

# **LAPORAN PRAKTIKUM APLIKASI MOBILE**

( Shared Preferences )



Disusun oleh

Nama : Lois Yemima Sari

NIM/Golongan : E31191447/ C

Jember , 30 - Mei - 2021  
Disetujui

---

**Jurusan Teknologi Informasi  
Politeknik Negeri Jember  
2021**

**BAB 1. PENDAHULUAN**

## 1.1 Latar Belakang

sekarang mobile phone telah mengalami perubahan, bisa digunakan untuk berbagai hal seperti bermain game, akses internet, menyimpan data, dan berbagai aplikasi lainnya. Dalam perkembangan mobile phone akan memberikan dampak yang ditimbulkan salah satu dampak positif dari perkembangan mobile phone adalah banyaknya diciptakannya aplikasi-aplikasi yang dapat membantu dan memudahkan dalam pembelajaran dibidang Teknik Telekomunikasi salah satunya Rangkaian Digital.

Pada Android, terdapat beberapa opsi penyimpanan (*storage*). Salah satunya adalah *Shared Preferences*. *Shared Preferences* adalah salah satu library untuk jenis data storage ,yang penyimpanannya menggunakan bentuk tipe data berpasangan key- value pairs (kunci nilai berpasangan ), yang berisi tipe data primitif, yang biasanya di simpan dalam mode private.

## 1.2 Tujuan

- a. Mahasiswa dapat memahami konsep dasar dari Shared preference
- b. Mahasiswa dapat mengimplementasikan shared preference pada pemrograman berbasis android

## 1.3 Manfaat

Laporan ini dibuat supaya dapat memahami tentang Shared preference, memahami mulai dari macam-macam Shared preference, pengertian, kegunaannya sampai pada penerapannya.



## BAB 2. TEORI

### Shared preference

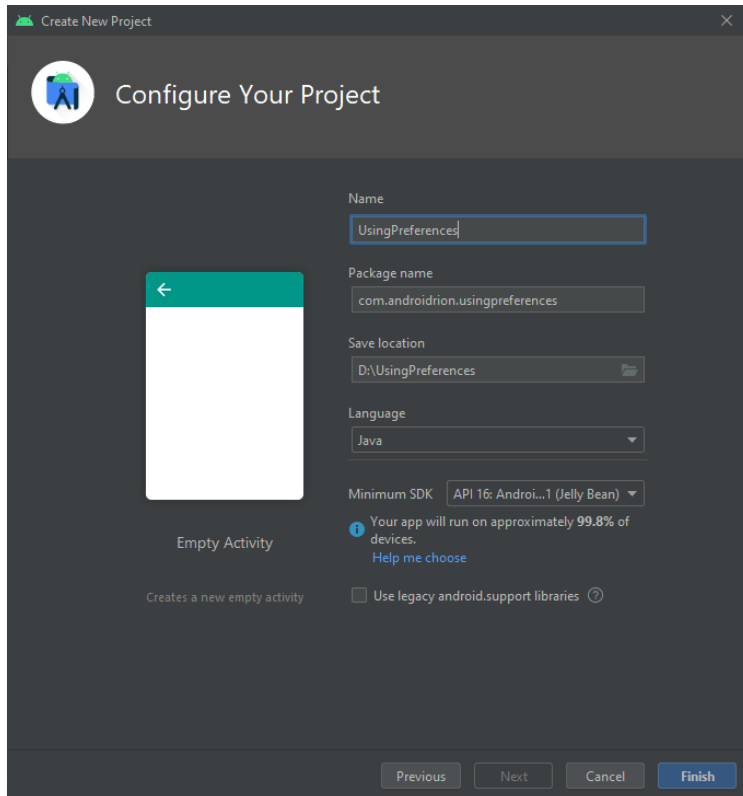
Pada Android, terdapat beberapa opsi penyimpanan (*storage*). Salah satunya adalah *Shared Preferences*. *Shared Preferences* adalah salah satu library untuk jenis data storage ,yang penyimpanannya menggunakan bentuk tipe data berpasangan key- value pairs (kunci nilai berpasangan ), yang berisi tipe data primitif, yang biasanya di simpan dalam mode private. Karakter dari penyimpanan (*storage*) ini adalah *key-value storage*, sehingga hanya bisa menyimpan data dengan tipe sederhana seperti *String*, *Double* atau *float*, *Integer*, *Boolean*, dan *Long*. *Shared Preferences* menyimpan data melalui penggunaan pasangan nama / nilai. Misalnya, tentukan nama untuk data yang ingin disimpan, dan kemudian keduanya beserta nilainya akan disimpan secara otomatis ke dalam file XML. Oleh karena itu, penyimpanan ini tergolong sangat minimalis dan sangat cocok untuk penyimpanan data yang sedikit.

Shared Preferences sering digunakan untuk cache , seperti menyimpan username dan password pada akun ,setting aplikasi, dan lain sebagainya. Yang pasti karena menggunakan tipe data primitif ,jadi ukuran tipe data relatif sangat kecil , serta ini tidak serumit saat kita membuat database SQLite. Jadi saat kita telah berhasil ,membuat data dengan menggunakan Shared Preferences , data tidak akan hilang meskipun kita telah keluar atau mendestroy Activity. Cara menghapusnya ,dengan meng-clear cache ,pada App info di Setting perangkat Android kalian, atau menguninstall Aplikasi kalian.

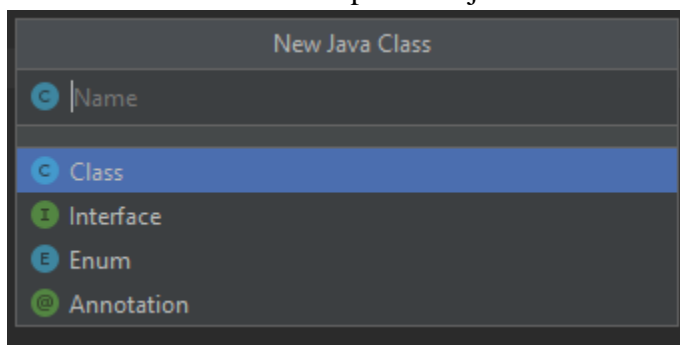
### BAB 3. HASIL dan PEMBAHASAN

Pada praktikum kali ini akan menggunakan Shared Preferences untuk pembuatan aplikasi login dan register sederhana

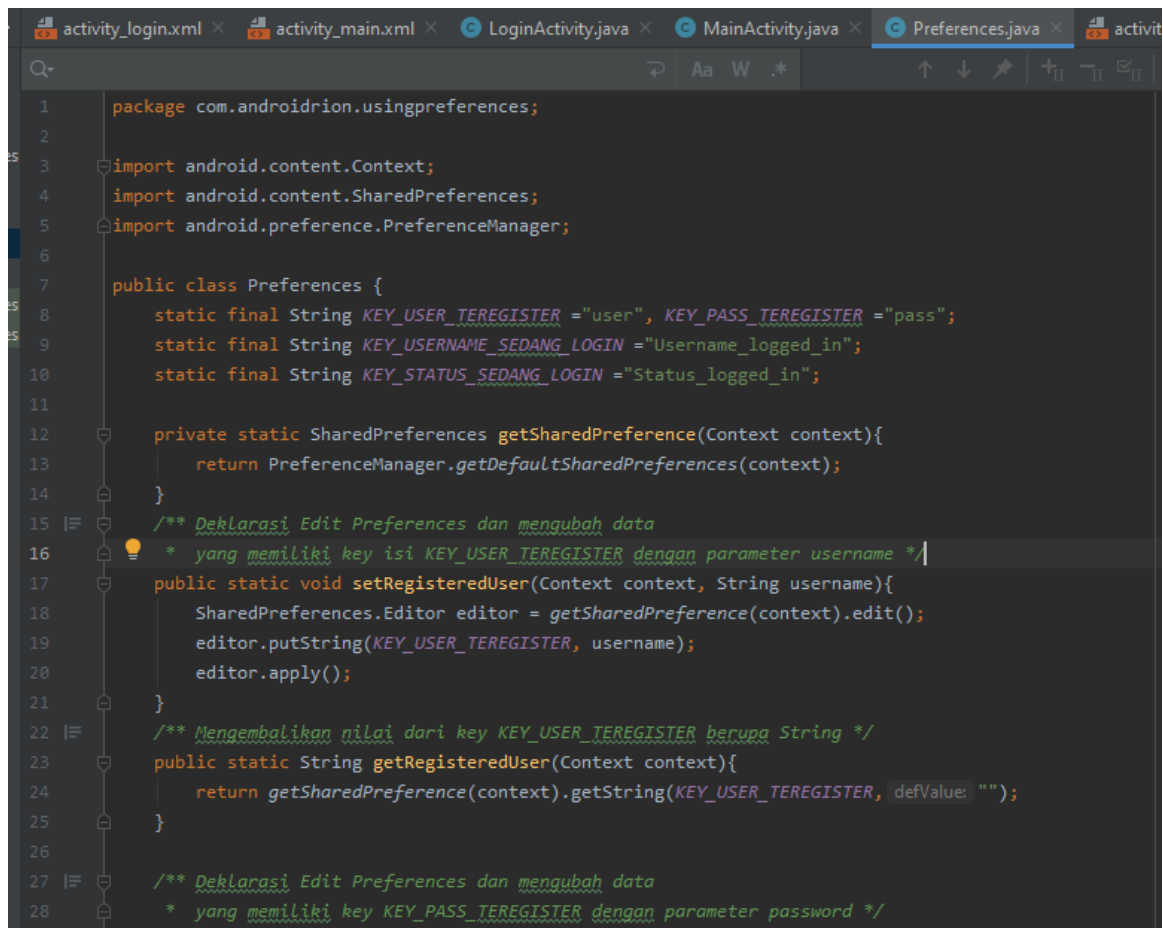
1. Buat project baru pada Android studio dengan nama UsingPreference lalu klik finish



2. Selanjutnya kita akan menambahkan file dengan ekstensi java, dengan klik file/new/java class. Berikan nama Preferences pada file java tersebut lalu klik oke.



3. Selanjutnya kita akan menambahkan source code dari class Preferences.
- getSharedPreferences() digunakan untuk lebih dari satu file yang diidentifikasi berdasarkan nama, dimana diatur pada parameter pertama. Untuk menyimpan data ke dalam SharedPreferences dibutuhkan class Editor yang bisa didapatkan dari method edit() dari instansiasi variable SharedPreferences. Jangan lupa untuk memanggil fungsi apply() untuk menyimpan ke SharedPreferences. Dan lakukan pendeklarasian key yang akan digunakan.



```
1 package com.androidrion.usingpreferences;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.preference.PreferenceManager;
6
7 public class Preferences {
8     static final String KEY_USER_TEREGISTER ="user", KEY_PASS_TEREGISTER ="pass";
9     static final String KEY_USERNAME_SEDANG_LOGIN ="Username_logged_in";
10    static final String KEY_STATUS_SEDANG_LOGIN ="Status_logged_in";
11
12    private static SharedPreferences getSharedPreference(Context context){
13        return PreferenceManager.getDefaultSharedPreferences(context);
14    }
15
16    /** Deklarasi Edit Preferences dan mengubah data
17     * yang memiliki key isi KEY_USER_TEREGISTER dengan parameter username */
18    public static void setRegisteredUser(Context context, String username){
19        SharedPreferences.Editor editor = getSharedPreference(context).edit();
20        editor.putString(KEY_USER_TEREGISTER, username);
21        editor.apply();
22    }
23
24    /** Mengembalikan nilai dari key KEY_USER_TEREGISTER berupa String */
25    public static String getRegisteredUser(Context context){
26        return getSharedPreference(context).getString(KEY_USER_TEREGISTER, defValue: "");
27    }
28
29    /** Deklarasi Edit Preferences dan mengubah data
30     * yang memiliki key KEY_PASS_TEREGISTER dengan parameter password */
```

```

29     public static void setRegisteredPass(Context context, String password){
30         SharedPreferences.Editor editor = getSharedPreference(context).edit();
31         editor.putString(KEY_PASS_TEREGISTER, password);
32         editor.apply();
33     }
34     /** Mengembalikan nilai dari key KEY_PASS_TEREGISTER berupa String */
35     public static String getRegisteredPass(Context context){
36         return getSharedPreference(context).getString(KEY_PASS_TEREGISTER, defValue: "");
37     }
38
39     /** Deklarasi Edit Preferences dan mengubah data
40      * yang memiliki key KEY_USERNAME_SEDANG_LOGIN dengan parameter username */
41     public static void setLoggedInUser(Context context, String username){
42         SharedPreferences.Editor editor = getSharedPreference(context).edit();
43         editor.putString(KEY_USERNAME_SEDANG_LOGIN, username);
44         editor.apply();
45     }
46     /** Mengembalikan nilai dari key KEY_USERNAME_SEDANG_LOGIN berupa String */
47     public static String getLoggedInUser(Context context){
48         return getSharedPreference(context).getString(KEY_USERNAME_SEDANG_LOGIN, defValue: "");
49     }
50
51     /** Deklarasi Edit Preferences dan mengubah data
52      * yang memiliki key KEY_STATUS_SEDANG_LOGIN dengan parameter status */
53     public static void setLoggedInStatus(Context context, boolean status){
54         SharedPreferences.Editor editor = getSharedPreference(context).edit();
55         editor.putBoolean(KEY_STATUS_SEDANG_LOGIN, status);
56         editor.apply();
57     }
58     /** Mengembalikan nilai dari key KEY_STATUS_SEDANG_LOGIN berupa boolean */
59     public static boolean getLoggedInStatus(Context context){
60         return getSharedPreference(context).getBoolean(KEY_STATUS_SEDANG_LOGIN, defValue: false);
61     }
62
63     /** Deklarasi Edit Preferences dan menghapus data, sehingga menjadikannya bernilai default
64      * khusus data yang memiliki key KEY_USERNAME_SEDANG_LOGIN dan KEY_STATUS_SEDANG_LOGIN */
65     public static void clearLoggedInUser (Context context){
66         SharedPreferences.Editor editor = getSharedPreference(context).edit();
67         editor.remove(KEY_USERNAME_SEDANG_LOGIN);
68         editor.remove(KEY_STATUS_SEDANG_LOGIN);
69         editor.apply();
70     }
71 }
72

```

4. Dalam class preferences terdapat mutator yang berguna untuk pendeklarasian SharedPreferences di activity yang berbeda-beda dan setiap kali menggunakan SharedPreferences harus melalui mutator.

```

12     private static SharedPreferences getSharedPreference(Context context){
13         return PreferenceManager.getDefaultSharedPreferences(context);
14     }

```

5. Selanjutnya kita akan membuat form login dan registrasi dengan meng klik file/new/activity/empty activity lalu masukkan nama LoginActivity dan RegisterActivity. Form ini berguna sebagai tampilan untuk melakukan login, registrasi dan main activity sebagai halaman utama.

### 1. Activity\_Login.xml

```
activity_login.xml activity_main.xml LoginActivity.java MainActivity.java
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:gravity="center"
7   android:padding="18dp"
8   android:orientation="vertical"
9   tools:context=".LoginActivity">
10  <EditText
11    android:id="@+id/et_emailSignIn"
12    android:layout_width="match_parent"
13    android:layout_height="wrap_content"
14    android:hint="Username"
15    android:inputType="textEmailAddress"
16    android:maxLines="1"
17    android:singleLine="true"
18    android:imeOptions="actionNext"/>
19  <EditText
20    android:id="@+id/et_passwordSignIn"
21    android:layout_width="match_parent"
22    android:layout_height="wrap_content"
23    android:layout_marginBottom="30dp"
24    android:hint="Password"
25    android:inputType="textPassword"
26    android:maxLines="1"
27    android:singleLine="true"
28    android:imeActionId="6"
29    android:imeActionLabel="Sign In"
30    android:imeOptions="actionDone" />
31  <LinearLayout
32    android:layout_width="match_parent"
33    android:layout_height="wrap_content">
34    <Button
35      android:id="@+id/button_signupSignIn"
36      android:layout_width="0dp"
37      android:layout_height="wrap_content"
38      android:text="Sign Up"
39      android:layout_weight="1"/>
40    <Button
41      android:id="@+id/button_signinSignIn"
42      android:layout_width="0dp"
43      android:layout_height="wrap_content"
44      android:text="Sign in"
45      android:layout_weight="1"/>
46  </LinearLayout>
47 </LinearLayout>
```

### 2. Activity\_register.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:tools="http://schemas.android.com/tools"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:gravity="center"
7   android:padding="18dp"
8   android:orientation="vertical"
9   tools:context=".RegisterActivity">
10  <EditText
11    android:id="@+id/et_emailSignup"
12    android:layout_width="match_parent"
13    android:layout_height="wrap_content"
14    android:hint="Username"
15    android:inputType="textEmailAddress"
16    android:maxLines="1"
17    android:singleLine="true"
18    android:imeOptions="actionNext"/>
19  <EditText
20    android:id="@+id/et_passwordSignup"
21    android:layout_width="match_parent"
22    android:layout_height="wrap_content"
23    android:hint="Password"
24    android:inputType="textPassword"
25    android:maxLines="1"
26    android:singleLine="true"
27    android:imeOptions="actionNext"/>
28  <EditText
29    android:id="@+id/et_passwordSignup2"
30    android:layout_width="match_parent"
31    android:layout_height="wrap_content"
32    android:layout_marginBottom="30dp"
33    android:hint="Confirm Password"
34    android:inputType="textPassword"
35    android:maxLines="1"
36    android:singleLine="true"
37    android:imeActionLabel="Sign Up"
38    android:imeOptions="actionDone" />
39  <Button
40    android:id="@+id/button_signupSignup"
41    android:layout_width="match_parent"
42    android:layout_height="wrap_content"
43    android:text="Sign Up" />
44 </LinearLayout>
```

### 3. Activity\_main.xml



```

1  <?xml version="1.0" encoding="utf-8"?>
2  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:padding="30dp"
7      tools:context=".MainActivity">
8      <LinearLayout
9          android:layout_width="match_parent"
10         android:layout_height="wrap_content"
11         android:orientation="vertical">
12         <TextView
13             android:layout_width="wrap_content"
14             android:layout_height="wrap_content"
15             android:textSize="20sp"
16             android:text="Masuk sebagai"/>
17         <TextView
18             android:id="@+id/tv_namaMain"
19             android:layout_width="wrap_content"
20             android:layout_height="wrap_content"
21             android:textSize="30sp"
22             tools:text="Namaku adalah"/>
23     </LinearLayout>
24     <Button
25         android:id="@+id/button_logoutMain"
26         android:layout_width="match_parent"
27         android:layout_height="wrap_content"
28         android:layout_alignParentBottom="true"
29         android:layout_marginTop="10dp"
30         android:text="Log Out" />
31 </RelativeLayout>

```

- Selanjutnya kita akan menambahkan sourcecode ke LoginActivity.java, untuk menambahkan proses login yang akan digunakan untuk form login dengan cara memasukkan username dan password yang sudah terdaftar.

```

1  package com.androidtrion.usingpreferences;
2
3  import androidx.appcompat.app.AppCompatActivity;
4
5  import android.os.Bundle;
6  import android.content.Intent;
7  import android.text.TextUtils;
8  import android.view.KeyEvent;
9  import android.view.View;
10 import android.view.inputmethod.EditorInfo;
11 import android.widget.EditText;
12 import android.widget.TextView;
13
14
15 public class LoginActivity extends AppCompatActivity {
16
17
18     private EditText mViewUser, mViewPassword;
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         setContentView(R.layout.activity_login);
24
25         /* Menginisialisasi variable dengan Form User dan Form Password dari Layout LoginActivity */
26         mViewUser=findViewById(R.id.et_emailSignin);
27         mViewPassword=findViewById(R.id.et_passwordSignin);
28         /* Menjalankan Method razia() Jika tombol SignIn di keyboard di sentuh */
29         mViewPassword.setOnEditorActionListener(new TextView.OnEditorActionListener() {
30             @Override

```

```

31 public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
32     if (actionId == EditorInfo.IME_ACTION_DONE || actionId == EditorInfo.IME_NULL) {
33         razia();
34         return true;
35     }
36     return false;
37 }
38 });
39
40 /* Menjalankan Method razia() jika merasakan tombol SignIn disentuh */
41 findViewById(R.id.button_signinSignin).setOnClickListener(new View.OnClickListener() {
42     @Override
43     public void onClick(View v) {
44         razia();
45     }
46 });
47
48 /* Ke RegisterActivity jika merasakan tombol SignUp disentuh */
49 findViewById(R.id.button_signupSignin).setOnClickListener(new View.OnClickListener() {
50     @Override
51     public void onClick(View v) {
52         startActivity(new Intent(getBaseContext(), RegisterActivity.class));
53     }
54 });
55 }
56
57 /** ke MainActivity jika data Status Login dari Data Preferences bernilai true */
58 @Override
59 protected void onStart() {
60     super.onStart();
61     if (Preferences.getLoggedInStatus(getBaseContext())){
62         startActivity(new Intent(getBaseContext(), MainActivity.class));
63         finish();
64     }
65 }
66
67 /** Men-check inputan User dan Password dan Memberikan akses ke MainActivity */
68 private void razia(){
69     /* Mereset semua Error dan fokus menjadi default */
70     mViewUser.setError(null);
71     mViewPassword.setError(null);
72     View fokus = null;
73     boolean cancel = false;
74
75     /* Mengambil text dari form User dan form Password dengan variable baru bertipe String*/
76     String user = mViewUser.getText().toString();
77     String password = mViewPassword.getText().toString();
78
79     /* Jika form user kosong atau TIDAK memenuhi kriteria di Method cekUser() maka, Set error
80      * di Form User dengan menset variable fokus dan error di Viewnya juga cancel menjadi true*/
81     if (TextUtils.isEmpty(user)){
82         mViewUser.setError("This field is required");
83         fokus = mViewUser;
84         cancel = true;
85     }else if (!cekUser(user)){
86         mViewUser.setError("This Username is not found");
87         fokus = mViewUser;
88         cancel = true;
89     }
90
91     /* Sama syarat percabangannya dengan User seperti di atas. Bedanya ini untuk Form Password*/

```

```

91         if (TextUtils.isEmpty(password)){
92             mViewPassword.setError("This field is required");
93             fokus = mViewPassword;
94             cancel = true;
95         }else if (!cekPassword(password)){
96             mViewPassword.setError("This password is incorrect");
97             fokus = mViewPassword;
98             cancel = true;
99         }
100
101         /* Jika cancel true, variable fokus mendapatkan fokus */
102         if (cancel) fokus.requestFocus();
103         else masuk();
104     }
105
106     /** Menuju ke MainActivity dan Set User dan Status sedang Login, di Preferences */
107     private void masuk(){
108         Preferences.setLoggedInUser(getBaseContext(),Preferences.getRegisteredUser(getBaseContext()));
109         Preferences.setLoggedInStatus(getBaseContext(), status: true);
110         startActivity(new Intent(getBaseContext(),MainActivity.class));finish();
111     }
112
113     /** True jika parameter password sama dengan data password yang terdaftar dari Preferences */
114     @ private boolean cekPassword(String password){
115         return password.equals(Preferences.getRegisteredPass(getBaseContext()));
116     }
117
118     /** True jika parameter user sama dengan data user yang terdaftar dari Preferences */
119     @ private boolean cekUser(String user){
120         return user.equals(Preferences.getRegisteredUser(getBaseContext()));
121     }
122 }

```

- Menambahkan sourcecode pada RegisterActivity.java untuk melakukan proses pendaftaran yang nantinya digunakan untuk melakukan proses pada form register dengan memasukkan username, password dan verifikasi password.

```

1  package com.androidrion.usingpreferences;
2
3  import androidx.appcompat.app.AppCompatActivity;
4  import android.os.Bundle;
5  import android.text.TextUtils;
6  import android.view.KeyEvent;
7  import android.view.View;
8  import android.view.inputmethod.EditorInfo;
9  import android.widget.EditText;
10 import android.widget.TextView;
11
12 public class RegisterActivity extends AppCompatActivity {
13
14     private EditText mViewUser, mViewPassword, mViewRepassword;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_register);
20
21         /* Menginisialisasi variable dengan Form User, Form Password, dan Form Repassword
22         dari Layout RegisterActivity */
23         mViewUser = findViewById(R.id.et_emailSignup);
24         mViewPassword = findViewById(R.id.et_passwordSignup);
25         mViewRepassword = findViewById(R.id.et_passwordSignup2);
26
27         /* Menjalankan Method razia() jika merasakan tombol SignUp di keyboard disentuh */
28         mViewRepassword.setOnEditorActionListener(new TextView.OnEditorActionListener() {
29             @Override
30             public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {

```

```

31         if (actionId == EditorInfo.IME_ACTION_DONE || actionId == EditorInfo.IME_NULL) {
32             razia();
33             return true;
34         }
35         return false;
36     }
37 });
38 /* Menjalankan Method razia() jika merasakan tombol SignUp disentuh */
39 findViewById(R.id.button_signupSignup).setOnClickListener(new View.OnClickListener() {
40     @Override
41     public void onClick(View v) {
42         razia();
43     }
44 });
45 }
46
47 /** Men-check inputan User dan Password dan Memberikan akses ke MainActivity */
48 private void razia(){
49     /* Meneset semua Error dan fokus menjadi default */
50     mViewUser.setError(null);
51     mViewPassword.setError(null);
52     mViewRepassword.setError(null);
53     View fokus = null;
54     boolean cancel = false;
55
56     /* Mengambil text dari Form User, Password, Repassword dengan variable baru bertipe String*/
57     String repassword = mViewRepassword.getText().toString();
58     String user = mViewUser.getText().toString();
59     String password = mViewPassword.getText().toString();
60
61     /* Jika form user kosong atau MEMENUHI kriteria di Method cekUser() maka, Set error di Form-
62     * User dengan menset variable fokus dan error di Viewnya juga cancel menjadi true*/
63     if (TextUtils.isEmpty(user)){
64         mViewUser.setError("This field is required");
65         fokus = mViewUser;
66         cancel = true;
67     }else if(cekUser(user)){
68         mViewUser.setError("This Username is already exist");
69         fokus = mViewUser;
70         cancel = true;
71     }
72
73     /* Jika form password kosong dan MEMENUHI kriteria di Method cekPassword maka,
74     * Reaksinya sama dengan percabangan User di atas. Bedanya untuk Password dan Repassword*/
75     if (TextUtils.isEmpty(password)){
76         mViewPassword.setError("This field is required");
77         fokus = mViewPassword;
78         cancel = true;
79     }else if (!cekPassword(password, repassword)){
80         mViewRepassword.setError("This password is incorrect");
81         fokus = mViewRepassword;
82         cancel = true;
83     }
84
85     /** Jika cancel true, variable fokus mendapatkan fokus. Jika false, maka
86     * Kembali ke LoginActivity dan Set User dan Password untuk data yang terdaftar */
87     if (cancel){
88         fokus.requestFocus();
89     }else{
90         Preferences.setRegisteredUser(getBaseContext(), user);

```

```

91         Preferences.setRegisteredPass(getBaseContext(),password);
92         finish();
93     }
94 }
95
96 /** True jika parameter password sama dengan parameter repassword */
97 @ private boolean cekPassword(String password, String repassword){
98     return password.equals(repassword);
99 }
100
101 /** True jika parameter user sama dengan data user yang terdaftar dari Preferences */
102 @ private boolean cekUser(String user){
103     return user.equals(Preferences.getRegisteredUser(getBaseContext()));
104 }
105 }

```

8. Menambahkan sourcecode pada MainActivity.java sebagai proses form activity\_main.xml, halaman utama yang menampilkan bahwa akun berhasil melakukan login dan pada halaman ini user dapat melakukan logout.

```

1  package com.androidrion.usingpreferences;
2
3
4  import androidx.appcompat.app.AppCompatActivity;
5  import android.content.Intent;
6  import android.os.Bundle;
7  import android.view.View;
8  import android.widget.TextView;
9
10 public class MainActivity extends AppCompatActivity {
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         /* Deklarasi dan Menginisialisasi variable nama dengan Label Nama dari Layout MainActivity */
18         TextView nama = findViewById(R.id.tv_namaMain);
19
20         /* Men-set Label Nama dengan data User sedang login dari Preferences */
21         nama.setText(Preferences.getLoggedInUser(getBaseContext()));
22
23         /* Men-set Status dan User yang sedang login menjadi default atau kosong di
24          * Data Preferences. Kemudian menuju ke LoginActivity*/
25         findViewById(R.id.button_logoutMain).setOnClickListener(new View.OnClickListener() {
26             @Override
27             public void onClick(View v) {
28                 //Menghapus Status login dan kembali ke Login Activity
29                 Preferences.clearLoggedInUser(getBaseContext());
30                 startActivity(new Intent(getBaseContext(),LoginActivity.class));
31                 finish();
32             }
33         });
34     }
35
36 }

```

**ALAMAT GITHUB**

[https://github.com/loisyemima/E31191447\\_Lois-Yemima-Sari\\_C.git](https://github.com/loisyemima/E31191447_Lois-Yemima-Sari_C.git)

## Hasil Tampilan

The image displays two side-by-side screenshots of a mobile application interface, both titled "UsingPreferences". The interface is set against a dark background. At the top, a status bar shows "4G" connectivity, signal strength, data speed (0.1K/s on the left and 2.6K/s on the right), time (16:42), and battery level (61%).

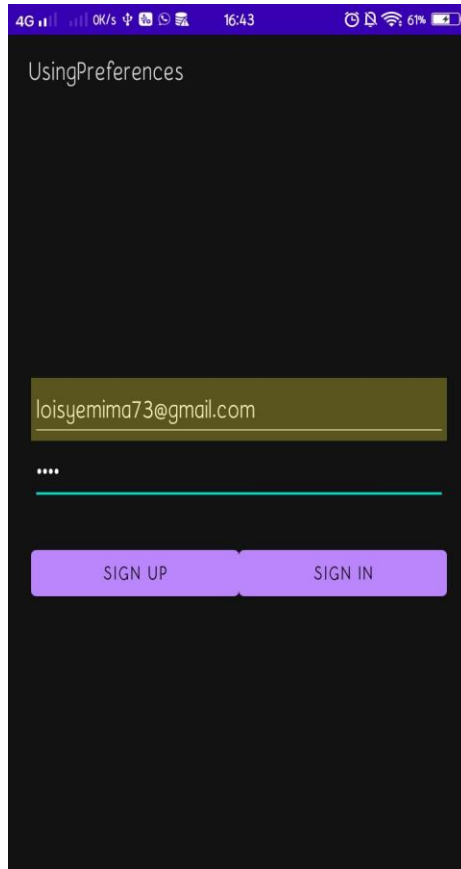
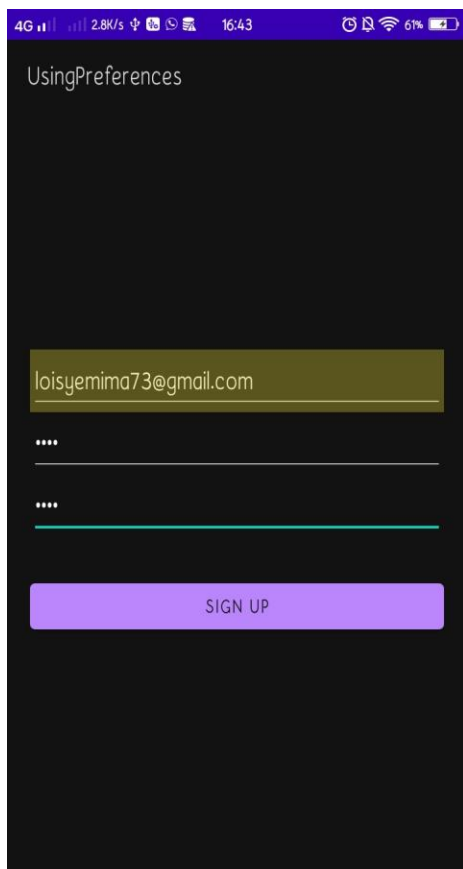
The main content area contains three input fields, each with a light gray label and a corresponding underline:

- Username**: The first input field.
- Password**: The second input field.
- Confirm Password**: The third input field.

Below the input fields, there are two distinct button layouts:

- Left Screenshot**: Features two separate, rounded rectangular buttons with a light blue background. The left button is labeled "SIGN UP" and the right button is labeled "SIGN IN".
- Right Screenshot**: Features a single, wider rounded rectangular button with a light blue background, labeled "SIGN UP".

At the bottom of each screenshot is a white navigation bar containing three standard Android icons: a hamburger menu (three horizontal lines), a home button (a house icon), and a back button (a curved arrow pointing left).







## **Bab 4. Kesimpulan**

Pada praktikum ini saya belajar mengenai penggunaan Shared Preference dalam pembuatan form login dan register .Data dalam SharedPreferences akan disimpan dalam sebuah file yang mengandung key-value secara berpasangan. Setiap file ini dikelola oleh Android framework yang aksesnya dapat diatur secara private atau dibagikan. Selain digunakan untuk penyimpanan data sederhana, pemanfaatan SharedPreferences digunakan juga untuk menyimpan pengaturan aplikasi. Dengan ini pengguna dapat mengatur aplikasi sesuai dengan keinginan.

## Daftar Pustaka

BKPM Aplikasi Mobile

Pengajar T. Login · Modul Pemrograman Mobile. Github.io. Published 2020.  
<https://polinema-mobile.github.io/08/02-login.html>