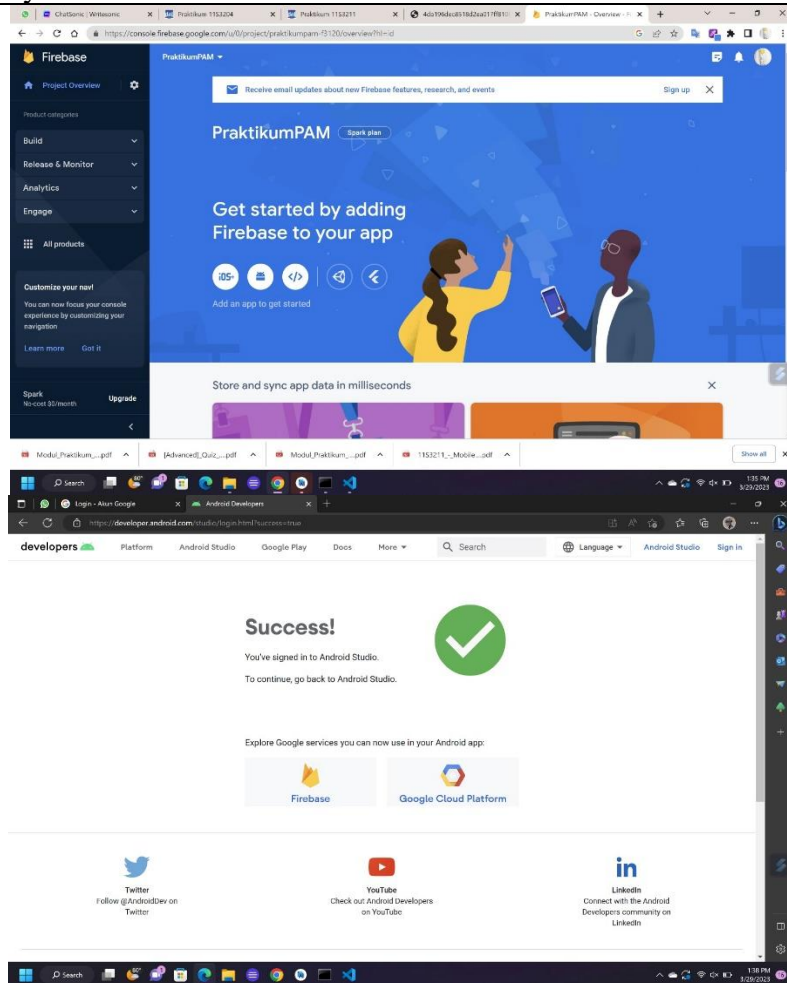


<b>NIM</b>	<b>:</b>	11S20018
<b>Nama</b>	<b>:</b>	Ruth Aulya Silalahi
<b>Kelas</b>	<b>:</b>	13F1
<b>Video Source Code</b>	<b>:</b>	<a href="https://youtu.be/n0S_0oqBcPs">https://youtu.be/n0S_0oqBcPs</a>

## A. Firebase Realtime Database

### Create Database

Laporkan hasilnya.



### Firestore – Android Studio Integration

Laporkan hasilnya.

The screenshot displays the process of integrating Firebase into an Android application using Android Studio. It includes the Firebase account selection interface, the terminal output confirming a successful build, the 'Add an Android app' configuration screen, and a file copy dialog for the 'google-services.json' file.

**Terminal Output:**

```
MD5: 4E:E0:02:DE:EF:1D:B4:1F:CC:F5:81:68:98:87:C1:6A
SHA1: 88:72:F7:87:03:53:DD:61:C6:DC:62:AD:68:8E:3D:8E:CB:DB:B0:EE
SHA-256: CD:23:13:25:41:81:39:6F:51:A0:56:A4:B6:70:78:18:00:71:28:8C:88:96:F1:E2:94:7F:6C:2B:D3:B8:28:47
Valid until: Wednesday, November 13, 2052
-----
BUILD SUCCESSFUL in 8s
1 actionable task: 1 executed
PS C:\Users\ASUS\AndroidStudioProjects\PAMPraktikum1011S20018>
```

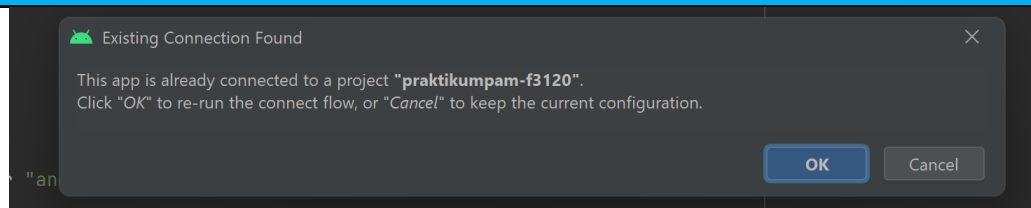
**Add an Android app configuration:**

- 1 Register app
- Android package name: `com.example.pampraktikum10_11s20018`
- App nickname (optional): `My Android App`
- Debug signing certificate SHA-1 (optional): `88:72:F7:87:03:53:DD:61:C6:DC:62:AD:68:8E:3D:8E:CB:DB:B0:EE`

**File Copy Dialog:**

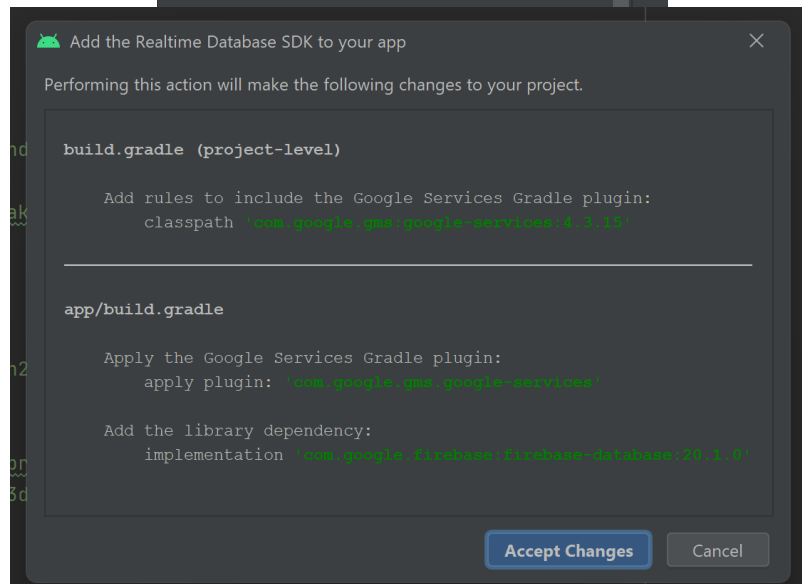
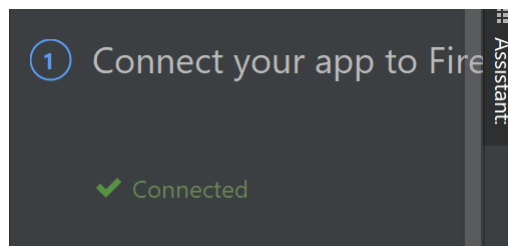
- Copy file: `C:\Users\ASUS\Downloads\hapus nanti\google-services.json`
- New name: `google-services.json`
- To directory: `C:\Users\ASUS\AndroidStudioProjects\PAMPraktikum1011S20018\app`
- Use Ctrl+Space for path completion
- ☒ Open in editor

### Firestore – Android Studio Integration

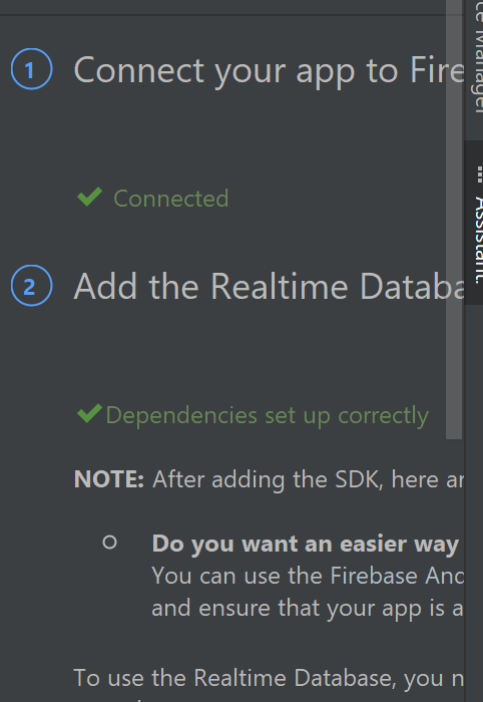


## Your Android Studio project is connected to your Firebase Android app

You can now use Firebase in your project! Go back to Android Studio to start using one of the  
Firestore SDKs.



### Firestore – Android Studio Integration



1 Connect your app to Firebase

✓ Connected

2 Add the Realtime Database

✓ Dependencies set up correctly


**NOTE:** After adding the SDK, here are some things you should know:

- Do you want an easier way? You can use the Firebase Android Studio plugin and ensure that your app is a valid Android application.

To use the Realtime Database, you need to add the following code to your app's `AndroidManifest.xml` file:

```

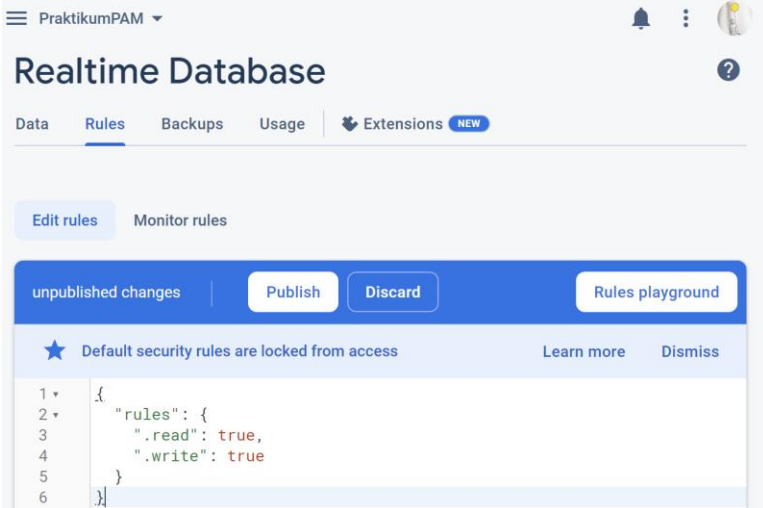
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    
```

Public-facing name ? project-72480949453 

Support email ?

### Adding Data into Real-time Database via Firebase Console

Laporkan hasilnya.



PraktikumPAM

## Realtime Database

Data Rules Backups Usage Extensions **NEW**

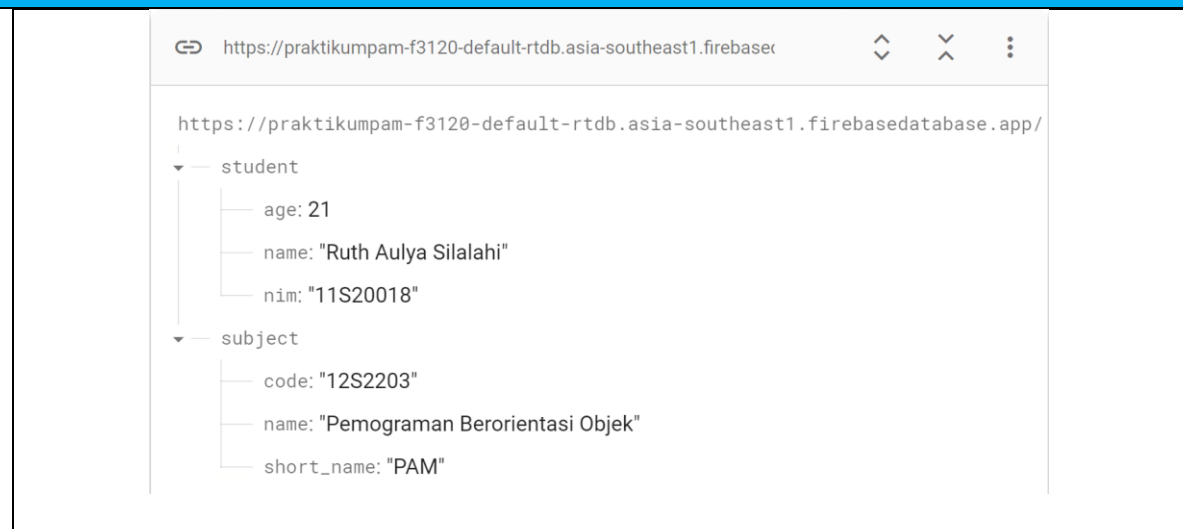
Edit rules Monitor rules

unpublished changes Publish Discard Rules playground

★ Default security rules are locked from access [Learn more](#) [Dismiss](#)

```

1 {
2   "rules": {
3     ".read": true,
4     ".write": true
5   }
6 }
    
```

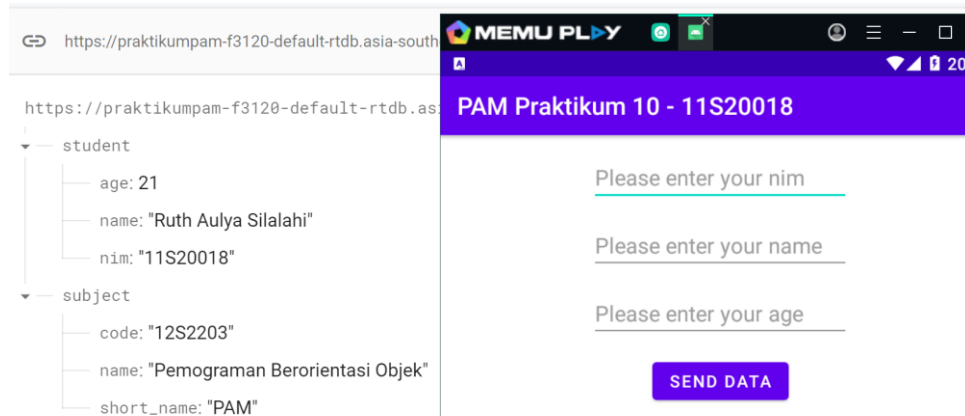
**Adding Data into Real-time Database via Firebase Console****Data Adding Data into Real-time Database via Android App**

Cari tahu kegunaan kode program pada line 17, 18, dan 34. Laporkan hasilnya.

Pertama, kode membuat instance dari `FirebaseDatabase` dengan memanggil metode `FirebaseDatabase.getInstance()`.

Selanjutnya, kode membuat `DatabaseReference` dengan memanggil metode `reference.child("student")`. Ini mengambil referensi ke node "student" dalam database Firebase.

Terakhir, kode menetapkan nilai untuk "nim" dengan memanggil `reference.child("nim").setValue(nim)`. Ini menetapkan nilai dari variabel `nim` ke node "nim" dalam "student" di database Firebase.



### Data Adding Data into Real-time Database via Android App

```

https://praktikumpam-f3120-default-rtdb.asia-southeast1.firebaseio.com/

├── student
│   ├── age: 21
│   ├── name: "Amelia Agustina Hutajulu"
│   └── nim: "11S20044"
└── subject
    ├── code: "12S2203"
    ├── name: "Pemograman Berorientasi Objek"
    └── short_name: "PAM"
        
```

PAM Praktikum 10 - 11S20018

SEND DATA

### Data Types Retrieving Real-Time Data From Database

Cari tahu kegunaan dari kode program pada line 38 sampai 57. Laporkan hasilnya.

Pertama, kode mengambil DatabaseReference dengan memanggil referenceGet yang sebelumnya sudah didefinisikan sebelumnya. Selanjutnya, kode menambahkan listener ke DatabaseReference dengan memanggil addValueEventListener. Ini akan memantau perubahan data di bawah node "student" di Firebase Realtime Database dan akan memanggil onDataChange setiap kali ada perubahan.

Di dalam onDataChange, kode membaca data yang tersimpan di bawah node "student" menggunakan snapshot. Ini kemudian diambil nilainya dan disimpan ke dalam variabel nim, name, dan age. Setelah itu, nilai dari variabel nim, name, dan age ditampilkan pada EditText etNIM, etName, dan etAge masing-masing, dan juga pada TextView tvNIM, tvName, dan tvAge. Jika terjadi kesalahan dalam membaca data dari Firebase Realtime Database, maka onCancelled akan dipanggil dan akan menampilkan pesan kesalahan.

```

https://praktikumpam-f3120-default-rtdb.asia-southeast1.firebaseio.com/

├── student
│   ├── age: 21
│   ├── name: "Amelia Agustina Hutajulu"
│   └── nim: "11S20044"
└── subject
    ├── code: "12S2203"
    ├── name: "Pemograman Berorientasi Objek"
    └── short_name: "PAM"
        
```

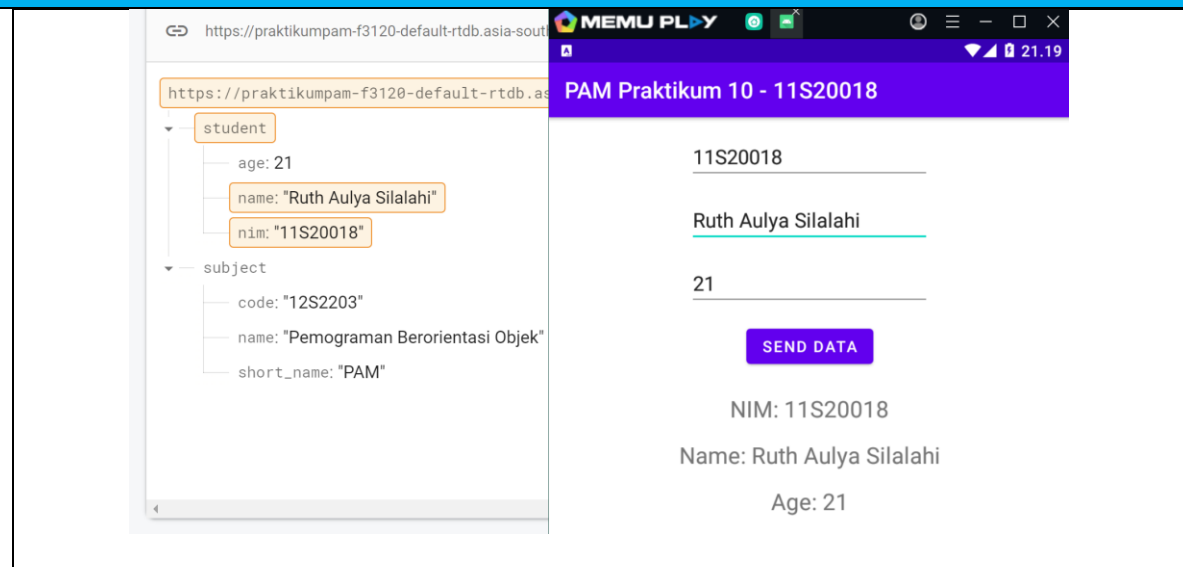
PAM Praktikum 10 - 11S20018

SEND DATA

NIM: 11S20044

Name: Amelia Agustina Hutajulu

Age: 21

**Data Types Retrieving Real-Time Data From Database****B. User Registration App using Firebase****Adding User**

Cari tahu kegunaan kode program pada line 13, 15, 16, 41 sampai 58 pada `AddUserActivity.kt`

`lateinit var addUserBinding: ActivityAddUserBinding` digunakan untuk mendeklarasikan variabel yang akan digunakan untuk mengikat view ke dalam Activity.

`val db : FirebaseDatabase = FirebaseDatabase.getInstance()` digunakan untuk membuat instance dari `FirebaseDatabase`.

`val dbRef : DatabaseReference = db.reference.child("users")` digunakan untuk membuat objek `DatabaseReference` yang merepresentasikan node "users" di dalam database `Firebase Realtime`.

`dbRef.child(id).setValue(user)` digunakan untuk menambahkan data pengguna baru ke dalam `Firebase Realtime Database` dengan menetapkan nilai dari variabel `user` ke node yang memiliki ID yang sama dengan nilai dari variabel `id`. Dalam hal ini, nilai dari `id` merupakan sebuah string yang merepresentasikan ID dari data pengguna, dan nilai dari `user` merupakan objek yang merepresentasikan data pengguna.

`addOnCompleteListener` digunakan untuk menambahkan listener yang akan dipanggil saat operasi menulis data ke database selesai dilakukan.

Jika `task.isSuccessful` maka akan menampilkan pesan `Toast` yang mengindikasikan bahwa pengguna baru telah ditambahkan ke database.

Jika terdapat kesalahan saat menambahkan data pengguna baru ke dalam database, maka akan menampilkan pesan `Toast` yang berisi informasi tentang kesalahan tersebut.

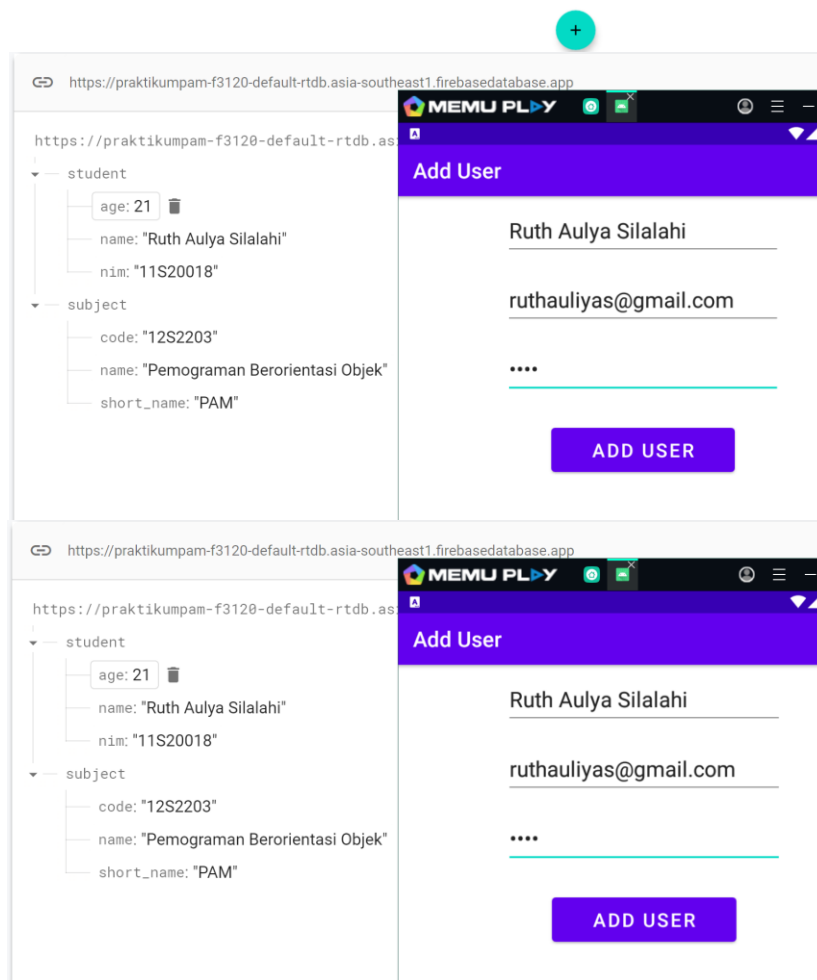
`val intent = Intent(this, MainActivity::class.java)` digunakan untuk membuat objek `Intent` yang akan memulai Activity `MainActivity` saat tombol "Add User" ditekan.

`startActivity(intent)` digunakan untuk memulai Activity `MainActivity`

### Adding User

Laporkan hasilnya

PAM Praktikum 10 - 11S20018





**Showing Retrieving Data on the RecyclerView****Analisis kegunaan dari kode program UserAdapter.kt**

Kode program di atas merupakan implementasi dari adapter untuk RecyclerView pada aplikasi Android. Adapter ini akan menampilkan daftar pengguna (users) yang terdiri dari nama, alamat email, dan kata sandi pengguna pada tampilan aplikasi. Dalam implementasinya, kelas UsersAdapter menerima dua parameter yaitu context dan userList. Parameter context digunakan untuk menyimpan konteks aplikasi, sedangkan parameter userList adalah daftar pengguna yang akan ditampilkan pada RecyclerView.

Kelas UsersViewHolder digunakan untuk menyimpan binding (ikatan) dari layout item pengguna pada RecyclerView, sehingga dapat mengakses elemen-elemen tampilan yang ada pada layout item tersebut. Metode onCreateViewHolder digunakan untuk membuat instance dari UsersViewHolder yang akan menampung layout item pengguna. Pada metode ini, layout item pengguna di-inflate dari file layout UsersItemBinding, yang kemudian diikat ke instance UsersViewHolder yang baru dibuat.

Metode onBindViewHolder digunakan untuk mengikat data pengguna pada posisi tertentu pada RecyclerView dengan instance UsersViewHolder yang sesuai. Pada metode ini, data pengguna diambil dari daftar userList pada posisi tertentu, dan kemudian ditampilkan pada elemen-elemen tampilan pada layout item pengguna yang sesuai. Metode getItemCount digunakan untuk mengembalikan jumlah item pada daftar userList. Metode getUserId digunakan untuk mendapatkan ID pengguna pada posisi tertentu pada daftar userList. Metode ini akan digunakan pada kelas MainActivity untuk menghapus pengguna yang dipilih dari database Firebase.

**Analisis kegunaan dari fungsi retrieveDataFromDatabase()**

Fungsi retrieveDataFromDatabase() digunakan untuk mengambil data pengguna dari database Firebase dan menampilkan data tersebut ke dalam RecyclerView. Fungsi ini menggunakan addValueEventListener untuk mendapatkan data secara real-time dan menghapus daftar pengguna sebelum menambahkan kembali pengguna yang diperbarui dari snapshot baru. Kemudian, fungsi ini membuat instance dari adapter UsersAdapter dengan data pengguna yang diperbarui dan menetapkan adapter ke RecyclerView. Dengan cara ini, setiap kali ada perubahan pada data di database, tampilan RecyclerView akan diperbarui secara otomatis.

Laporkan hasilnya.

**PAM Praktikum 10 - 11S20018**

**Ruth Aulya Silalahi**

**ruthauliyas@gmail.com**

**1234**

**Showing Retrieving Data on the RecyclerView**

<div><div>PAM Praktikum 10 - 11S20018</div><div>Ruth Aulya Silalahi ruthauliyas@gmail.com 1234</div><div>Grace Natalia Br Sitepu ifs20051@students.del.ac.id 1234</div></div>
---

**Updating Data****Analisis kegunaan dari fungsi getAndSetData() dan fungsi updateData()**

Fungsi getAndSetData() digunakan untuk mengambil data pengguna dari intent dan menentukannya ke EditText di dalam tampilan. Ini memungkinkan pengguna untuk melihat informasi yang sudah ada sebelum mereka memutuskan untuk mengubahnya. Sedangkan fungsi updateData() berfungsi untuk mengambil data pengguna yang telah diubah dari EditText di dalam tampilan dan memperbarui informasi tersebut pada Firebase Realtime Database. Fungsi ini menggunakan id pengguna untuk memperbarui data yang sesuai pada database dan menampilkan pesan Toast ketika operasi selesai berhasil.

Laporkan hasilnya.

<div><div>Update User</div><div>Ruth Aulya Silalahi</div><div>ruthauliyas@gmail.com</div><div>....</div><div>UPDATE USER</div><div>PAM Praktikum 10 - 11S20018</div><div>Ruth Silalahi ruthauliyas@gmail.com 1234</div></div>
---

**Deleting Selected User**

Analisis kode program pada line 37 sampai 58.

Pada `ItemTouchHelper.SimpleCallback`, `0` dan `ItemTouchHelper.LEFT` or `ItemTouchHelper.RIGHT` adalah argumen konstruktor. `0` digunakan untuk menentukan jenis aksi yang didukung oleh `ItemTouchHelper`. Pada kasus ini, tidak ada aksi penggeseran item ke posisi lain yang didukung. `ItemTouchHelper.LEFT` or `ItemTouchHelper.RIGHT` digunakan untuk menentukan jenis gerakan penggeseran item yang akan didukung. Di sini, item akan dihapus dengan menggeser ke arah kanan atau kiri. Metode `onMove` tidak diimplementasikan di dalam kode, karena tidak mendukung aksi penggeseran item ke posisi lain.

Sementara itu, metode `onSwiped` digunakan untuk menangani aksi penggeseran item. Di dalam metode ini, kode mengambil id dari pengguna yang dipilih untuk dihapus, kemudian menghapus item tersebut dari database Firebase menggunakan `dbRef.child(id).removeValue()`. Akhirnya, kode menampilkan pesan Toast yang memberitahukan bahwa pengguna telah dihapus dari database. Terakhir, `ItemTouchHelper` dipasangkan dengan `RecyclerView` menggunakan `attachToRecyclerView(mainBinding.rvUsers)` untuk menerapkan gesture penghapusan pada daftar pengguna yang terdapat pada `RecyclerView`.

Laporkan hasilnya.

**PAM Praktikum 10 - 11S20018**

**Ruth Silalahi**

**ruthauliyas@gmail.com**

**1234**

**Grace Natalia Br Sitepu**

**ifs20051@students.del.ac.id**

**1234**

**PAM Praktikum 10 - 11S20018**

**Grace Natalia Br Sitepu**

**ifs20051@students.del.ac.id**

**1234**

**Deleting All User**

Analisis kegunaan dari fungsi `showDialogMessage()` dan kode program pada line 41 sampai 62. Laporkan hasilnya.

Fungsi `ItemTouchHelper()` digunakan untuk menambahkan gestur penghapusan pada `RecyclerView`. Saat pengguna menggeser elemen ke kanan atau ke kiri, kode akan menghapus item tersebut dari Firebase Realtime Database menggunakan id yang diperoleh dari `ViewHolder`. Setelah item dihapus, pesan Toast akan ditampilkan untuk memberitahu pengguna bahwa item telah dihapus.

Sementara itu, fungsi `showDialogMessage()` digunakan untuk menampilkan dialog konfirmasi penghapusan semua pengguna. Jika pengguna mengonfirmasi penghapusan semua pengguna, kode akan menghapus seluruh item dari Firebase Realtime Database menggunakan `dbRef.removeValue()`. Setelah semua item dihapus, adapter akan diberitahu dengan menggunakan `usersAdapter.notifyDataSetChanged()`. Akhirnya, pesan Toast akan ditampilkan untuk memberitahu pengguna bahwa seluruh item telah dihapus.

**PAM Praktikum 10 - 11S20018**



**Grace Natalia Br Sitepu**

**ifs20051@students.del.ac.id**

**1234**

**Delete All Users**

If click Yes, all users will be deleted, If  
you want to delete a specific user, you  
can swipe the item you want to delete  
right or left

**CANCEL YES**

**PAM Praktikum 10 - 11S20018**

