

**LAPORAN AKTIVITAS LAB
PEMROGRAMAN PIRANTI BERGERAK**

UJIAN AKHIR SEMESTER



OLEH:
412020011 – Vaneza Angelica Citra

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KRISTEN KRIDA WACANA
JAKARTA
2022**

DAFTAR ISI

DAFTAR ISI.....	ii
Splash Screen.....	1
ActivitySplash.xml	1
SplashActivity.java	1
AndroidManifest.xml.....	2
Onboarding Screen Login and Signup.....	3
strings.xml.....	3
activity_main.xml.....	3
slides_layout.xml.....	5
SliderAdapter.java.....	5
Set up Firebase Database	7
MainActivity.java.....	8
Emulator	11
User.java	12
activity_signup.xml	13
SignupActivity.java.....	14
Menu	16
bottom_nav.xml.....	16
option_menu.xml.....	17
activity_menu.xml.....	17
gradiant1.xml	18
fragment_menu.xml.....	18
fragment_profile.xml	21
MenuFragment.java	22
ProfileFragment.java.....	24
MenuActivity.java.....	25
Emulator	27
activity_pilih_suit.xml	28
PilihSuit.java	29
Emulator	30
Suit Game	31
activity_suit_player.xml & activity_suit_npc.xml.....	31
dialog_winner.xml.....	32
Music	32
SuitPlayer.java	33

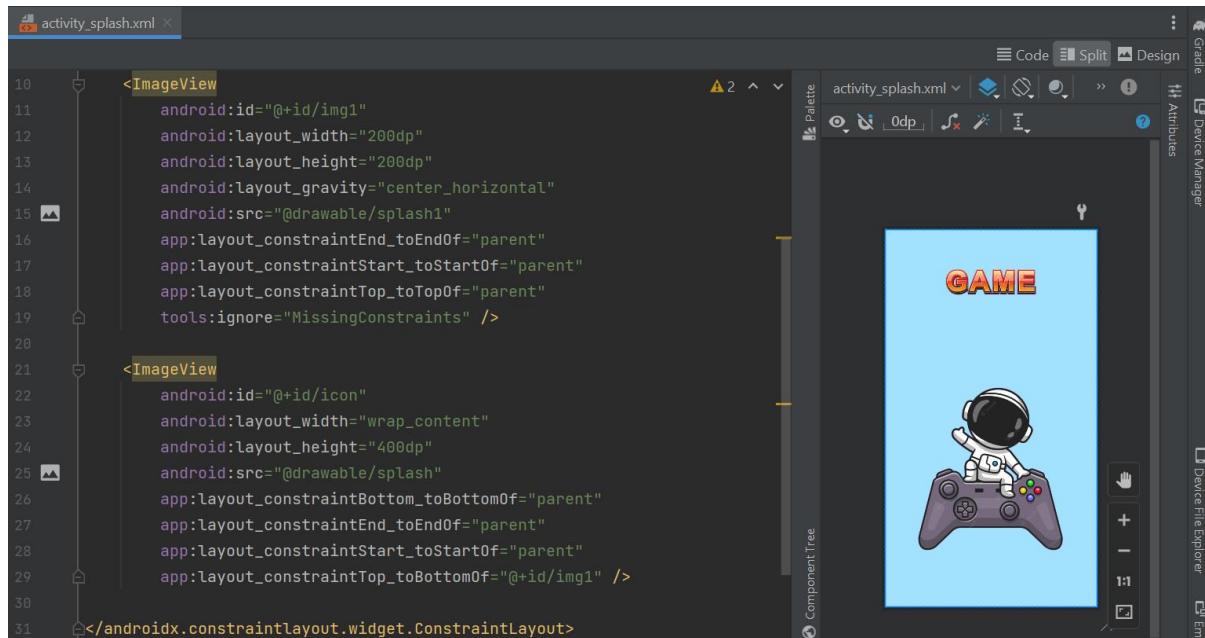
Emulator	37
SuitNPC.java	38
Tic Tac Toe Game	40
activity_tic_tac_toe.xml	40
TicTacToe.java	42
Emulator	44
About Developer	45
Menu	45
activity_about_developer	45
fragment_contact.xml	47
fragment_personal.xml	47
AboutDeveloper.java	48
Emulator	48
Edit Profile User	49
dialog_update_password.xml	49
activity_edit_profile_page.xml	50
EditProfilePage.java	51
Emulator	57
Comment	58
Komentar.java	58
activity_create_memo.xml	59
CreateMemo.java	59
activity_update.xml	60
UpdateActivity.java	61
KomentarAdapter.java	63
activity_data_users.xml	64
DataUsers.java	65
Emulator	66

Splash Screen

ActivitySplash.xml



Dalam project game ini, pertama kali yang saya lakukan adalah membuat activity baru bernama "SplashActivity" yang diletakkan di dalam directory baru "Splash". Activity ini untuk membuat sebuah Splash Screen yaitu tampilan paling awal saat program dijalankan.



Kemudian di dalam file .xml, diletakkan 2 ImageView yang berbeda untuk mendesain Splash Screen.

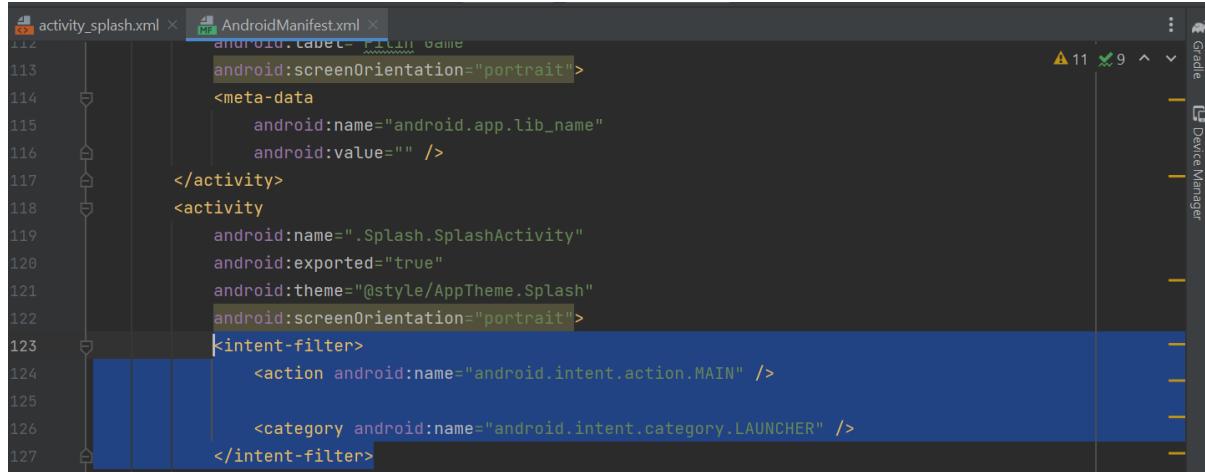
SplashActivity.java

```
1 package com.example.uas.Splash;
2
3 import ...
4
5 public class SplashActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_splash);
11
12        new Handler(Looper.getMainLooper()).postDelayed( new Runnable() {
13            @Override
14            public void run() {
15                Intent intentRadio = new Intent( packageContext: SplashActivity.this, MainActivity.class);
16
17                startActivity(intentRadio);
18            }
19        }, delayMillis: 1000);
20    }
21}
```

The code in the Java file defines a new activity called "SplashActivity" that extends "AppCompatActivity". It overrides the "onCreate" method to set the main layout to "activity_splash". It then uses a Handler to post a delayed Runnable. This Runnable creates an Intent to start the "MainActivity" and calls "startActivity" on it after a delay of 1000 milliseconds.

Selanjutnya dalam file **SplashActivity.java**, digunakan Handler untuk melakukan delay dengan menggunakan method `.postDelayed()` kemudian masukkan fungsi intent untuk berpindah activity ke MainActivity dengan lama splash interval yaitu 1000 ms.

AndroidManifest.xml

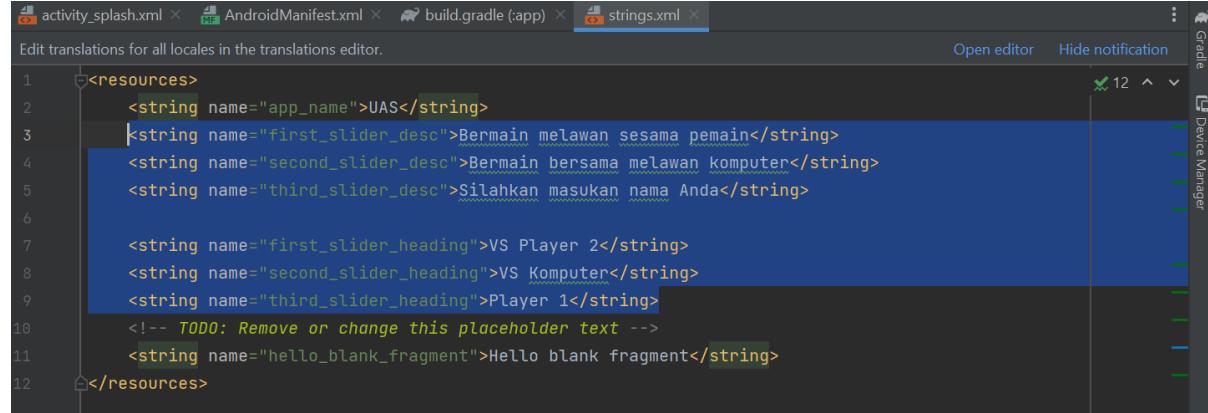


```
activity_main.xml x AndroidManifest.xml x
113     android:label="Main Name"
114         android:screenOrientation="portrait">
115             <meta-data
116                 android:name="android.app.lib_name"
117                 android:value="" />
118         </activity>
119         <activity
120             android:name=".Splash.SplashActivity"
121             android:exported="true"
122             android:theme="@style/AppTheme.Splash"
123             android:screenOrientation="portrait">
124                 <intent-filter>
125                     <action android:name="android.intent.action.MAIN" />
126                     <category android:name="android.intent.category.LAUNCHER" />
127                 </intent-filter>
```

Kemudian pada file `AndroidManifest.xml`, tambahkan code yang diblock di atas agar saat aplikasi dirun, maka akan menjalankan `SplashActivity` pertama kali. Selanjutnya letakkan `android:screenOrientation = "portrait"` pada line 122 agar aplikasi hanya dapat dijalankan secara portrait, sehingga meskipun layar android di miringkan tidak akan berubah ke landscape.

Onboarding Screen Login and Signup

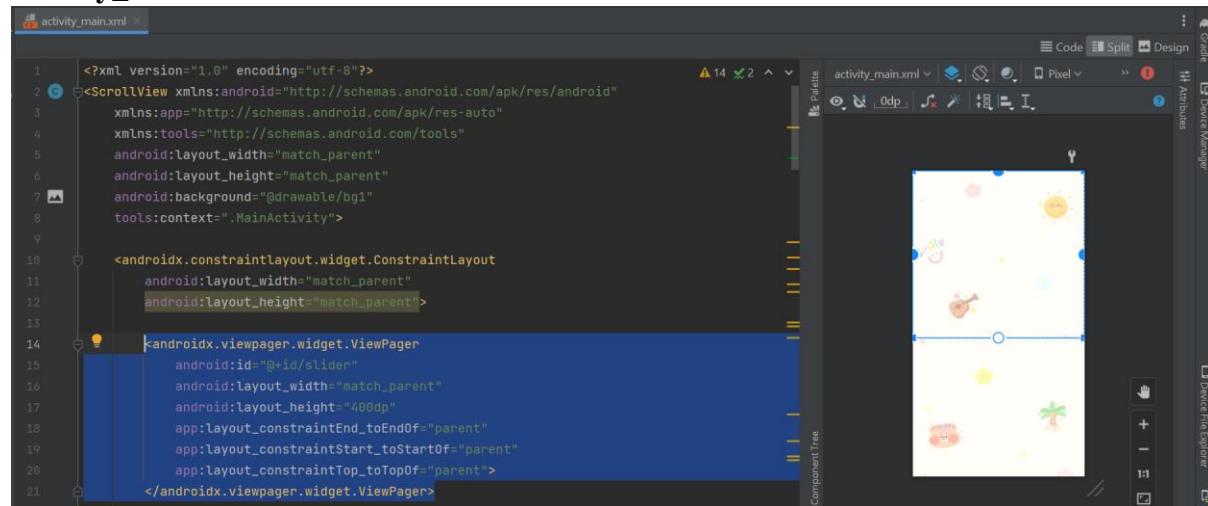
strings.xml



```
1 <resources>
2     <string name="app_name">UAS</string>
3     <string name="first_slider_desc">Bermain melawan sesama pemain</string>
4     <string name="second_slider_desc">Bermain bersama melawan komputer</string>
5     <string name="third_slider_desc">Silahkan masukan nama Anda</string>
6
7     <string name="first_slider_heading">VS Player 2</string>
8     <string name="second_slider_heading">VS Komputer</string>
9     <string name="third_slider_heading">Player 1</string>
10    <!-- TODO: Remove or change this placeholder text -->
11    <string name="hello_blank_fragment">Hello blank fragment</string>
12
13 </resources>
```

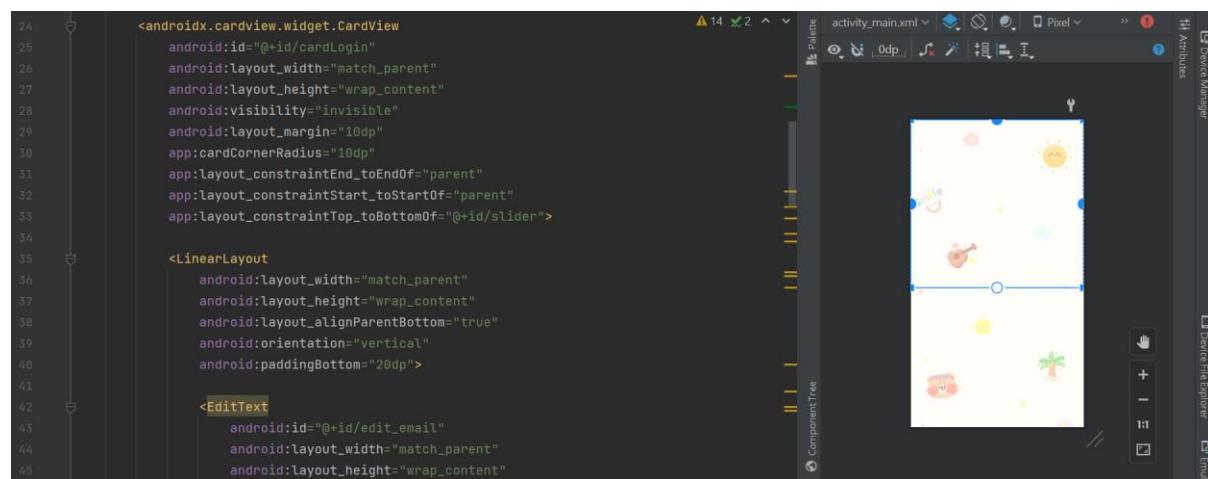
Pertama tambahkan beberapa string di atas di dalam file strings.xml beserta nama textnya.

activity_main.xml

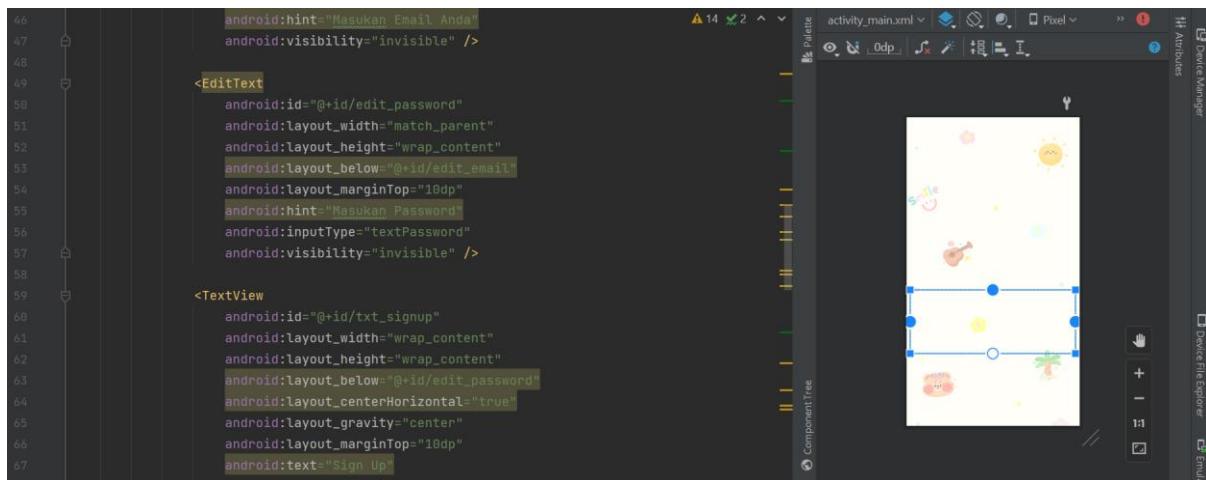


```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@drawable/bg1"
8     tools:context=".MainActivity">
9
10    <androidx.constraintlayout.widget.ConstraintLayout
11        android:layout_width="match_parent"
12        android:layout_height="match_parent">
13
14        <androidx.viewpager.widget.ViewPager
15            android:id="@+id/slides"
16            android:layout_width="match_parent"
17            android:layout_height="400dp"
18            app:layout_constraintEnd_toEndOf="parent"
19            app:layout_constraintStart_toStartOf="parent"
20            app:layout_constraintTop_toTopOf="parent">
21                </androidx.viewpager.widget.ViewPager>
```

Tambahkan code diatas sebagai tempat untuk menampung gambar dan text untuk onboarding screen.

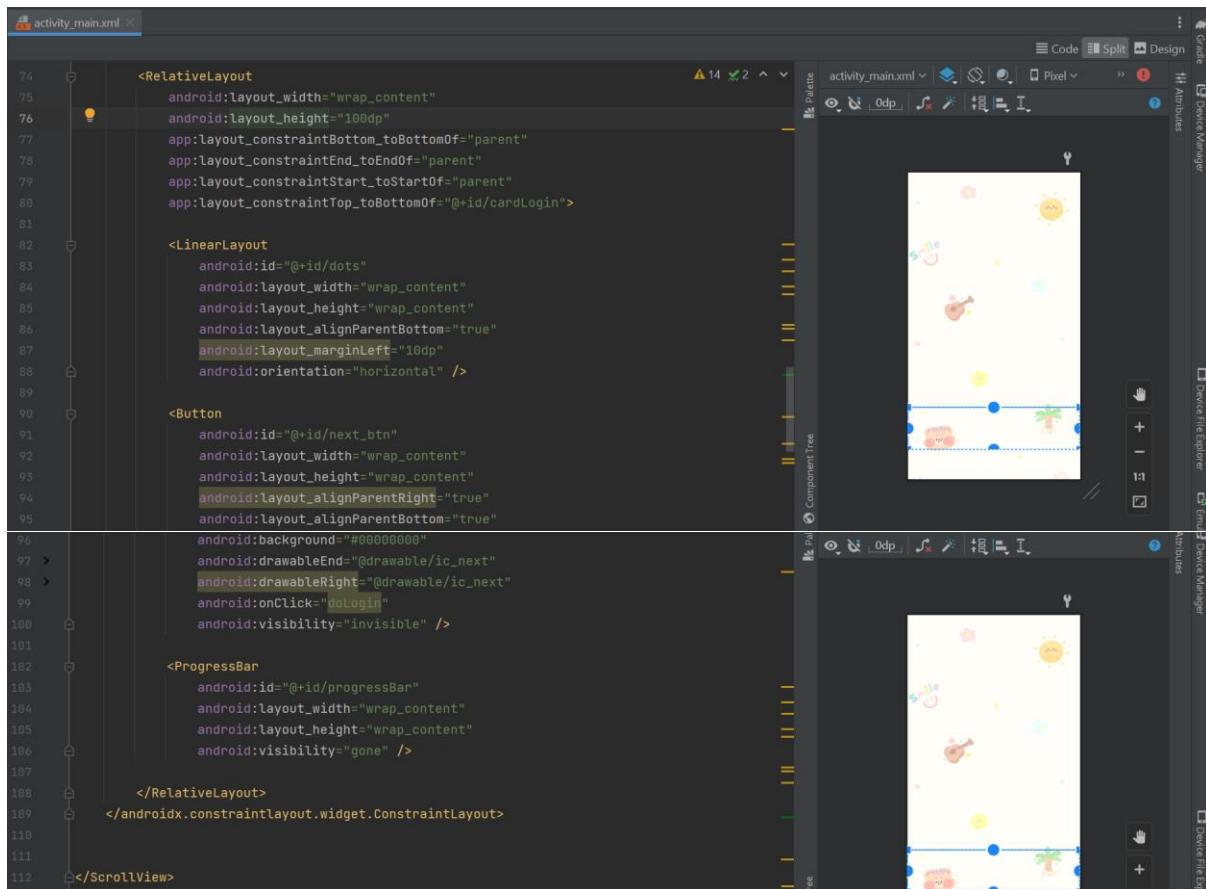


```
24 <androidx.cardview.widget.CardView
25     android:id="@+id/cardLogin"
26     android:layout_width="match_parent"
27     android:layout_height="wrap_content"
28     android:visibility="invisible"
29     android:layout_margin="10dp"
30     app:cardCornerRadius="10dp"
31     app:layout_constraintEnd_toEndOf="parent"
32     app:layout_constraintStart_toStartOf="parent"
33     app:layout_constraintTop_toBottomOf="@+id/slider">
34
35     <LinearLayout
36         android:layout_width="match_parent"
37         android:layout_height="wrap_content"
38         android:layout_alignParentBottom="true"
39         android:orientation="vertical"
40         android:paddingBottom="20dp">
41
42         <EditText
43             android:id="@+id/edit_email"
44             android:layout_width="match_parent"
45             android:layout_height="wrap_content"
```



```
46    android:hint="Masukan Email Anda"
47    android:visibility="invisible" />
48
49    <EditText
50        android:id="@+id/edit_password"
51        android:layout_width="match_parent"
52        android:layout_height="wrap_content"
53        android:layout_below="@+id/edit_email"
54        android:layout_marginTop="10dp"
55        android:hint="Masukan Password"
56        android:inputType="textPassword"
57        android:visibility="invisible" />
58
59    <TextView
60        android:id="@+id/txt_signup"
61        android:layout_width="wrap_content"
62        android:layout_height="wrap_content"
63        android:layout_below="@+id/edit_password"
64        android:layout_centerHorizontal="true"
65        android:layout_gravity="center"
66        android:layout_marginTop="10dp"
67        android:text="Sign Up"
```

Selanjutnya tambahkan 2 EditText yang akan digunakan pada slide layout ke 3 untuk melakukan login dan 1 TextView untuk melakukan signup. Tambahkan atribut android:visibility="invisible" agar View tidak langsung terlihat.



```
74    <RelativeLayout
75        android:layout_width="wrap_content"
76        android:layout_height="100dp"
77        app:layout_constraintBottom_toBottomOf="parent"
78        app:layout_constraintEnd_toEndOf="parent"
79        app:layout_constraintStart_toStartOf="parent"
80        app:layout_constraintTop_toBottomOf="@+id/cardLogin">
81
82        <LinearLayout
83            android:id="@+id/dots"
84            android:layout_width="wrap_content"
85            android:layout_height="wrap_content"
86            android:layout_alignParentBottom="true"
87            android:layout_marginLeft="10dp"
88            android:orientation="horizontal" />
89
90        <Button
91            android:id="@+id/next_btn"
92            android:layout_width="wrap_content"
93            android:layout_height="wrap_content"
94            android:layout_alignParentRight="true"
95            android:layout_alignParentBottom="true"
96            android:background="#00000000"
97            android:drawableEnd="@drawable/ic_next"
98            android:drawableRight="@drawable/ic_next"
99            android:onClick="doLogin"
100           android:visibility="invisible" />
101
102        <ProgressBar
103            android:id="@+id/progressBar"
104            android:layout_width="wrap_content"
105            android:layout_height="wrap_content"
106            android:visibility="gone" />
107
108    </RelativeLayout>
109    </androidx.constraintlayout.widget.ConstraintLayout>
110
111
112 </ScrollView>
```

Kemudian tambahkan LinearLayout sebagai tempat untuk meletakkan dots yang akan memberi tahu *user* berada di page mana. Selanjutnya tambahkan Button dan ProgressBar untuk melakukan proses login dan berpindah activity.

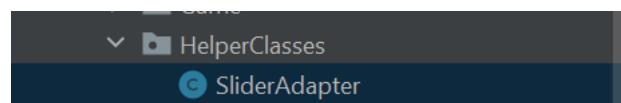
slides_layout.xml



```
1 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
2     xmlns:app="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:padding="10dp">
6
7     <ImageView
8         android:id="@+id/slider_image"
9         android:layout_width="wrap_content"
10        android:layout_height="wrap_content"
11        app:layout_constraintBottom_toBottomOf="parent"
12        app:layout_constraintEnd_toEndOf="parent"
13        app:layout_constraintStart_toStartOf="parent"
14        app:layout_constraintTop_toTopOf="parent"
15        app:srcCompat="@drawable/landing_page_1" />
16
17     <TextView
18         android:id="@+id/slider_heading"
19         android:layout_width="wrap_content"
20         android:layout_height="wrap_content"
21         android:fontFamily="@font/dryer_brush"
22         android:text="VS Player 2"
23         android:textSize="25sp"/>
24
25     <TextView
26         android:id="@+id/slider_desc"
27         android:layout_width="wrap_content"
28         android:layout_height="wrap_content"
29         android:padding="10dp"
30         android:text="Bermain melawan sesama pemain"
31         android:textSize="14sp"/>
32
33 </androidx.constraintlayout.widget.ConstraintLayout>
```

Selanjutnya buat sebuah file .xml baru dalam file directory **layout** dan beri nama **slides_layout**. Kemudian letakkan 1 ImageView dan 2 TextView untuk menampung image dan text berupa judul dan description yang akan dideklarasi dalam file SliderAdapter.java.

SliderAdapter.java



Kemudian dibuat sebuah java class yang diletakkan dalam directory **HelperClasses** untuk menampung gambar dan text yang dibutuhkan.

```
17 public class SliderAdapter extends PagerAdapter {
18     Context context;
19     LayoutInflater layoutInflater;
20
21     public SliderAdapter(Context context) { this.context = context; }
22
23     int images[] = {
24         R.drawable.landing_page_1,
25         R.drawable.landing_page_2,
26         R.drawable.person1
27     };
28
29     int headings[] = {
30         "VS Player 2",
31         "VS Komputer",
32         "Player 1",
33     };
34
35     int descriptions[] = {
36         "Bermain melawan sesama pemain",
37         "Bermain bersama melawan komputer",
38         "Silahkan masukan nama Anda"
39     };
40
41 }
42
```

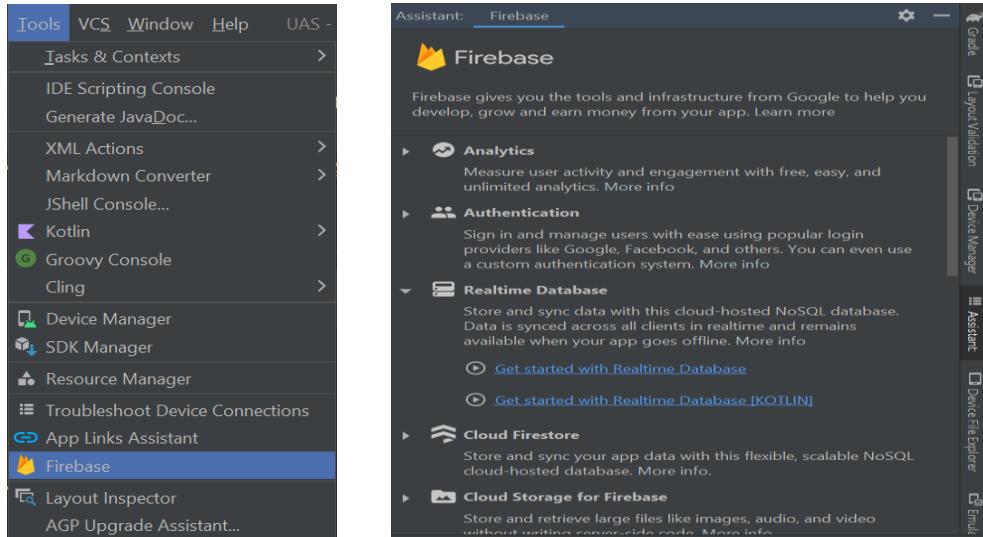
Dibuat 3 variabel array untuk menampung text dan gambar yang ingin kita tampilkan dalam Onboarding.

```
42
43     @Override
44     public int getCount() { return headings.length; }
45
46     @Override
47     public boolean isViewFromObject(@NonNull View view, @NonNull Object object) {
48         return view == (ConstraintLayout) object;
49     }
50
51     @NonNull
52     @Override
53     public Object instantiateItem(@NonNull ViewGroup container, int position) {
54
55         layoutInflater = (LayoutInflater) context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
56         View view = layoutInflater.inflate(R.layout.slides_layout, container, attachToRoot: false);
57
58         //Hooks
59         ImageView imageView = view.findViewById(R.id.slider_image);
60         TextView heading = view.findViewById(R.id.slider_heading);
61         TextView dsc = view.findViewById(R.id.slider_desc);
62
63         imageView.setImageResource(images[position]);
64         heading.setText(headings[position]);
65         dsc.setText(descriptions[position]);
66
67         container.addView(view);
68
69         return view;
70     }
71
72     @Override
73     public void destroyItem(@NonNull ViewGroup container, int position, @NonNull Object object) {
74         container.removeView((ConstraintLayout) object);
75     }
76
77 }
```

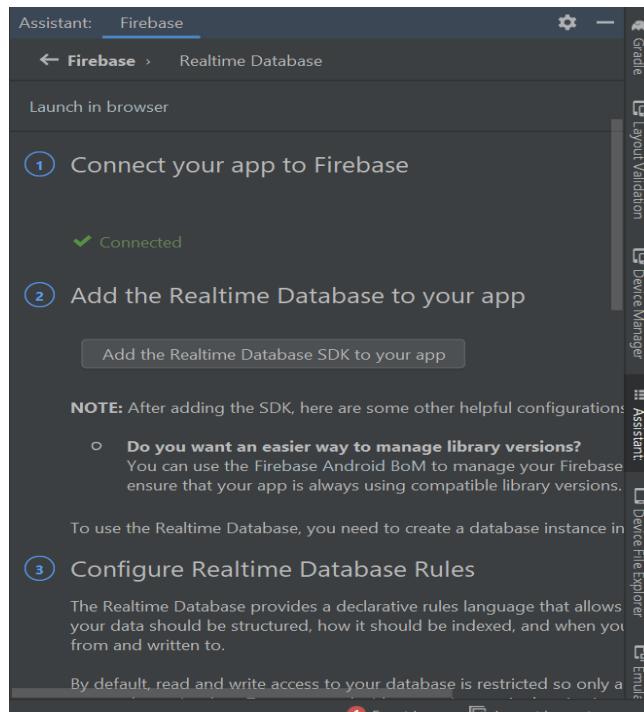
Selanjutnya tambahkan beberapa method di atas untuk membuat beberapa fungsi yang akan digunakan dalam file MainActivity.java. Dengan menggunakan LayoutInflater maka kita dapat menyambungkan layout slides_layout sehingga variabel array sebelumnya dapat ditempatkan

ke dalam View yang berada di layout tersebut dan akan ditampilkan bergantian berdasarkan *position*.

Set up Firebase Database



Untuk menghubungkan firebase dengan aplikasi android, kita dapat melakukannya dengan menggunakan Firebase Assistant yang ada pada menu Tools > Firebase. Kemudian pilih **Realtime Database** dan klik **save and retrieve data**. Selanjutnya klik **Get started with Realtime Database** dan klik **Connect to Firebase**.



Ketika firebase selesai disconnect akan menampilkan tampilan di atas. Kemudian klik **Add the Realtime Database SDK to your app** dan klik **Accept Changes**. Selanjutnya pada website firebase buatlah Realtime Database dengan pilihan **Start in test mode**.

The screenshot shows the Firebase Authentication console. On the left, there's a sidebar with project settings like Realtime Database, Authentication, Storage, and Firestore Database. The main area is titled "Authentication" and has tabs for "Users", "Sign-in method", "Templates", "Usage", and "Settings". Under "Sign-in method", it says "Sign-in providers" and shows a table with one row: "Provider" (Email/Password) and "Status" (Enabled). There's also an "Add new provider" button. Below this, under "Advanced", there's a section for "SMS Multi-factor Authentication" with a note about enabling two-step verification.

Selanjutnya dalam website firebase buka ke **Firebase Console > Authentication > Sign-In Method** dan pilih provider Email/Password kemudian ubah status ke **enabled**.

MainActivity.java

```

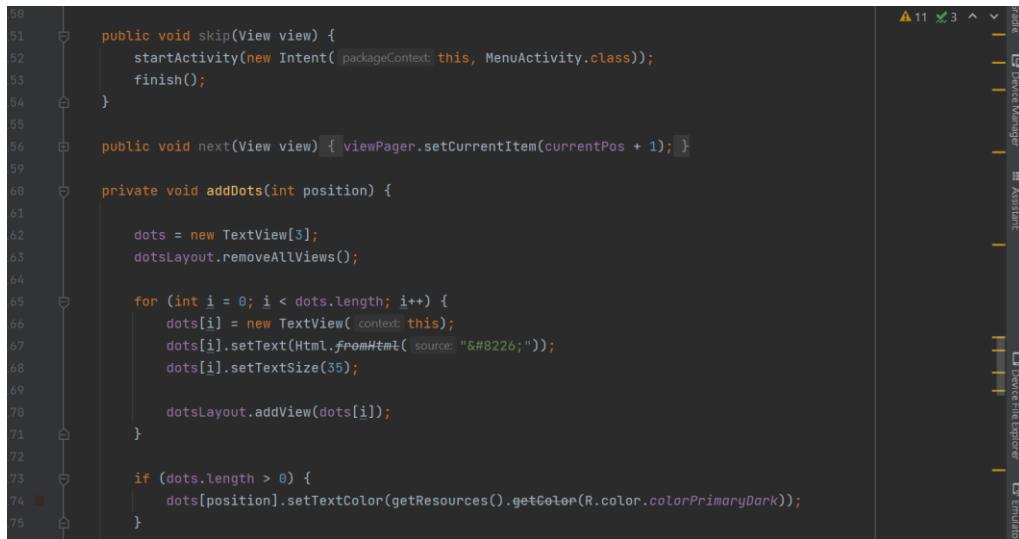
28
29
30     public class MainActivity extends AppCompatActivity {
31
32         //Variables
33         ViewPager viewPager;
34         LinearLayout dotsLayout;
35         SliderAdapter sliderAdapter;
36         TextView[] dots;
37         TextView txtSignUp;
38         Button imgNext;
39         private EditText emailEditText, passwordTextView;
40         private Button Btn;
41         private ProgressBar progressBar;
42         private FirebaseAuth mAuth;
43         int currentPos;
44
45
46         @Override
47         protected void onCreate(Bundle savedInstanceState) {
48             super.onCreate(savedInstanceState);
49             getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
50             setContentView(R.layout.activity_main);
51
52             // taking instance of FirebaseAuth
53             mAuth = FirebaseAuth.getInstance();
54
55             //Hooks
56             viewPager = findViewById(R.id.slider);
57             dotsLayout = findViewById(R.id.dots);
58             // initialising all views through id defined above
59             emailEditText = findViewById(R.id.edit_email);
60             passwordTextView = findViewById(R.id.edit_password);
61             Btn = findViewById(R.id.next_btn);
62             progressBar = findViewById(R.id.progressBar);
63             txtSignup = findViewById(R.id.txt_signup);
64
65             //Call adapter
66             sliderAdapter = new SliderAdapter(context:this);
67             viewPager.setAdapter(sliderAdapter);
68
69             //Dots
70             dots = new TextView[dotsLayout.getChildCount()];
    
```

Selanjutnya dalam MainActivity.java deklarasi id dari View yang berada dalam layout dan beberapa kelas lain yang kita butuhkan. FirebaseAuth digunakan mengautentikasi data melalui Google, dll (line 53).

```
78     addDots(position: 0);
79     viewPager.addOnPageChangeListener(changeListener);
80
81 }
82
83 private void loginUserAccount()
84 {
85
86     // show the visibility of progress bar to show loading
87     progressBar.setVisibility(View.VISIBLE);
88
89     // Take the value of two edit texts in Strings
90     String email, password;
91     email = editTextView.getText().toString();
92     password = passwordTextView.getText().toString();
93
94     // validations for input email and password
95     if (TextUtils.isEmpty(email)) {
96         Toast.makeText(getApplicationContext(),
97             text: "Please enter email!!",
98             Toast.LENGTH_LONG)
99             .show();
100    return;
101 }
102
103 if (TextUtils.isEmpty(password)) {
104     Toast.makeText(getApplicationContext(),
105         text: "Please enter password!!",
106         Toast.LENGTH_LONG)
107         .show();
108    return;
109 }
110
111 // signin existing user
112 mAuth.signInWithEmailAndPassword(email, password)
113     .addOnCompleteListener(
114         new OnCompleteListener<AuthResult>() {
115             @Override
116             public void onComplete(
117                 @NonNull Task<AuthResult> task)
118             {
119                 if (task.isSuccessful()) {
120                     Toast.makeText(getApplicationContext(),
121                         text: "Login successful!!",
122                         Toast.LENGTH_LONG)
123                         .show();
124
125                     // hide the progress bar
126                     progressBar.setVisibility(View.GONE);
127
128                     // if sign-in is successful
129                     // intent to home activity
130                     Intent intent
131                         = new Intent(packageContext: MainActivity.this,
132                         MenuActivity.class);
133                     startActivity(intent);
134                 }
135
136             else {
137
138                 // sign-in failed
139                 Toast.makeText(getApplicationContext(),
140                     text: "Login failed!!",
141                     Toast.LENGTH_LONG)
142                     .show();
143
144                 // hide the progress bar
145                 progressBar.setVisibility(View.GONE);
146
147             }
148
149         }
150
151     }
152 }
```

Line 74-80 berfungsi ketika Button next diklik akan memanggil *method* **loginUserAccount()**.

Method `loginUserAccount()` berfungsi untuk mengambil hasil inputan *user* (line 91-92) dan mengecek apakah inputan tersebut kosong atau tidak. Jika tidak, maka akan dilakukan pengecekan apakah email dan password yang dimasukkan telah terdaftar dalam *database* atau tidak. Jika email telah terdaftar, maka proses login berhasil dan *user* akan berpindah *activity* ke *MenuActivity*. Jika tidak terdaftar, maka *user* akan mendapatkan informasi berupa *toast*.



The screenshot shows the Java code for a class named `OnBoardingActivity`. The code includes methods for skipping slides, navigating to the next slide, and adding dots to a dot indicator layout. The `addDots` method creates three `TextView` objects and adds them to a `LinearLayout`, then sets the text color of the specified dot to a dark primary color if it exists.

```
51     public void skip(View view) {
52         startActivity(new Intent(packageContext: this, MenuActivity.class));
53         finish();
54     }
55
56     public void next(View view) { viewPager.setCurrentItem(currentPos + 1); }
57
58     private void addDots(int position) {
59
60         dots = new TextView[3];
61         dotsLayout.removeAllViews();
62
63         for (int i = 0; i < dots.length; i++) {
64             dots[i] = new TextView(context: this);
65             dots[i].setText(Html.fromHtml(source: "\u2022"));
66             dots[i].setTextSize(36);
67
68             dotsLayout.addView(dots[i]);
69         }
70
71         if (dots.length > 0) {
72             dots[position].setTextColor(getResources().getColor(R.color.colorPrimaryDark));
73         }
74     }
75 }
```

Method `addDots()` berfungsi untuk menambah dots pada onboarding agar *user* dapat mengetahui posisi slidenya. Dots akan mengambil position dari slide sehingga posisi *slide user* akan ditandai dengan warna yang berbeda.

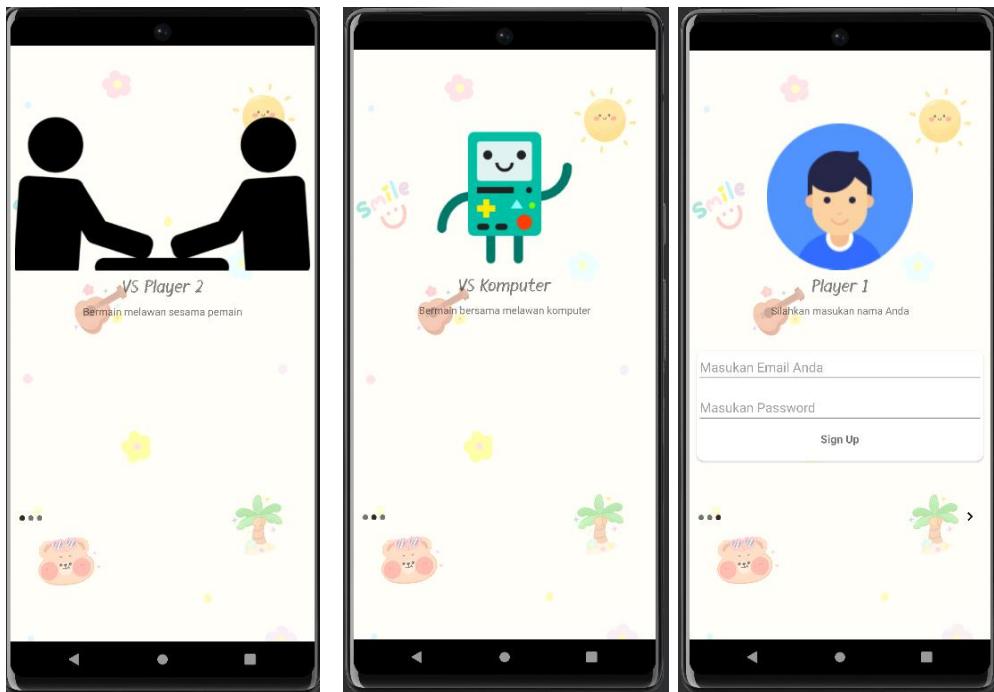
```

179     //mengatur visibility View disetiap slide layout
180     ViewPager.OnPageChangeListener changeListener = new ViewPager.OnPageChangeListener() {
181         @Override
182         public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
183
184     }
185     @Override
186     public void onPageSelected(int position) {
187         addDots(position);
188         currentPos = position;
189
190         if (position == 0) {
191             emailTextView.setVisibility(View.INVISIBLE);
192             passwordTextView.setVisibility(View.INVISIBLE);
193             Btn.setVisibility(View.INVISIBLE);
194             txtSignUp.setVisibility(View.INVISIBLE);
195         }else if (position == 1) {
196             emailTextView.setVisibility(View.INVISIBLE);
197             passwordTextView.setVisibility(View.INVISIBLE);
198             Btn.setVisibility(View.INVISIBLE);
199             txtSignUp.setVisibility(View.INVISIBLE);
200         }else {
201             emailTextView.setVisibility(View.VISIBLE);
202             passwordTextView.setVisibility(View.VISIBLE);
203             Btn.setVisibility(View.VISIBLE);
204             txtSignUp.setVisibility(View.VISIBLE);
205             //doLogin(imgNext);
206             txtSignUp.setOnClickListener(new View.OnClickListener() {
207                 @Override
208                 public void onClick(View view) {
209                     Intent signup = new Intent(getApplicationContext(), SignupActivity.class);
210                     startActivity(signup);
211                 }
212             });
213         }
214     }
215
216     @Override
217     public void onPageScrolled(int state) {
218
219     }
220     @Override
221     public void onPageSelected(int state) {
222
223     }
224 };

```

Code di atas berfungsi untuk mengambil *position* sehingga dapat ditentukan *visibility* dari beberapa View di setiap *slide*.

Emulator



Hasil emulator dari MainActivity.java akan menghasil tampilan seperti di atas. Gambar dan text ditampilkan berbeda-beda sesuai dengan positionnya. Kemudian EditText dan Button beserta text Signup akan ditampilkan apabila *user* berada di posisi terakhir.

User.java

```
6  public class User implements Parcelable {
7      public String name,email;
8
9      public User(String name, String email){
10         this.name = name;
11         this.email = email;
12     }
13
14
15     public String getEmail() { return email; }
16
17     public void setEmail(String email) { this.email = email; }
18
19     public String getName() { return name; }
20
21     public void setName(String name) { this.name = name; }
22
23
24
25
26
27
28
29
30
31
32
33     protected User(Parcel in) {
34         this.name = in.readString();
35         this.email = in.readString();
36     }
37
38     public static final Parcelable.Creator<User> CREATOR = new Parcelable.Creator<User>() {
39
40         @Override
41         public User createFromParcel(Parcel source) { return new User(source); }
42
43
44         @Override
45         public User[] newArray(int size) { return new User[size]; }
46     };
47
48
49
50         @Override
51         public int describeContents() { return 0; }
52
53
54         @Override
55         public void writeToParcel(Parcel dest, int flags) {
56             dest.writeString(name);
57             dest.writeString(email);
58         }
59     }
60 }
```

Selanjutnya buatlah sebuah kelas baru dan beri nama User.java. Kelas ini berfungsi untuk memudahkan pengiriman obyek secara langsung berupa data name dan email dengan menggunakan **Parcelable**.

activity_signup.xml

The screenshot shows the Android Studio interface with the XML code for `activity_signup.xml` on the left and a preview of the layout on the right. The layout consists of a `ScrollView` containing a `LinearLayout` with three `EditText` fields for Name, Email, and Password, and a `Button` labeled "Submit". The preview shows the layout with placeholder text and a purple "Submit" button.

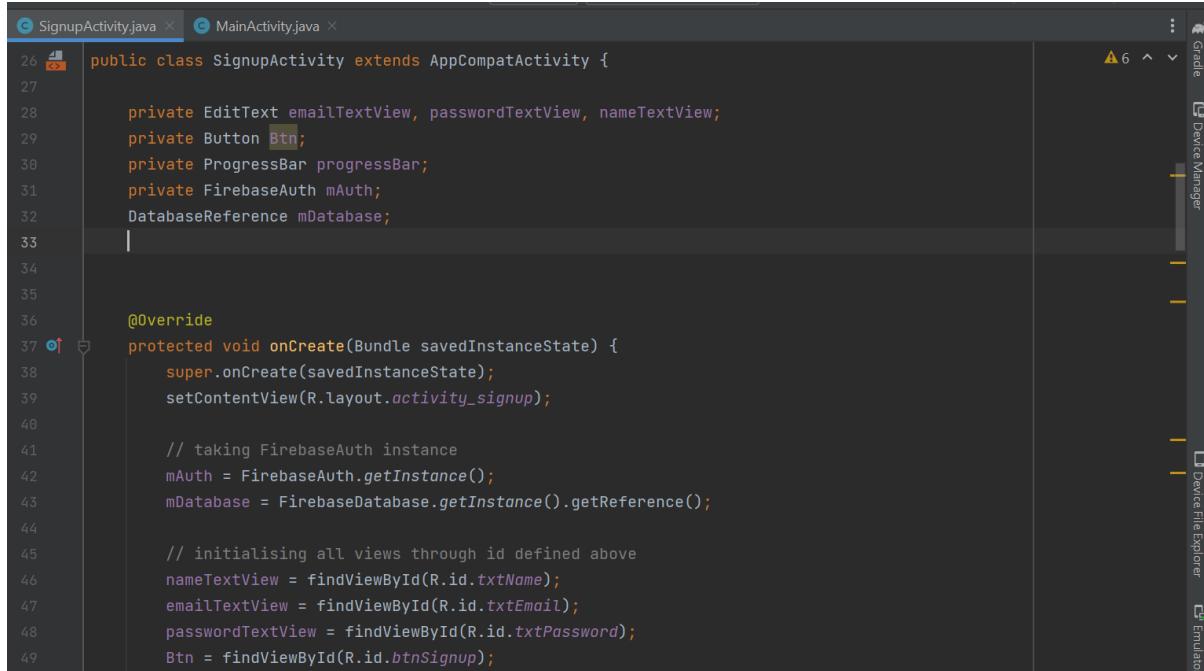
```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:padding="15dp"
    android:orientation="vertical"
    tools:context=".Profile.SignupActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <TextView
            android:id="@+id/textView"
            android:layout_width="wrap_content"
            android:layout_marginBottom="20dp"
            android:layout_gravity="center"
            android:textStyle="bold"
            android:textColor="@color/black"
            android:layout_height="wrap_content"
            android:text="Sign Up"
            android:textSize="30sp" />
        <TextView
            android:id="@+id/textView2"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Name" />
        <EditText
            android:id="@+id/txtName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPersonName" />
        <TextView
            android:id="@+id/textView3"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Email" />
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textEmailAddress" />
        <TextView
            android:id="@+id/textView4"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:text="Password" />
        <EditText
            android:id="@+id/txtPassword"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:inputType="textPassword" />
        <Button
            android:id="@+id/btnSignup"
            android:layout_marginTop="20dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Submit"
            android:onClick="doRegister"/>
        <!-- ProgressBar for loading time -->
        <ProgressBar
            android:id="@+id/progressbar"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:visibility="gone" />
    </LinearLayout>
</ScrollView>
```

Dalam file ini, diletakkan 3 `EditText` yang nantinya berfungsi sebagai tempat inputan *user* berupa nama, email, dan password *user*. Kemudian berisi sebuah `Button` untuk mengsubmit

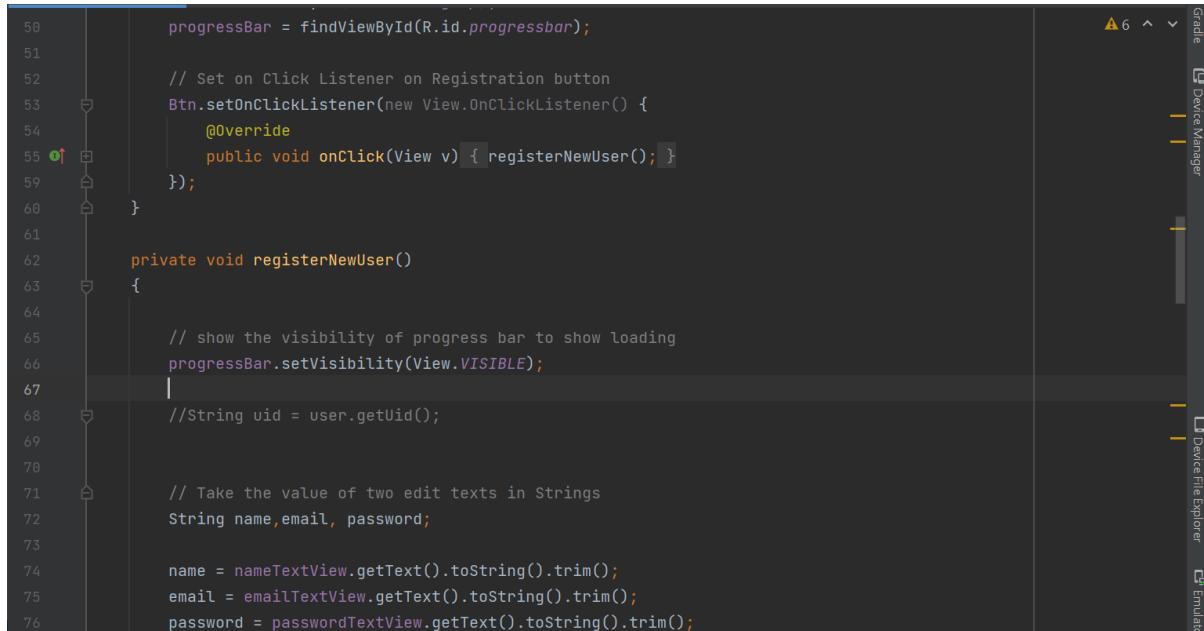
hasil inputan *user*. Terdapat ProgressBar juga yang nantinya akan muncul setelah *user* mengklik *button* submit untuk memberi info kepada *user* bahwa aplikasi tidak hang.

SignupActivity.java



```
26 public class SignupActivity extends AppCompatActivity {
27
28     private EditText emailTextView, passwordTextView, nameTextView;
29     private Button Btn;
30     private ProgressBar progressBar;
31     private FirebaseAuth mAuth;
32     DatabaseReference mDatabase;
33
34
35
36     @Override
37     protected void onCreate(Bundle savedInstanceState) {
38         super.onCreate(savedInstanceState);
39         setContentView(R.layout.activity_signup);
40
41         // taking FirebaseAuth instance
42         mAuth = FirebaseAuth.getInstance();
43         mDatabase = FirebaseDatabase.getInstance().getReference();
44
45         // initialising all views through id defined above
46         nameTextView = findViewById(R.id.txtName);
47         emailTextView = findViewById(R.id.txtEmail);
48         passwordTextView = findViewById(R.id.txtPassword);
49         Btn = findViewById(R.id.btnSignup);
```

Untuk membuat signup, deklarasi terlebih dahulu beberapa kelas Firebase yang dibutuhkan untuk melakukan signup. Kemudian deklarasi View yang ada di file .xml untuk mengambil inputan *user*.



```
50     progressBar = findViewById(R.id.progressbar);
51
52     // Set on Click Listener on Registration button
53     Btn.setOnClickListener(new View.OnClickListener() {
54         @Override
55         public void onClick(View v) { registerNewUser(); }
56     });
57
58
59     private void registerNewUser()
60     {
61
62         // show the visibility of progress bar to show loading
63         progressBar.setVisibility(View.VISIBLE);
64
65         //String uid = user.getUid();
66
67
68         // Take the value of two edit texts in Strings
69         String name, email, password;
70
71         name = nameTextView.getText().toString().trim();
72         email = emailTextView.getText().toString().trim();
73         password = passwordTextView.getText().toString().trim();
```

Kemudian buatlah sebuah **method registerNewUser()** untuk meletakkan code-code agar *user* dapat melakukan *signup*. Saat *user* mengklik *button* Submit, maka ProgressBar akan ditampilkan. Kemudian hasil inputan *user* akan diambil menggunakan code 74-76.

```

78     // Validations for input email and password
79     if (TextUtils.isEmpty(email)) {
80         Toast.makeText(getApplicationContext(),
81             text: "Please enter email!!",
82             Toast.LENGTH_LONG)
83             .show();
84         return;
85     }
86     if (TextUtils.isEmpty(password)) {
87         Toast.makeText(getApplicationContext(),
88             text: "Please enter password!!",
89             Toast.LENGTH_LONG)
90             .show();
91         return;
92     }
93
94     if(password.length()< 6 ){
95         Toast.makeText(getApplicationContext(),
96             text: "Min password length should be 6 characters!!",
97             Toast.LENGTH_LONG)
98             .show();
99         return;
100    }

```

Selanjutnya masukkan code di atas untuk membuat validasi dari inputan *user*. Line 79 – 92 berfungsi untuk mengecek apakah inputan email dan password kosong atau tidak, apabila kosong maka akan memberi informasi untuk mengisi email dan password dalam bentuk *toast*. Kemudian line 94 – 100 berfungsi untuk mengecek apakah password yang dimasukkan memiliki Panjang > 5 atau tidak, jika tidak maka akan mengeluarkan informasi berupa *toast*.

```

102     // create new user or register new user
103     mAuth
104         .createUserWithEmailAndPassword(email, password)
105         .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
106
107
108     @Override
109     public void onComplete(@NonNull Task<AuthResult> task)
110     {
111         if (task.isSuccessful()) {
112             User user = new User(name, email);
113
114             FirebaseDatabase.getInstance().getReference( path: "Users")
115                 .getCurrentUser().getUid()).setValue(user).addOnCompleteListener(new OnCompleteListener<ValueSnapshot>()
116
117             @Override
118             public void onComplete(@NonNull Task<Void> task) {
119                 if(task.isSuccessful()){
120                     Toast.makeText(getApplicationContext(),
121                         text: "Registration successful!!",
122                         Toast.LENGTH_LONG)
123                         .show();
124
125                     // hide the progress bar
126                     progressBar.setVisibility(View.GONE);
127                 } else {
128

```

Untuk membuat akun baru kira dapat mengirimkan alamat email dan kata sandi pengguna baru ke `createUserWithEmailAndPassword`. `FirebaseDatabase` berfungsi untuk mengakses *Firebase Database*. *Method* `getReference` berfungsi untuk mengakses lokasi *Users* di *database*. *Method* `getInstance` berfungsi untuk mendapatkan *instance Firebase Database* dengan URL yang telah ditentukan. Line 113 berfungsi untuk mengambil uid dari Authentication ketika dilakukan registrasi untuk dibuat tabel baru dalam *Users* dengan uid yang sama. Kemudian hasil inputan *user* akan dimasukan ke dalam tabel tersebut. Ketika registrasi berhasil akan menampilkan *toast* dan *ProgressBar* akan menghilang

```

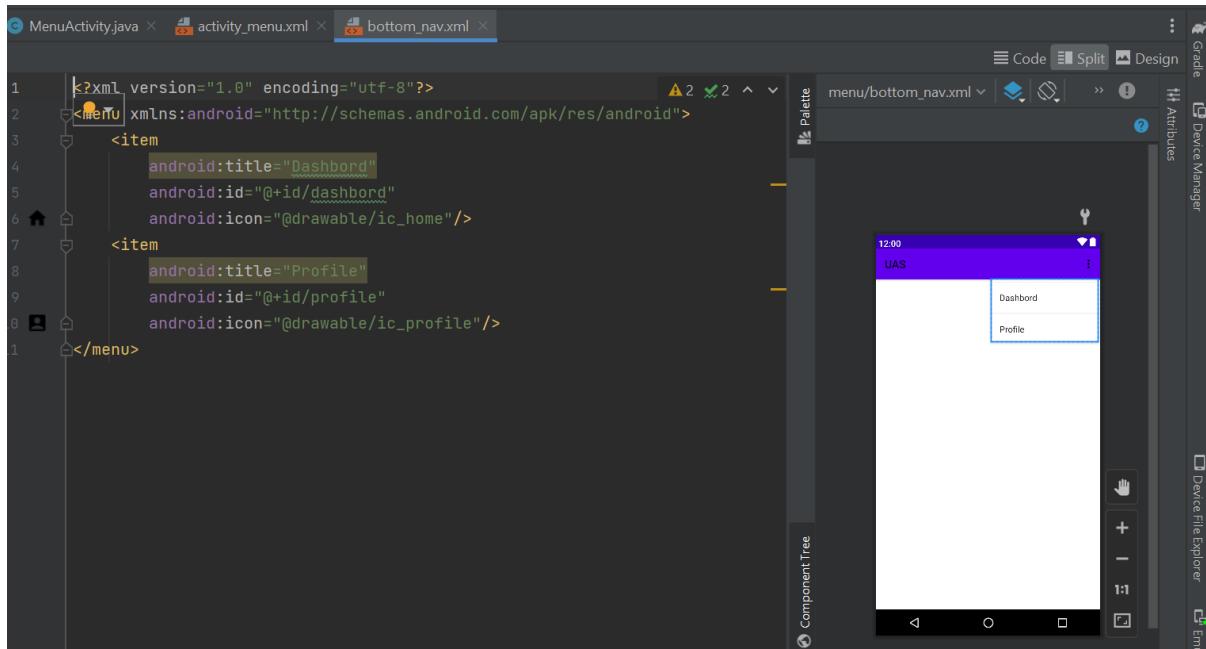
127         // Registration failed
128         Toast.makeText(
129             getApplicationContext(),
130             "Registration failed!!"
131             + " Please try again later",
132             LENGTH_LONG)
133         .show();
134
135         // hide the progress bar
136         progressBar.setVisibility(View.GONE);
137     }
138 }
139
140
141
142     // if the user created intent to login activity
143     Intent intent
144         = new Intent(packageContext: SignupActivity.this,
145             MainActivity.class);
146     startActivity(intent);
147

```

Apabila registrasi gagal, maka *user* akan mendapatkan informasi berupa *toast* untuk mengulang resgistrasi lagi. Ketika *user* telah berhasil melakukan login, maka *user* akan berpindah *activity* kembali ke *MainActivity* lagi.

Menu

bottom_nav.xml



Pertama buatlah sebuah file .xml baru dalam folder res > menu dan beri nama bottom_nav. Kemudian masukkan icon untuk gambar yang nantinya akan digunakan pada BottomNavigation.

option_menu.xml

The screenshot shows the Android Studio interface with the code editor and design preview side-by-side. The code editor contains the XML for an options menu:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <item
        android:id="@+id/about"
        android:title="About"
        app:showAsAction="never"/>
    <item
        android:id="@+id/setting"
        android:title="Memo"
        app:showAsAction="never" />
    <item
        android:id="@+id/home"
        android:title="Home"
        app:showAsAction="never"/>
    <item
        android:id="@+id/logout"
        android:title="Logout"
        app:showAsAction="never"/>
</menu>
```

The design preview shows a dropdown menu with items: About, Memo, Home, and Logout.

Selanjutnya buat lagi file .xml baru dalam folder res > menu dan beri nama option_menu. Sintaks ini berfungsi untuk membuat option menu di atas dalam bentuk 3 titik di ujung kanan yang dapat di dropdown.

activity_menu.xml

The screenshot shows the Android Studio interface with the code editor and design preview side-by-side. The code editor contains the XML for an activity layout:

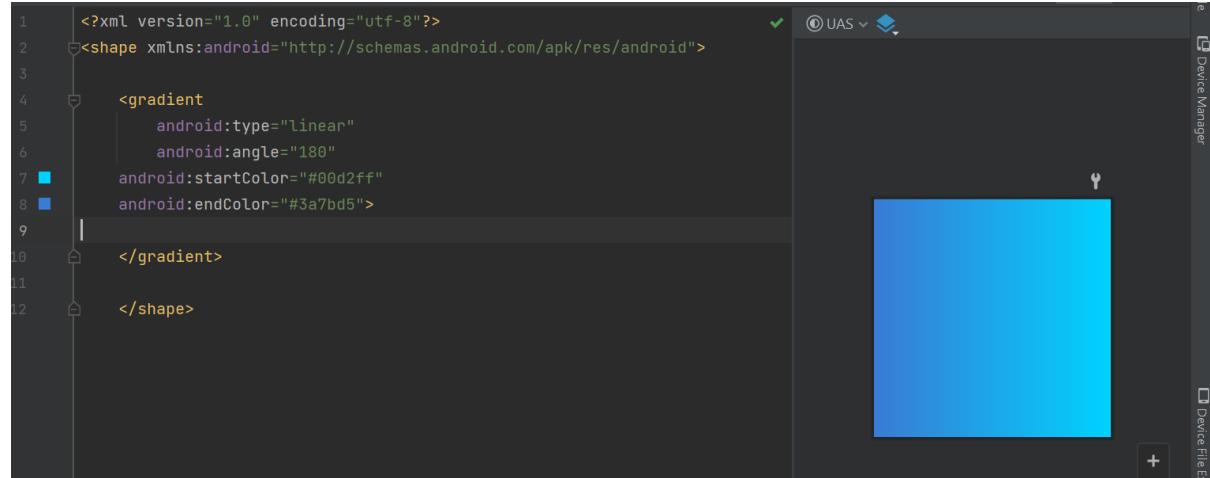
```
<FrameLayout
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    app:layout_constraintBottom_toTopOf="@+id/bottomNavigationView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<com.google.android.material.bottomnavigation.BottomNavigationView
    android:id="@+id/bottomNavigationView"
    android:layout_width="match_parent"
    android:layout_height="75dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.5"
    app:layout_constraintStart_toStartOf="parent"
    app:menu="@menu/bottom_nav" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

The design preview shows a bottom navigation bar with two items: Eambar and Profile.

Selanjutnya buatlah sebuah activity_baru dan beri nama MenuActivity. Activity ini berfungsi untuk menampilkan menu permainan dan profile *user* yang dapat dilihat dengan mengklik BottomNavigation. FrameLayout berfungsi sebagai tempat untuk meletakkan fragment dari menu game dan profile.

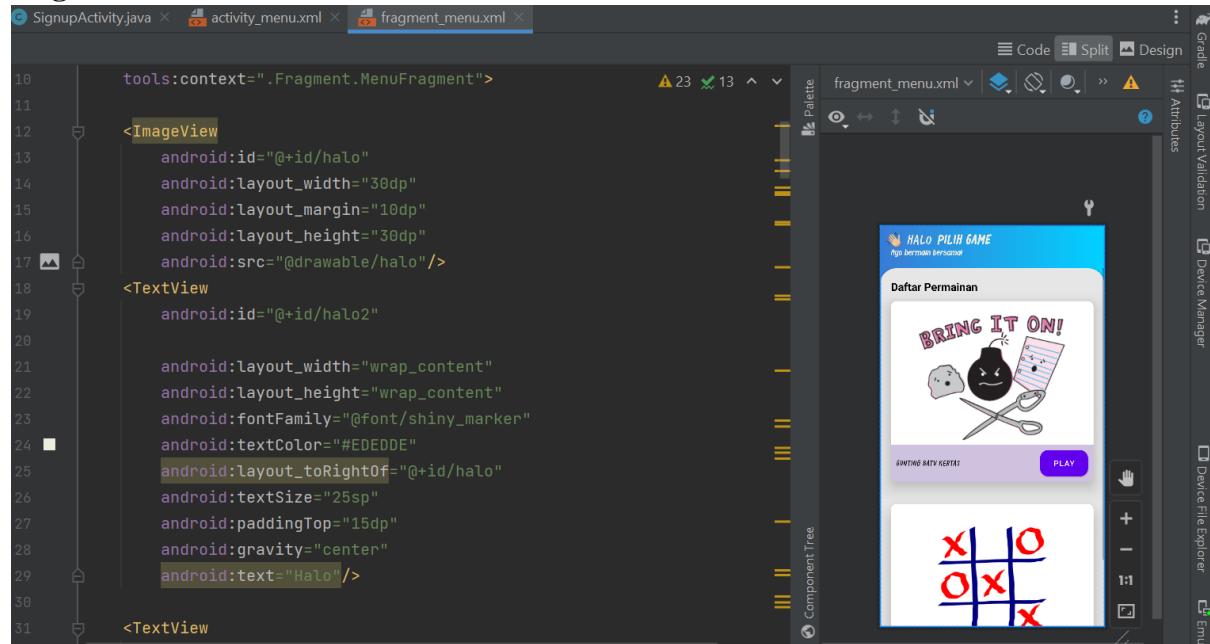
gradiant1.xml



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <shape xmlns:android="http://schemas.android.com/apk/res/android">
3
4     <gradient
5         android:type="linear"
6         android:angle="180"
7         android:startColor="#00d2ff"
8         android:endColor="#3a7bd5">
9     </gradient>
10 </shape>
```

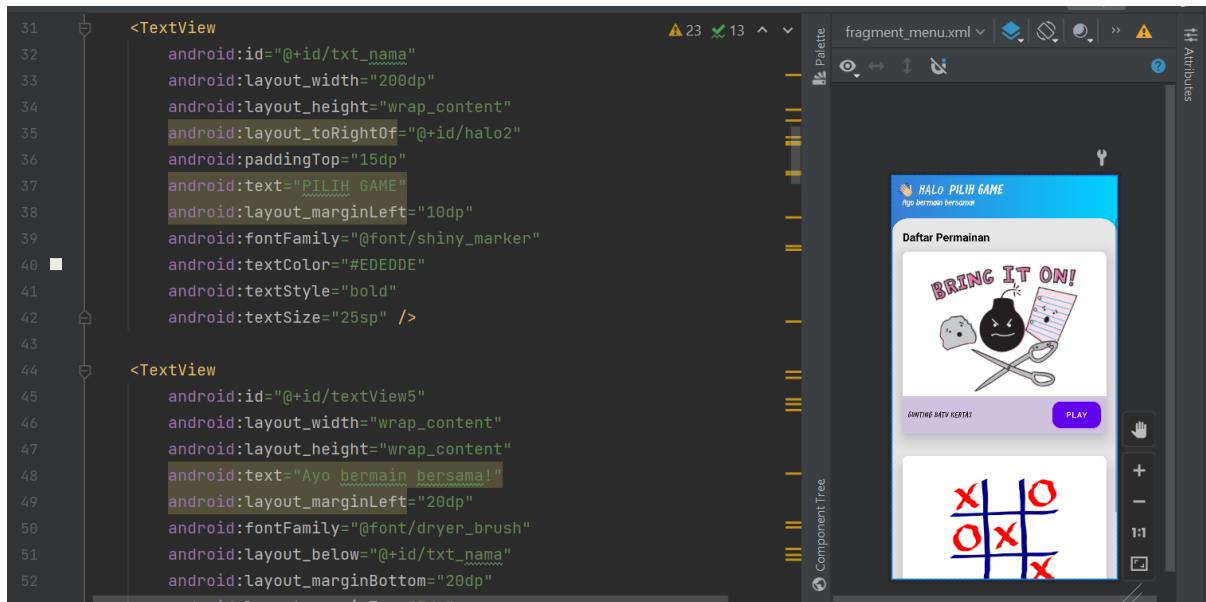
Buatlah file .xml pada drawable untuk membuat warna lebih dari satu dalam satu objek yang sama.

fragment_menu.xml



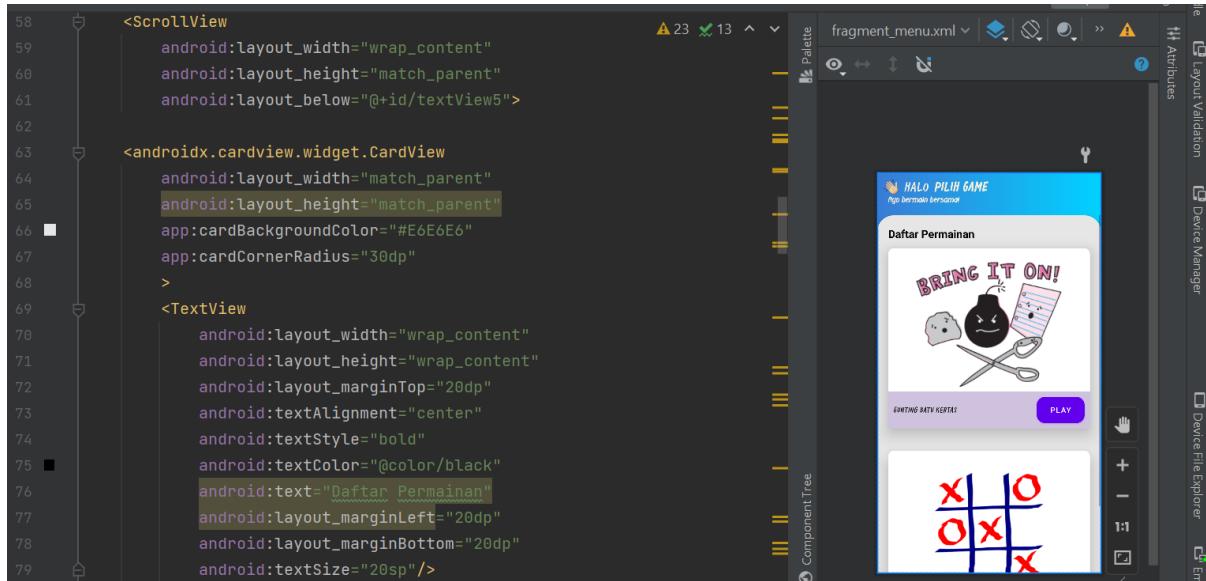
```
10 <tools:context=".Fragment.MenuFragment">
11
12     <ImageView
13         android:id="@+id/halo"
14         android:layout_width="30dp"
15         android:layout_margin="10dp"
16         android:layout_height="30dp"
17         android:src="@drawable/halo"/>
18
19     <TextView
20         android:id="@+id/halo2"
21
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:fontFamily="@font/shiny_marker"
25         android:textColor="#EDEDDE"
26         android:layout_toRightOf="@+id/halo"
27         android:textSize="25sp"
28         android:paddingTop="15dp"
29         android:gravity="center"
30         android:text="Halo"/>
31
32     <TextView
```

Kemudian buatlah sebuah fragment baru dan beri nama MenuFragment untuk diletakkan pada MainActivity agar *user* dapat memilih game yang mereka inginkan. Kemudian tambahkan code-code berikut. Pada background parent tambahkan gradiant yang dibuat sebelumnya.



```
51 <TextView
52     android:id="@+id/txt_nama"
53     android:layout_width="200dp"
54     android:layout_height="wrap_content"
55     android:layout_toRightOf="@+id/halo2"
56     android:paddingTop="15dp"
57     android:text="PILIH GAME"
58     android:layout_marginLeft="10dp"
59     android:fontFamily="@font/shiny_marker"
60     android:textColor="#EDEDEE"
61     android:textStyle="bold"
62     android:textSize="25sp" />
63
64 <TextView
65     android:id="@+id/textView5"
66     android:layout_width="wrap_content"
67     android:layout_height="wrap_content"
68     android:text="Ayo bermain bersama!"
69     android:layout_marginLeft="20dp"
70     android:fontFamily="@font/dryer_brush"
71     android:layout_below="@+id/txt_nama"
72     android:layout_marginBottom="20dp" data-bbox="190 100 450 340"/>
```

Tambahkan 1 TextView di samping text “Halo” untuk menempatkan nama *user* nantinya.



```
58 <ScrollView
59     android:layout_width="wrap_content"
60     android:layout_height="match_parent"
61     android:layout_below="@+id/textView5">
62
63     <androidx.cardview.widget.CardView
64         android:layout_width="match_parent"
65         android:layout_height="match_parent"
66         app:cardBackgroundColor="#E6E6E6"
67         app:cardCornerRadius="30dp"
68         >
69
70         <TextView
71             android:layout_width="wrap_content"
72             android:layout_height="wrap_content"
73             android:layout_marginTop="20dp"
74             android:textAlignment="center"
75             android:textStyle="bold"
76             android:textColor="@color/black"
77             android:text="Daftar Permainan"
78             android:layout_marginLeft="20dp"
79             android:layout_marginBottom="20dp"
80             android:textSize="20sp" />
81
82     </CardView>
83
84 </ScrollView>
```

Letakkan CardView yang akan diletakkan pilihan game ke dalam ScrollView agar *user* dapat mengscroll tampilan di layar yang lebih kecil.

```

82 <LinearLayout
83     android:layout_width="match_parent"
84     android:layout_height="match_parent"
85     android:layout_marginTop="40dp"
86     android:orientation="vertical">
87
88
89     <androidx.cardview.widget.CardView
90         android:id="@+id/cardSuit"
91         android:layout_width="match_parent"
92         android:layout_height="match_parent"
93         android:layout_margin="20dp"
94         android:clickable="true"
95         android:foreground="?android:attr/selectableItemBackground"
96         app:cardCornerRadius="10dp"
97         app:cardElevation="20dp"
98         app:cardPreventCornerOverlap="true">
99
100        <RelativeLayout
101            android:layout_width="wrap_content"
102            android:layout_height="wrap_content">
103
104            <ImageView
105                android:id="@+id/img_suit"
106                android:layout_width="367dp"
107                android:layout_height="match_parent"
108                android:layout_centerHorizontal="true"
109                android:src="@drawable/pilih_suit" />
110
111            <RelativeLayout
112                android:layout_width="match_parent"
113                android:layout_height="wrap_content"
114                android:layout_below="@+id/img_suit"
115                android:background="@color/purple"
116                android:orientation="horizontal">
117
118                <TextView
119                    android:id="@+id/suit"
120                    android:layout_width="wrap_content"
121                    android:layout_height="wrap_content"
122                    android:layout_centerVertical="true"
123                    android:layout_marginLeft="10dp"
124                    android:fontFamily="@font/shiny_marker"
125                    android:text="Gunting Batu Kertas" />
126
127                <Button
128                    android:id="@+id/btnSuit"
129                    android:layout_width="wrap_content"
130                    android:layout_height="wrap_content"
131                    android:layout_alignParentRight="true"
132                    android:layout_marginVertical="10dp"
133                    android:layout_marginRight="10dp"
134                    android:background="@drawable/radius"
135                    android:text="Play" />
136
137
138            </RelativeLayout>
139        </RelativeLayout>
140    </androidx.cardview.widget.CardView>
141
142
143

```

Kemudian tambahkan CardView lagi di dalam CardView sebelumnya untuk membuat pilihan menu gamenya. Dalam CardView tersebut tambahkan ImageView untuk gambar permainan, 1 TextView untuk nama permainannya, dan 1 Button untuk berpindah activity ke game.

fragment_profile.xml



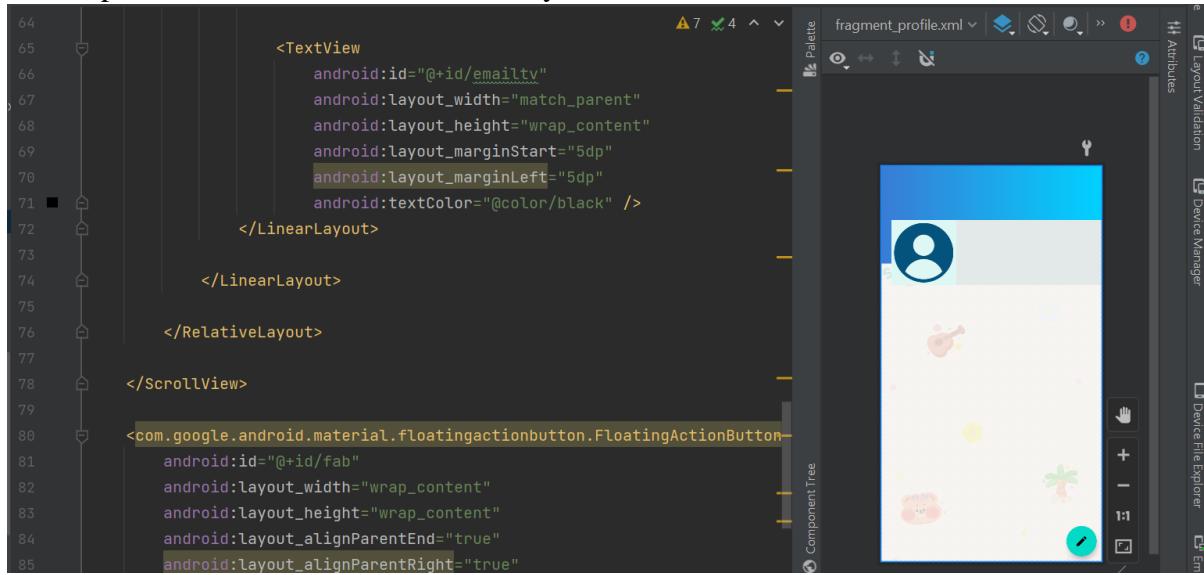
```
fragment_menu.xml x fragment_profile.xml x
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="#F1EDED"
8     tools:context=".Fragment.ProfileFragment">
9
10    <ImageView
11        android:layout_width="match_parent"
12        android:layout_height="match_parent"
13        android:alpha="0.4"
14        android:src="@drawable/bg1" />
15
16    <ScrollView
17        android:layout_width="match_parent"
18        android:layout_height="wrap_content">
19
20        <RelativeLayout
21            android:layout_width="match_parent"
22            android:layout_height="wrap_content">
```

Kemudian buatlah sebuah fragment baru dan beri nama ProfileFragment. Pada file fragment_profile.xml tambahkan sintaks-sintaks berikut untuk menampilkan detail *user* nantinya.



```
24
25    <ImageView
26        android:id="@+id/caverty"
27        android:layout_width="match_parent"
28        android:layout_height="180dp"
29        android:background="@drawable/gradiant1"
30        android:scaleType="fitXY">
31    </ImageView>
32
33    <LinearLayout
34        android:id="@+id/linearLayout"
35        android:layout_width="match_parent"
36        android:layout_height="120dp"
37        android:layout_marginTop="100dp"
38        android:orientation="horizontal">
39
40        <ImageView
41            android:id="@+id/avatarv"
42            android:layout_width="120dp"
43            android:layout_height="120dp"
44            android:background="@color/baby_blue"
45            android:layout_gravity="center_horizontal"
46
47            android:layout_marginStart="20dp"
48            android:padding="5dp"
49            android:src="@drawable/landing_page_login" />
50
51        <LinearLayout
52            android:layout_width="match_parent"
53            android:layout_height="match_parent"
54            android:background="@color/grey"
55            android:orientation="vertical">
56
57            <TextView
58                android:id="@+id/nametv"
59                android:layout_width="match_parent"
60                android:layout_height="wrap_content"
61                android:layout_marginStart="5dp"
62                android:layout_marginLeft="5dp"
63                android:layout_marginTop="20dp"
64                android:textColor="@color/black"
65                android:textSize="25sp" />
```

Tambahkan ImageView untuk meletakkan foto profile *user* dan 2 TextView untuk menampilkan email dan nama *user* nantinya.



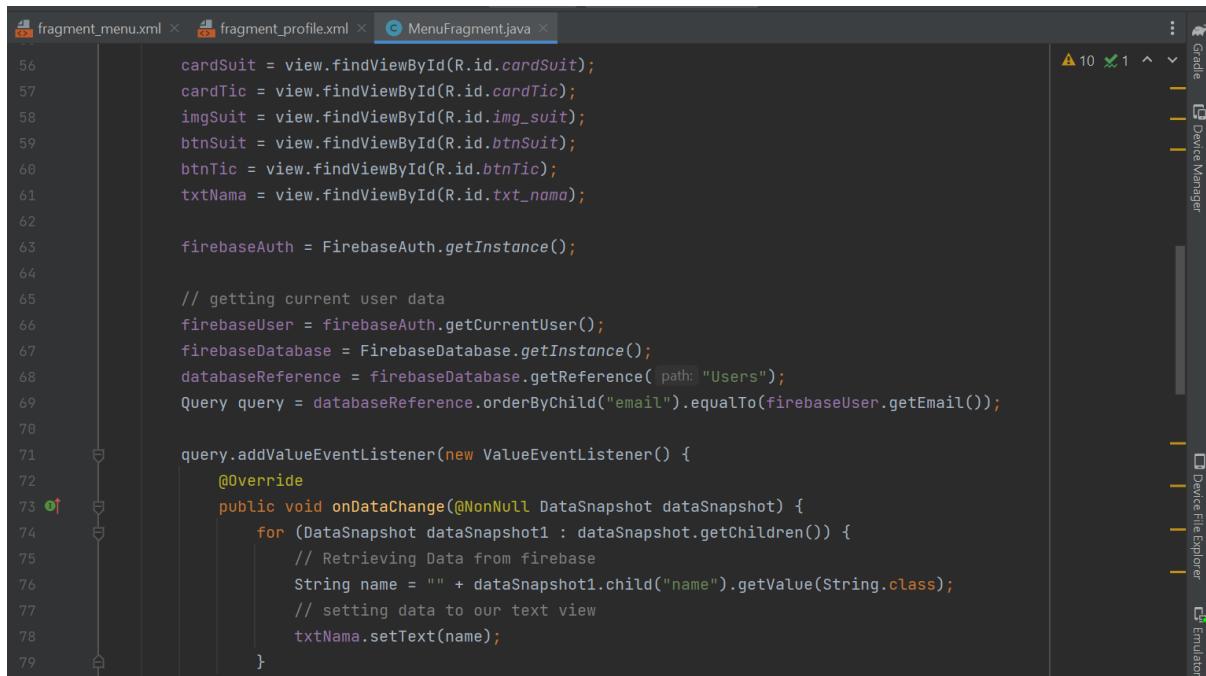
Kemudian letakkan FloatingActionButton agar gambar yang diinginkan dapat melayang dan selalu muncul dibawah. Gambar ini berfungsi untuk melakukan update pada *profile user*.

MenuFragment.java

```
fragment_menu.xml x fragment_profile.xml x MenuFragment.java x
31
32     public class MenuFragment extends Fragment {
33
34         ImageView imgSuit;
35         ImageView imgTic;
36         TextView txtNama;
37         CardView cardsSuit;
38         CardView cardTic;
39         Button btnSuit,btnTic;
40         private FirebaseAuth firebaseAuth;
41         FirebaseUser firebaseUser;
42         FirebaseDatabase firebaseDatabase;
43         DatabaseReference databaseReference;
44
45
46     public MenuFragment() {
47         // require a empty public constructor
48     }
49
50
51     @Override
52     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
53         // Inflate the layout for this fragment
54         View view = inflater.inflate(R.layout.fragment_menu, container, attachToRoot: false);
```

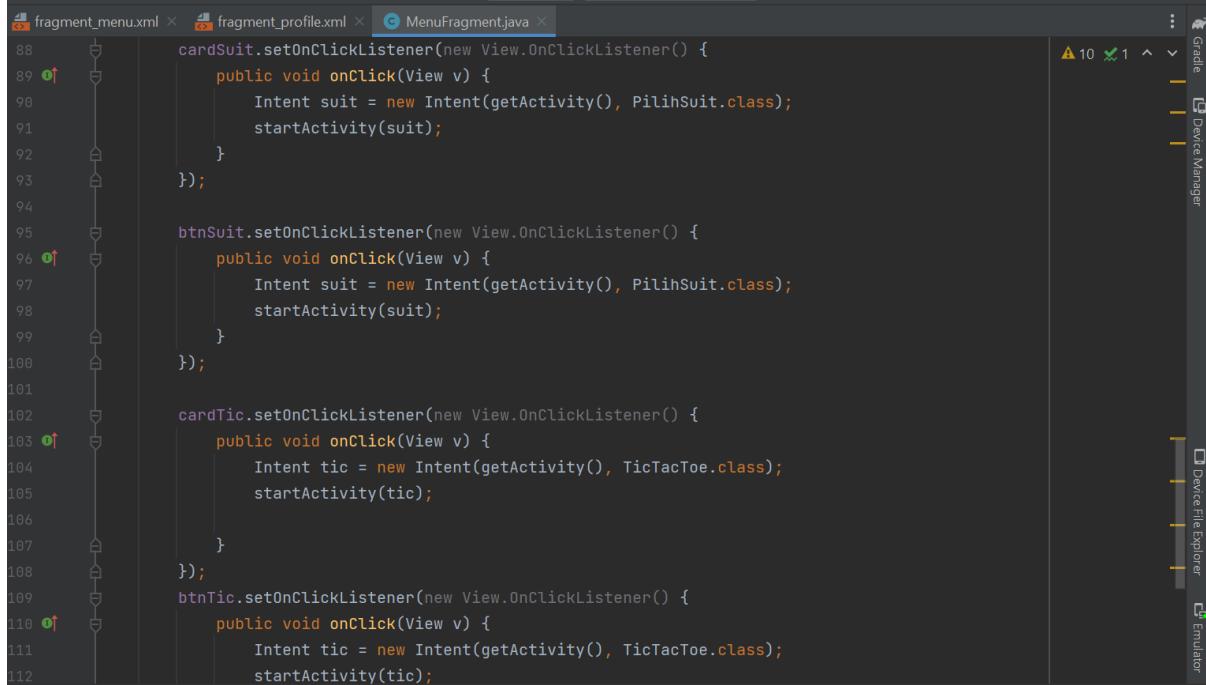
The screenshot shows the Java code for the MenuFragment. It starts with a standard fragment constructor. Then it defines several fields: imgSuit, imgTic, txtNama, cardsSuit, cardTic, btnSuit, btnTic, firebaseAuth, firebaseUser, firebaseDatabase, and databaseReference. Finally, it overrides the onCreateView method to inflate the fragment_menu.xml layout using an inflater.

Pertama panggil beberapa kelas firebase yang dibutuhkan untuk mengambil data dan beberapa View untuk memanggil View dalam file xml. Kemudian gunakan inflater untuk menyambungkan java class dengan file xml yang ditentukan.



```
fragment_menu.xml fragment_profile.xml MenuFragment.java
56     cardSuit = view.findViewById(R.id.cardSuit);
57     cardTic = view.findViewById(R.id.cardTic);
58     imgSuit = view.findViewById(R.id.img_suit);
59     btnSuit = view.findViewById(R.id.btnSuit);
60     btnTic = view.findViewById(R.id.btnTic);
61     txtNama = view.findViewById(R.id.txt_nama);
62
63     firebaseAuth = FirebaseAuth.getInstance();
64
65     // getting current user data
66     firebaseUser = firebaseAuth.getCurrentUser();
67     FirebaseDatabase database = FirebaseDatabase.getInstance();
68     DatabaseReference reference = database.getReference("Users");
69     Query query = reference.orderByChild("email").equalTo(firebaseUser.getEmail());
70
71     query.addValueEventListener(new ValueEventListener() {
72         @Override
73         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
74             for (DataSnapshot dataSnapshot1 : dataSnapshot.getChildren()) {
75                 // Retrieving Data from firebase
76                 String name = "" + dataSnapshot1.child("name").getValue(String.class);
77                 // setting data to our text view
78                 txtNama.setText(name);
79             }
80         }
81     });
82 }
```

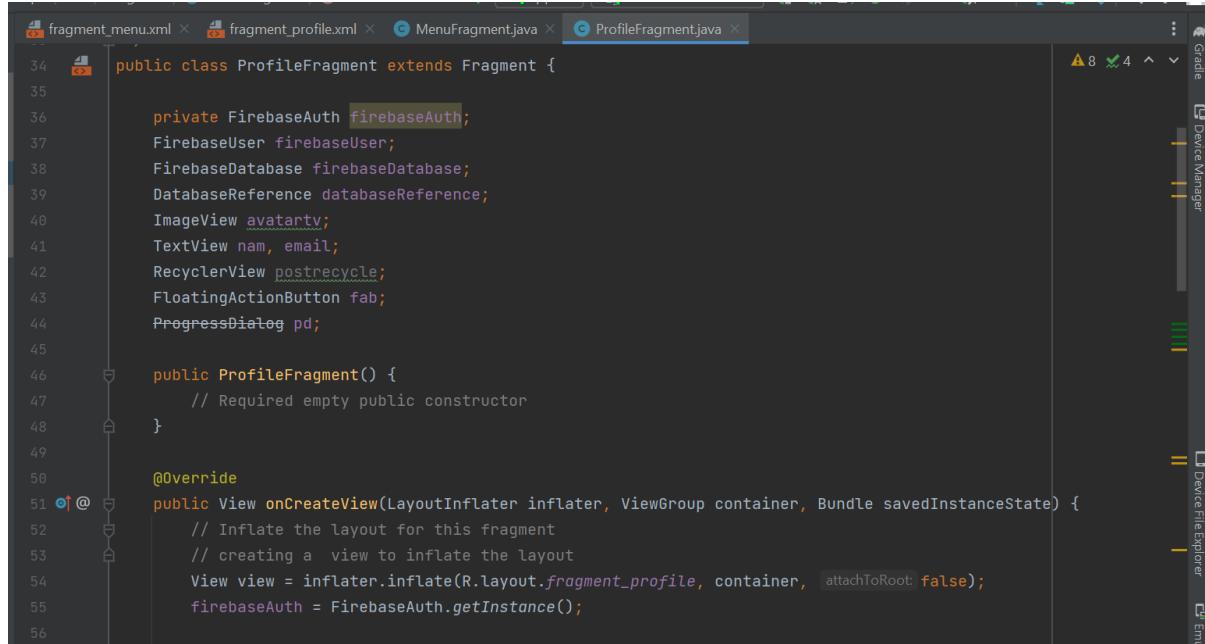
Selanjutnya panggil id dari View .xml yang dibutuhkan dan letakkan ke dalam variabel. Kemudian gunakan Database Reference untuk membaca data yang ada di lokasi “Users” dalam firebase. Kemudian kelas FirebaseAuth berfungsi untuk mengambil informasi profil pengguna yaitu email. Pada line 71-79 berfungsi untuk mengambil value dari “name” pada firebase berdasarkan email user kemudian mengganti txtName sesuai dengan value yang diambil.



```
fragment_menu.xml fragment_profile.xml MenuFragment.java
88     cardSuit.setOnClickListener(new View.OnClickListener() {
89         public void onClick(View v) {
90             Intent suit = new Intent(getActivity(), PilihSuit.class);
91             startActivity(suit);
92         }
93     });
94
95     btnSuit.setOnClickListener(new View.OnClickListener() {
96         public void onClick(View v) {
97             Intent suit = new Intent(getActivity(), PilihSuit.class);
98             startActivity(suit);
99         }
100    });
101
102    cardTic.setOnClickListener(new View.OnClickListener() {
103        public void onClick(View v) {
104            Intent tic = new Intent(getActivity(), TicTacToe.class);
105            startActivity(tic);
106        }
107    });
108    btnTic.setOnClickListener(new View.OnClickListener() {
109        public void onClick(View v) {
110            Intent tic = new Intent(getActivity(), TicTacToe.class);
111            startActivity(tic);
112        }
113    });
114 }
```

Selanjutnya code diatas berfungsi untuk memindahkan user ke activity yang telah ditentukan berdasarkan Button atau Card yang mereka klik dengan menggunakan kelas intent.

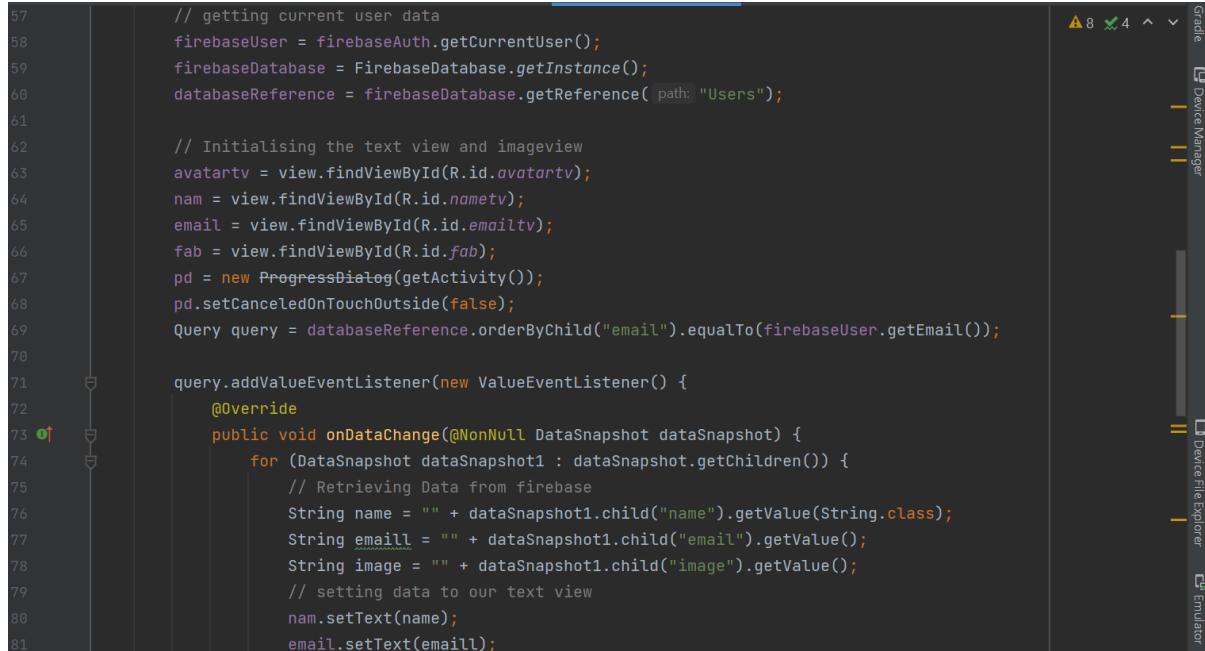
ProfileFragment.java



```
34 public class ProfileFragment extends Fragment {
35
36     private FirebaseAuth firebaseAuth;
37     FirebaseAuth firebaseUser;
38     FirebaseDatabase firebaseDatabase;
39     DatabaseReference databaseReference;
40     ImageView avatartv;
41     TextView nam, email;
42     RecyclerView postrecycle;
43     FloatingActionButton fab;
44     ProgressDialog pd;
45
46     public ProfileFragment() {
47         // Required empty public constructor
48     }
49
50     @Override
51     public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {
52         // Inflate the layout for this fragment
53         // creating a view to inflate the layout
54         View view = inflater.inflate(R.layout.fragment_profile, container, attachToRoot: false);
55         firebaseAuth = FirebaseAuth.getInstance();
56     }
57 }
```

The screenshot shows the Android Studio interface with the ProfileFragment.java file open in the main editor. The code defines a Fragment class named ProfileFragment. It imports several Firebase-related classes: FirebaseAuth, FirebaseAuthUser, FirebaseDatabase, DatabaseReference, ImageView, TextView, RecyclerView, FloatingActionButton, and ProgressDialog. The constructor is empty. The onCreateView method inflates the layout from R.layout.fragment_profile and sets the attachToRoot parameter to false. It also initializes the FirebaseAuth instance. The code is annotated with line numbers and some JavaDoc-style annotations.

Sama seperti fragment sebelumnya, pertama panggil beberapa kelas firebase yang dibutuhkan untuk mengambil data *user* dan beberapa View untuk memanggil View beserta FloatingActionButton dalam file xml. Kemudian digunakan inflater untuk menyambungkan java class dengan file xml yang ditentukan.



```
57     // getting current user data
58     firebaseUser = FirebaseAuth.getCurrentUser();
59     FirebaseDatabase firebaseDatabase = FirebaseDatabase.getInstance();
60     databaseReference = firebaseDatabase.getReference(path: "Users");
61
62     // Initialising the text view and imageview
63     avatartv = view.findViewById(R.id.avatartv);
64     nam = view.findViewById(R.id.nametv);
65     email = view.findViewById(R.id.emaitv);
66     fab = view.findViewById(R.id.fab);
67     pd = new ProgressDialog(getActivity());
68     pd.setCanceledOnTouchOutside(false);
69     Query query = databaseReference.orderByChild("email").equalTo(firebaseUser.getEmail());
70
71     query.addValueEventListener(new ValueEventListener() {
72         @Override
73         public void onDataChange(@NotNull DataSnapshot dataSnapshot) {
74             for (DataSnapshot dataSnapshot1 : dataSnapshot.getChildren()) {
75                 // Retrieving Data from firebase
76                 String name = "" + dataSnapshot1.child("name").getValue(String.class);
77                 String email = "" + dataSnapshot1.child("email").getValue();
78                 String image = "" + dataSnapshot1.child("image").getValue();
79                 // setting data to our text view
80                 nam.setText(name);
81                 email.setText(email);
82         }
83     }
84 }
```

This part of the code implements the ValueEventListener interface. It retrieves the child nodes "name", "email", and "image" under the "Users" reference. The retrieved values are then set to the corresponding TextViews (nam and email) and ImageView (avatartv).

Selanjutnya panggil id dari View .xml yang dibutuhkan dan letakkan ke dalam variabel. Kemudian gunakan Database Reference untuk membaca data yang ada di lokasi “Users” dalam firebase. Kemudian kelas FirebaseAuth berfungsi untuk mengambil informasi profil pengguna yaitu email. Pada line 71-81 berfungsi untuk mengambil *value* dari “name”, “image” dan “email” pada firebase berdasarkan email *user* kemudian mengganti TextView yaitu nam dan email sesuai dengan *value* yang diambil.

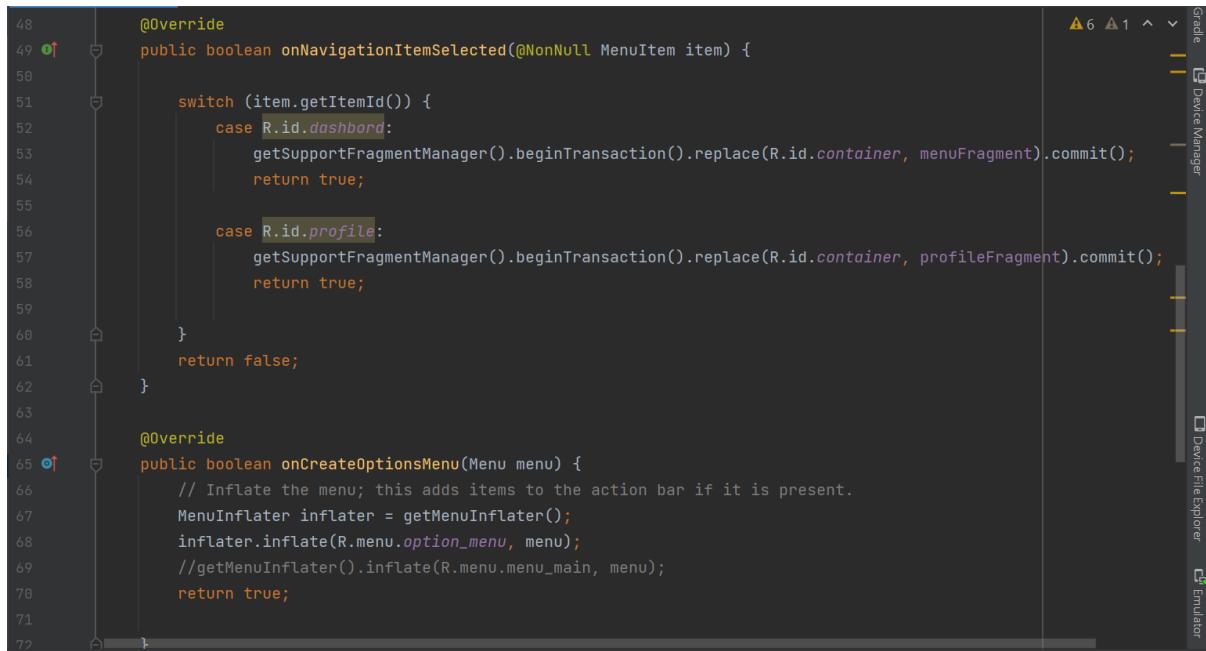
```
82     try {
83         Glide.with(getActivity()).load(image).into(avatartv);
84     } catch (Exception e) {
85     }
86 }
87 }
88 }
89
90 @Override
91 public void onCancelled(@NonNull DatabaseError databaseError) {
92 }
93 }
94 });
95
96 // On click we will open EditProfileActivity
97 fab.setOnClickListener(new View.OnClickListener() {
98     @Override
99     public void onClick(View v) {
100         startActivity(new Intent(getActivity(), EditProfilePage.class));
101     }
102 });
103
104 return view;
105 }
106
107 @Override
108 public void onCreate(@Nullable Bundle savedInstanceState) {
109     setHasOptionsMenu(true);
110     super.onCreate(savedInstanceState);
111 }
```

Untuk value dari image akan diletakkan ke dalam ImageView yaitu avatartv (line 83). Kemudian apabila FloatingActionButton fab diklik, maka *user* akan berpindah activity ke EditProfilPage.

MenuActivity.java

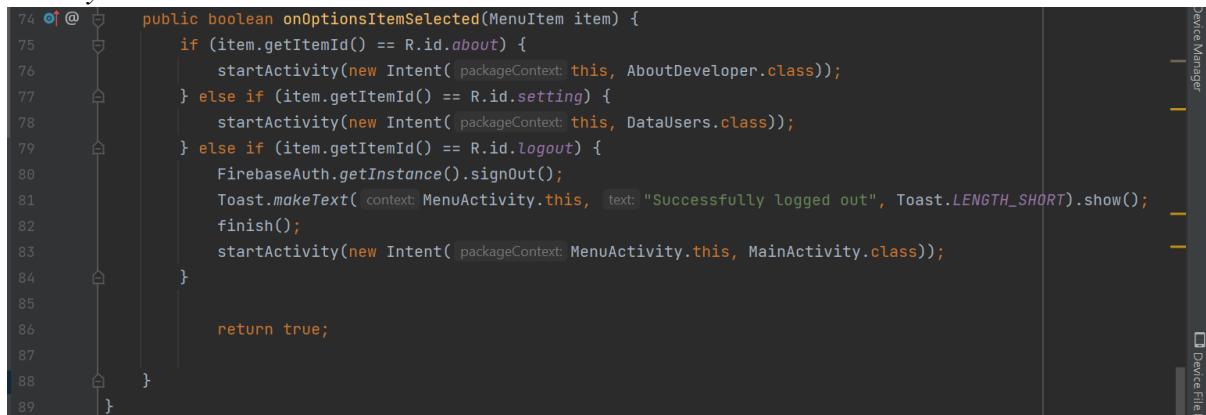
```
24 public class MenuActivity extends AppCompatActivity implements BottomNavigationView.OnNavigationItemSe
25
26     BottomNavigationView bottomNavigationView;
27     //MediaPlayer menu;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_menu);
33         //menu = MediaPlayer.create(this, R.raw.menu);
34         //menu.start();
35
36
37         bottomNavigationView = findViewById(R.id.bottomNavigationView);
38
39         bottomNavigationView.setOnNavigationItemSelectedListener(this);
40         bottomNavigationView.setSelectedItemId(R.id.dashboard);
41
42     }
43
44     MenuFragment menuFragment = new MenuFragment();
45     ProfileFragment profileFragment = new ProfileFragment();
```

Selanjutnya pada file MenuActivity.java, panggil id dari bottomNavigationView agar bisa diakses ke kelas java (line 37). Line 37 berfungsi untuk memberi listener saat menu BottomNavigation diklik. Kemudian panggil class Fragment yang telah dibuat sebelumnya (line 45 - 46).



```
48     @Override
49     public boolean onNavigationItemSelected(@NonNull MenuItem item) {
50
51         switch (item.getItemId()) {
52             case R.id.dashboard:
53                 getSupportFragmentManager().beginTransaction().replace(R.id.container, menuFragment).commit();
54                 return true;
55
56             case R.id.profile:
57                 getSupportFragmentManager().beginTransaction().replace(R.id.container, profileFragment).commit();
58                 return true;
59
60         }
61         return false;
62     }
63
64     @Override
65     public boolean onCreateOptionsMenu(Menu menu) {
66         // Inflate the menu; this adds items to the action bar if it is present.
67         MenuInflater inflater = getMenuInflater();
68         inflater.inflate(R.menu.option_menu, menu);
69         //getMenuInflater().inflate(R.menu.menu_main, menu);
70         return true;
71     }
72 }
```

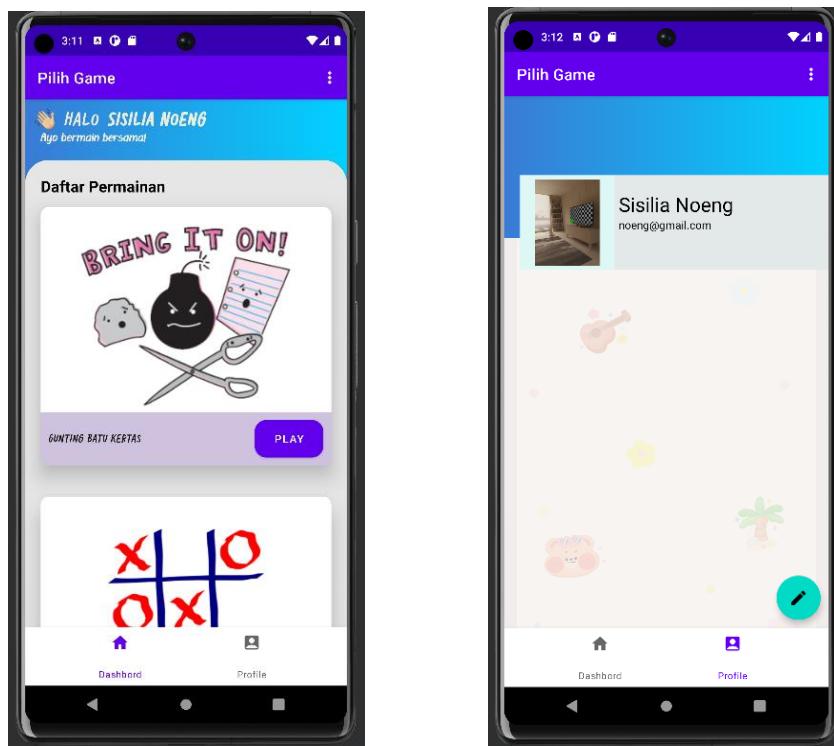
Method pada line 49 – 62 berfungsi untuk berpindah-pindah fragment saat menu BottomNavigation diklik berdasarkan id yang ada di bottom_nav sebelumnya. Fragment tersebut akan diletakkan di FrameLayout pada activity_menu sebelumnya yang diberi id “container”. Kemudian *method* pada line 65 – 72 berfungsi untuk memanggil file option_menu.xml yang telah dibuat sebelumnya agar menu 3 titik dapat ditampilkan dalam *activity* ini.



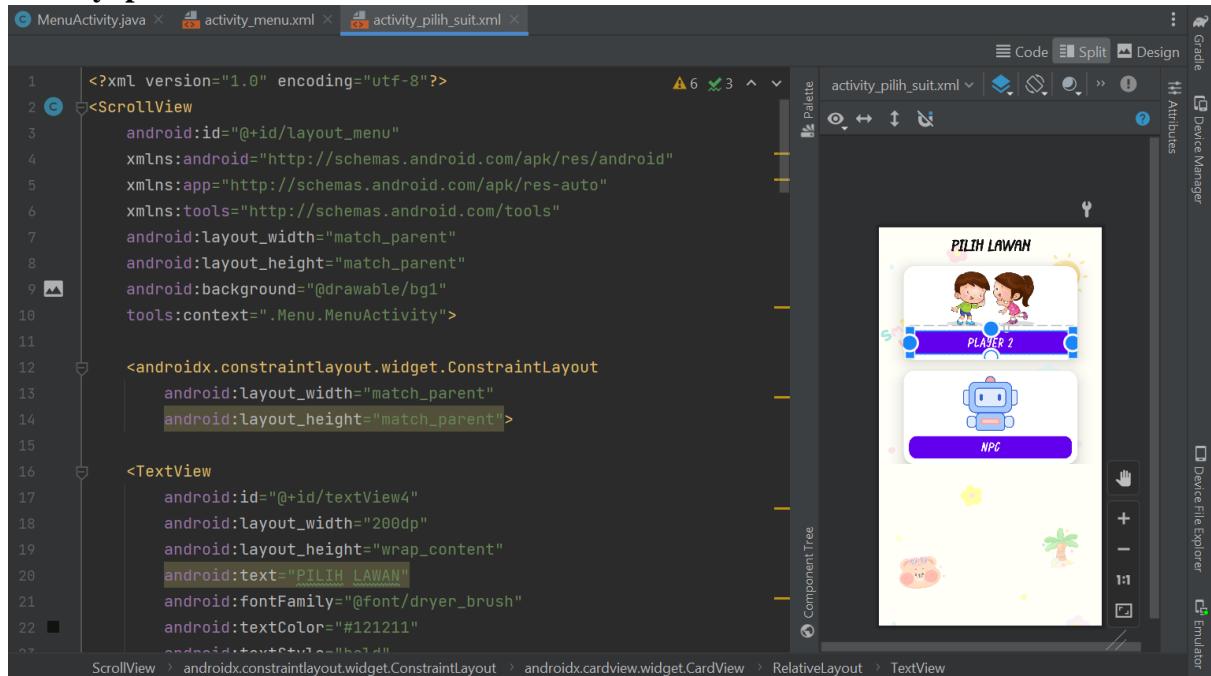
```
74     @Override
75     public boolean onOptionsItemSelected(MenuItem item) {
76         if (item.getItemId() == R.id.about) {
77             startActivity(new Intent(getApplicationContext(), AboutDeveloper.class));
78         } else if (item.getItemId() == R.id.setting) {
79             startActivity(new Intent(getApplicationContext(), DataUsers.class));
80         } else if (item.getItemId() == R.id.logout) {
81             FirebaseAuth.getInstance().signOut();
82             Toast.makeText(getApplicationContext(), "Successfully logged out", Toast.LENGTH_SHORT).show();
83             finish();
84             startActivity(new Intent(getApplicationContext(), MainActivity.class));
85         }
86
87         return true;
88     }
89 }
```

Method di atas berfungsi untuk berpindah *activity* berdasarkan menu item yang diklik *user*. Kemudian pada line 80 – 83 berfungsi untuk menglogout akun *user* dan akan langsung berpindah ke MainActivity ketika telah berhasil melakukan logout.

Emulator



activity_pilih_suit.xml



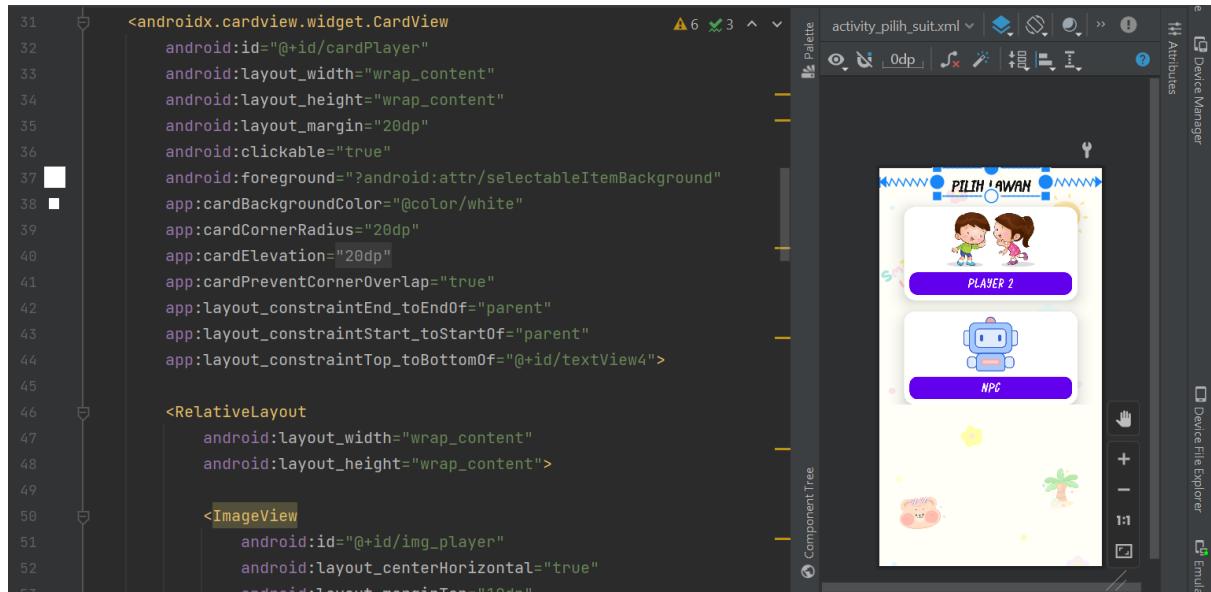
The screenshot shows the Android Studio interface with the XML code for `activity_pilih_suit.xml` in the left pane and the design preview in the right pane. The XML code defines a `ScrollView` containing a `ConstraintLayout` with a `TextView` labeled "PILIH LAWAN". The design preview shows a colorful interface with two children at the top, a blue robot icon in the center, and text labels "PLAYER 2" and "NPC" below it. The background features a sun, clouds, and palm trees.

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    android:id="@+id/layout_menu"
    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:background="@drawable/bg1"
    tools:context=".Menu.MenuActivity">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/textView4"
            android:layout_width="200dp"
            android:layout_height="wrap_content"
            android:text="PILIH LAWAN"
            android:fontFamily="@font/dryer_brush"
            android:textColor="#121211"
            android:layout_margin="10dp" />
    
```

Selanjutnya buatlah sebuah *activity* baru dan beri nama PilihSuit. *Activity* ini berfungsi untuk memberi pilihan lawan kepada *user*. Ketika pada *MenuActivity*, *user* memilih bermain gunting batu kertas maka *user* akan dialihkan ke *activity* ini. Gunakan *ScrollView* agar *user* bisa melihat keseluruhan tampilan meskipun menggunakan layar kecil atau dalam *landscape*.



The screenshot shows the Android Studio interface with the XML code for `activity_pilih_suit.xml` in the left pane and the design preview in the right pane. The XML code defines a `CardView` with a purple background and white text, containing a `RelativeLayout` with an `ImageView` and a `TextView` labeled "PILIH LAWAN". The design preview shows the same colorful interface as the previous screenshot, with the title bar "PILIH LAWAN" and the "PLAYER 2" and "NPC" options.

```
<androidx.cardview.widget.CardView
    android:id="@+id/cardPlayer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="20dp"
    android:clickable="true"
    android:foreground="?android:attr/selectableItemBackground"
    app:cardBackgroundColor="@color/white"
    app:cardCornerRadius="20dp"
    app:cardElevation="20dp"
    app:cardPreventCornerOverlap="true"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView4">

    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/img_player"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="10dp" />
    
```

```
55     android:layout_height="100dp"
56     android:src="@drawable/suit_player" />
57
58 <TextView
59     android:id="@+id/suit"
60     android:layout_width="300dp"
61     android:layout_height="wrap_content"
62     android:layout_below="@+id/img_player"
63     android:layout_centerHorizontal="true"
64     android:layout_marginStart="10dp"
65     android:layout_marginTop="10dp"
66     android:layout_marginEnd="10dp"
67     android:layout_marginBottom="10dp"
68     android:background="@drawable/radius"
69     android:fontFamily="@font/shiny_marker"
70     android:padding="10dp"
71     android:text="Player 2"
72     android:textAlignment="center"
73     android:textColor="@color/white"
74     android:textSize="25sp" />
75 </RelativeLayout>
76 </androidx.cardview.widget.CardView>
```

Untuk pilihan menu lawannya, digunakan CardView untuk menampung ImageView dan TextView.

PilihSuit.java

```
35  
36  
37     @Override  
38     protected void onCreate(Bundle savedInstanceState) {  
39         super.onCreate(savedInstanceState);  
40         setContentView(R.layout.activity_pilih_suit);  
41  
42         cardPemain = findViewById(R.id.cardPlayer);  
43         cardNPC = findViewById(R.id.cardNPC);  
44  
45         cardPemain.setOnClickListener(new View.OnClickListener() {  
46             public void onClick(View v) {  
47                 Intent player = new Intent(getApplicationContext(), SuitPlayer.class);  
48                 startActivity(player);  
49             }  
50         });  
51  
52         cardNPC.setOnClickListener(new View.OnClickListener() {  
53             public void onClick(View v) {  
54                 Intent NPC = new Intent(getApplicationContext(), SuitNPC.class);  
55                 startActivity(NPC);  
56             }  
57         });  
58     }  
59  
60 }
```

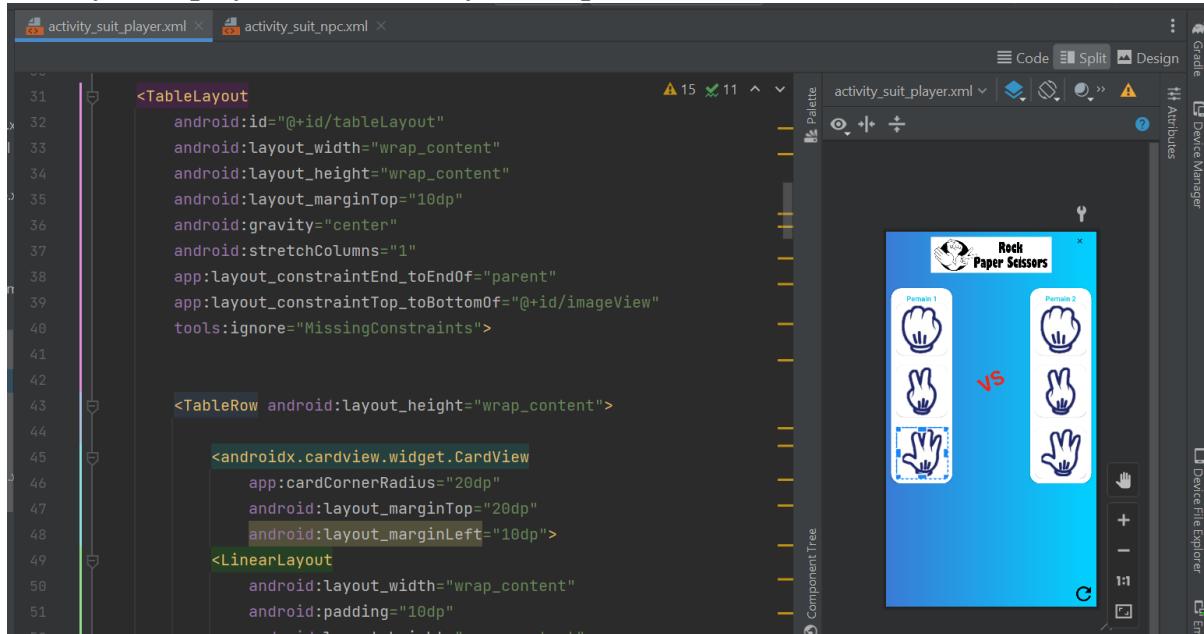
Code di atas berfungsi agar *user* dapat berpindah *activity* ketika mengklik CardView sesuai dengan lawan yang mereka inginkan.

Emulator



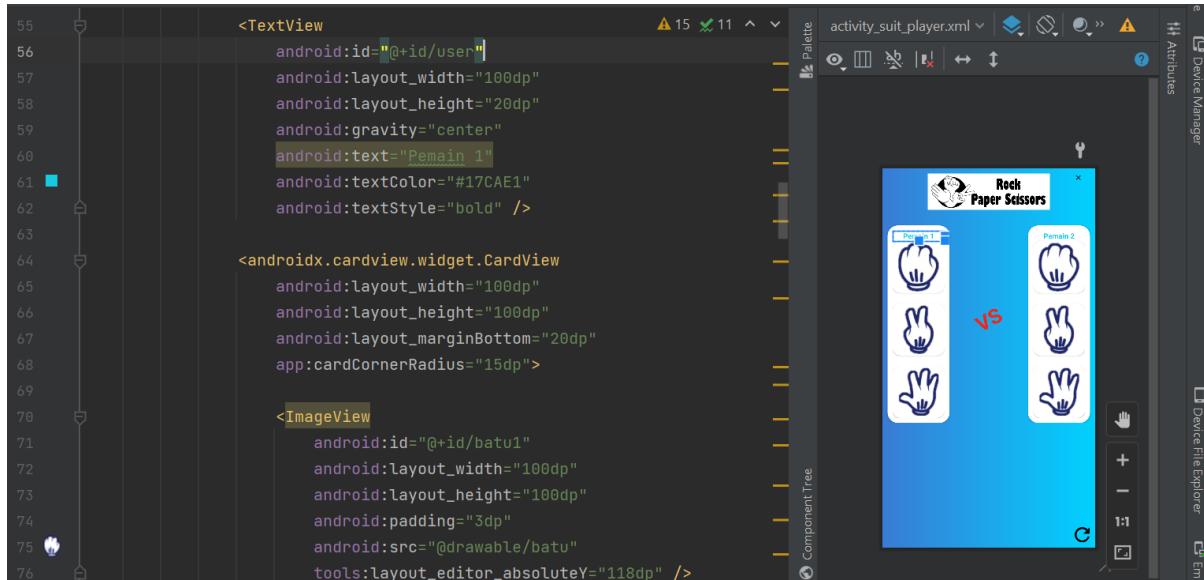
Suit Game

activity_suit_player.xml & activity_suit_npc.xml



```
31 <TableLayout
32     android:id="@+id/tableLayout"
33     android:layout_width="wrap_content"
34     android:layout_height="wrap_content"
35     android:layout_marginTop="10dp"
36     android:gravity="center"
37     android:stretchColumns="1"
38     app:layout_constraintEnd_toEndOf="parent"
39     app:layout_constraintTop_toBottomOf="@+id/imageView"
40     tools:ignore="MissingConstraints">
41
42     <TableRow android:layout_height="wrap_content">
43
44         <androidx.cardview.widget.CardView
45             app:cardCornerRadius="20dp"
46             android:layout_marginTop="20dp"
47             android:layout_marginLeft="10dp">
48             <LinearLayout
49                 android:layout_width="wrap_content"
50                 android:padding="10dp">
51
52             <Image>
```

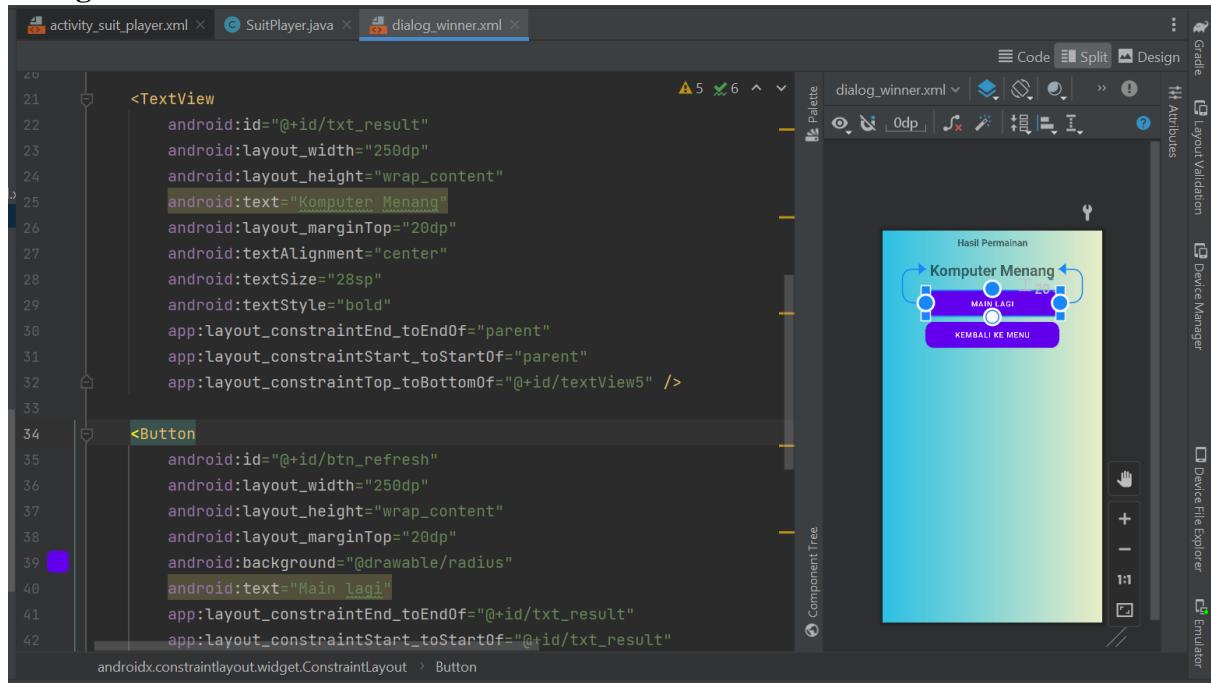
Untuk bagian tampilan game, saya menggunakan Table Layout agar bisa dibuat row sehingga pilihan *user* dan pilihan lawan dapat sejajar. Kemudian saya meletakkan CardView untuk CardView pilihan.



```
55 <TextView
56     android:id="@+id/user1"
57     android:layout_width="100dp"
58     android:layout_height="20dp"
59     android:gravity="center"
60     android:text="Pemain 1"
61     android:textColor="#17CAE1"
62     android:textStyle="bold" />
63
64 <androidx.cardview.widget.CardView
65     android:layout_width="100dp"
66     android:layout_height="100dp"
67     android:layout_marginBottom="20dp"
68     app:cardCornerRadius="15dp">
69
70     <ImageView
71         android:id="@+id/batu1"
72         android:layout_width="100dp"
73         android:layout_height="100dp"
74         android:padding="3dp"
75         android:src="@drawable/batu"
76         tools:layout_editor_absoluteY="118dp" />
```

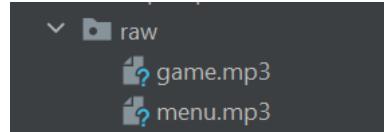
Untuk setiap pilihan suit digunakan CardView yang berisi ImageView dari gambar gunting batu kertas.

dialog_winner.xml

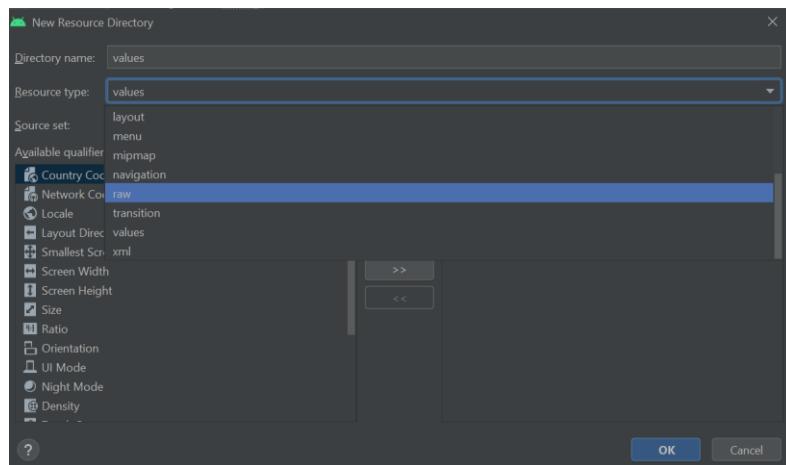


Selanjutnya buatlah sebuah file .xml baru dan beri nama dialog_winner.xml. File ini berfungsi untuk menampilkan hasil permainan yang akan ditampilkan dalam bentuk dialog menggunakan AlertDialog. Dalam file ini terdapat 2 button untuk mengarahkan *user* ke game lagi atau kembali ke menu. Terdapat TextView untuk menampilkan pemenang dari game.

Music



Untuk memberikan music pada game, kita perlu membuat direktori baru dengan nama raw pada folder res. Direktori dapat dibuat dengan cara mengklik kanan pada folder res > New > Android Resource Directory.



Kemudian ubahlah Resouce type menjadi raw. Folder ini untuk menyimpan video atau musik dalam project.

SuitPlayer.java

```
activity_suit_player.xml x SuitPlayer.java x dialog_winner.xml x
37     ImageView imgClose;
38     TextView userGame;
39     int pilihSuit = 0;
40     int hasil = 1;
41     int pilihSuitLawan = 0;
42     MediaPlayer game;
43
44     @Override
45     protected void onCreate(Bundle savedInstanceState) {
46         super.onCreate(savedInstanceState);
47         setContentView(R.layout.activity_suit_player);
48
49         batuPemain = findViewById(R.id.batu1);
50         guntingPemain = findViewById(R.id.gunting1);
51         kertasPemain = findViewById(R.id.kertas1);
52         batuLawan = findViewById(R.id.batu2);
53         guntingLawan = findViewById(R.id.gunting2);
54         kertasLawan = findViewById(R.id.kertas2);
55         hasilSuit = findViewById(R.id.hasil);
56         ulang = findViewById(R.id.refresh);
57         imgClose = findViewById(R.id.img_close);
58         userGame = findViewById(R.id.user);
59         game = MediaPlayer.create(context: this, R.raw.game);
60         game.start();
```

Pertama deklarasi beberapa View yang dibutuhkan dari file .xml sebelumnya. Digunakan kelas MediaPlayer agar dapat mengintegrasikan audio ke dalam aplikasi. Line 60 berfungsi untuk memutar audio yang telah ditentukan pada line 59.

```
62     batuPemain.setOnClickListener(new View.OnClickListener() {
63         public void onClick(View v) {
64             Log.d(tag: "MainActivity", msg: "Pemain 1 memilih batu");
65             pilihSuit = 1;
66             selectSuit();
67             batuPemain.setBackgroundColor(Color.parseColor(colorString: "#FF709EB3"));
68             Toast.makeText(context: SuitPlayer.this, text: "Pemain 1 memilih batu", Toast.LENGTH_SHORT).show();
69         }
70     });
71
72     guntingPemain.setOnClickListener(new View.OnClickListener() {
73         public void onClick(View v) {
74             Log.d(tag: "MainActivity", msg: "Pemain 1 memilih gunting");
75             pilihSuit = 2;
76             selectSuit();
77             guntingPemain.setBackgroundColor(Color.parseColor(colorString: "#FF709EB3"));
78             Toast.makeText(context: SuitPlayer.this, text: "Pemain 1 memilih gunting", Toast.LENGTH_SHORT).show();
79         }
80     });
81
82     kertasPemain.setOnClickListener(new View.OnClickListener() {
83         public void onClick(View v) {
84             Log.d(tag: "MainActivity", msg: "Pemain 1 memilih kertas");
85             pilihSuit = 3;
86             selectSuit();
```

Pada code diatas berfungsi untuk mengambil hasil klik *user* apakah mereka memilih batu, gunting, atau kertas. Setiap pilihan memiliki nilai pilihSuit yang berbeda dalam rentang 1 - 3. Nilai 1 untuk batu, nilai 2 untuk gunting, dan nilai 3 untuk kertas. Hal tersebut digunakan agar memudahkan algoritma saat pemilihan hasil permainan. Ketika *user* mengklik salah satu gambar suit, maka akan tercatat dalam log dan akan menampilkan informasi berupa *toast* tentang apa yang dipilih *user*. Background dari gambar juga akan berubah kemudian akan dipanggil *method* selectSuit().

```
92 batuLawan.setOnClickListener(new View.OnClickListener() {
93     public void onClick(View v) {
94         Log.d( tag: "MainActivity", msg: "Pemain 2 memilih batu");
95         pilihSuitLawan = 1;
96         selectSuit();
97         batuLawan.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
98         Toast.makeText( context: SuitPlayer.this, text: "Pemain 2 memilih batu", Toast.LENGTH_SHORT).show();
99     }
100 });
101
102 guntingLawan.setOnClickListener(new View.OnClickListener() {
103     public void onClick(View v) {
104         Log.d( tag: "MainActivity", msg: "Pemain 2 memilih batu");
105         pilihSuitLawan = 2;
106         selectSuit();
107         guntingLawan.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
108         Toast.makeText( context: SuitPlayer.this, text: "Pemain 2 memilih gunting", Toast.LENGTH_SHORT).show();
109     }
110 });
111
112 kertasLawan.setOnClickListener(new View.OnClickListener() {
113     public void onClick(View v) {
114         Log.d( tag: "MainActivity", msg: "Pemain 2 memilih kertas");
115         pilihSuitLawan = 3;
116         selectSuit();
```

Untuk pilihan lawan memiliki output yang sama seperti pilihan pemain 1. Namun untuk pilihan lawan akan disimpan ke dalam pilihSuitLawan dengan nilai yang sama seperti pemain 1 untuk setiap pilihan suit.

```
123 ulang.setOnClickListener(new View.OnClickListener() {
124     public void onClick(View v) { closeDialog(); }
125 );
126
127 imgClose.setOnClickListener(new View.OnClickListener() {
128     public void onClick(View v) {
129         game.stop();
130         finish();
131     }
132 );
133
134 }
135
136
137 }
138
139 private void closeDialog(){
140     hapusPemain();
141     hapusLawan();
142     reset();
143     pilihSuit = 0;
144     pilihSuitLawan = 0;
145 }
```

Method closeDialog() berfungsi untuk mereset kembali seluruh nilai pilihan pemain 1 maupun pemain lawan dan seluruh warna background dari text maupun gambar dengan memanggil beberapa *method* lain (line 140-142).

```

147     // mengembalikan hasil suit textview ke awal
148     private void reset(){
149         hasilSuit.setTextColor(Color.parseColor( colorString: "#F0A415"));
150         hasilSuit.setText("VS");
151         hasilSuit.setBackgroundColor(0);
152         hasilSuit.setTextSize(40F);
153     }
154
155     private void selectSuit(){
156         final Handler handler = new Handler(Looper.getMainLooper());
157         if(pilihSuitLawan==0) {
158             handler.postDelayed(new Runnable() {
159                 @Override
160                 public void run() { hapusPemain(); }
161             }, delayMillis: 500);
162         }
163     }
164

```

Method reset() pada line 148 – 153 berfungsi untuk mereset text hasil permainan yang berada ditengah-tengah gambar suit. Kemudian *method* selectSuit() pada line 156-164 berfungsi apabila ketika pilihan lawan belum ada atau sesudah pemain 1 mengklik pilihan mereka, background dari gambar suit yang dipilih akan langsung dihapus dalam interval 500 ms. Hal ini berfungsi agar pemain 2 tidak mengetahui pilihan dari pemain 1.

```

166     // hapusLawan();
167     if(pilihSuitLawan>0){
168         switch(pilihSuit){
169             case 1 : batuPemain.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
170             case 2 : guntingPemain.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
171             case 3 : kertasPemain.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
172         }
173     }
174     suit();
175     //hapusPemain();
176

```

Kemudian untuk line 167 – 173 berfungsi untuk mengubah background gambar suit yang dipilih *user*. Kemudian setelah lawan memilih pilihan suit, maka akan dipanggil *method* suit() untuk menentukan pemenang game.

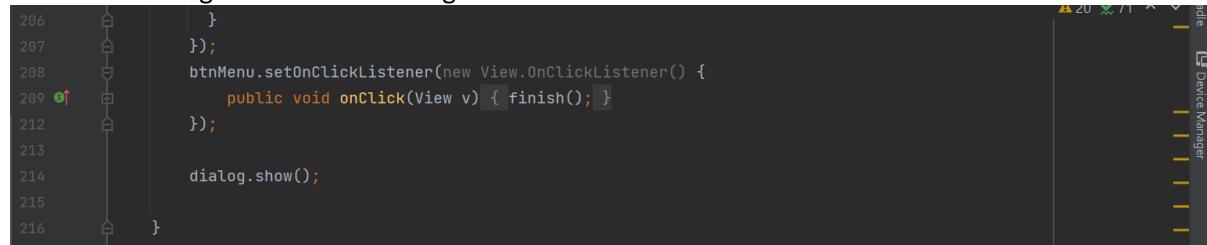
```

182     private void showDialog(){
183
184         AlertDialog.Builder dialogBuilder = new AlertDialog.Builder( context: this);
185         View viewCustome = LayoutInflater.from(this).inflate(R.layout.dialog_winner, root: null, attachToRoot: false);
186
187         dialogBuilder.setView(viewCustome);
188
189
190         final AlertDialog dialog = dialogBuilder.create();
191
192         Button btnRefresh = viewCustome.findViewById(R.id.btn_refresh);
193         Button btnMenu = viewCustome.findViewById(R.id.btn_menu);
194         TextView txtResult = viewCustome.findViewById(R.id.txt_result);
195         //TextView namaUser = intent.getStringExtra("DATA_USER_NAME")
196
197         if(hasil==1) txtResult.setText("SERI");
198         if(hasil==2)txtResult.setText("Pemain 1 MENANG!");
199         if(hasil==3) txtResult.setText("Pemain 2 MENANG!");
200
201
202         btnRefresh.setOnClickListener(new View.OnClickListener() {
203             public void onClick(View v) {
204                 closeDialog();
205                 dialog.dismiss();
206             }
207         });

```

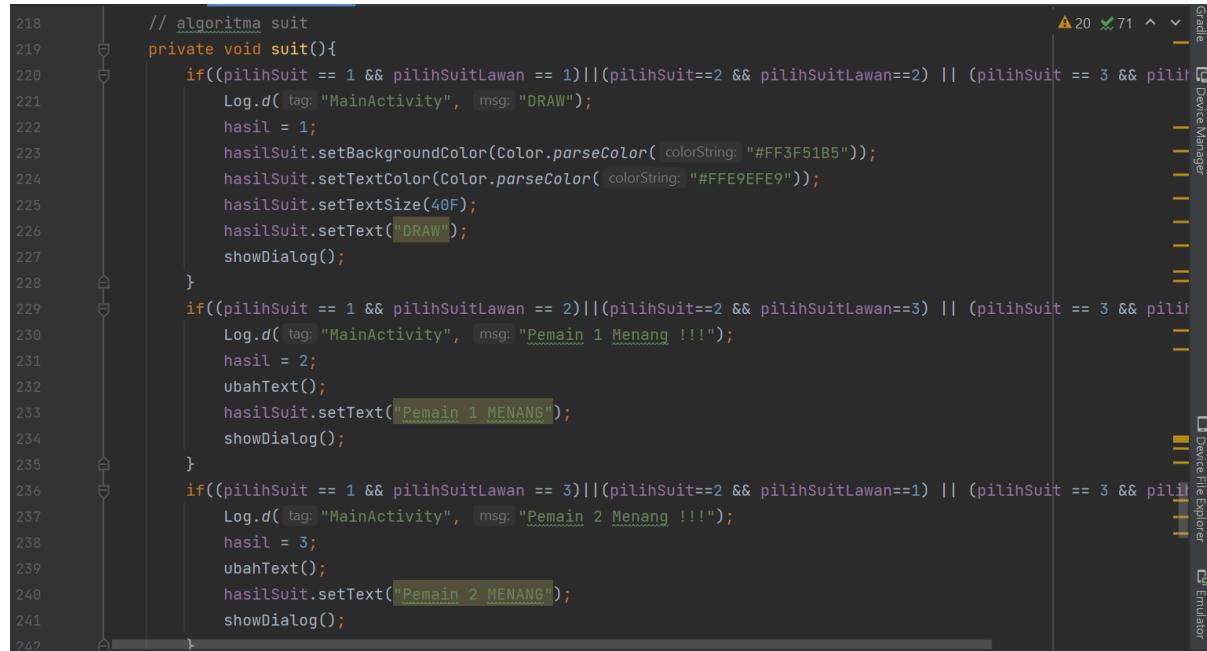
AlertDialog berfungsi untuk membuat dialog berdasarkan layout yang telah dibuat sebelumnya yaitu dialog_winner. Pada line 197 – 199 berfungsi untuk mengganti text hasil permainan berdasarkan value dari hasil yang dilakukan pada *method* suit() nantinya. Kemudian pada line 202-207 berfungsi untuk

mereset permainan dengan *method* `closeDialog()` yang telah dibuat sebelumnya dan menutup dialog ketika *user* mengklik button Main Lagi.



```
206     }
207     });
208     btnMenu.setOnClickListener(new View.OnClickListener() {
209         @Override
210         public void onClick(View v) { finish(); }
211     });
212
213     dialog.show();
214 }
215 }
216 }
```

Selanjutnya ketika *user* mengklik button Menu, maka *activity* akan berakhir dan otomatis tertutup hingga kembali ke menu PilihanSuit.



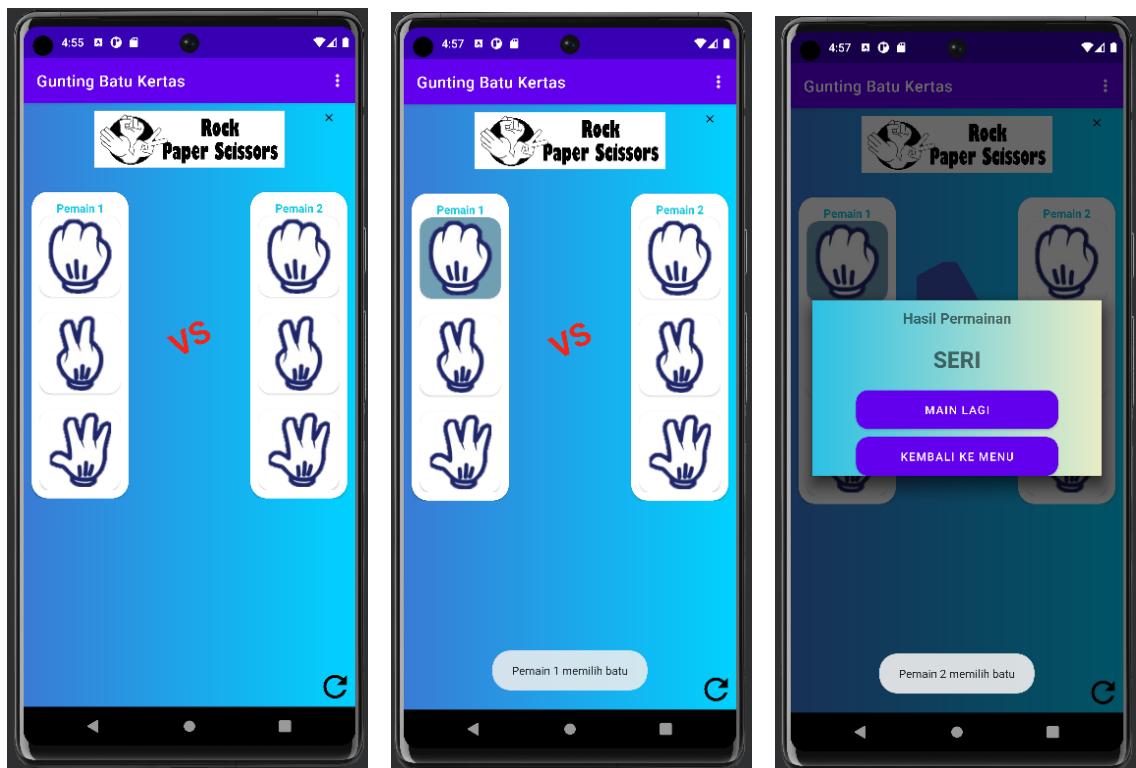
```
218     // algoritma suit
219     private void suit(){
220         if((pilihSuit == 1 & pilihSuitLawan == 1)|| (pilihSuit==2 & pilihSuitLawan==2) || (pilihSuit == 3 & pilihSuitLawan==3)) {
221             Log.d( tag: "MainActivity", msg: "DRAW");
222             hasil = 1;
223             hasilSuit.setBackgroundColor(Color.parseColor( colorString: "#FF3F51B5"));
224             hasilSuit.setTextColor(Color.parseColor( colorString: "#FFE9E9E9"));
225             hasilSuit.setTextSize(40F);
226             hasilSuit.setText("DRAW");
227             showDialog();
228         }
229         if((pilihSuit == 1 & pilihSuitLawan == 2)|| (pilihSuit==2 & pilihSuitLawan==3) || (pilihSuit == 3 & pilihSuitLawan==1)) {
230             Log.d( tag: "MainActivity", msg: "Pemain 1 Menang !!!");
231             hasil = 2;
232             ubahText();
233             hasilSuit.setText("Pemain 1 MENANG");
234             showDialog();
235         }
236         if((pilihSuit == 1 & pilihSuitLawan == 3)|| (pilihSuit==2 & pilihSuitLawan==1) || (pilihSuit == 3 & pilihSuitLawan==2)) {
237             Log.d( tag: "MainActivity", msg: "Pemain 2 Menang !!!");
238             hasil = 3;
239             ubahText();
240             hasilSuit.setText("Pemain 2 MENANG");
241             showDialog();
242         }
243     }
244 }
```

Code di atas merupakan algoritma dari suit yang dipilih *user* untuk menentukan pemenang game. Karena nilai 1 merupakan batu, 2 gunting, dan 3 kertas, sehingga apabila *user* dan lawan memilih pilihan suit yang sama maka akan memiliki nilai yang sama juga. Misalnya *user* memilih batu dan lawan juga memilih batu maka hasil permainan akan draw. Hasil draw akan menghasilkan nilai hasil = 1, sedangkan apabila pemain 1 memilih batu (1) dan lawan memilih gunting (2) maka pemain 1 memenangkan permainan dan menghasilkan nilai hasil = 2. Apabila lawan yang menang maka akan menghasilkan nilai hasil = 3. Ketika pemenang ditentukan akan langsung menampilkan dialog dan mengubah text dari `hasilSuit` serta mengubah backgroundnya dengan memanggil *method* `ubahText()`.

```
245     private void ubahText(){
246         hasilSuit.setBackgroundColor(Color.parseColor( colorString: "#FF4CAF50"));
247         hasilSuit.setTextColor(Color.parseColor( colorString: "#FFE9E9E9"));
248         hasilSuit.setTextSize(15F);
249     }
250
251     private void hapusPemain(){
252         batuPemain.setBackgroundColor(0);
253         guntingPemain.setBackgroundColor(0);
254         kertasPemain.setBackgroundColor(0);
255     }
256
257     private void hapusLawan(){
258         batuLawan.setBackgroundColor(0);
259         guntingLawan.setBackgroundColor(0);
260         kertasLawan.setBackgroundColor(0);
261     }
```

Line 245 – 249 berfungsi untuk mengubah *background* dan *text size* dari *TextView* hasilSuit. Kemudian pada line 251 – 255 berfungsi untuk menghapus *background* pilihan pemain 1, sedangkan *method* hapusLawan() berfungsi untuk menghapus *background* pilihan pemain 2.

Emulator



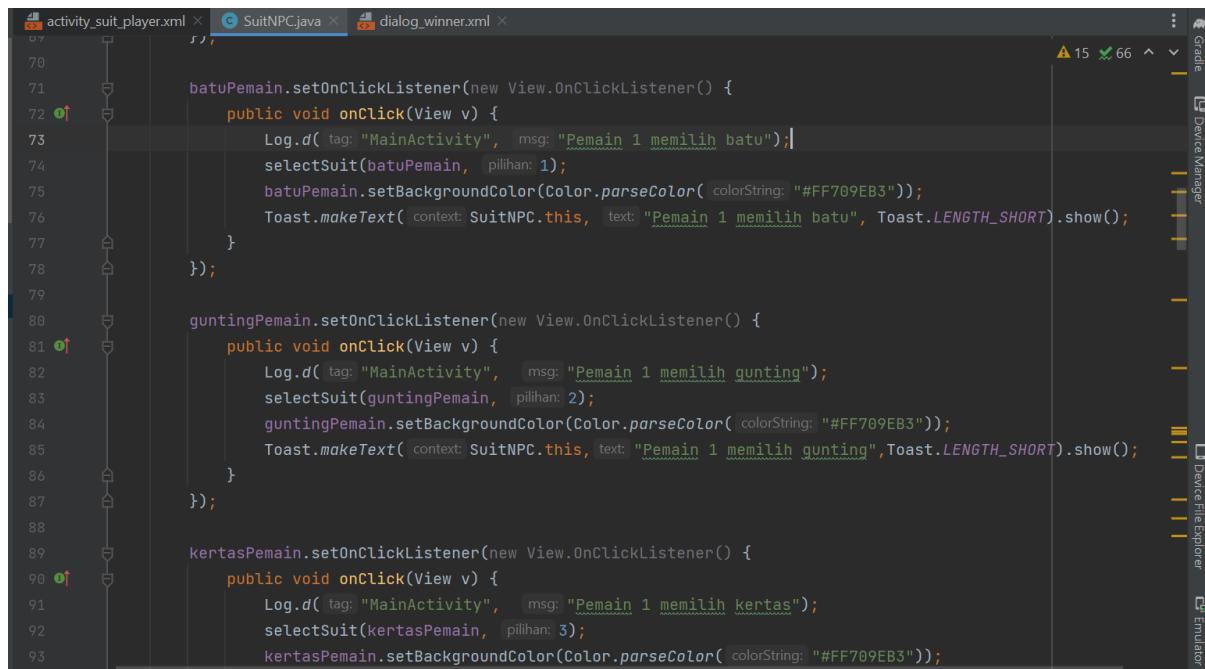
Ketika emulator dijalankan, akan menghasilkan tampilan seperti gambar di kiri, kemudian apabila pemain 1 memilih pilihan suit akan menghasilkan tampilan seperti di tengah. Ketika pemain 2 selesai memilih pilihan suit, maka tampilan dialog akan langsung keluar untuk mengumumkan hasil permainan.

SuitNPC.java

```
public class SuitNPC extends AppCompatActivity {

    MediaPlayer game;
    ImageView batuPemain;
    ImageView guntingPemain;
    ImageView kertasPemain;
    ImageView batuCom;
    ImageView guntingCom;
    ImageView kertasCom;
    TextView hasilSuit;
    ImageView ulang;
    ImageView imgClose;
    TextView namaPemain;
    int pilihSuit = 0;
    int hasil = 1;
    int pilihCom = new Random().nextInt( bound: 3 - 1 + 1) + 1;
```

Untuk code game dengan lawan NPC hamper memiliki algoritma yang sama dengan lawan pemain 2. Pilihan lawan NPC akan dipilih berdasarkan angka random dengan rentang 1-3 yang disimpan dalam variabel pilihCom.



```
activity_suit_player.xml x SuitNPCjava x dialog_winner.xml x
70
71     batuPemain.setOnClickListener(new View.OnClickListener() {
72         public void onClick(View v) {
73             Log.d( tag: "MainActivity", msg: "Pemain 1 memilih batu");
74             selectSuit(batuPemain, pilihan: 1);
75             batuPemain.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
76             Toast.makeText( context: SuitNPC.this, text: "Pemain 1 memilih batu", Toast.LENGTH_SHORT).show();
77         }
78     });
79
80     guntingPemain.setOnClickListener(new View.OnClickListener() {
81         public void onClick(View v) {
82             Log.d( tag: "MainActivity", msg: "Pemain 1 memilih gunting");
83             selectSuit(guntingPemain, pilihan: 2);
84             guntingPemain.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
85             Toast.makeText( context: SuitNPC.this, text: "Pemain 1 memilih gunting", Toast.LENGTH_SHORT).show();
86         }
87     });
88
89     kertasPemain.setOnClickListener(new View.OnClickListener() {
90         public void onClick(View v) {
91             Log.d( tag: "MainActivity", msg: "Pemain 1 memilih kertas");
92             selectSuit(kertasPemain, pilihan: 3);
93             kertasPemain.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));

Gradle Device Manager Device File Explorer Emulator
```

Ketika *user* mengklik pilihan suit mereka, hasil permainan akan langsung dikeluarkan karena secara otomatis com akan langsung memilih secara random setelah *user* mengklik pilihan suit. Hasil permainan tersebut ditampilkan dengan *method* selectSuit().

```
// menampilkan pilihan hasil acak com
private void com(){
    if(pilihSuit>0) {
        if (pilihCom == 1){
            Log.d( tag: "MainActivity", msg: "Com memilih batu");
            batuCom.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
            Toast.makeText( context: this, text: "CPU memilih batu", Toast.LENGTH_SHORT).show();
        }
        if (pilihCom == 2){
            Log.d( tag: "MainActivity", msg: "Com memilih gunting");
            guntingCom.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
            Toast.makeText( context: this, text: "CPU memilih gunting", Toast.LENGTH_SHORT).show();
        }
        if (pilihCom == 3){
            Log.d( tag: "MainActivity", msg: "Com memilih kertas");
            kertasCom.setBackgroundColor(Color.parseColor( colorString: "#FF709EB3"));
            Toast.makeText( context: this, text: "CPU memilih kertas", Toast.LENGTH_SHORT).show();
        }
    }
}
```

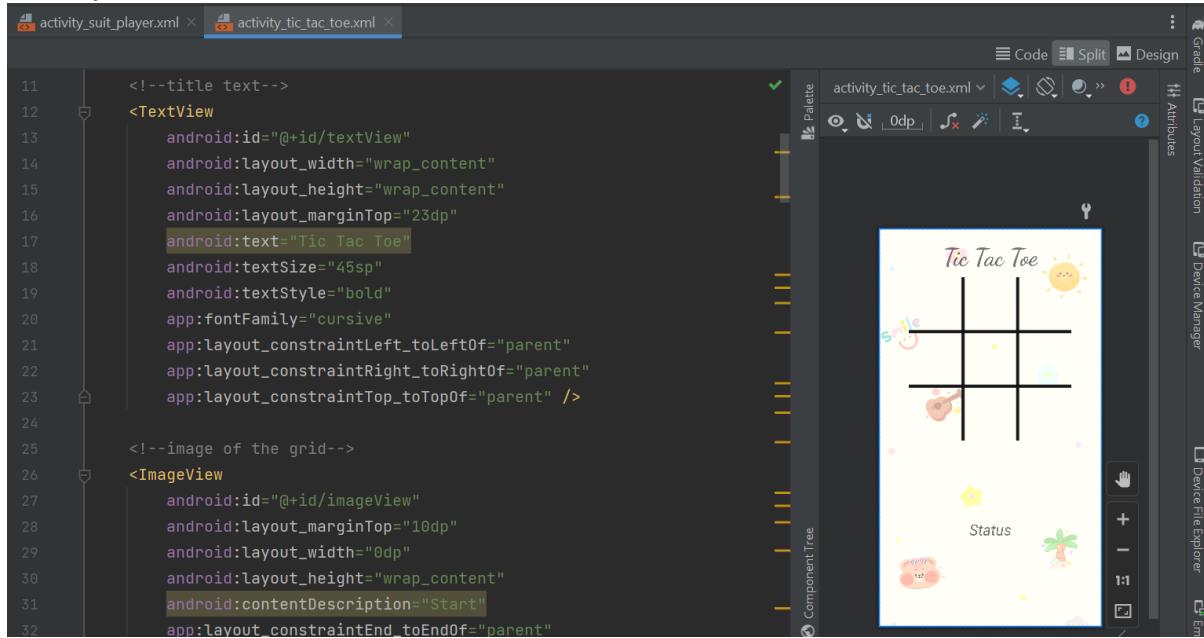
Method di atas untuk memberi tanda pilihan apa yang dipilih oleh komputer berdasarkan angka acak sebelumnya dengan mengubah background gambar dan memberi informasi lewat *toast*.

```
private void hapusCom(){
    batuCom.setBackgroundColor(0);
    guntingCom.setBackgroundColor(0);
    kertasCom.setBackgroundColor(0);
    pilihCom = new Random().nextInt( bound: 3 - 1 + 1) + 1;
}
```

Method di atas untuk mereset pilihan komputer dengan menghapus background dan memberi nomor angka acak yang baru. Ketika emulator ditampilkan, akan menghasilkan tampilan yang sama seperti melawan pemain 2.

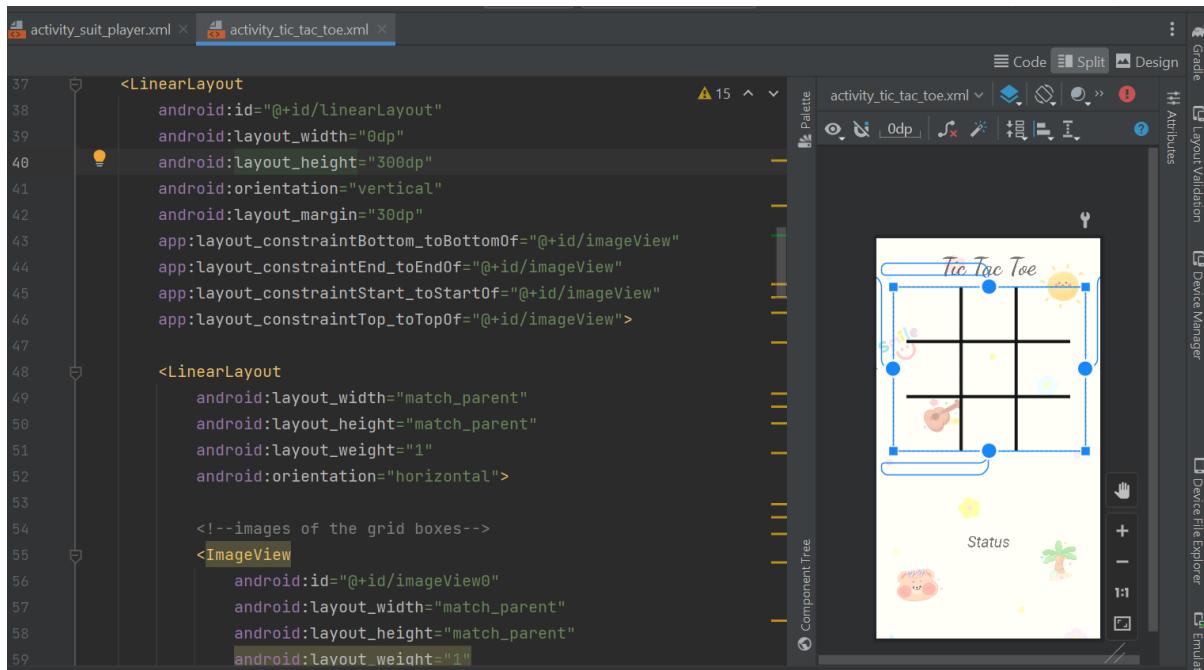
Tic Tac Toe Game

activity_tic_tac_toe.xml



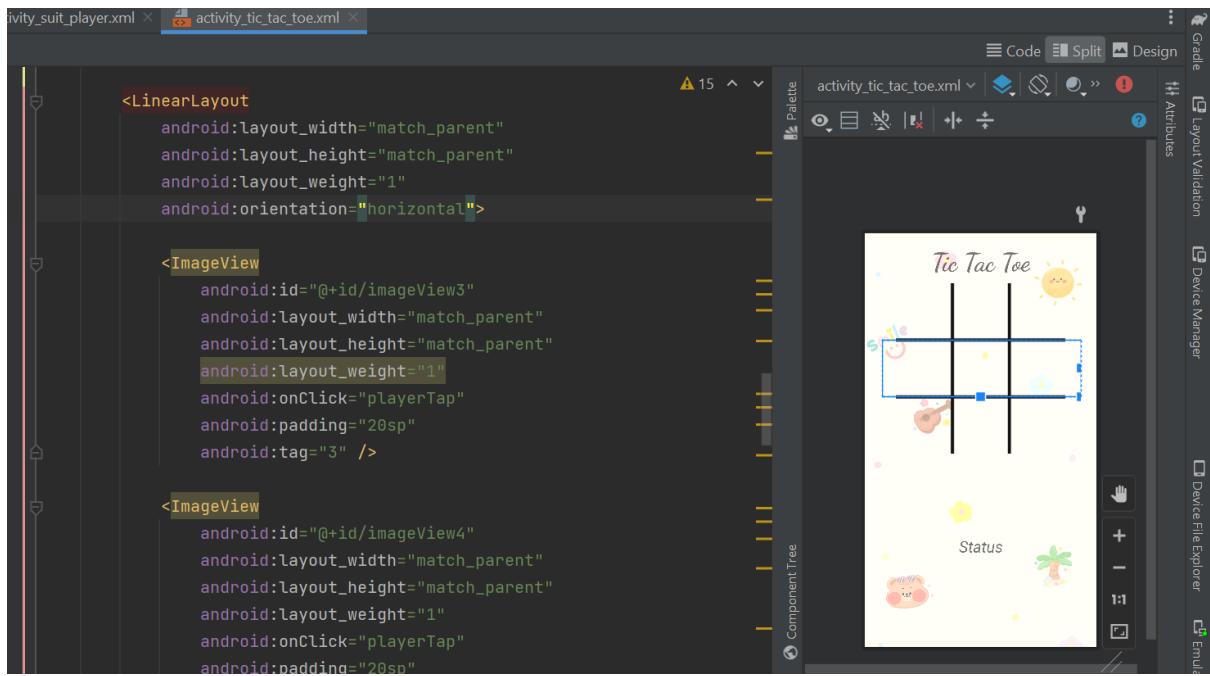
```
11     <!--title text-->
12     <TextView
13         android:id="@+id/textView"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"
16         android:layout_marginTop="23dp"
17         android:text="Tic Tac Toe"
18         android:textSize="45sp"
19         android:textStyle="bold"
20         app:fontFamily="cursive"
21         app:layout_constraintLeft_toLeftOf="parent"
22         app:layout_constraintRight_toRightOf="parent"
23         app:layout_constraintTop_toTopOf="parent" />
24
25     <!--image of the grid-->
26     <ImageView
27         android:id="@+id/imageView"
28         android:layout_marginTop="10dp"
29         android:layout_width="0dp"
30         android:layout_height="wrap_content"
31         android:contentDescription="Start"
32         app:layout_constraintEnd_toEndOf="parent"
```

Ketika pada MenuActivity *user* memilih bermain TicTacToe, maka *user* akan berpindah *activity* ke TicTacToe. Pada file .xml letakkan ImageView untuk garis-garis kotak untuk menempati gambar o dan x *user*.

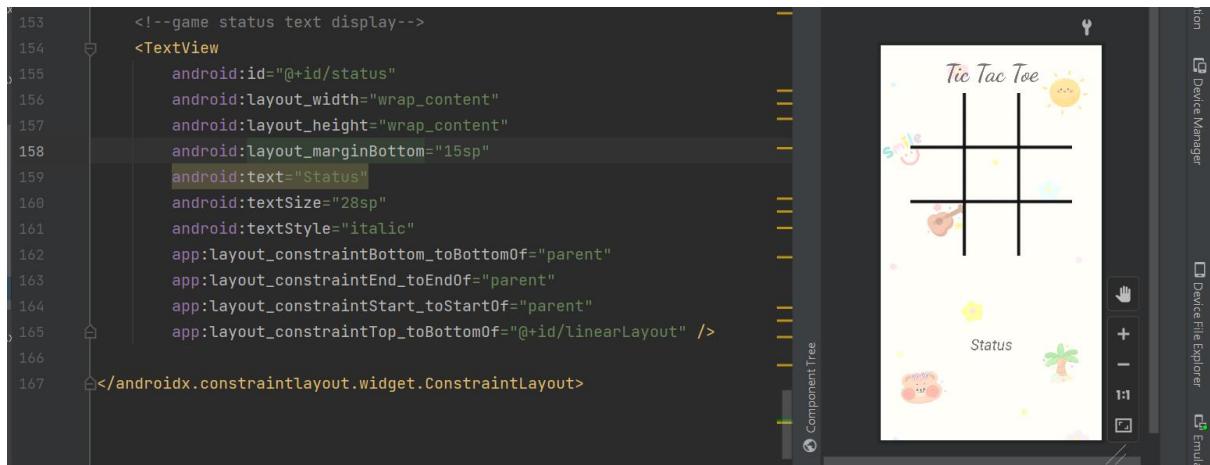


```
37     <LinearLayout
38         android:id="@+id/linearLayout"
39         android:layout_width="0dp"
40         android:layout_height="300dp"
41         android:orientation="vertical"
42         android:layout_margin="30dp"
43         app:layout_constraintBottom_toBottomOf="@+id/imageView"
44         app:layout_constraintEnd_toEndOf="@+id/imageView"
45         app:layout_constraintStart_toStartOf="@+id/imageView"
46         app:layout_constraintTop_toTopOf="@+id/imageView">
47
48     <LinearLayout
49         android:layout_width="match_parent"
50         android:layout_height="match_parent"
51         android:layout_weight="1"
52         android:orientation="horizontal">
53
54     <!--images of the grid boxes-->
55     <ImageView
56         android:id="@+id/imageView0"
57         android:layout_width="match_parent"
58         android:layout_height="match_parent"
59         android:layout_weight="1" />
```

Kemudian letakkan LinearLayout untuk membatasi garis agar o dan x *user* tidak keluar berlebihan dari gambar kotak.



Kemudian dalam LinearLayout dibuat lagi 3 LinearLayout dengan orientation landscape untuk mengisi setiap 3 baris kotak sebagai grid dari imageView o dan x dapat menempati kotak dengan bagus.



Kemudian letakkan 1 TextView untuk memberikan informasi siapa pemenang dari game Tictactoe.

TicTacToe.java

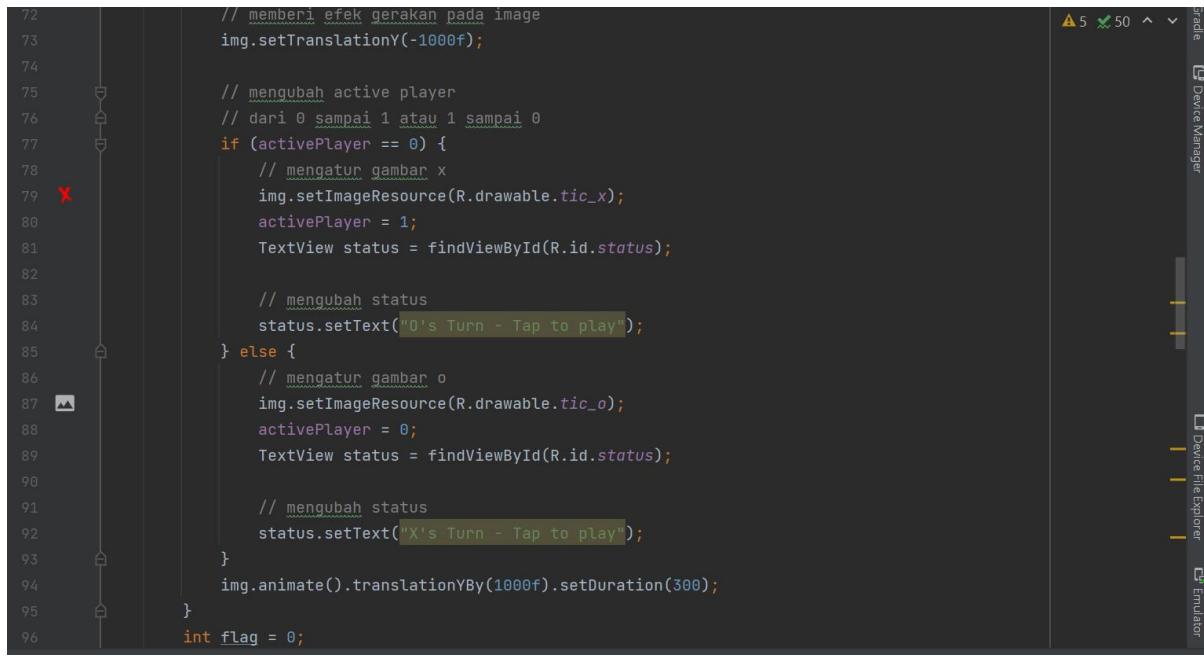
```
19 public class TicTacToe extends AppCompatActivity {
20     boolean gameActive = true;
21     MediaPlayer game;
22
23     // Player
24     // 0 - X
25     // 1 - O
26     int activePlayer = 0;
27     int[] gameState = {2, 2, 2, 2, 2, 2, 2, 2, 2, 2};
28
29     // arti state:
30     // 0 - X
31     // 1 - O
32     // 2 - Null
33     // array untuk posisi menang
34     int[][] winPositions = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8},
35                             {0, 3, 6}, {1, 4, 7}, {2, 5, 8},
36                             {0, 4, 8}, {2, 4, 6}};
37     public static int counter = 0;
38
39     @Override
40     protected void onCreate(Bundle savedInstanceState) {
41         super.onCreate(savedInstanceState);
42         setContentView(R.layout.activity_tic_tac_toe);
43         game = MediaPlayer.create(context: this, R.raw.aame);
44     }
45 }
```

Pertama buatlah sebuah variabel line 20 untuk mengecek apakah permainan masih dapat berjalan atau tidak. Kemudian buat variabel line 26 untuk memberi giliran pada pemain secara bergantian. Selanjutnya buatlah array untuk memberi nilai pada setiap state dalam kotak dengan 2 yang berarti null. Kemudian buat array baru (line 34) untuk menentukan posisi menang dalam box.

```
47     public void playerTap(View view) {
48         ImageView img = (ImageView) view;
49         int tappedImage = Integer.parseInt(img.getTag().toString());
50
51         // memanggil fungsi game resetgame
52         // ketika salah satu player menang atau semua kotak telah penuh
53         if (!gameActive) {
54             gameReset(view);
55         }
56
57         // if untuk klik image kosong
58         if (gameState[tappedImage] == 2) {
59             // increase the counter
60             // after every tap
61             counter++;
62
63             // mengecek kotak terakhir
64             if (counter == 9) {
65                 // reset game
66                 gameActive = false;
67             }
68
69             // menandai posisi ini
70             gameState[tappedImage] = activePlayer;
71         }
72     }
73 }
```

Kemudian buatlah sebuah *method* baru untuk mengecek permainan. Line 53 – 55 berfungsi untuk mengecek apakah permainan bisa dilanjutkan atau tidak dengan melihat isi kotak telah penuh atau ada player yang memenangkan permainan. Kemudian if pada line 58 untuk mengecek apakah kotak image yang diklik *user* kosong atau tidak. Jika kosong, maka counter akan bertambah sehingga menandakan bahwa box tersebut telah diisi. Kemudian line 64-67

berfungsi untuk mengecek apakah seluruh kotak box telah penuh atau tidak, jika penuh maka value gameActive akan menjadil false.

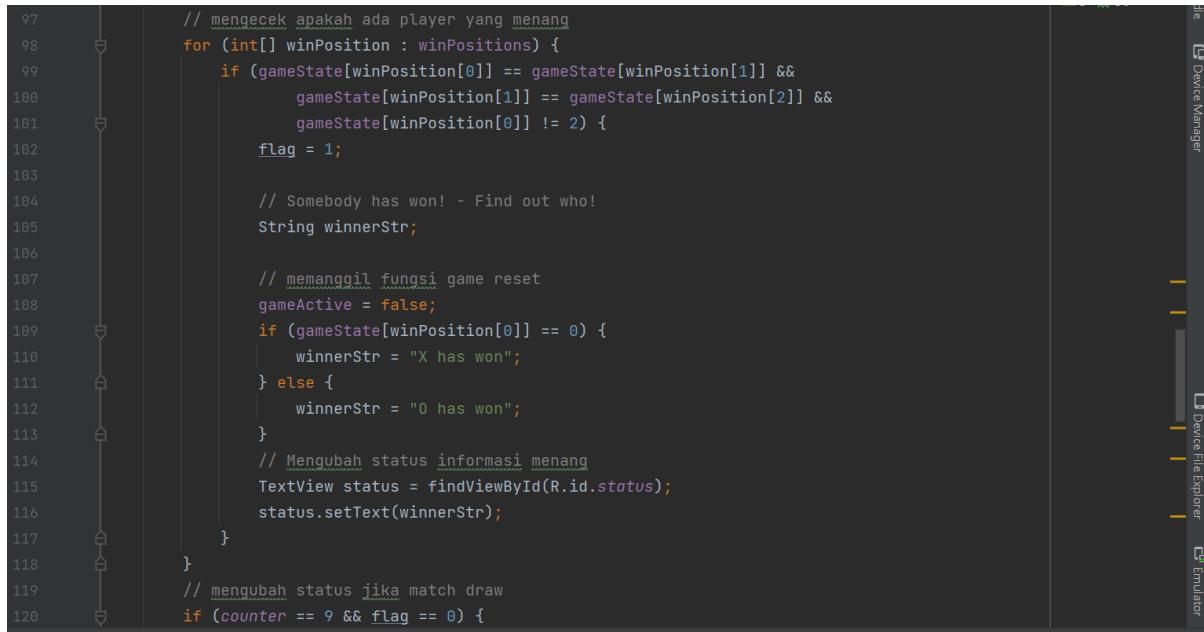


```

72     // memberi efek gerakan pada image
73     img.setTranslationY(-1000f);
74
75     // mengubah active player
76     // dari 0 sampai 1 atau 1 sampai 0
77     if (activePlayer == 0) {
78         // mengatur gambar x
79         img.setImageResource(R.drawable.tic_x);
80         activePlayer = 1;
81         TextView status = findViewById(R.id.status);
82
83         // mengubah status
84         status.setText("O's Turn - Tap to play");
85     } else {
86         // mengatur gambar o
87         img.setImageResource(R.drawable.tic_o);
88         activePlayer = 0;
89         TextView status = findViewById(R.id.status);
90
91         // mengubah status
92         status.setText("X's Turn - Tap to play");
93     }
94     img.animate().translationYBy(1000f).setDuration(300);
95
96     int flag = 0;

```

Pada line 77 – 85 berfungsi untuk menandakan giliran pemain x dan akan mengubah gambar yang ditap menjadi gambar x serta activatePlayer juga akan berubah menjadi 1 untuk mengubah giliran menjadi pemain o. Untuk line 85 – 93 berfungsi yang sama hanya saja mengganti activePlayer menjadi 0 yang berarti berganti pemain menjadi gilrian pemain x.



```

97     // mengecek apakah ada player yang menang
98     for (int[] winPosition : winPositions) {
99         if (gameState[winPosition[0]] == gameState[winPosition[1]] &&
100            gameState[winPosition[1]] == gameState[winPosition[2]] &&
101            gameState[winPosition[0]] != 2) {
102             flag = 1;
103
104             // Somebody has won! - Find out who!
105             String winnerStr;
106
107             // memanggil fungsi game reset
108             gameActive = false;
109             if (gameState[winPosition[0]] == 0) {
110                 winnerStr = "X has won";
111             } else {
112                 winnerStr = "O has won";
113             }
114             // Mengubah status informasi menang
115             TextView status = findViewById(R.id.status);
116             status.setText(winnerStr);
117         }
118     }
119     // mengubah status jika match draw
120     if (counter == 9 && flag == 0) {

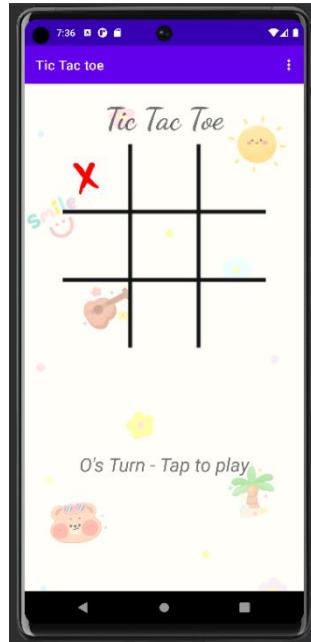
```

Kemudian perulangan pada line 98 – 118 berfungsi untuk mengecek apakah ada salah satu player yang memenangkan permainan berdasarkan array posisi menang sebelumnya. Jika ada yang menang, maka variabel flag akan berubah menjadi 1 pertanda ada yang menang. Kemudian gameActive akan menjadi false karena game sudah selesai. Jika posisi menangnya adalah 0, maka pemain X lah yang memenangkan permainan, begitupun sebaliknya untuk posisi menang 1.

```
120     if (counter == 9 && flag == 0) {
121         TextView status = findViewById(R.id.status);
122         status.setText("Match Draw");
123     }
124 }
125
126 // reset the game
127 public void gameReset(View view) {
128     gameActive = true;
129     activePlayer = 0;
130     for (int i = 0; i < gameState.length; i++) {
131         gameState[i] = 2;
132     }
133     // menghapus semua gambar dari kotak dalam grid
134     ((ImageView) findViewById(R.id.imageView0)).setImageResource(0);
135     ((ImageView) findViewById(R.id.imageView1)).setImageResource(0);
136     ((ImageView) findViewById(R.id.imageView2)).setImageResource(0);
137     ((ImageView) findViewById(R.id.imageView3)).setImageResource(0);
138     ((ImageView) findViewById(R.id.imageView4)).setImageResource(0);
139     ((ImageView) findViewById(R.id.imageView5)).setImageResource(0);
140     ((ImageView) findViewById(R.id.imageView6)).setImageResource(0);
141     ((ImageView) findViewById(R.id.imageView7)).setImageResource(0);
142     ((ImageView) findViewById(R.id.imageView8)).setImageResource(0);
143
144     TextView status = findViewById(R.id.status);
145     status.setText("X's Turn - Tap to play");
146 }
```

Kemudian line 120 – 123 berfungsi untuk mengecek apakah permainan draw atau tidak dengan melihat value dari counter dan flag. Kemudian untuk *method* 127 – 146, berfungsi untuk mereset permainan dengan menghapus seluruh image pada kotak yang ada.

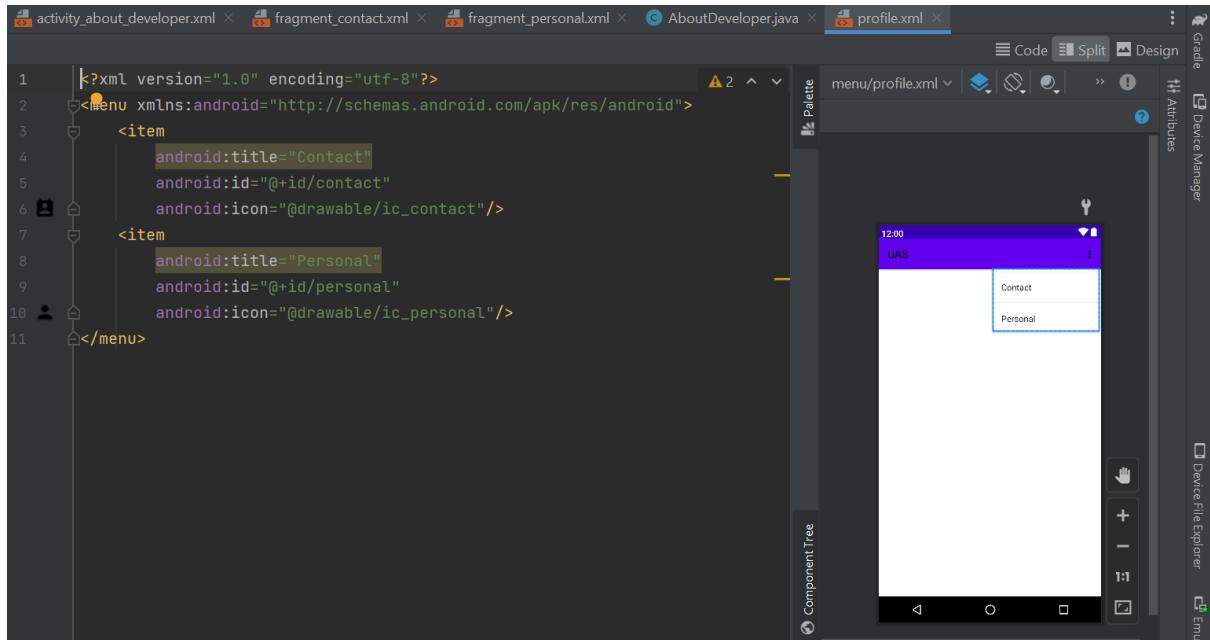
Emulator



Ketika emulator dirun, maka tampilan game tactictoe akan seperti di atas dengan gambar x dan o yang dapat diklik disetiap grid kotak.

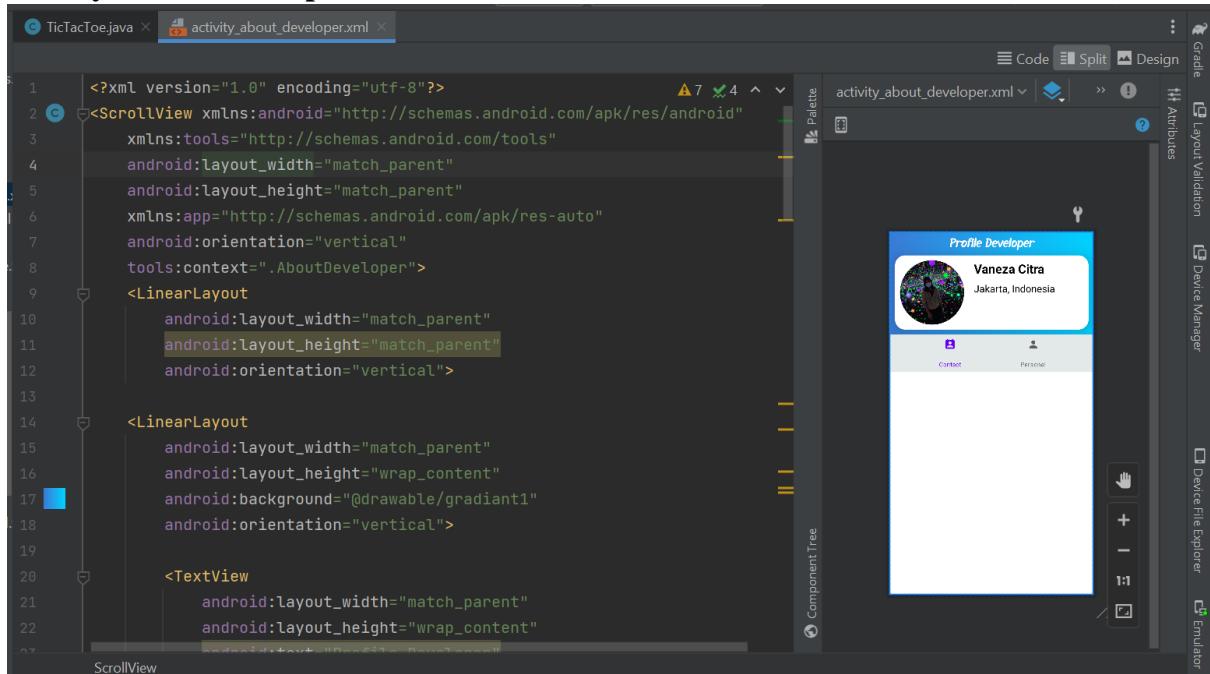
About Developer

Menu

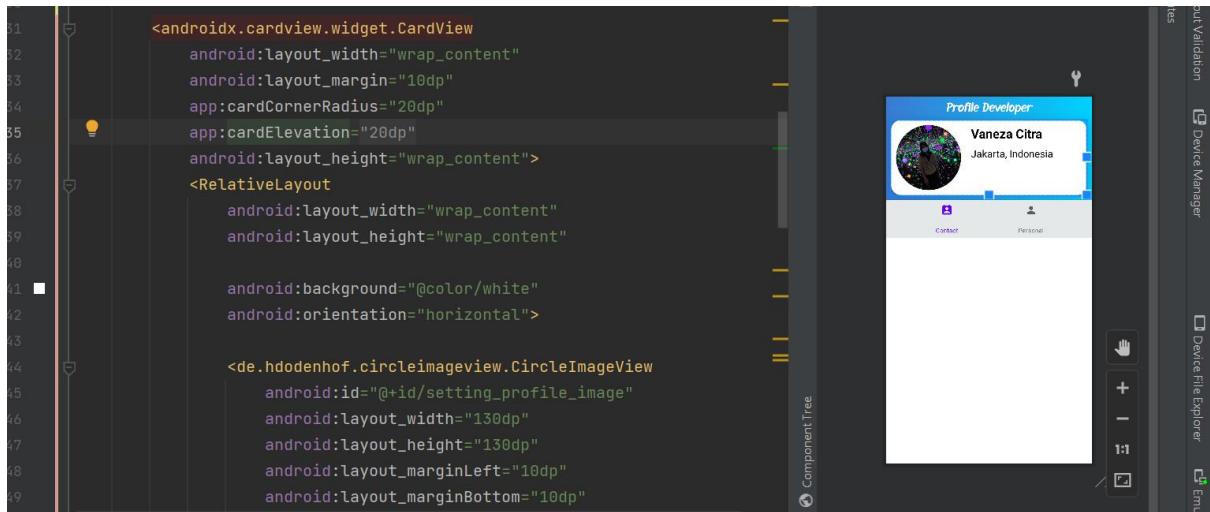


Buatlah sebuah menu baru pada res > menu dan beri nama profile. Untuk setiap item menu tambahkan icon gambar untuk digunakan pada BottomNavbar.

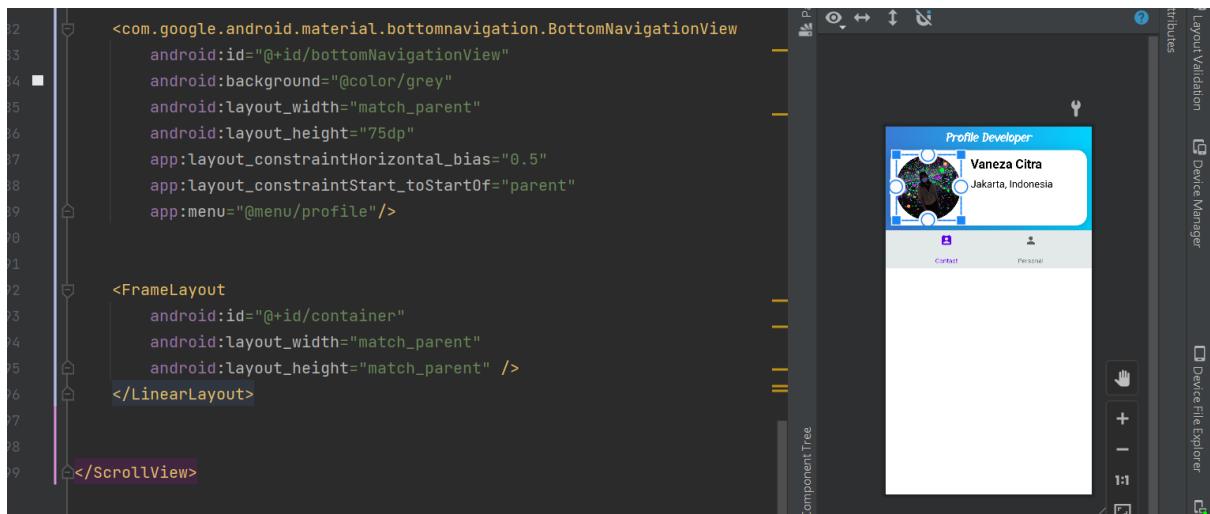
activity_about_developer



Kemudian untuk profile developer dibuat dalam ScrollView sehingga *user* dapat melihatnya meskipun dalam ukuran layar yang kecil. Kemudian diletakkan LinearLayout untuk menampung seluruh tampilan dan 1 lagi LinearLayout untuk menampung CardView untuk tampilan profil.

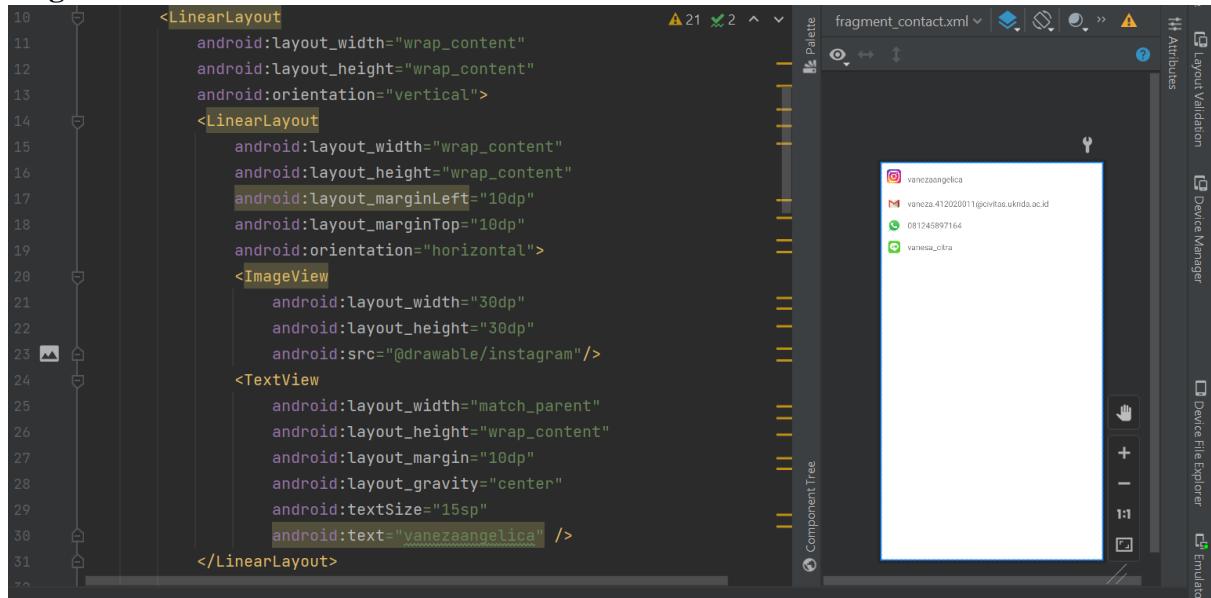


Dalam CardView terdapat CircleImageView agar gambar dapat berbentuk lingkaran kemudian terdapat 2 TextView untuk meletakkan data *developer* seperti nama dan tempat tinggal.



Kemudian diletakkan BottomNavigationView agar *user* dapat melihat profil *user* tanpa perlu berganti *activity* dan hanya berganti fragment saja. Fragment akan diletakkan ke dalam FrameLayout yang telah disediakan.

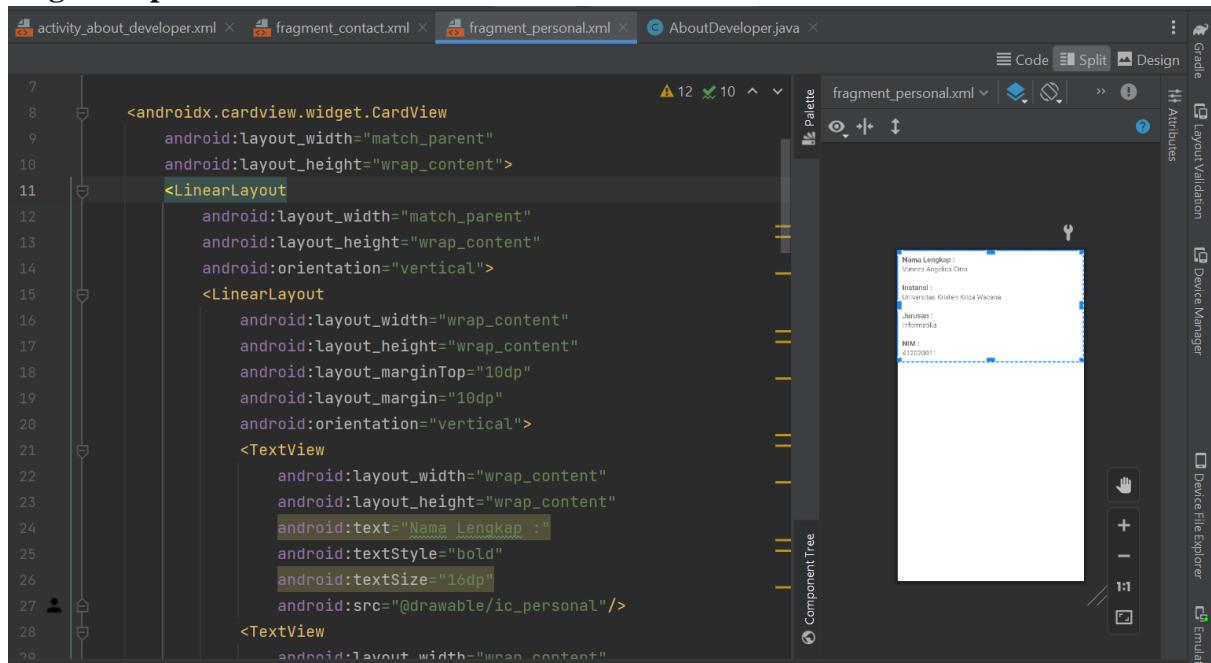
fragment_contact.xml



```
10 <LinearLayout
11     android:layout_width="wrap_content"
12     android:layout_height="wrap_content"
13     android:orientation="vertical">
14         <LinearLayout
15             android:layout_width="wrap_content"
16             android:layout_height="wrap_content"
17             android:layout_marginLeft="10dp"
18             android:layout_marginTop="10dp"
19             android:orientation="horizontal">
20                 <ImageView
21                     android:layout_width="30dp"
22                     android:layout_height="30dp"
23                     android:src="@drawable/instagram"/>
24                 <TextView
25                     android:layout_width="match_parent"
26                     android:layout_height="wrap_content"
27                     android:layout_margin="10dp"
28                     android:layout_gravity="center"
29                     android:textSize="15sp"
30                     android:text="vanezaangelica" />
31             </LinearLayout>
32         </LinearLayout>
33     </LinearLayout>
```

Dibuat sebuah fragment baru dan diberi nama ContactFragment. Dalam file ini diletakkan LinearLayout orientation vertical untuk menampung setiap informasi contact developer. Dibuat lagi LinearLayout dengan orientation horizontal untuk menampung setiap data *contact developer* berupa text *contact* dan gambar *contact*.

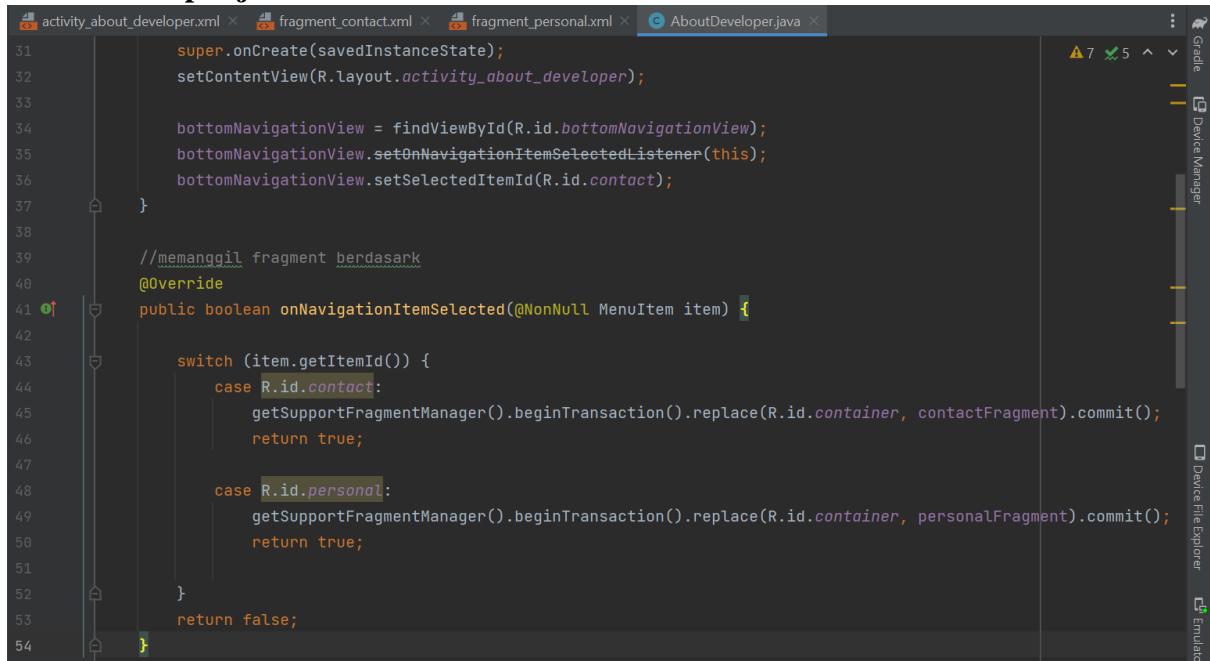
fragment_personal.xml



```
7 <androidx.cardview.widget.CardView
8     android:layout_width="match_parent"
9     android:layout_height="wrap_content">
10    <LinearLayout
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:orientation="vertical">
14            <LinearLayout
15                android:layout_width="wrap_content"
16                android:layout_height="wrap_content"
17                android:layout_marginTop="10dp"
18                android:layout_margin="10dp"
19                android:orientation="vertical">
20                    <TextView
21                        android:layout_width="wrap_content"
22                        android:layout_height="wrap_content"
23                        android:text="Nama Lengkap :"
24                        android:textStyle="bold"
25                        android:textSize="16dp"/>
26                    <TextView
27                        android:layout_width="wrap_content"
28                        android:layout_height="wrap_content"
29                        android:src="@drawable/ic_personal"/>
30                </LinearLayout>
31            </LinearLayout>
32        </LinearLayout>
33    </CardView>
```

Dibuat fragment baru lagi dengan nama fragment_personal. Pada file ini data personal developer dimasukkan ke dalam CardView. Untuk datanya ditulis secara manual tidak dari database.

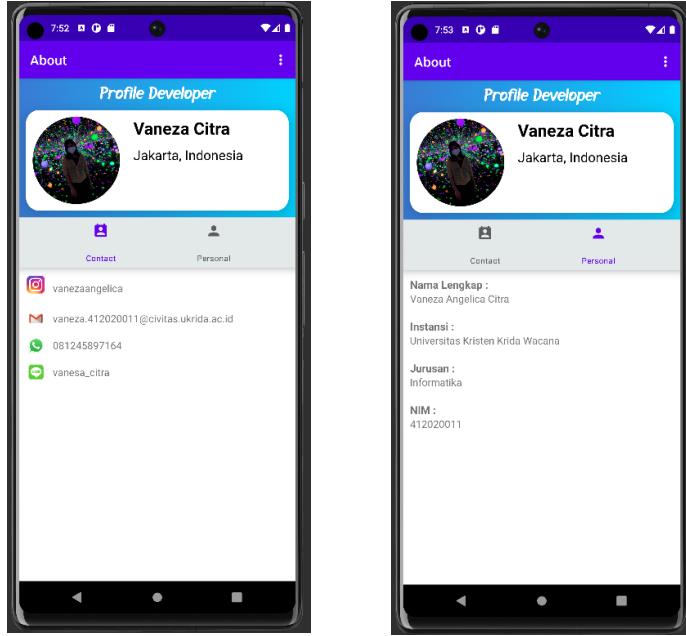
AboutDeveloper.java



```
activity_about_developer.xml fragment_contact.xml fragment_personal.xml AboutDeveloper.java
31     super.onCreate(savedInstanceState);
32     setContentView(R.layout.activity_about_developer);
33
34     bottomNavigationView = findViewById(R.id.bottomNavigationView);
35     bottomNavigationView.setOnNavigationItemSelected(this);
36     bottomNavigationView.setSelectedItemId(R.id.contact);
37 }
38
39 //memanggil fragment berdasarkan
40 @Override
41 public boolean onNavigationItemSelected(@NonNull MenuItem item) {
42
43     switch (item.getItemId()) {
44         case R.id.contact:
45             getSupportFragmentManager().beginTransaction().replace(R.id.container, contactFragment).commit();
46             return true;
47
48         case R.id.personal:
49             getSupportFragmentManager().beginTransaction().replace(R.id.container, personalFragment).commit();
50             return true;
51     }
52     return false;
53 }
54 }
```

Selanjutnya pada file AboutDeveloper.java, panggil id dari bottomNavigationView agar bisa diakses ke kelas java (line 34). Line 35 berfungsi untuk memberi listener saat menu BottomNavigation diklik. Kemudian line 41 – 54 berfungsi untuk mengubah fragment sesuai apa yang diklik *user* pada BottomNavbar.

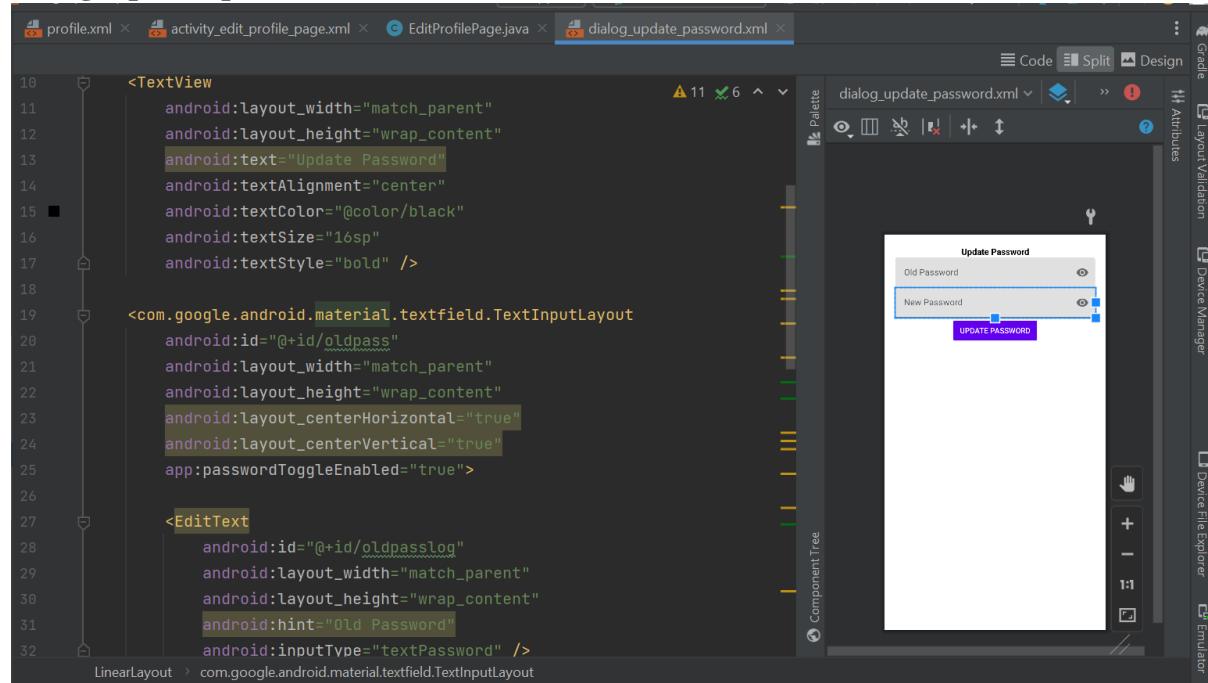
Emulator



Ketika emulator dijalankan akan menghasil tampilan seperti di atas dengan fragment yang berbeda sesuai posisi yang diklik *user* pada menu Bottom Navigation.

Edit Profile User

dialog_update_password.xml



The screenshot shows the Android Studio interface with the 'dialog_update_password.xml' file open in the code editor. The XML code defines a dialog layout with a title bar, two text input fields (Old Password and New Password), and a single 'UPDATE PASSWORD' button.

```
10 <TextView
11     android:layout_width="match_parent"
12     android:layout_height="wrap_content"
13     android:text="Update Password"
14     android:textAlignment="center"
15     android:textColor="@color/black"
16     android:textSize="16sp"
17     android:textStyle="bold" />
18
19 <com.google.android.material.textfield.TextInputLayout
20     android:id="@+id/oldpass"
21     android:layout_width="match_parent"
22     android:layout_height="wrap_content"
23     android:layout_centerHorizontal="true"
24     android:layout_centerVertical="true"
25     app:passwordToggleEnabled="true">
26
27     <EditText
28         android:id="@+id/oldpasslog"
29         android:layout_width="match_parent"
30         android:layout_height="wrap_content"
31         android:hint="Old Password"
32         android:inputType="textPassword" />
```

The preview window on the right shows the 'Update Password' dialog with the title 'Update Password'. It contains two text input fields labeled 'Old Password' and 'New Password', and a blue 'UPDATE PASSWORD' button at the bottom right.

Pertama dibuat sebuah file .xml baru dan beri nama dialog_update_password. File ini digunakan untuk membuat alert box saat *user* ingin melakukan update password. Letakkan 2 EditText dalam TextInputLayout kemudian berikan 1 Button untuk mengsubmit.

activity_edit_profile_page.xml

The screenshot shows the Android Studio interface with two panes. The left pane displays the XML code for `activity_edit_profile_page.xml`, and the right pane shows a preview of the UI.

XML Code:

```
<ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_marginTop="18dp"
    android:layout_height="match_parent">
    <TextView
        android:id="@+id/judul_edit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:text="Edit Profile Data"
        android:textAlignment="center"
        android:textSize="26sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="MissingConstraints" />
    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/setting_profile_image"
        android:layout_width="130dp"/>
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/profilepic"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/setting_profile_image"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="Update Profile Pic"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/setting_profile_image" />
    <TextView
        android:id="@+id/editname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/profilepic"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="Update Name"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editname" />
    <TextView
        android:id="@+id/changepass"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editname"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:text="Change Password"
        android:textAlignment="center"
        android:textColor="@android:color/black"
        android:textSize="20sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editname" />

```

UI Preview:

The preview shows a dark-themed mobile application screen titled "Edit Profile Data". It features a circular profile picture placeholder at the top. Below it are three text buttons: "Update Profile Pic", "Update Name", and "Change Password". Each button has a small icon next to it. The layout uses a ConstraintLayout with various constraints like centerHorizontal, centerVertical, and layout_below to position the elements.

Kemudian dibuat sebuah activity baru bernama EditProfilePage. Dalam file .xml ini tambahkan CircleImageView agar profil *user* berbentuk bulat. Kemudian tambahkan 3 TextView yang nantinya dapat diklik untuk melakukan update data.

EditProfilePage.java

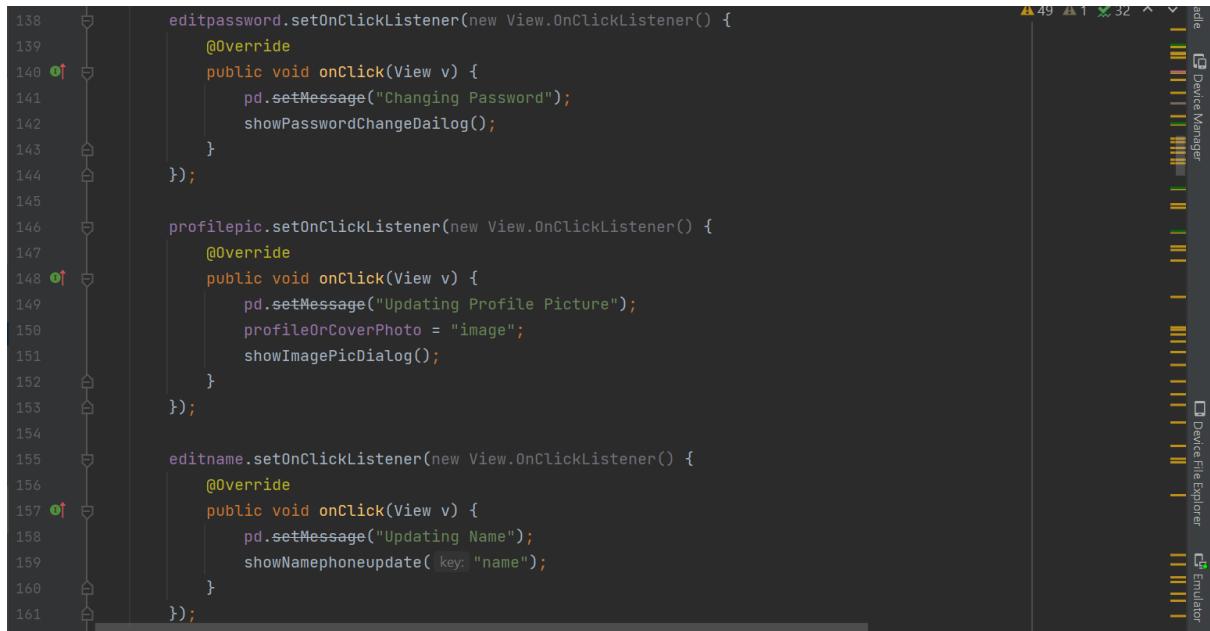
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_edit_profile_page);
    profilepic = findViewById(R.id.profilepic);
    editname = findViewById(R.id.editname);
    btnBack = findViewById(R.id.back);
    set = findViewById(R.id.setting_profile_image);
    pd = new ProgressDialog(context: this);
    pd.setCanceledOnTouchOutside(false);
    editpassword = findViewById(R.id.changepassword);
    firebaseAuth = FirebaseAuth.getInstance();
    firebaseUser = firebaseAuth.getCurrentUser();
    String uid = firebaseUser.getUid();
    FirebaseDatabase database = FirebaseDatabase.getInstance("https://uaspbb-2f830-default.firebaseio.com/");
    storageReference = FirebaseStorage.getInstance().getReference();
    databaseReference = FirebaseDatabase.getInstance().getReference("Users");
    mDatabase = FirebaseDatabase.getInstance();
    cameraPermission = new String[]{Manifest.permission.CAMERA, Manifest.permission.WRITE_EXTERNAL_STORAGE};
    storagePermission = new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE};
    Query query = databaseReference.orderByChild("email").equalTo(firebaseUser.getEmail());
    user = getIntent().getParcelableExtra(EXTRA_DATA);
}
```

Seperti biasa pertama deklarasi id dari file .xml agar dapat digunakan pada kelas java dan deklarasi beberapa kelas firebase agar dapat mengakses database. Line 100 berfungsi untuk mengambil data tambahan dari intent.

```
query.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        DatabaseReference dbKomen = mDatabase.child("Users");
        for (DataSnapshot dataSnapshot1 : dataSnapshot.getChildren()) {
            String image = "" + dataSnapshot1.child("image").getValue();
            try {
                Glide.with(activity: EditProfilePage.this).load(image).into(set);
            } catch (Exception e) {
            }
        }
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
    }
});
```

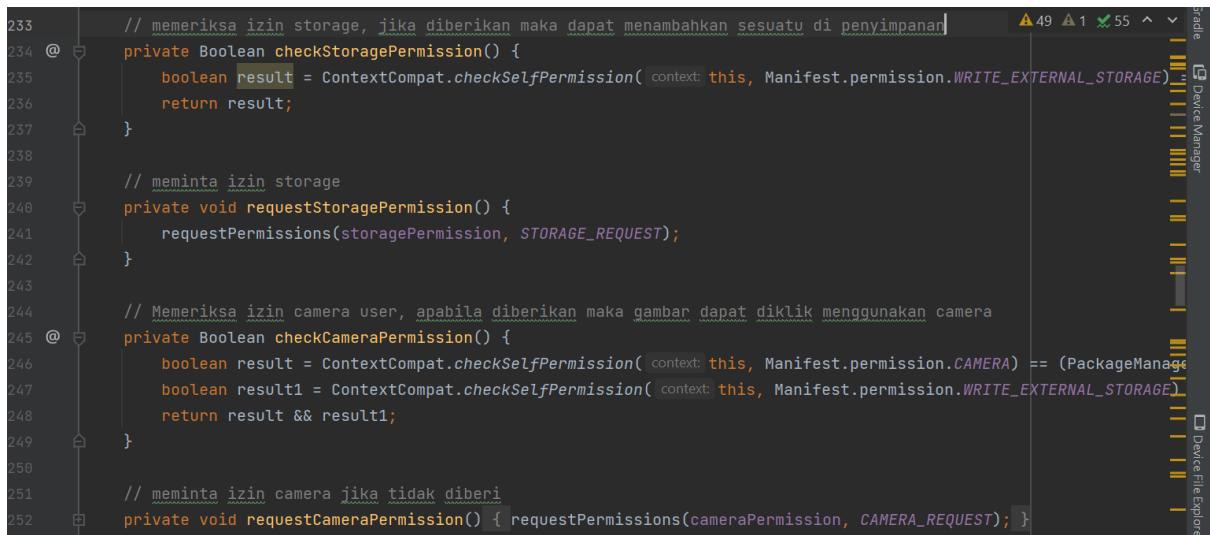
Code di atas berfungsi untuk mengambil data image dalam bentuk String pada “Users” dan meletakkannya ke dalam CircleImageView Set.



```
138     editpassword.setOnClickListener(new View.OnClickListener() {
139         @Override
140         public void onClick(View v) {
141             pd.setMessage("Changing Password");
142             showPasswordChangeDialog();
143         }
144     );
145
146     profilepic.setOnClickListener(new View.OnClickListener() {
147         @Override
148         public void onClick(View v) {
149             pd.setMessage("Updating Profile Picture");
150             profileOrCoverPhoto = "image";
151             showImagePicDialog();
152         }
153     );
154
155     editname.setOnClickListener(new View.OnClickListener() {
156         @Override
157         public void onClick(View v) {
158             pd.setMessage("Updating Name");
159             showNamePhoneupdate(key: "name");
160         }
161     );

```

Code di atas berfungsi ketika *user* mengklik salah satu *TextView* untuk melakukan *update*, maka akan dipanggil *method* untuk menampilkan sebuah dialog.



```
233     // memeriksa izin storage, jika diberikan maka dapat menambahkan sesuatu di penyimpanan
234     @
235     private Boolean checkStoragePermission() {
236         boolean result = ContextCompat.checkSelfPermission(context: this, Manifest.permission.WRITE_EXTERNAL_STORAGE);
237         return result;
238     }
239
240     // meminta izin storage
241     private void requestStoragePermission() {
242         requestPermissions(storagePermission, STORAGE_REQUEST);
243     }
244
245     // Memeriksa izin camera user, apabila diberikan maka gambar dapat diklik menggunakan camera
246     @
247     private Boolean checkCameraPermission() {
248         boolean result = ContextCompat.checkSelfPermission(context: this, Manifest.permission.CAMERA) == (PackageManager.PERMISSION_GRANTED);
249         boolean result1 = ContextCompat.checkSelfPermission(context: this, Manifest.permission.WRITE_EXTERNAL_STORAGE);
250         return result && result1;
251     }
252
253     // meminta izin camera jika tidak diberi
254     private void requestCameraPermission() { requestPermissions(cameraPermission, CAMERA_REQUEST); }
```

Code-code di atas digunakan untuk meminta izin *user* dalam penggunaan camera dan penyimpanan. *Method* *requestPerminssions()* untuk menunjukkan izin yang diperlukan aplikasi kepada pengguna, tetapi metode ini tidak menyebutkan alasan penggunaan izin. Untuk memastikan izin dari *user*, izin akan diteruskan ke *ContextCompat.checkSelfPerminssion()*. Metode *ContextCompat.checkSelfPerminssion()* menampilkan *PERMISSION_GRANTED* yang berarti izin telah diberikan.

```

257     private void showPasswordChangeDialog() {
258         View view = LayoutInflater.from(this).inflate(R.layout.dialog_update_password, root: null);
259         final EditText oldpass = view.findViewById(R.id.oldpasslog);
260         final EditText newpass = view.findViewById(R.id.newpasslog);
261         Button editpass = view.findViewById(R.id.updatepass);
262         AlertDialog.Builder builder = new AlertDialog.Builder(context: this);
263         builder.setView(view);
264         final AlertDialog dialog = builder.create();
265         dialog.show();
266         editpass.setOnClickListener(new View.OnClickListener() {
267             @Override
268             public void onClick(View v) {
269                 String oldp = oldpass.getText().toString().trim();
270                 String newp = newpass.getText().toString().trim();
271                 if (TextUtils.isEmpty(oldp)) {
272                     Toast.makeText(context: EditProfilePage.this, text: "Current Password cant be empty", Toast.LENGTH_SHORT).show();
273                     return;
274                 }
275                 if (TextUtils.isEmpty(newp)) {
276                     Toast.makeText(context: EditProfilePage.this, text: "New Password cant be empty", Toast.LENGTH_LONG).show();
277                     return;
278                 }
279                 dialog.dismiss();
280                 updatePassword(oldp, newp);
281             }
282         });
283     }

```

Code di atas berfungsi untuk menampilkan alert box dari layout dialog_update_password yang telah dibuat sebelumnya. Apabila *user* mengklik button submit, maka akan diambil inputannya (line 269-270) kemudian dilakukan pengecekan apakah salah satu EditText kosong atau tidak dan mengecek password lama menggunakan method updatePassword.

```

285     // Mengecek apakah kata sandi lama telah diautentifikasi
286     // Jika benar, maka kata sandi dapat diperbarui
287     private void updatePassword(String oldp, final String newp) {
288         pd.show();
289         final FirebaseAuth user = firebaseAuth.getCurrentUser();
290         AuthCredential authCredential = EmailAuthProvider.getCredential(user.getEmail(), oldp);
291         user.reauthenticate(authCredential)
292             .addOnSuccessListener(new OnSuccessListener<Void>() {
293                 @Override
294                 public void onSuccess(Void aVoid) {
295                     user.updatePassword(newp)
296                         .addOnSuccessListener(new OnSuccessListener<Void>() {
297                             @Override
298                             public void onSuccess(Void aVoid) {
299                                 pd.dismiss();
300                                 Toast.makeText(context: EditProfilePage.this, text: "Changed Password", Toast.LENGTH_SHORT).show();
301                             }
302                         })
303                         .addOnFailureListener(new OnFailureListener() {
304                             @Override
305                             public void onFailure(@NonNull Exception e) {
306                                 pd.dismiss();
307                                 Toast.makeText(context: EditProfilePage.this, text: "Failed", Toast.LENGTH_LONG).show();
308                             }
309                         });
310                 }
311             })
312             .addOnFailureListener(new OnFailureListener() {
313                 @Override
314                 public void onFailure(@NonNull Exception e) {
315                     pd.dismiss();
316                     Toast.makeText(context: EditProfilePage.this, text: "Failed", Toast.LENGTH_LONG).show();
317                 }
318             });
319     }

```

AuthCredential digunakan untuk mewakili kredensial supaya server Firebase Authentication dapat mengautentifikasi pengguna. Jika password lama berhasil diautentikasi, maka *new password* dapat dilakukan. Jika gagal, maka akan ditampilkan toast “Failed”.

```
320     private void showNamePhoneupdate(final String key) {
321         AlertDialog.Builder builder = new AlertDialog.Builder( context: this );
322         builder.setTitle("Update " + key);
323
324         // creating a layout to write the new name
325         LinearLayout layout = new LinearLayout( context: this );
326         layout.setOrientation(LinearLayout.VERTICAL);
327         layout.setPadding( left: 10, top: 10, right: 10, bottom: 10 );
328         final EditText editText = new EditText( context: this );
329         editText.setHint("Enter " + key);
330         layout.addView(editText);
331         builder.setView(layout);
332
333         builder.setPositiveButton( text: "Update", new DialogInterface.OnClickListener() {
334             @Override
335             public void onClick(DialogInterface dialog, int which) {
336                 final String value = editText.getText().toString().trim();
337                 //namaUser.setText(value);
338                 if (!TextUtils.isEmpty(value)) {
339                     pd.show();
340
341                     // Here we are updating the new name
342                     HashMap<String, Object> result = new HashMap<>();
343                     result.put(key, value);
344
345                     // Now we are updating the database
346                     FirebaseDatabase database = FirebaseDatabase.getInstance();
347                     DatabaseReference reference = database.getReference("Users");
348                     reference.child(key).setValue(result);
349                     pd.dismiss();
350                     Toast.makeText(context, "Name updated successfully", Toast.LENGTH_SHORT).show();
351                 }
352             }
353         });
354     }
355 }
```

Code di atas untuk membuat sebuah alert box agar *user* dapat menginput nama baru pada EditText (line 328). Apabila *user* mengklik tombol update, maka akan dilakukan pengecekan apakah EditText yang dimasukkan kosong atau tidak.

```
344 databaseReference.child(firebaseUser.getUid()).updateChildren(result).addOnSucce49  
345 ssion((  
346     @Override  
347     public void onSuccess(Void aVoid) {  
348         pd.dismiss();  
349  
350         // after updated we will show updated  
351         Toast.makeText(context: EditProfilePage.this, text: "updated", Toast.LENGTH_LONG).show();  
352     }  
353 }).addOnFailureListener(new OnFailureListener() {  
354     @Override  
355     public void onFailure(@NonNull Exception e) {  
356         pd.dismiss();  
357         Toast.makeText(context: EditProfilePage.this, text: "Unable to update", Toast.LENGTH_LONG).show();  
358     }  
359 });  
360 if (key.equals("name")) {  
361     DatabaseReference databaser = FirebaseDatabase.getInstance().getReference(path: "Users");  
362     DatabaseReference dbData = mDatabase.child("Users");  
363     Query query = databaser.orderByChild("email").equalTo(firebaseUser.getEmail());  
364     query.addValueEventListener(new ValueEventListener() {  
365         @Override  
366         public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
367             for (DataSnapshot dataSnapshot : dataSnapshot.getChildren()) {  
368                 user.setName(value);  
369  
370                 dbData.child(dataId).setValue(user);  
371             }  
372         }  
373     });  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }  
1001 }  
1002 }  
1003 }  
1004 }  
1005 }  
1006 }  
1007 }  
1008 }  
1009 }  
1010 }  
1011 }  
1012 }  
1013 }  
1014 }  
1015 }  
1016 }  
1017 }  
1018 }  
1019 }  
1020 }  
1021 }  
1022 }  
1023 }  
1024 }  
1025 }  
1026 }  
1027 }  
1028 }  
1029 }  
1030 }  
1031 }  
1032 }  
1033 }  
1034 }  
1035 }  
1036 }  
1037 }  
1038 }  
1039 }  
1040 }  
1041 }  
1042 }  
1043 }  
1044 }  
1045 }  
1046 }  
1047 }  
1048 }  
1049 }  
1050 }  
1051 }  
1052 }  
1053 }  
1054 }  
1055 }  
1056 }  
1057 }  
1058 }  
1059 }  
1060 }  
1061 }  
1062 }  
1063 }  
1064 }  
1065 }  
1066 }  
1067 }  
1068 }  
1069 }  
1070 }  
1071 }  
1072 }  
1073 }  
1074 }  
1075 }  
1076 }  
1077 }  
1078 }  
1079 }  
1080 }  
1081 }  
1082 }  
1083 }  
1084 }  
1085 }  
1086 }  
1087 }  
1088 }  
1089 }  
1090 }  
1091 }  
1092 }  
1093 }  
1094 }  
1095 }  
1096 }  
1097 }  
1098 }  
1099 }  
1100 }  
1101 }  
1102 }  
1103 }  
1104 }  
1105 }  
1106 }  
1107 }  
1108 }  
1109 }  
1110 }  
1111 }  
1112 }  
1113 }  
1114 }  
1115 }  
1116 }  
1117 }  
1118 }  
1119 }  
1120 }  
1121 }  
1122 }  
1123 }  
1124 }  
1125 }  
1126 }  
1127 }  
1128 }  
1129 }  
1130 }  
1131 }  
1132 }  
1133 }  
1134 }  
1135 }  
1136 }  
1137 }  
1138 }  
1139 }  
1140 }  
1141 }  
1142 }  
1143 }  
1144 }  
1145 }  
1146 }  
1147 }  
1148 }  
1149 }  
1150 }  
1151 }  
1152 }  
1153 }  
1154 }  
1155 }  
1156 }  
1157 }  
1158 }  
1159 }  
1160 }  
1161 }  
1162 }  
1163 }  
1164 }  
1165 }  
1166 }  
1167 }  
1168 }  
1169 }  
1170 }  
1171 }  
1172 }  
1173 }  
1174 }  
1175 }  
1176 }  
1177 }  
1178 }  
1179 }  
1180 }  
1181 }  
1182 }  
1183 }  
1184 }  
1185 }  
1186 }  
1187 }  
1188 }  
1189 }  
1190 }  
1191 }  
1192 }  
1193 }  
1194 }  
1195 }  
1196 }  
1197 }  
1198 }  
1199 }  
1200 }  
1201 }  
1202 }  
1203 }  
1204 }  
1205 }  
1206 }  
1207 }  
1208 }  
1209 }  
1210 }  
1211 }  
1212 }  
1213 }  
1214 }  
1215 }  
1216 }  
1217 }  
1218 }  
1219 }  
1220 }  
1221 }  
1222 }  
1223 }  
1224 }  
1225 }  
1226 }  
1227 }  
1228 }  
1229 }  
1230 }  
1231 }  
1232 }  
1233 }  
1234 }  
1235 }  
1236 }  
1237 }  
1238 }  
1239 }  
1240 }  
1241 }  
1242 }  
1243 }  
1244 }  
1245 }  
1246 }  
1247 }  
1248 }  
1249 }  
1250 }  
1251 }  
1252 }  
1253 }  
1254 }  
1255 }  
1256 }  
1257 }  
1258 }  
1259 }  
1260 }  
1261 }  
1262 }  
1263 }  
1264 }  
1265 }  
1266 }  
1267 }  
1268 }  
1269 }  
1270 }  
1271 }  
1272 }  
1273 }  
1274 }  
1275 }  
1276 }  
1277 }  
1278 }  
1279 }  
1280 }  
1281 }  
1282 }  
1283 }  
1284 }  
1285 }  
1286 }  
1287 }  
1288 }  
1289 }  
1290 }  
1291 }  
1292 }  
1293 }  
1294 }  
1295 }  
1296 }  
1297 }  
1298 }  
1299 }  
1300 }  
1301 }  
1302 }  
1303 }  
1304 }  
1305 }  
1306 }  
1307 }  
1308 }  
1309 }  
1310 }  
1311 }  
1312 }  
1313 }  
1314 }  
1315 }  
1316 }  
1317 }  
1318 }  
1319 }  
1320 }  
1321 }  
1322 }  
1323 }  
1324 }  
1325 }  
1326 }  
1327 }  
1328 }  
1329 }  
1330 }  
1331 }  
1332 }  
1333 }  
1334 }  
1335 }  
1336 }  
1337 }  
1338 }  
1339 }  
1340 }  
1341 }  
1342 }  
1343 }  
1344 }  
1345 }  
1346 }  
1347 }  
1348 }  
1349 }  
1350 }  
1351 }  
1352 }  
1353 }  
1354 }  
1355 }  
1356 }  
1357 }  
1358 }  
1359 }  
1360 }  
1361 }  
1362 }  
1363 }  
1364 }  
1365 }  
1366 }  
1367 }  
1368 }  
1369 }  
1370 }  
1371 }  
1372 }  
1373 }  
1374 }  
1375 }  
1376 }  
1377 }  
1378 }  
1379 }  
1380 }  
1381 }  
1382 }  
1383 }  
1384 }  
1385 }  
1386 }  
1387 }  
1388 }  
1389 }  
1390 }  
1391 }  
1392 }  
1393 }  
1394 }  
1395 }  
1396 }  
1397 }  
1398 }  
1399 }  
1400 }  
1401 }  
1402 }  
1403 }  
1404 }  
1405 }  
1406 }  
1407 }  
1408 }  
1409 }  
1410 }  
1411 }  
1412 }  
1413 }  
1414 }  
1415 }  
1416 }  
1417 }  
1418 }  
1419 }  
1420 }  
1421 }  
1422 }  
1423 }  
1424 }  
1425 }  
1426 }  
1427 }  
1428 }  
1429 }  
1430 }  
1431 }  
1432 }  
1433 }  
1434 }  
1435 }  
1436 }  
1437 }  
1438 }  
1439 }  
1440 }  
1441 }  
1442 }  
1443 }  
1444 }  
1445 }  
1446 }  
1447 }  
1448 }  
1449 }  
1450 }  
1451 }  
1452 }  
1453 }  
1454 }  
1455 }  
1456 }  
1457 }  
1458 }  
1459 }  
1460 }  
1461 }  
1462 }  
1463 }  
1464 }  
1465 }  
1466 }  
1467 }  
1468 }  
1469 }  
1470 }  
1471 }  
1472 }  
1473 }  
1474 }  
1475 }  
1476 }  
1477 }  
1478 }  
1479 }  
1480 }  
1481 }  
1482 }  
1483 }  
1484 }  
1485 }  
1486 }  
1487 }  
1488 }  
1489 }  
1490 }  
1491 }  
1492 }  
1493 }  
1494 }  
1495 }  
1496 }  
1497 }  
1498 }  
1499 }  
1500 }  
1501 }  
1502 }  
1503 }  
1504 }  
1505 }  
1506 }  
1507 }  
1508 }  
1509 }  
1510 }  
1511 }  
1512 }  
1513 }  
1514 }  
1515 }  
1516 }  
1517 }  
1518 }  
1519 }  
1520 }  
1521 }  
1522 }  
1523 }  
1524 }  
1525 }  
1526 }  
1527 }  
1528 }  
1529 }  
1530 }  
1531 }  
1532 }  
1533 }  
1534 }  
1535 }  
1536 }  
1537 }  
1538 }  
1539 }  
1540 }  
1541 }  
1542 }  
1543 }  
1544 }  
1545 }  
1546 }  
1547 }  
1548 }  
1549 }  
1550 }  
1551 }  
1552 }  
1553 }  
1554 }  
1555 }  
1556 }  
1557 }  
1558 }  
1559 }  
1560 }  
1561 }  
1562 }  
1563 }  
1564 }  
1565 }  
1566 }  
1567 }  
1568 }  
1569 }  
1570 }  
1571 }  
1572 }  
1573 }  
1574 }  
1575 }  
1576 }  
1577 }  
1578 }  
1579 }  
1580 }  
1581 }  
1582 }  
1583 }  
1584 }  
1585 }  
1586 }  
1587 }  
1588 }  
1589 }  
1590 }  
1591 }  
1592 }  
1593 }  
1594 }  
1595 }  
1596 }  
1597 }  
1598 }  
1599 }  
1600 }  
1601 }  
1602 }  
1603 }  
1604 }  
1605 }  
1606 }  
1607 }  
1608 }  
1609 }  
1610 }  
1611 }  
1612 }  
1613 }  
1614 }  
1615 }  
1616 }  
1617 }  
1618 }  
1619 }  
1620 }  
1621 }  
1622 }  
1623 }  
1624 }  
1625 }  
1626 }  
1627 }  
1628 }  
1629 }  
1630 }  
1631 }  
1632 }  
1633 }  
1634 }  
1635 }  
1636 }  
1637 }  
1638 }  
1639 }  
1640 }  
1641 }  
1642 }  
1643 }  
1644 }  
1645 }  
1646 }  
1647 }  
1648 }  
1649 }  
1650 }  
1651 }  
1652 }  
1653 }  
1654 }  
1655 }  
1656 }  
1657 }  
1658 }  
1659 }  
1660 }  
1661 }  
1662 }  
1663 }  
1664 }  
1665 }  
1666 }  
1667 }  
1668 }  
1669 }  
1670 }  
1671 }  
1672 }  
1673 }  
1674 }  
1675 }  
1676 }  
1677 }  
1678 }  
1679 }  
1680 }  
1681 }  
1682 }  
1683 }  
1684 }  
1685 }  
1686 }  
1687 }  
1688 }  
1689 }  
1690 }  
1691 }  
1692 }  
1693 }  
1694 }  
1695 }  
1696 }  
1697 }  
1698 }  
1699 }  
1700 }  
1701 }  
1702 }  
1703 }  
1704 }  
1705 }  
1706 }  
1707 }  
1708 }  
1709 }  
1710 }  
1711 }  
1712 }  
1713 }  
1714 }  
1715 }  
1716 }  
1717 }  
1718 }  
1719 }  
1720 }  
1721 }  
1722 }  
1723 }  
1724 }  
1725 }  
1726 }  
1727 }  
1728 }  
1729 }  
1730 }  
1731 }  
1732 }  
1733 }  
1734 }  
1735 }  
1736 }  
1737 }  
1738 }  
1739 }  
1740 }  
1741 }  
1742 }  
1743 }  
1744 }  
1745 }  
1746 }  
1747 }  
1748 }  
1749 }  
1750 }  
1751 }  
1752 }  
1753 }  
1754 }  
1755 }  
1756 }  
1757 }  
1758 }  
1759 }  
1760 }  
1761 }  
1762 }  
1763 }  
1764 }  
1765 }  
1766 }  
1767 }  
1768 }  
1769 }  
1770 }  
1771 }  
1772 }  
1773 }  
1774 }  
1775 }  
1776 }  
1777 }  
1778 }  
1779 }  
1780 }  
1781 }  
1782 }  
1783 }  
1784 }  
1785 }  
1786 }  
1787 }  
1788 }  
1789 }  
1790 }  
1791 }  
1792 }  
1793 }  
1794 }  
1795 }  
1796 }  
1797 }  
1798 }  
1799 }  
1800 }  
1801 }  
1802 }  
1803 }  
1804 }  
1805 }  
1806 }  
1807 }  
1808 }  
1809 }  
1810 }  
1811 }  
1812 }  
1813 }  
1814 }  
1815 }  
1816 }  
1817 }  
1818 }  
1819 }  
1820 }  
1821 }  
1822 }  
1823 }  
1824 }  
1825 }  
1826 }  
1827 }  
1828 }  
1829 }  
1830 }  
1831 }  
1832 }  
1833 }  
1834 }  
1835 }  
1836 }  
1837 }  
1838 }  
1839 }  
1840 }  
1841 }  
1842 }  
1843 }  
1844 }  
1845 }  
1846 }  
1847 }  
1848 }  
1849 }  
1850 }  
1851 }  
1852 }  
1853 }  
1854 }  
1855 }  
1856 }  
1857 }  
1858 }  
1859 }  
1860 }  
1861 }  
1862 }  
1863 }  
1864 }  
1865 }  
1866 }  
1867 }  
1868 }  
1869 }  
1870 }  
1871 }  
1872 }  
1873 }  
1874 }  
1875 }  
1876 }  
1877 }  
1878 }  
1879 }  
1880 }  
1881 }  
1882 }  
1883 }  
1884 }  
1885 }  
1886 }  
1887 }  
1888 }  
1889 }  
1890 }  
1891 }  
1892 }  
1893 }  
1894 }  
1895 }  
1896 }  
1897 }  
1898 }  
1899 }  
1900 }  
1901 }  
1902 }  
1903 }  
1904 }  
1905 }  
1906 }  
1907 }  
1908 }  
1909 }  
1910 }  
1911 }  
1912 }  
1913 }  
1914 }  
1915 }  
1916 }  
1917 }  
1918 }  
1919 }  
1920 }  
1921 }  
1922 }  
1923 }  
1924 }  
1925 }  
1926 }  
1927 }  
1928 }  
1929 }  
1930 }  
1931 }  
1932 }  
1933 }  
1934 }  
1935 }  
1936 }  
1937 }  
1938 }  
1939 }  
1940 }  
1941 }  
1942 }  
1943 }  
1944 }  
1945 }<
```

Kemudian akan diambilUid database yang sama dengan uid authentication dan dicari berdasarkan key yaitu name. Apabila ditemukan key name, maka akan diambil inputan *user* untuk update name.

```
394     // Menampilkan dialog pilihan gambar akan diambil dari camera atau gallery
395     private void showImagePicDialog() {
396         String options[] = {"Camera", "Gallery"};
397         AlertDialog.Builder builder = new AlertDialog.Builder( context: this );
398         builder.setTitle("Pick Image From");
399         builder.setItems(options, new DialogInterface.OnClickListener() {
400             @Override
401             public void onClick(DialogInterface dialog, int which) {
402                 // if access is not given then we will request for permission
403                 if (which == 0) {
404                     if (!checkCameraPermission()) {
405                         requestCameraPermission();
406                     } else {
407                         pickFromCamera();
408                     }
409                 } else if (which == 1) {
410                     if (!checkStoragePermission()) {
411                         requestStoragePermission();
412                     } else {
413                         pickFromGallery();
414                     }
415                 }
416             }
417         });
418         builder.create().show();
```

Code 396 untuk membuat pilihan untuk *user* berupa Cameraa atau Galerry, sedangkan code 398 berfungsi untuk memberi title pada alert box. Jika *user* memilih Camera, maka akan dilakukan pengecekan izin. Jika tidak diberi izin maka akan dijalankan *method* requestCameraPermission, namun jika diberi izin akan dijalankan *method* pickFromCamera(). Jika *user* memilih Gallery, maka akan dilakukan pengecekan izin juga. Apabila diberi izin maka dijalankan *method* pickFromGallery(), namun jika tidak akan dijalankan *method* requestStoragePermission().

```
421     @Override
422     public void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
423         if (resultCode == Activity.RESULT_OK) {
424             if (requestCode == IMAGEPICK_GALLERY_REQUEST) {
425                 imageuri = data.getData();
426                 uploadProfileCoverPhoto(imageuri);
427             }
428             if (requestCode == IMAGE_PICKCAMERA_REQUEST) {
429                 uploadProfileCoverPhoto(imageuri);
430             }
431         }
432         super.onActivityResult(requestCode, resultCode, data);
433     }
```

Method ini memberi requestCode, resultCode, dan data tambahan berfungsi untuk memproses data yang dikembalikan di aktivitas pembuatnya. resultCode merupakan kode hasil yang disetel pada aktivitas yang dilauch, requestCode merupakan kode permintaan pribadi untuk pengirim yang akan dikaitkan dengan hasil data saat dikembalikan, kemudian data berisi data yang dikembalikan dari *activity* peluncur.

```

465     // Mengklik foto kemudian memulai activity untuk memperbarui data
466     private void pickFromCamera() {
467         ContentValues contentValues = new ContentValues();
468         contentValues.put(MediaStore.Images.Media.TITLE, "Temp_pic");
469         contentValues.put(MediaStore.Images.Media.DESCRIPTION, "Temp Description");
470         imageuri = this.getContentResolver().insert(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, contentValues);
471         Intent camerIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
472         camerIntent.putExtra(MediaStore.EXTRA_OUTPUT, imageuri);
473         startActivityForResult(camerIntent, IMAGE_PICKCAMERA_REQUEST);
474     }
475
476     // memilih gambar dari galerry
477     private void pickFromGallery() {
478         Intent galleryIntent = new Intent(Intent.ACTION_PICK);
479         galleryIntent.setType("image/*");
480         startActivityForResult(galleryIntent, IMAGEPICK_GALLERY_REQUEST);
481     }

```

ContentValues merupakan kelas untuk menyimpan sekumpulan values yang dapat diproses oleh ContentResolver. Line 468 – 469 untuk menambahkan value ke set. Kemudian akan berpindah activity untuk memperbarui data.

```

483     // upload image
484     private void uploadProfileCoverPhoto(final Uri uri) {
485         pd.show();
486
487         // We are taking the filepath as storagepath + firebaseauth.getUid()+"png"
488         String pathname = storagepath + "" + profileOrCoverPhoto + "_" + firebaseUser.getUid();
489         StorageReference storageReference1 = storageReference.child(pathname);
490         storageReference1.putFile(uri).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
491             @Override
492             public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
493                 Task<Uri> uritask = taskSnapshot.getStorage().getDownloadUrl();
494                 while (!uritask.isSuccessful());
495
496                 // We will get the url of our image using uritask
497                 final Uri downloadUri = uritask.getResult();
498                 if (uritask.isSuccessful()) {
499
500                     // updating our image url into the realtime database
501                     HashMap<String, Object> hashMap = new HashMap<>();
502                     hashMap.put(profileOrCoverPhoto, downloadUri.toString());
503                     databaseReference.child(firebaseUser.getUid()).updateChildren(hashMap).addOnSuccessListener(new OnSuccessListener< DatabaseReference.CompletionListener >() {
504                         @Override
505                         public void onSuccess(Void aVoid) {
506                             pd.dismiss();
507                             Toast.makeText(EditProfilePage.this, "Updated", Toast.LENGTH_LONG).show();
508                         }
509                     });
510                 }
511             }
512         });
513     }

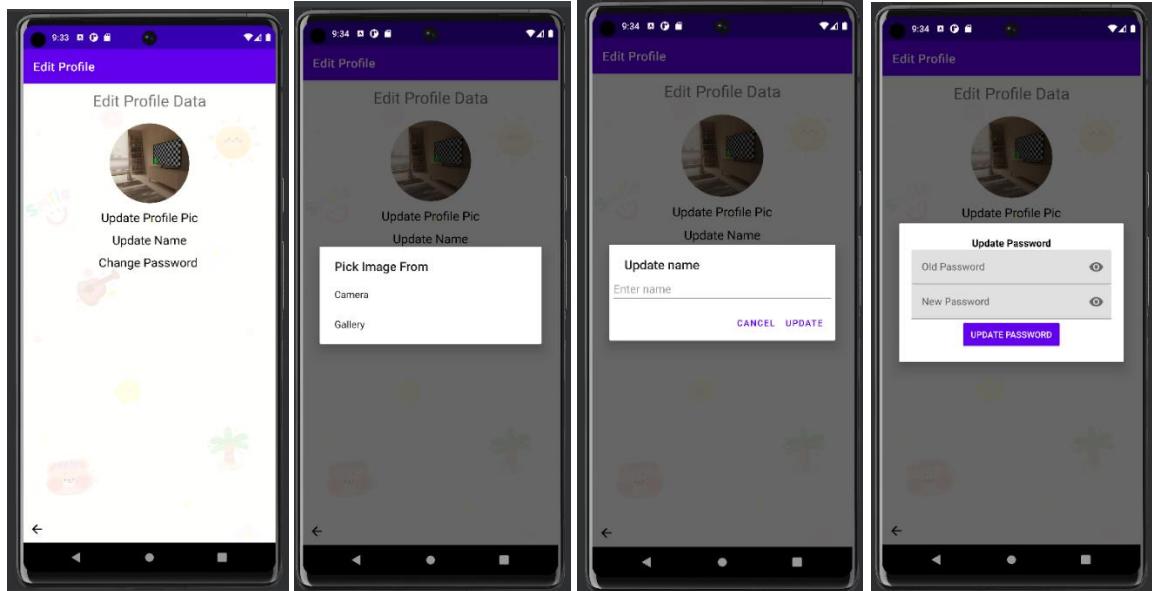
```

Code diatas berfungsi untuk mengupload data image yang diubah berdasarkan pathname dalam database.

```
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
```

Apabila upload gagal, maka akan ditampilkan *toast* bahwa update mengalami eror.

Emulator



Ketika emulator dijalankan, maka akan menampilkan tampilan seperti gambar pertama, kemudian jika update profil diklik maka akan ditampilkan tampilan seperti gambar kedua. Kemudian apabila update name diklik, maka akan menampilkan tampilan seperti gambar ketiga, sedangkan change password seperti gambar keempat.

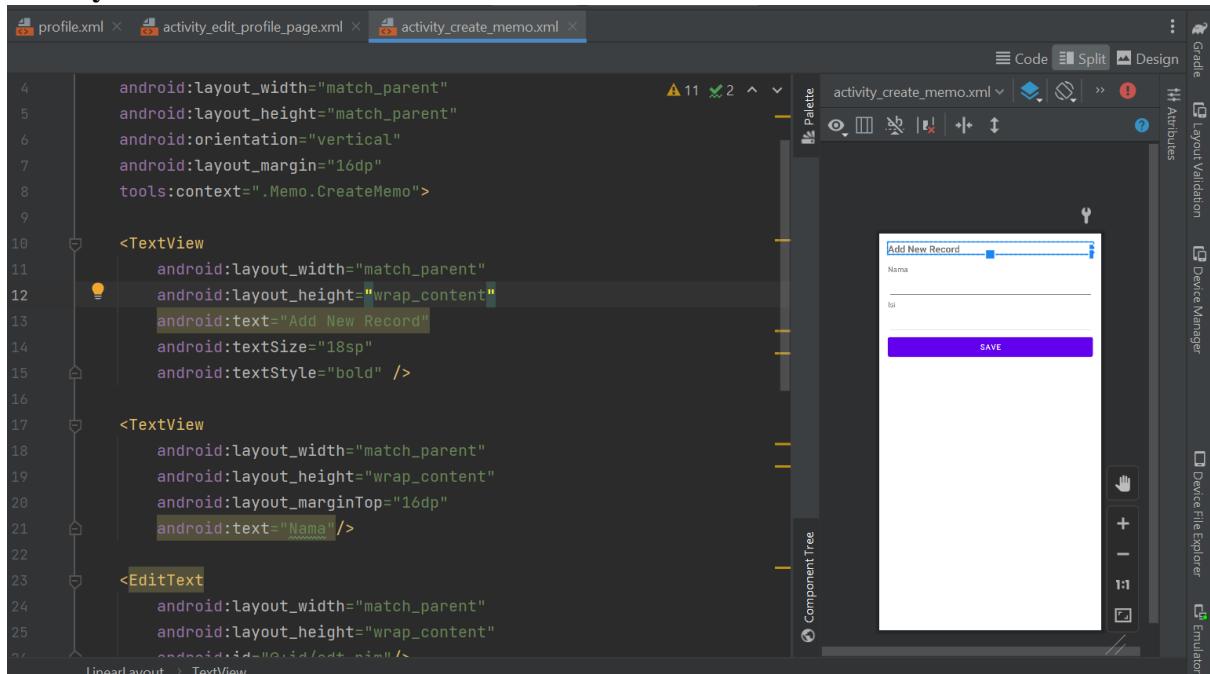
Comment

Komentar.java

```
5  public class Komentar implements Parcelable {
6      private String id;
7      private String nama;
8      private String komen;
9
10     public Komentar() {}
11
12     public String getId() { return id; }
13
14     public void setId(String id) { this.id = id; }
15
16     public String getKomen() { return komen; }
17
18     public void setKomen(String komen) { this.komen = komen; }
19
20     public String getNama() { return nama; }
21
22     public void setNama(String nama) { this.nama = nama; }
23
24
25     @Override
26     public int describeContents() { return 0; }
27
28     public int describeContents() { return 0; }
29
30
31     @Override
32     public void writeToParcel(Parcel dest, int flags) {
33         dest.writeString(this.id);
34         dest.writeString(this.komen);
35         dest.writeString(this.nama);
36     }
37
38     @Protected
39     protected Komentar(Parcel in) {
40         this.id = in.readString();
41         this.komen = in.readString();
42         this.nama = in.readString();
43     }
44
45     public static final Parcelable.Creator<Komentar> CREATOR = new Parcelable.Creator<Komentar>() {
46         @Override
47         public Komentar createFromParcel(Parcel source) { return new Komentar(source); }
48
49         @Override
50         public Komentar[] newArray(int size) { return new Komentar[size]; }
51     }
52 }
```

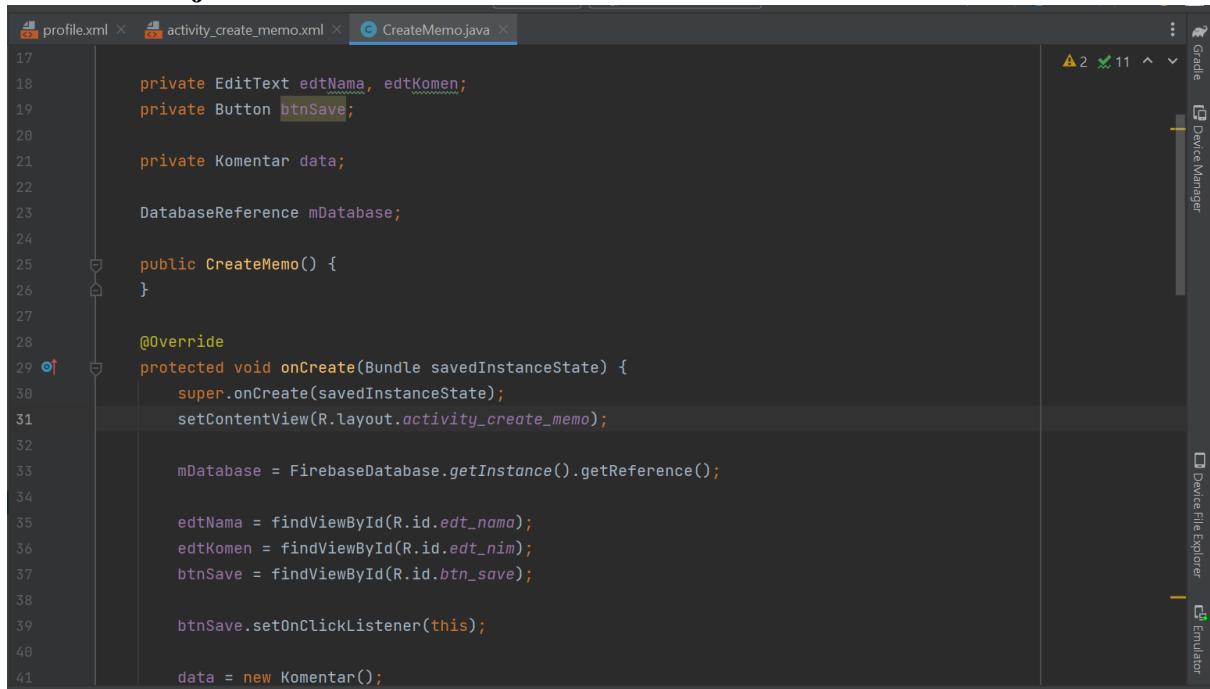
Selanjutnya buatlah sebuah kelas baru dan beri nama Komentar.java. Kelas ini berfungsi untuk memudahkan pengiriman obyek secara langsung berupa data id, nama dan komen dengan menggunakan **Parcelable**.

activity_create_memo.xml



Pertama buatlah sebuah activity baru dan beri nama CreateMemo. Kemudian tambahkan 2 EditText agar *user* bisa menambah text pada database.

CreateMemo.java



Pertama deklarasi EditText dan Button ke dalam kelas java agar bisa diakses. Kemudian import dan tambahkan kelas Komentar.

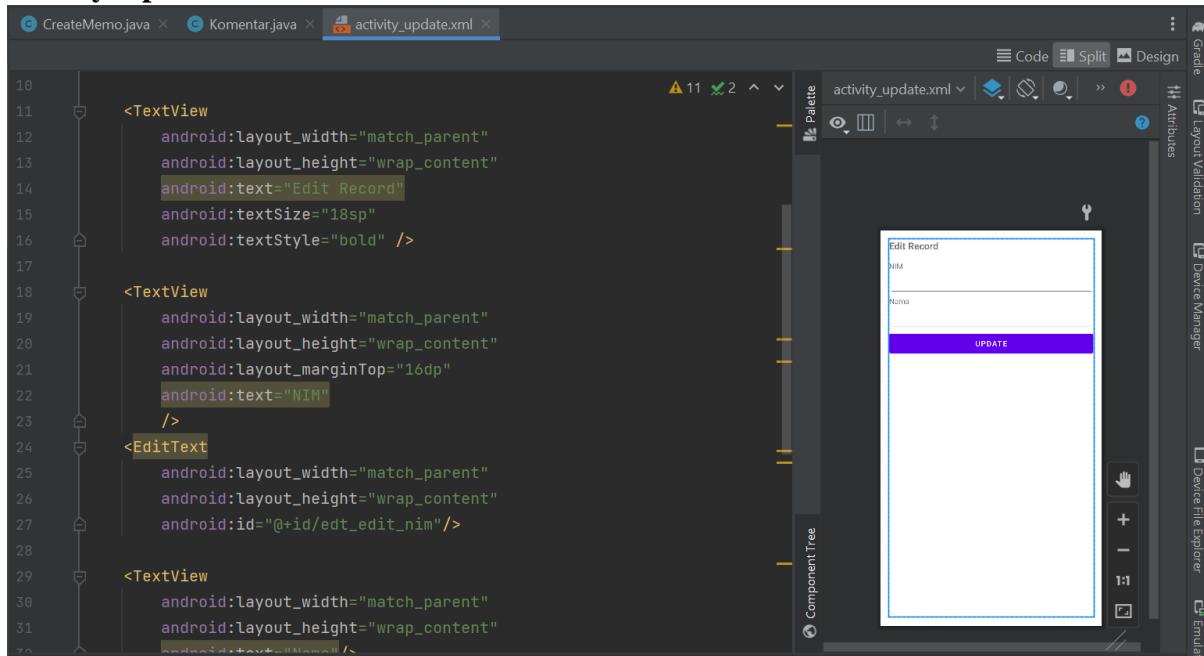
```

53     private void saveData()
54     {
55         String nama = edtNama.getText().toString().trim();
56         String komen = edtKomen.getText().toString().trim();
57
58         boolean isEmptyFields = false;
59
60         if (TextUtils.isEmpty(nama)) {
61             isEmptyFields = true;
62             edtNama.setError("Field ini tidak boleh kosong");
63         }
64
65         if (TextUtils.isEmpty(komen)) {
66             isEmptyFields = true;
67             edtKomen.setError("Field ini tidak boleh kosong");
68         }
69
70         if (! isEmptyFields) {
71
72             Toast.makeText( context: CreateMemo.this, text: "Saving Data...", Toast.LENGTH_SHORT).show();
73
74             DatabaseReference dbKomen = mDatabase.child("comment");
75
76             String id = dbKomen.push().getKey();
77             data.setId(id);
78             data.setKomen(komen);
79             data.setNama(nama);
80
81             //insert data
82             dbKomen.child(id).setValue(data);
83
84             finish();
85
86         }
87     }
88 }

```

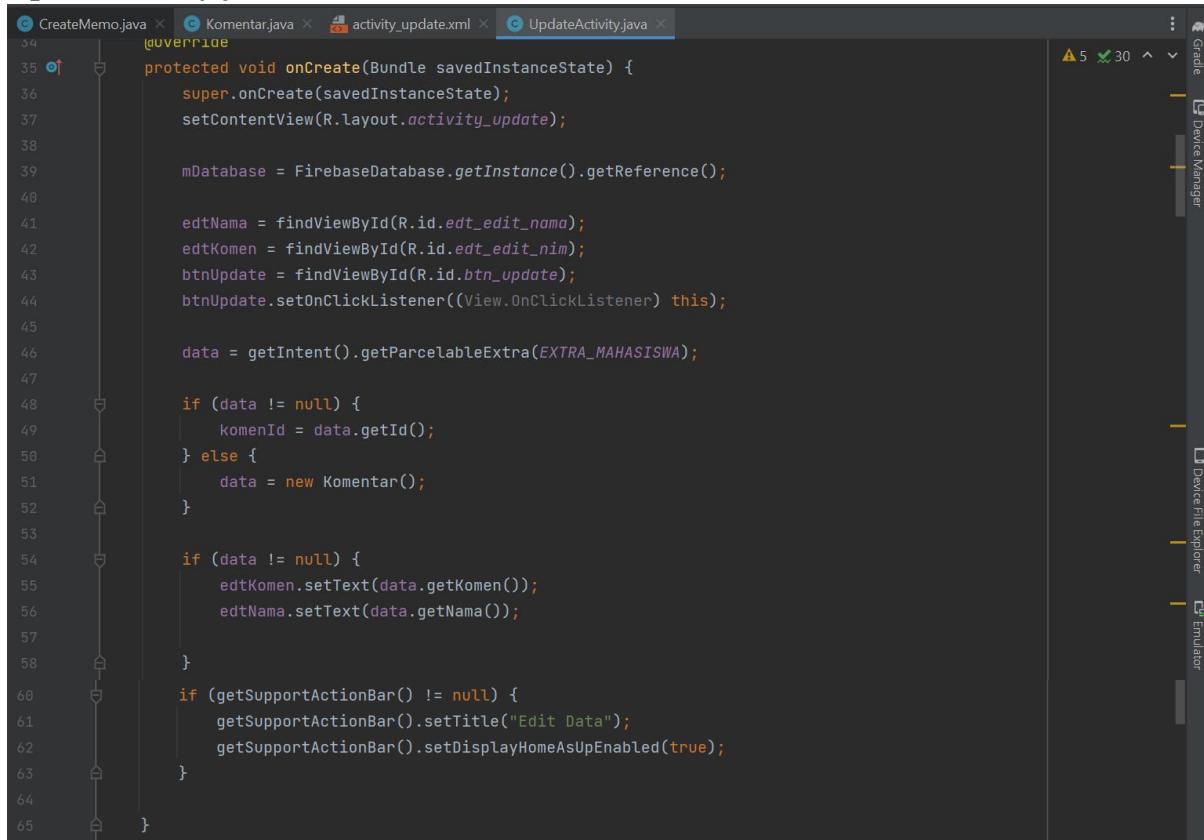
Untuk menambahkan data ke firebase, kita perlu mengambil terlebih dahulu inputan *user* kemudian dicek apakah inputannya kosong atau tidak. Jika tidak, maka tambahkan inputan *user* ke dalam comment dengan mengambil *method* dari kelas Komentar.

activity_update.xml



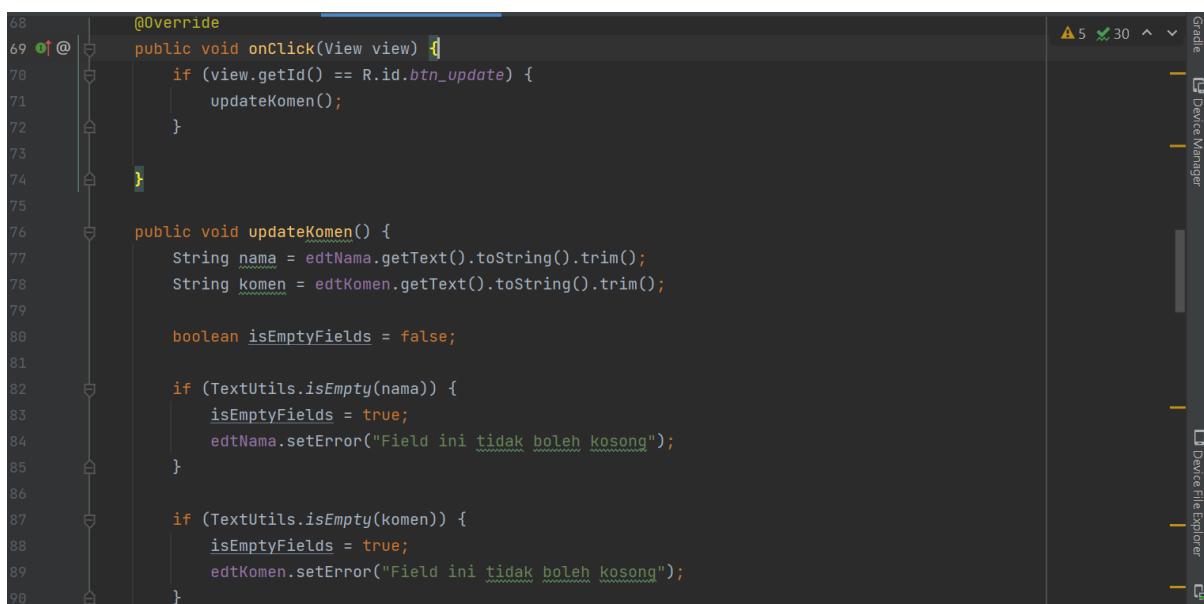
Selanjutnya buatlah sebuah activity baru yang Bernama UpdateActivity. Dalam file .xml tambahkan 2 EditText untuk mengupdate data name dan komen. Tambahkan 1 Button untuk melakukan submit

UpdateActivity.java



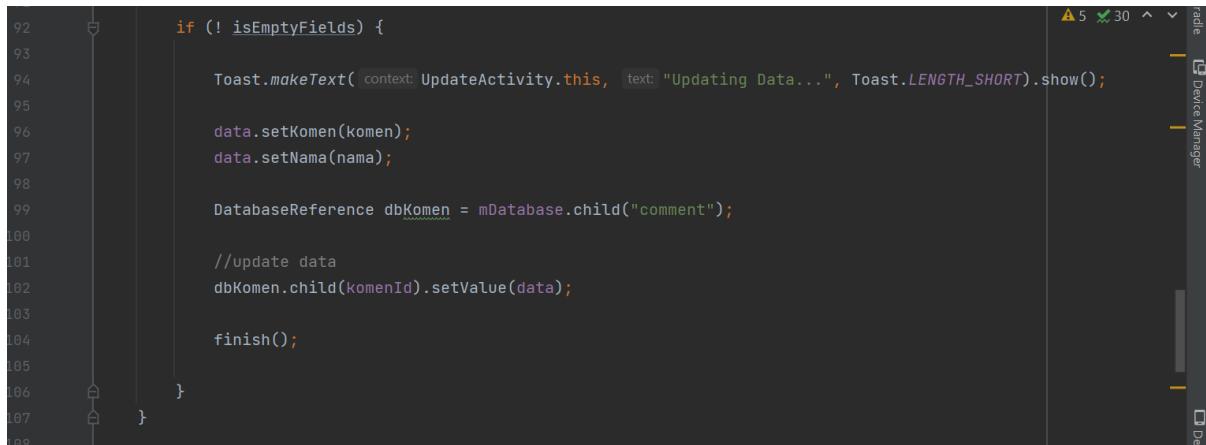
```
34  @Override
35  protected void onCreate(Bundle savedInstanceState) {
36      super.onCreate(savedInstanceState);
37      setContentView(R.layout.activity_update);
38
39      mDatabase = FirebaseDatabase.getInstance().getReference();
40
41      edtNama = findViewById(R.id.edt_edit_nama);
42      edtKomen = findViewById(R.id.edt_edit_nim);
43      btnUpdate = findViewById(R.id.btn_update);
44      btnUpdate.setOnClickListener((View.OnClickListener) this);
45
46      data = getIntent().getParcelableExtra(EXTRA_MAHASISWA);
47
48      if (data != null) {
49          komenId = data.getId();
50      } else {
51          data = new Komentar();
52      }
53
54      if (data != null) {
55          edtKomen.setText(data.getKomen());
56          edtNama.setText(data.getNama());
57      }
58
59      if (getSupportActionBar() != null) {
60          getSupportActionBar().setTitle("Edit Data");
61          getSupportActionBar().setDisplayHomeAsUpEnabled(true);
62      }
63
64  }
65 }
```

Apabila data Komentar yang diambil tidak null, maka akan diambil id dari data tersebut. Kemudian EditText akan diganti textnya berdasarkan nama dan komen yang didapat.



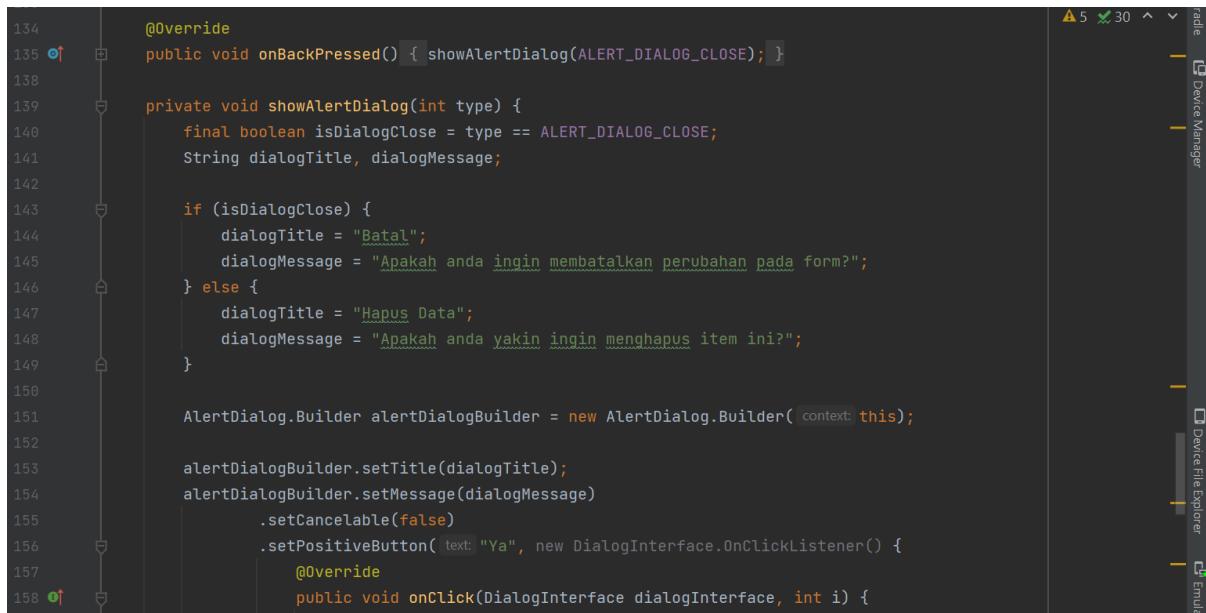
```
66  @Override
67  public void onClick(View view) {
68      if (view.getId() == R.id.btn_update) {
69          updateKomen();
70      }
71  }
72
73
74
75
76  public void updateKomen() {
77      String nama = edtNama.getText().toString().trim();
78      String komen = edtKomen.getText().toString().trim();
79
80      boolean isEmptyFields = false;
81
82      if (TextUtils.isEmpty(nama)) {
83          isEmptyFields = true;
84          edtNama.setError("Field ini tidak boleh kosong");
85      }
86
87      if (TextUtils.isEmpty(komen)) {
88          isEmptyFields = true;
89          edtKomen.setError("Field ini tidak boleh kosong");
90      }
91  }
92 }
```

Untuk melakukan update, inputan *user* akan dicek apakah kosong atau tidak. Jika kosong maka akan diberi error bahwa field tidak boleh kosong



```
92     if (! isEmptyFields) {
93
94         Toast.makeText( context: UpdateActivity.this, text: "Updating Data...", Toast.LENGTH_SHORT).show();
95
96         data.setKomen(komen);
97         data.setNama(nama);
98
99         DatabaseReference dbKomen = mDatabase.child("comment");
100
101        //update data
102        dbKomen.child(komenId).setValue(data);
103
104        finish();
105
106    }
107 }
```

Jika EditText tidak kosong, maka akan dilakukan update data dengan memanggil *method* dari kelas Komentar.



```
134     @Override
135     public void onBackPressed() { showAlertDialog(ALERT_DIALOG_CLOSE); }
136
137
138     private void showAlertDialog(int type) {
139         final boolean isDialogClose = type == ALERT_DIALOG_CLOSE;
140         String dialogTitle, dialogMessage;
141
142         if (isDialogClose) {
143             dialogTitle = "Batal";
144             dialogMessage = "Apakah anda ingin membatalkan perubahan pada form?";
145         } else {
146             dialogTitle = "Hapus Data";
147             dialogMessage = "Apakah anda yakin ingin menghapus item ini?";
148         }
149
150
151         AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: this);
152
153         alertDialogBuilder.setTitle(dialogTitle);
154         alertDialogBuilder.setMessage(dialogMessage)
155             .setCancelable(false)
156             .setPositiveButton( text: "Ya", new DialogInterface.OnClickListener() {
157                 @Override
158                 public void onClick(DialogInterface dialogInterface, int i) {
```

Apabila *user* mengklik button back, maka akan menampilkan alert box berupa meyakinkan *user* untuk membatalkan update. Sedangkan apabila *user* mengklik button x, maka akan menampilkan alert box untuk meyakinkan *user* untuk menghapus item.

```
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
```

```
if (isDialogClose) {
    finish();
} else {
    //hapus data
    DatabaseReference dbKomen =
        mDatabase.child("comment").child(komenId);

    dbKomen.removeValue();

    Toast.makeText(context, UpdateActivity.this, text: "Deleting data...", Toast.LENGTH_SHORT).show();
    finish();
}

.setNegativeButton(text: "Tidak", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        dialogInterface.cancel();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
```

Jika *user* menjawab “ya” pada hapus data, maka data yang dipilih akan terhapus menggunakan *method .removeValue()*. Sedangkan jika *user* mengklik “tidak” maka akan dibatalkan.

KomentarAdapter.java

```
14 public class KomentarAdapter extends BaseAdapter {
15     private Context context;
16     private ArrayList<Komentar> komenList = new ArrayList<>();
17
18     public void setKomentarList(ArrayList<Komentar> mahasiswaList) {
19         this.komenList = mahasiswaList;
20     }
21
22     public KomentarAdapter(Context context) { this.context = context; }
23
24
25     @Override
26     public int getCount() { return komenList.size(); }
27
28     @Override
29     public Object getItem(int i) { return komenList.get(i); }
30
31     @Override
32     public long getItemId(int i) { return i; }
33
34     @Override
35     public View getView(int i, View view, ViewGroup viewGroup) {
36         View itemView = view;
37
38         if (itemView == null) {
39             itemView = LayoutInflater.from(context).inflate(R.layout.item_komentar, viewGroup, false);
40
41             TextView tvKomentar = itemView.findViewById(R.id.tv_komentar);
42             tvKomentar.setText(komenList.get(i).getKomentar());
43
44             TextView tvNama = itemView.findViewById(R.id.tv_nama);
45             tvNama.setText(komenList.get(i).getNama());
46
47             ImageView ivProfile = itemView.findViewById(R.id.iv_profile);
48             ivProfile.setImageResource(komenList.get(i).getProfile());
49
50             itemView.setTag(komenList.get(i));
51         }
52
53         return itemView;
54     }
55 }
```

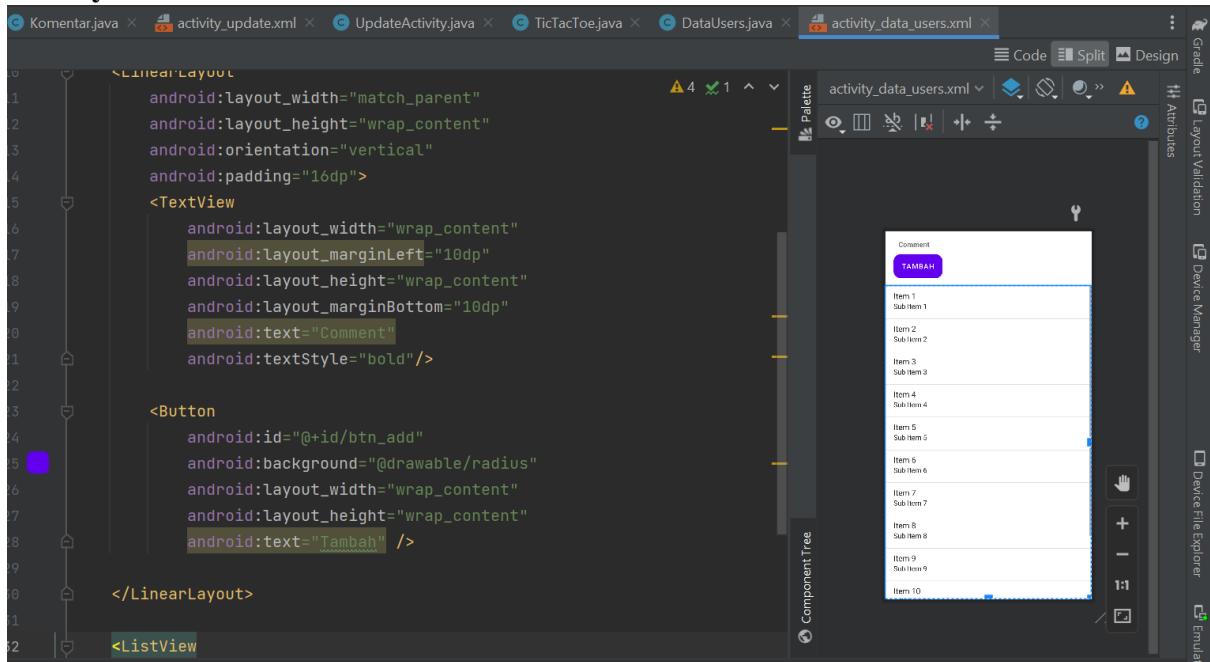
```

46         itemView = LayoutInflater.from(context)
47             .inflate(R.layout.item_data, viewGroup, attachToRoot: false);
48     }
49
50     ViewHolder viewHolder = new ViewHolder(itemView);
51
52     Komentar data = (Komentar) getItem(i);
53     viewHolder.bind(data);
54     return itemView;
55 }
56
57 private class ViewHolder {
58     private TextView txtKomen, txtName;
59
60     @ViewInject
61     ViewHolder(View view) {
62         txtName = view.findViewById(R.id.txt_nama);
63         txtKomen = view.findViewById(R.id.txt_nim);
64     }
65     void bind(Komentar data) {
66         txtName.setText(data.getNama());
67         txtKomen.setText(data.getKomen());
68     }
69 }
70 }

```

Base Adapter merupakan implementasi umum untuk adaptor yang dapat digunakan pada ListView. Kelas ini berguna untuk mempermudah menampilkan data menggunakan ListView.

activity_data_users.xml



Selanjutnya membuat activity baru dan diberi nama DataUsers.java. Dalam file .xml letakkan 1 Button untuk melakukan penambahan data dan ListView untuk menempatkan list dari data.

DataUsers.java

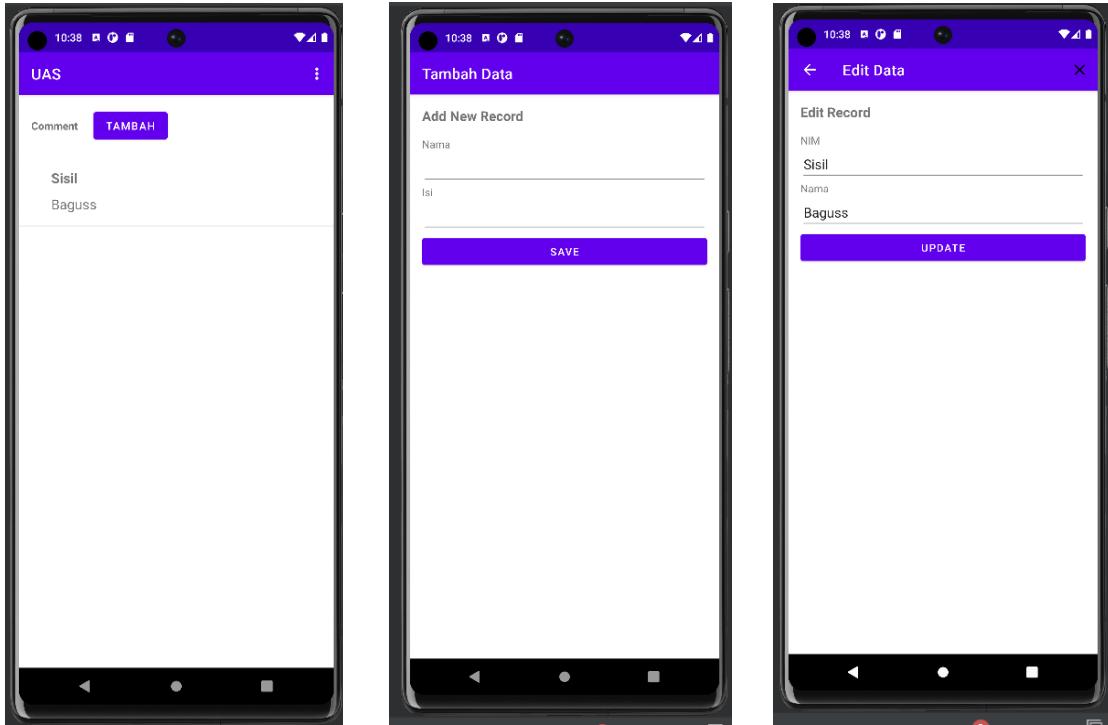
```
39     @Override
40     protected void onCreate(Bundle savedInstanceState) {
41         super.onCreate(savedInstanceState);
42         setContentView(R.layout.activity_data_users);
43
44         dbKomen = FirebaseDatabase.getInstance().getReference("comment");
45
46         listView = findViewById(R.id.lv_list);
47         btnAdd = findViewById(R.id.btn_add);
48         btnAdd.setOnClickListener(this);
49
50         //list mahasiswa
51         komenList = new ArrayList<>();
52
53         //kode yang ditambahkan
54         listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
55             @Override
56             public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
57                 Intent intent = new Intent(packageContext: DataUsers.this, UpdateActivity.class);
58                 intent.putExtra(UpdateActivity.EXTRA_MAHASISWA, komenList.get(i));
59
60                 startActivity(intent);
61             }
62         });
63     }
64 }
```

Pertama deklarasi View yang ada di file .xml dan kelas KomentarAdapter. Jika salah satu list dipilih, maka akan berpindah *activity* ke *UpdateActivity*.

```
66     @Override
67     protected void onStart() {
68         super.onStart();
69
70         dbKomen.addValueEventListener(new ValueEventListener() {
71             @Override
72             public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
73                 komenList.clear();
74
75                 for (DataSnapshot komentarSnapshot : dataSnapshot.getChildren()) {
76                     Komentar komentar = komentarSnapshot.getValue(Komentar.class);
77                     komenList.add(komentar);
78                 }
79
80                 KomentarAdapter adapter = new KomentarAdapter(context: DataUsers.this);
81                 adapter.setKomentarList(komenList);
82                 listView.setAdapter(adapter);
83             }
84
85             @Override
86             public void onCancelled(@NonNull DatabaseError databaseError) {
87                 Toast.makeText(context: DataUsers.this, text: "Terjadi kesalahan.", Toast.LENGTH_SHORT).show();
88             }
89         });
90     }
91 }
```

Code diatas untuk mengambil value dari database dengan memanggil kelas Komentar dan kelas KomentarAdapter. Kemudian value tersebut akan ditambahkan ke dalam listView.

Emulator



Apabila emulator dijalankan, maka akan menampilkan tampilan seperti gambar pertama, kemudian apabila tombol tambah diklik maka akan berpindah activity seperti gambar ke 2. Kemudian apabila salah satu item diklik, akan berpindah ke Update Activity seperti gambar 3.

Link Drive : <https://drive.google.com/drive/folders/1q5FeGkMvlxhvZGp-xRoNdTdlV-Yeffe1>