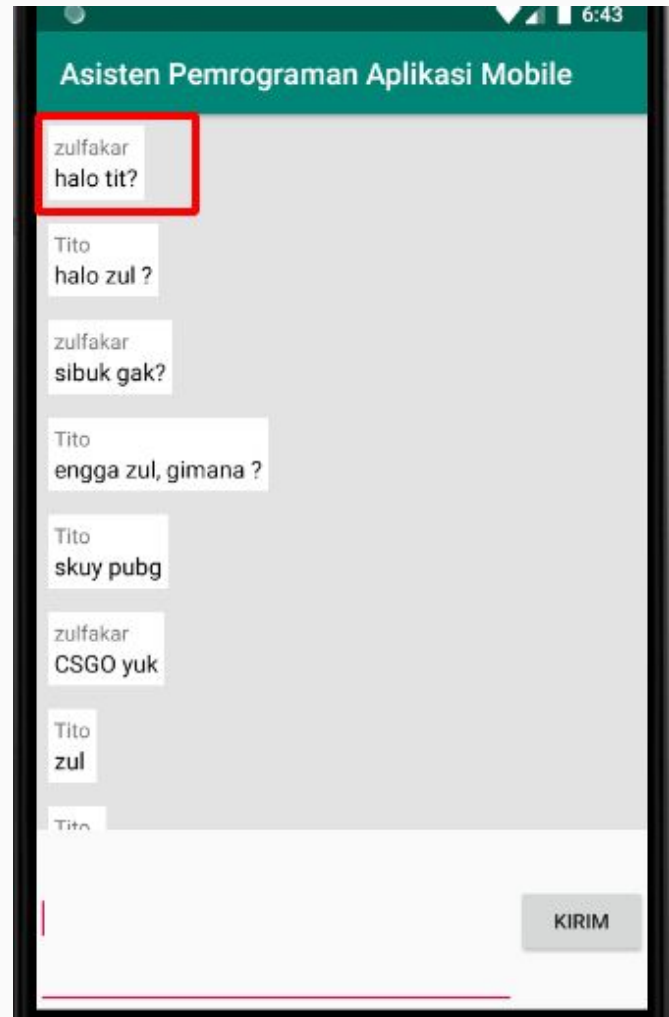
Source code Chat.java : <https://gist.github.com/muzafakar/78eabeb3886904f301285e8d3737ca52>

Setting up the RecyclerView #1

Karena kita menggunakan RecyclerView maka kita juga harus membuat layout yang akan mewakili 1 baris dari keseluruhan chat.

Buat file layout baru di folder res > layout. Dengan nama item_chat.xml
Source code item_chat.xml:
<https://gist.github.com/muzafakar/52ba5dff6144dc93e67b3d1589420ba8>

Jika diperhatikan di gambar, yang setiap chat itu diwakili oleh layout item_chat.xml yang di populate dari data chat.



Setting up the RecyclerView #2

Step berikutnya adalah membuat adapter untuk RecyclerView nya.

Adapter ini berfungsi untuk mengubah data JSON ke dalam bentuk UI yang bisa dilihat oleh user.

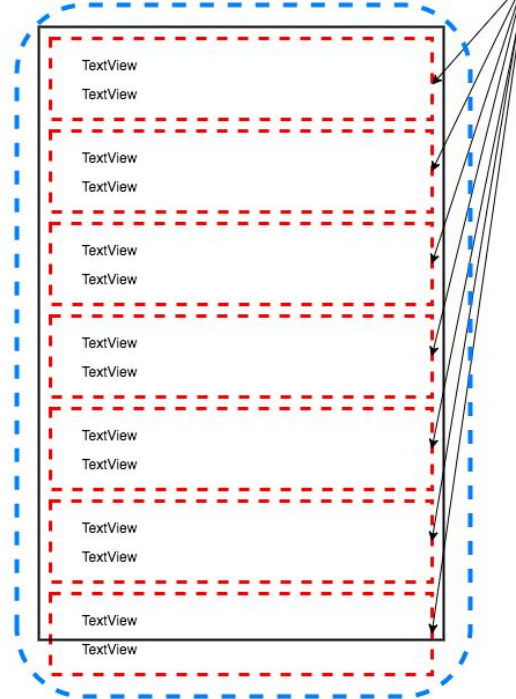
Buat sebuah file java baru dengan nama ChatAdapter.java.

Source codenya :

<https://gist.github.com/muzafakar/78eabeb3886904f301285e8d3737ca52>

RecyclerView

Recycles data within a set of ViewHolders



ViewHolders

Populates the View elements in the UI

Adapter

Binds data to ViewHolders



Data Collection

Populates the views in the ViewHolders



Setelah ini kita ngoding bareng.

Pastikan progress code di MainActivity.java sudah deklarasi dan inisialisasi objek ya.

```
public class MainActivity extends AppCompatActivity {  
    private Button btnSend;  
    private EditText etMessage;  
    private RecyclerView rvChat;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        btnSend = findViewById(R.id.btnSend);  
        etMessage = findViewById(R.id.etMessage);  
        rvChat = findViewById(R.id.rvChat);  
    }  
}
```