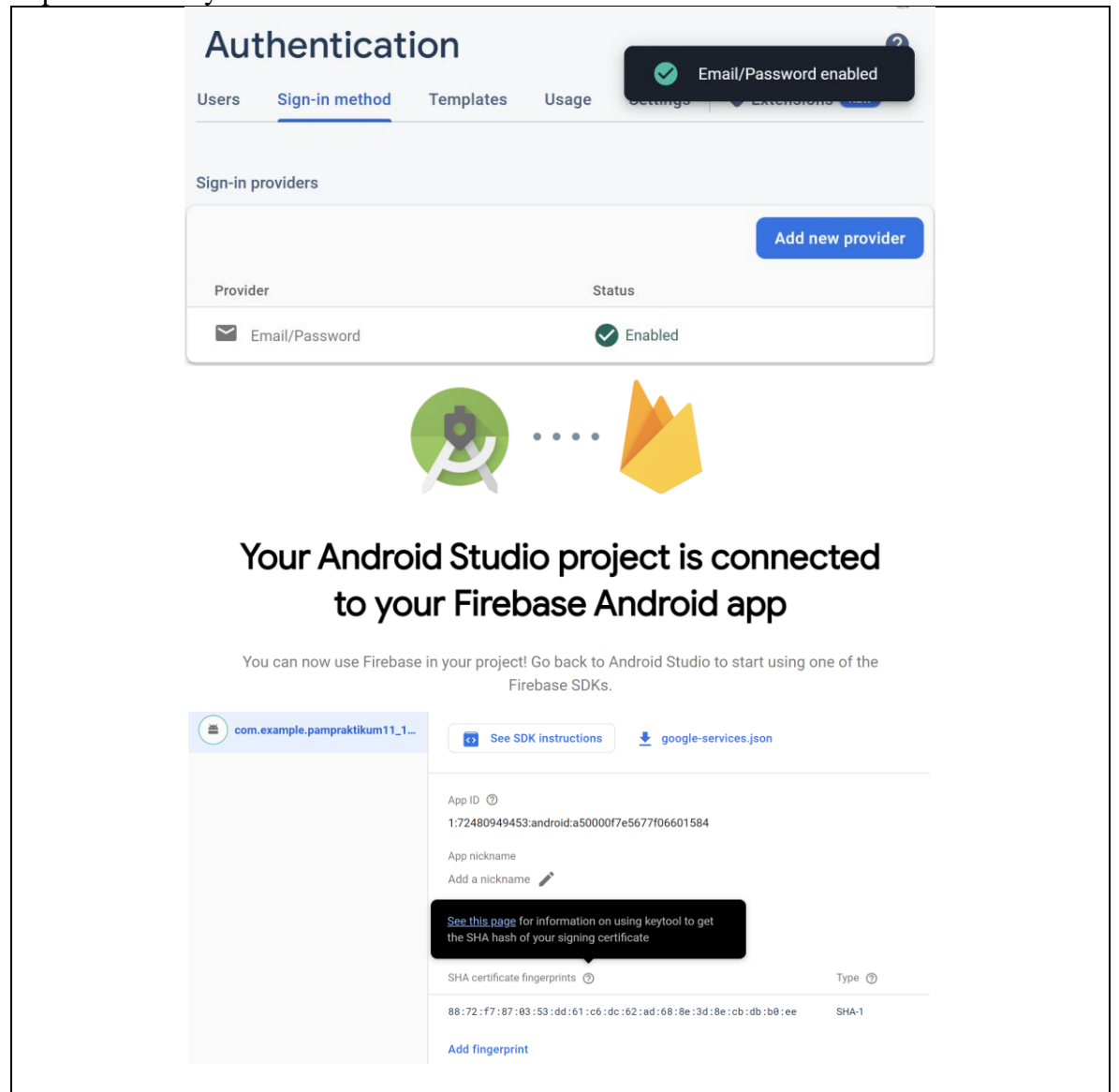


| | | |
|-------------------|---|---|
| NIM | : | 11S20018 |
| Nama | : | Ruth Aulya Silalahi |
| Kelas | : | 13IF1 |
| Video Source Code | : | https://youtu.be/2oJVfGHNwi0 |

A. Authenticaatin with Firebase**Email and Password Authentication**

Laporkan hasilnya.



The screenshot displays the Firebase Authentication console. At the top, a blue header reads "Email and Password Authentication". Below it, a notification banner states "Email/Password enabled" with a green checkmark. The "Sign-in method" tab is selected, showing a table of "Sign-in providers". The table has two columns: "Provider" and "Status". The "Email/Password" provider is listed with a status of "Enabled". Below the table, there is a section titled "Your Android Studio project is connected to your Firebase Android app". This section includes a message: "You can now use Firebase in your project! Go back to Android Studio to start using one of the Firebase SDKs." Below this message, there are links for "See SDK instructions" and "google-services.json". At the bottom, the "App ID" is displayed as "1:72480949453:android:a500007e5677f06601584". The "App nickname" is "Add a nickname". A tooltip points to the "App ID" with the text: "See this page for information on using keytool to get the SHA hash of your signing certificate". Below the app information, the "SHA certificate fingerprints" are listed as "88:72:f7:87:03:53:dd:61:c6:dc:62:ad:68:8e:3d:8e:cb:db:b0:ee" with a type of "SHA-1". A link "Add fingerprint" is also present.

Email and Password Authentication**PAM Praktikum 11 -11S20018****SIGN IN****SIGN UP**

Your account has been created

PAM Praktikum 11 -11S20018

Selamat kamu telah berhasil login.

User Remembering with Firebase

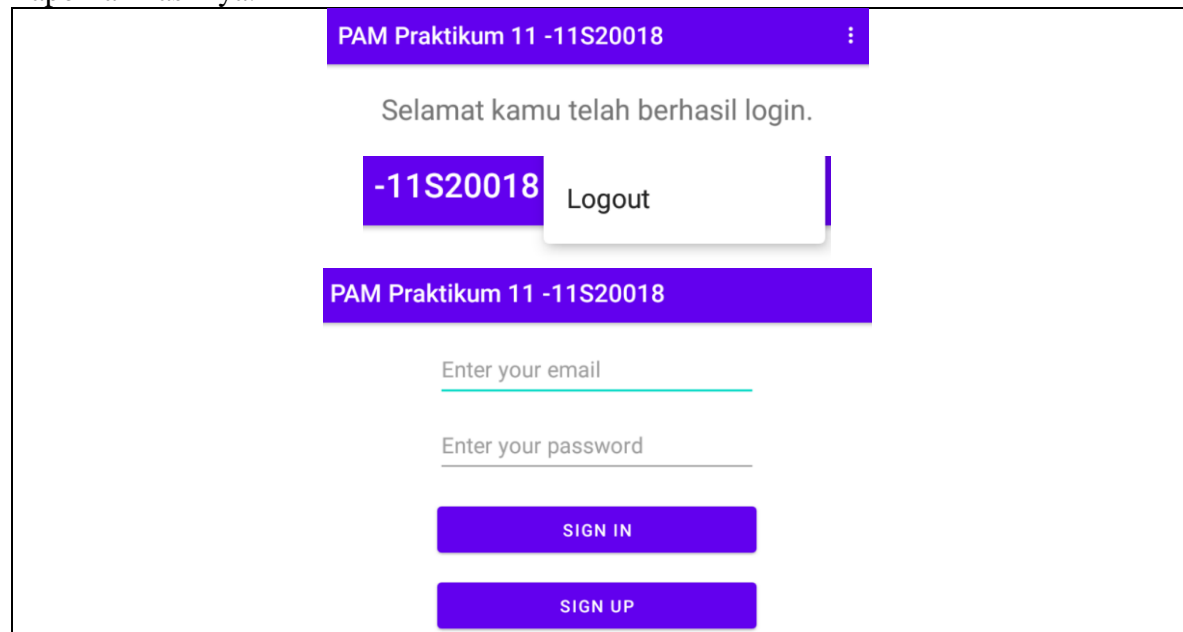
Analisis kegunaan dari kode program pada line 15 sampai 18 dan 21 sampai 26.

onOptionsItemSelected(menu: Menu?): Boolean: Fungsi ini dipanggil ketika menu untuk activity tersebut harus dibuat. Fungsi ini mengambil objek Menu sebagai parameter dan mengembalikan nilai boolean. Fungsi ini menggunakan objek MenuInflater untuk mengambil tata letak menu dari file XML dan memasangnya pada objek Menu yang diberikan sebagai parameter. Kemudian, fungsi mengembalikan nilai true untuk menunjukkan bahwa menu telah berhasil dibuat.

onOptionsItemSelected(item: MenuItem): Boolean: Fungsi ini dipanggil ketika salah satu item di menu yang telah dibuat di atas, dipilih oleh pengguna. Fungsi ini mengambil objek MenuItem sebagai parameter dan mengembalikan nilai boolean. Fungsi ini memeriksa apakah item yang dipilih adalah item dengan ID miLogout. Jika ya, maka fungsi melakukan logout pengguna dengan memanggil `FirebaseAuth.getInstance().signOut()`. Selanjutnya, fungsi membuka LoginActivity dengan membuat objek Intent, memanggil `startActivity(intent)`, dan menutup activity saat ini dengan memanggil `finish()`. Jika item yang dipilih bukan miLogout, maka fungsi mengembalikan nilai false.

User Remembering with Firebase

Laporkan hasilnya.

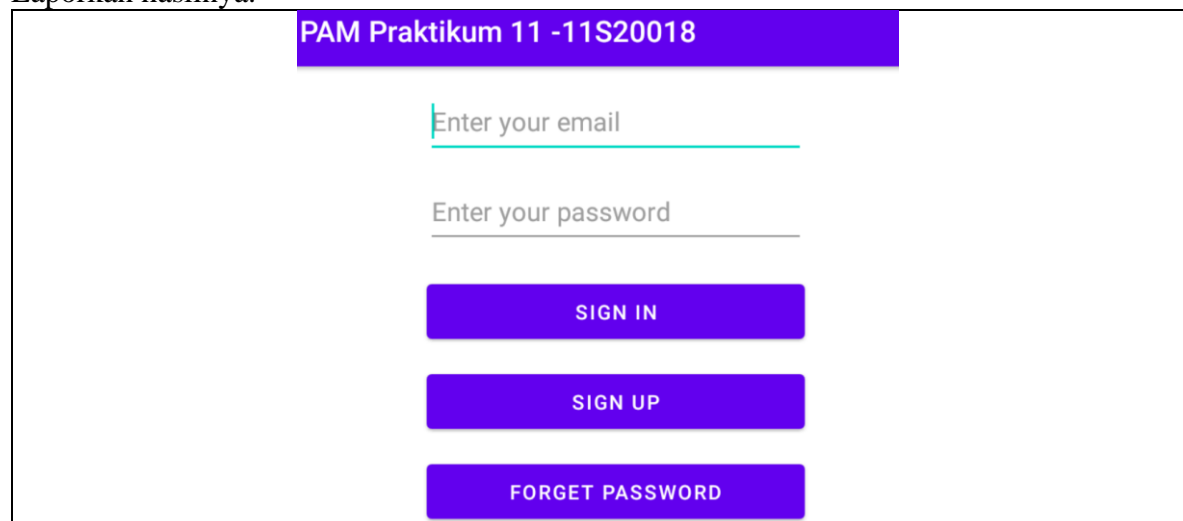


The screenshot shows a mobile application interface. At the top, there is a purple header bar with the text "PAM Praktikum 11 -11S20018" and a three-dot menu icon. Below the header, a message "Selamat kamu telah berhasil login." is displayed. Underneath the message, there is a purple button with the text "-11S20018" and a white button with the text "Logout". Below these buttons, there is another purple header bar with the text "PAM Praktikum 11 -11S20018". Underneath this header, there are two input fields: "Enter your email" and "Enter your password". Below the input fields, there are two purple buttons: "SIGN IN" and "SIGN UP".

Adding Forgot Password OptionKegunaan kode program pada line 23 sampai 32 pada ForgerActivity.kt

Kode tersebut melakukan pengiriman email reset password kepada pengguna yang meminta reset password dengan memanggil fungsi `sendPasswordResetEmail(email)` pada objek `auth`. Fungsi `addOnCompleteListener` dipanggil dengan objek `task` sebagai parameter. Fungsi ini akan dieksekusi setelah proses pengiriman email reset password selesai, baik itu berhasil atau gagal. Jika proses berhasil, maka akan menampilkan pesan singkat pada layar dengan menggunakan `Toast.makeText` dan kemudian menutup activity saat ini dengan memanggil `finish()`. Jika proses gagal, maka tidak ada tindakan yang diambil dalam kode tersebut.

Laporkan hasilnya.



The screenshot shows a mobile application interface. At the top, there is a purple header bar with the text "PAM Praktikum 11 -11S20018". Below the header, there are two input fields: "Enter your email" and "Enter your password". Below the input fields, there are three purple buttons: "SIGN IN", "SIGN UP", and "FORGET PASSWORD".

Adding Forgot Password Option**PAM Praktikum 11 -11S20018**ruthauliyas@gmail.com**RESET MY PASSWORD**

We sent a password reset mail to
your mail address

Hello,

Follow this link to reset your project-72480949453 password for your ruthauliyas@gmail.com account.

https://praktikumpam-f3120.firebaseio.com/_/auth/action?mode=resetPassword&oobCode=IGaJe4AG&apiKey=AlzaSyB-J9DWxEqN6B2lwt28VZUTgNmN-LgthZ8&lang=en

If you didn't ask to reset your password, you can ignore this email.

Thanks,

Your project-72480949453 team

Reset your password

for ruthauliyas@gmail.com

New password

.....

**SAVE****Password changed**

You can now sign in with your new
password

PAM Praktikum 11 -11S20018

Selamat kamu telah berhasil login.

Authentication with Phone

Analisis kegunaan dari fungsi `signingSMSCode()` dan `signingWithPhoneAuthCredential()`.

Fungsi `signingSMSCode()` dan `signingWithPhoneAuthCredential(credential: PhoneAuthCredential)` digunakan untuk memverifikasi kode SMS yang diterima oleh pengguna pada aktivitas pendaftaran nomor telepon. Pada fungsi `signingSMSCode()`, kode verifikasi yang diterima dari SMS akan dibandingkan dengan kode yang dimasukkan oleh pengguna melalui `EditText`. Jika kode yang dimasukkan oleh pengguna sesuai dengan kode verifikasi, maka fungsi `signingWithPhoneAuthCredential(credential: PhoneAuthCredential)` akan dipanggil dengan memasukkan kredensial telepon sebagai parameter.

Fungsi `signingWithPhoneAuthCredential(credential: PhoneAuthCredential)` kemudian digunakan untuk masuk dengan menggunakan kredensial telepon yang valid. Jika masuk berhasil, maka pengguna akan diarahkan ke `MainActivity`. Namun, jika masuk gagal, maka pesan kesalahan akan ditampilkan melalui `Toast`.

Laporkan hasilnya.

PAM Praktikum 11 -11S20018

Enter your email

Enter your password

SIGN IN

SIGN UP

FORGET PASSWORD

SIGNIN WITH PHONE NUMBER

20:37 87,7KB/d 4G

120.firebaseio.com

Verifying you're not a robot...

Authentication with Phone

20:37

6,6KB/d

4G

VoLTE

46

←

Google

1

20:37

Enter: [897216](#) App: [praktikumpa-m-f3120.firebaseio.com](#)
jK3LnRN3jkF

20:38

19,9KB/d

4G

VoLTE

46

PAM Praktikum 11 -11S20018

+62

SEND SMS CODE

897216

VERIFIY TO SIGN IN

20:38

5,8KB/d

4G

VoLTE

46

PAM Praktikum 11 -11S20018

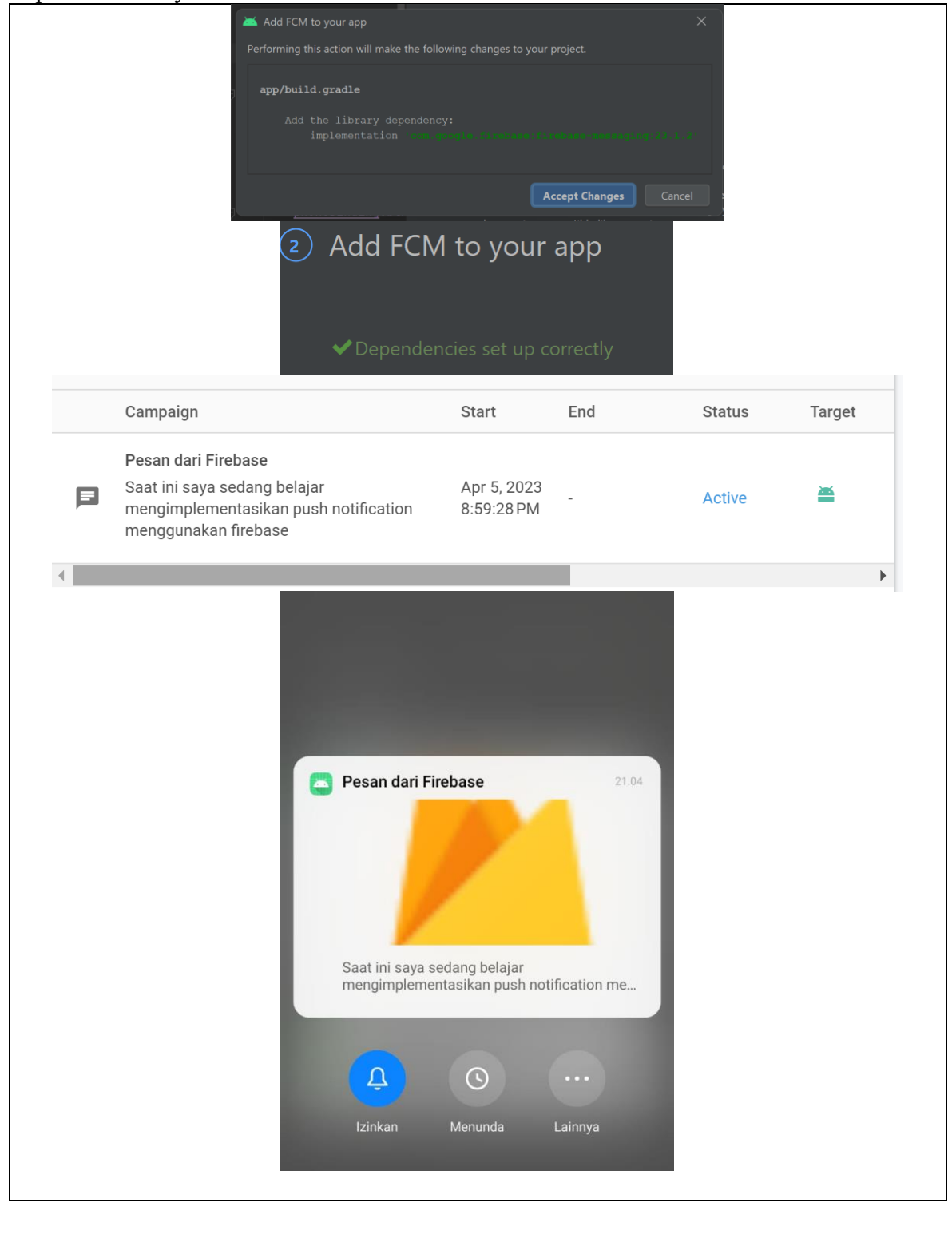
⋮

Selamat kamu telah berhasil login.

B. Push Notification with Firebase



Push Notification

Laporkan hasilnya.



The screenshot displays the Android Studio environment. At the top, a blue header reads "Push Notification". Below it, the text "Laporkan hasilnya." is visible. The main workspace shows a dialog box titled "Add FCM to your app" with a close button (X). The dialog contains the text: "Performing this action will make the following changes to your project." and a code block for `app/build.gradle` showing the addition of the dependency: `implementation 'com.google.firebase:firebase-messaging:21.1.0'`. Below the dialog, a green checkmark indicates "Dependencies set up correctly".

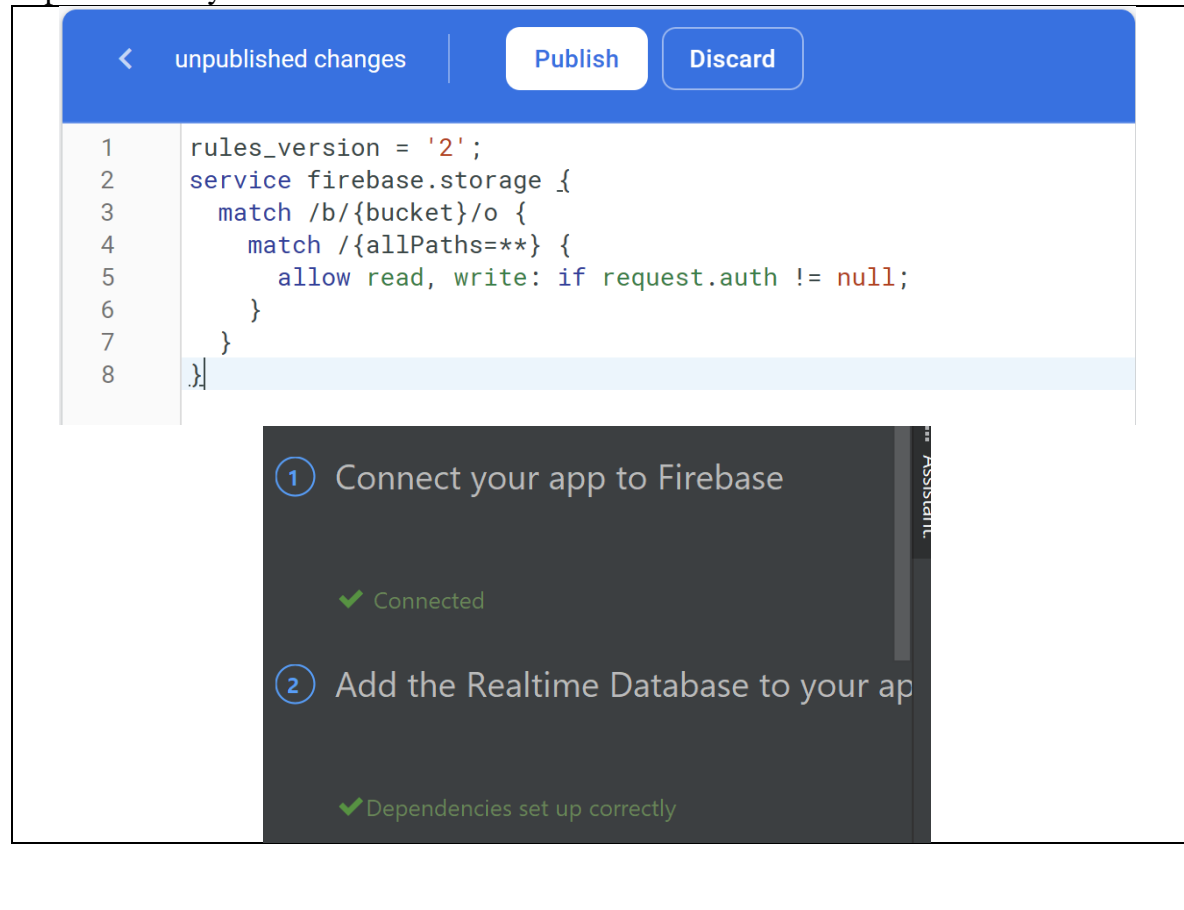
Below the dialog, a table lists Firebase messages:

| Campaign | Start | End | Status | Target |
|---|------------------------|-----|--------|---|
| Pesan dari Firebase  Saat ini saya sedang belajar mengimplementasikan push notification menggunakan firebase | Apr 5, 2023 8:59:28 PM | - | Active |  |

At the bottom, a preview of a push notification is shown. The notification has a title "Pesan dari Firebase" and a timestamp "21.04". The body text reads: "Saat ini saya sedang belajar mengimplementasikan push notification me...". At the bottom of the notification card are three buttons: "Izinkan" (with a bell icon), "Menunda" (with a clock icon), and "Lainnya" (with a three-dot icon).

C. Cloud Storage**Setup Cloud Storage**

Laporkan hasilnya.

**Selecting an Image from the Gallery**


Analisis kegunaan fungsi `chooseImage()`. Laporkan hasilnya.

Dari implementasi kode yang diberikan, fungsi `chooseImage()` berfungsi untuk memeriksa apakah izin untuk membaca penyimpanan eksternal telah diberikan kepada aplikasi. Jika izin belum diberikan, fungsi ini akan meminta izin. Jika izin telah diberikan, fungsi ini akan memulai aktivitas untuk memilih gambar dari penyimpanan eksternal dan hasil pemilihan gambar kemudian akan dikembalikan ke aplikasi melalui `onActivityResult()`. Dengan demikian, fungsi `chooseImage()` adalah bagian penting dalam proses memilih gambar dari penyimpanan eksternal pada aplikasi Android.

Selecting an Image from the Gallery

PAM Praktikum 11 -11S20018


Label gambar ...



SAVE

PAM Praktikum 11 -11S20018

Label gambar ...



SAVE

Saving the selected image to Cloud Storage

Laporkan hasilnya.

Get started with Cloud Storage

Cloud Storage for Firebase provides secure file uploads and other user-generated content.

Launch in browser

1 Connect your app to Firebase


✓ Connected

2 Add Cloud Storage to your app

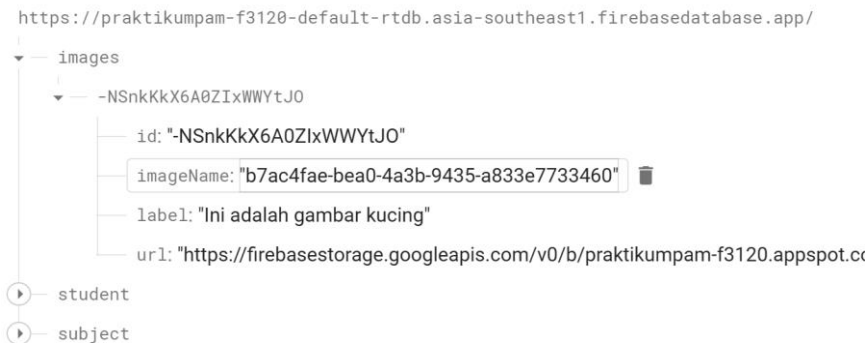
✓ Dependencies set up correctly

PAM Praktikum - 11S20018

Ini adalah gambar kucing



SAVE

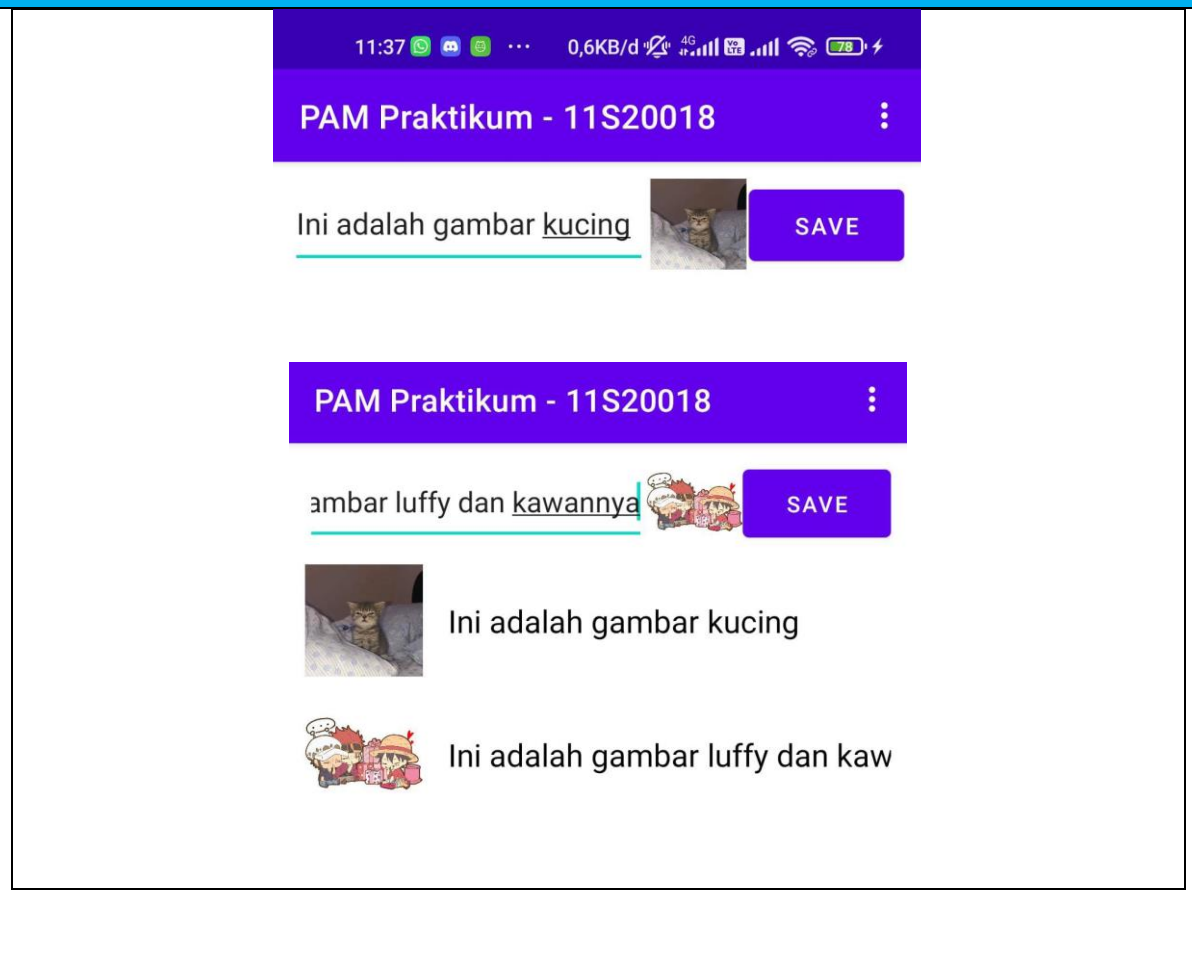
Saving the selected image to Cloud Storage**Showing the image saved to Cloud Storage in Recycler View**

Analisis kegunaan dari fungsi `retrieveDatafromDatabase()`. Laporkan hasilnya.

Fungsi `retrieveDatafromDatabase()` digunakan untuk mengambil data dari database Firebase menggunakan Firebase Realtime Database pada platform Android. Fungsi ini memiliki beberapa tujuan penting. Mengambil data dari Firebase Realtime Database: Fungsi ini menggunakan `dbRef` (objek `DatabaseReference`) untuk menambahkan `ValueEventListener` yang akan mengambil data dari database Firebase. `ValueEventListener` akan memantau perubahan data pada database, dan saat ada perubahan, `onDataChange()` akan dipanggil untuk mengambil data yang baru dari `DataSnapshot` yang diterima.

Membersihkan dan memperbarui daftar data: Setelah data baru diambil dari database, fungsi ini akan membersihkan daftar data `myImageList` dan mengisi ulang dengan data yang baru diambil menggunakan loop for untuk setiap child dalam `snapshot.children`. Data yang diambil dari snapshot akan diubah menjadi objek `MyImage` menggunakan `getValue(MyImage::class.java)` dan ditambahkan ke `myImageList` jika tidak null. Mengatur adapter dan layout manager: Setelah `myImageList` diperbarui, fungsi ini akan mengatur adapter `MyImageAdapter` dengan menggunakan `myImageList` sebagai sumber data, dan mengatur layout manager sebagai `LinearLayoutManager` pada `RecyclerView` dengan menggunakan `mainBinding.rvImages.layoutManager = LinearLayoutManager(this@MainActivity)`.

Menampilkan data pada `RecyclerView`: Terakhir, fungsi ini akan mengatur adapter yang sudah diatur sebelumnya ke `RecyclerView` menggunakan `mainBinding.rvImages.adapter = myImageAdapter`, sehingga data yang sudah diambil dari database akan ditampilkan pada `RecyclerView` di dalam `MainActivity`. Namun, perlu diingat bahwa fungsi ini belum mengimplementasikan metode `onCancelled()` yang diberikan dalam `ValueEventListener`. Anda masih perlu mengimplementasikan aksi yang sesuai ketika operasi dibatalkan, seperti menampilkan pesan kesalahan atau mengambil tindakan yang relevan.

Showing the image saved to Cloud Storage in Recycler View**Updating the Image save to Cloud Storage**

Analisis kegunaan dari fungsi `uploadPhoto()` dan `updateData()`. Laporkan hasilnya.

Fungsi `uploadPhoto()` digunakan untuk mengunggah foto yang dipilih oleh pengguna ke Firebase Storage. Foto diambil dari `imageUri` yang telah dipilih menggunakan `intent.ACTION_GET_CONTENT` pada fungsi `chooseImage()`. Setelah foto diunggah, URL download foto tersebut diambil menggunakan `myUploadImageRef.downloadUrl.addOnSuccessListener`, dan kemudian dilanjutkan untuk memperbarui data di Firebase Realtime Database menggunakan fungsi `updateData()`.

Fungsi `updateData()` digunakan untuk memperbarui data di Firebase Realtime Database setelah foto berhasil diunggah ke Firebase Storage. Data yang diperbarui meliputi label, URL foto, dan nama file foto yang diunggah. Fungsi ini mengambil data dari `updateMyImageBinding.etUpdateLabel` (untuk label baru) dan `imageUrl` (hasil dari URL download foto setelah diunggah ke Firebase Storage), serta `imageName` (nama file foto yang diunggah). Data tersebut dikirim ke Firebase Realtime Database menggunakan `dbRef.child(id).updateChildren()` untuk memperbarui data pada node yang sesuai dengan id yang diterima dari intent. Setelah data berhasil diperbarui, pesan "The image has been updated" ditampilkan dan activity diakhiri (`finish()`).

Updating the Image save to Cloud Storage

PAM Praktikum - 11S20018

Label gambar ...



SAVE



Ini adalah gambar kucing



Ini adalah gambar luffy dan kaw

← Update Image



Ini adalah gambar kucing

UPDATE DATA

← Update Image





Poster The Promised Neverland


UPDATE DATA

Updating the Image save to Cloud Storage

PAM Praktikum - 11S20018

Label gambar ...  **SAVE**

 Poster The Promised Neverland

 Ini adalah gambar luffy dan kaw

Deleting an Image We Added to Cloud Storage

Analisis kode program line 63 sampai 94. Laporkan hasilnya.

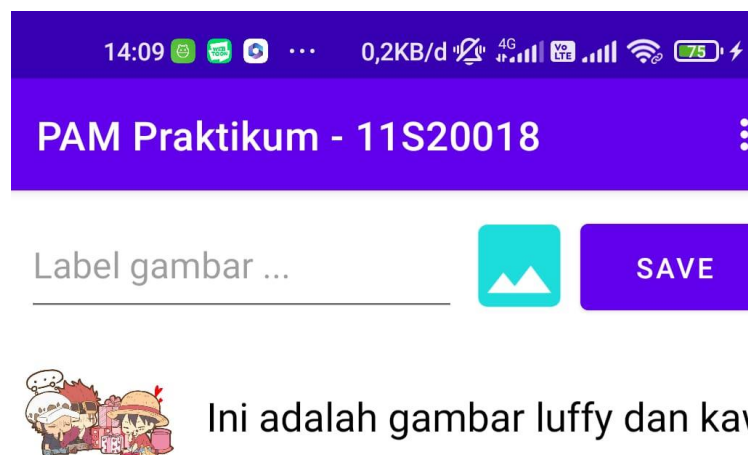
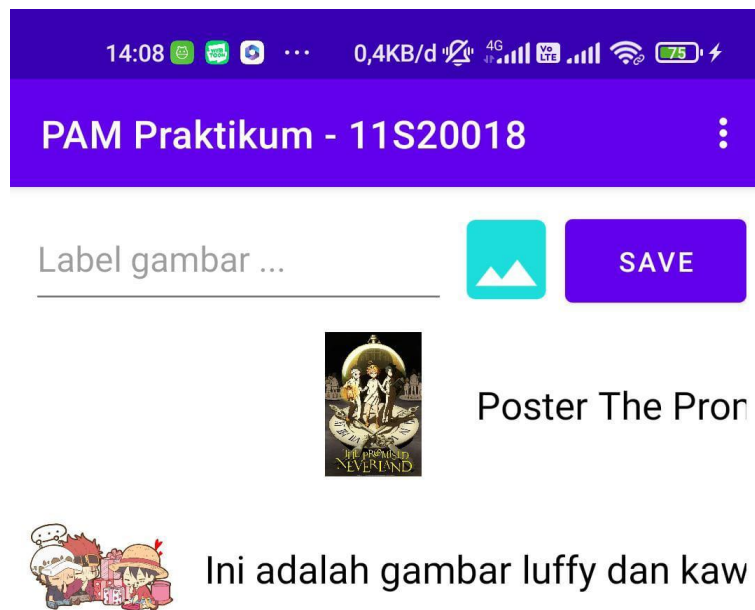
Kode program di atas merupakan penggunaan ItemTouchHelper dalam sebuah RecyclerView yang digunakan untuk meng-handle aksi swipe (geser) pada item-item dalam RecyclerView. Analisis kode programnya sebagai berikut:

1. ItemTouchHelper: ItemTouchHelper adalah kelas bawaan dari Android yang digunakan untuk mengatur interaksi pengguna terhadap item-item dalam RecyclerView, seperti aksi swipe, drag, atau drop. Dalam kode di atas, ItemTouchHelper digunakan untuk mengatur aksi swipe (geser) pada item-item dalam RecyclerView.
2. ItemTouchHelper.SimpleCallback: ItemTouchHelper.SimpleCallback adalah kelas turunan dari ItemTouchHelper.Callback yang menyediakan implementasi default untuk aksi swipe dan drag dalam RecyclerView. Dalam kode di atas, kita membuat objek ItemTouchHelper.SimpleCallback dengan meng-override dua metode yaitu onMove() dan onSwiped().
3. onMove(): Metode ini digunakan untuk mengatur aksi drag, yaitu ketika pengguna melakukan geser (drag) pada item dalam RecyclerView. Pada kode di atas, metode ini belum diimplementasikan dan hanya menampilkan komentar "Not yet implemented".
4. onSwiped(): Metode ini digunakan untuk mengatur aksi swipe, yaitu ketika pengguna melakukan geser (swipe) pada item dalam RecyclerView. Pada kode di atas, kita menghapus item yang ditempati oleh viewHolder (item yang digeser) dari database Firebase Realtime Database dengan menggunakan dbRef.child(id).removeValue(), dan juga menghapus file gambar yang terkait dengan item tersebut dari Firebase Storage dengan menggunakan

Deleting an Image We Added to Cloud Storage

`imageReference.delete()`. Selain itu, kita juga menampilkan pesan toast yang memberitahukan bahwa item telah dihapus.

5. `mainBinding.rvImages`: `mainBinding` adalah objek yang digunakan untuk mengakses komponen UI dalam layout yang terhubung dengan Activity ini, sedangkan `rvImages` adalah `RecyclerView` yang digunakan dalam layout tersebut. Dalam kode di atas, `ItemTouchHelper` di-attach (dipasang) dengan `RecyclerView` tersebut menggunakan metode `attachToRecyclerView()` untuk mengaktifkan aksi swipe pada item-item dalam `RecyclerView`.



Delete All Images We Added to Cloud Storage

Analisis kegunaan dari fungsi `showDialogForDeleteAllData()`. Laporkan hasilnya.

Kode program di atas adalah sebuah fungsi yang digunakan untuk menampilkan dialog konfirmasi penghapusan seluruh data gambar (images) dalam aplikasi. Kode tersebut menggunakan `AlertDialog.Builder` untuk membuat dialog konfirmasi dengan tombol "Cancel" dan "Yes". Ketika tombol "Yes" ditekan, data gambar akan dihapus dari Firebase Realtime Database dan Firebase Storage, serta tampilan `RecyclerView` akan diperbarui. Pesan toast juga ditampilkan untuk memberitahukan pengguna bahwa data gambar telah dihapus.

