

Tutorial 4

Android Shared Preferences

Shared Preferences adalah penyimpanan key-value, Sehingga hanya bisa menyimpan data bertipe primitif (float, int, long, string, boolean). Penyimpanan ini tergolong sangat minimalis dan sangat cocok untuk penyimpanan data yang sedikit.

Dengan memanfaatkan Shared Preference, kita bisa melakukan login, logout dan register, yang membedakannya adalah, ketika kita menutup aplikasi, kita tidak akan logout secara otomatis kecuali kita melakukan aksi logout, hal ini dapat mempermudah seseorang dalam penggunaan aplikasi yang sering dibuka agar tidak perlu login berulang kali.

Apakah semua aplikasi perlu selalu dalam keadaan login sampai kita pilih logout?

Apakah aplikasi perlu diberi session untuk logout dalam rentang waktu tertentu? Agar apa?

Buatlah sebuah project baru dengan nama ApplicationPreferences, lalu buat sebuah java baru dengan nama **Preferences.java**, isikan kodingan berikut:

```
package com.example.tutorial4;

import ...

public class Preferences {
    /* Mendeklarasikan key-value berupa String, sebagai variabel penyimpan data,
    sehingga setiap data memiliki key yang berbeda satu sama lain */
    static final String KEY_USER_TEREGISTER = "user", KEY_PASS_TEREGISTER = "pass";
    static final String KEY_USERNAME_SEDANG_LOGIN = "Username_logged_in";
    static final String KEY_STATUS_SEDANG_LOGIN = "Status_logged_in";

    /* Mendeklarasikan Shared Preferences dengan parameter context */
    private static SharedPreferences getSharedPreferences(Context context){
        return PreferenceManager.getDefaultSharedPreferences(context);
    }

    /* Mendeklarasikan Edit Preferences dan mengubah data KEY_USER_TEREGISTER dengan parameter username */
    public static void setRegisteredUser(Context context, String username){
        SharedPreferences.Editor editor = getSharedPreferences(context).edit();
        editor.putString(KEY_USER_TEREGISTER, username);
        editor.apply();
    }

    /* Mengembalikan nilai dari key KEY_USER_TEREGISTER berupa String */
    public static String getRegisteredUser(Context context){
        return getSharedPreferences(context).getString(KEY_USER_TEREGISTER, s1: "");
    }

    /* Mendeklarasikan Edit Preferences KEY_PASS_TEREGISTER dengan parameter password */
    public static void setRegisteredPass(Context context, String password){
        SharedPreferences.Editor editor = getSharedPreferences(context).edit();
        editor.putString(KEY_PASS_TEREGISTER, password);
        editor.apply();
    }

    /* Mengembalikan nilai dari key KEY_PASS_TEREGISTER berupa String */
    public static String getRegisteredPass(Context context){
        return getSharedPreferences(context).getString(KEY_PASS_TEREGISTER, s1: "");
    }

    /* Mendeklarasikan Edit Preferences KEY_USERNAME_SEDANG_LOGIN dengan parameter username */
    public static void setLoggedInUser(Context context, String username){
        SharedPreferences.Editor editor = getSharedPreferences(context).edit();
        editor.putString(KEY_USERNAME_SEDANG_LOGIN, username);
        editor.apply();
    }

    /* Mengembalikan nilai dari key KEY_USERNAME_SEDANG_LOGIN berupa String */
    public static String getLoggedInUser(Context context){
        return getSharedPreferences(context).getString(KEY_USERNAME_SEDANG_LOGIN, s1: "");
    }

    /* Mendeklarasikan Edit Preferences KEY_STATUS_SEDANG_LOGIN dengan parameter status */
    public static void setLoggedInStatus(Context context, boolean status){
        SharedPreferences.Editor editor = getSharedPreferences(context).edit();
        editor.putBoolean(KEY_STATUS_SEDANG_LOGIN, status);
        editor.apply();
    }

    /* Mengembalikan nilai dari key KEY_STATUS_SEDANG_LOGIN berupa boolean */
    public static boolean getLoggedInStatus(Context context){
        return getSharedPreferences(context).getBoolean(KEY_STATUS_SEDANG_LOGIN, b: false);
    }

    /* Mendeklarasikan Edit Preferences dan menghapus data,
    sehingga menjadikannya bernilai default. Khusus untuk data yang memiliki key KEY_USERNAME_SEDANG_LOGIN
    dan KEY_STATUS_SEDANG_LOGIN */
    public static void clearLoggedInUser (Context context){
        SharedPreferences.Editor editor = getSharedPreferences(context).edit();
        editor.remove(KEY_USERNAME_SEDANG_LOGIN);
        editor.remove(KEY_STATUS_SEDANG_LOGIN);
        editor.apply();
    }
}
```

Selanjutnya buatlah sebuah Activity baru untuk menampilkan halaman login dan register, beri nama **ActivityLogin.java** dan **ActivityRegister.java** dengan baris kodingan berikut:

ActivityLogin.java

```
public class ActivityLogin extends AppCompatActivity {
    private EditText mViewUser, mViewPassword;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        /* Melakukan Inisialisasi variabel dengan nilai dari UI */
        mViewUser=findViewById(R.id.et_emailsignin);
        mViewPassword=findViewById(R.id.et_passwordsignin);
        /* Menjalankan fungsi razia() Jika tombol SignIn di keyboard di sentuh */
        mViewPassword.setOnEditorActionListener((v, actionId, event) -> {
            if (actionId == EditorInfo.IME_ACTION_DONE || actionId == EditorInfo.IME_NULL) {
                razia();
                return true;
            }
            return false;
        });
        /* Menjalankan fungsi razia() jika tombol SignIn disentuh */
        findViewById(R.id.button_signinsignin).setOnClickListener((v) -> {
            razia();
        });
        /* Ke RegisterActivity jika tombol SignUp disentuh */
        findViewById(R.id.button_signupsignin).setOnClickListener((v) -> {
            startActivity(new Intent(getApplicationContext(), ActivityRegister.class));
        });
    }

    /* ke MainActivity jika data Status Login dari Data Preferences bernilai true */
    @Override
    protected void onStart() {
        super.onStart();
        if (Preferences.getLoginStatus(getApplicationContext())) {
            startActivity(new Intent(getApplicationContext(), MainActivity.class));
            finish();
        }
    }
}
```

```

        /* Jika cancel true, variable fokus mendapatkan fokus */
        if (cancel) fokus.requestFocus();
        else masuk();
    }

    /* Menuju ke MainActivity dan Set User dan Status menjadi sedang login di Preferences */
    private void masuk() {
        Preferences.setLoggedInUser(getBaseContext(), Preferences.getRegisteredUser(getBaseContext()));
        Preferences.setLoggedInStatus(getBaseContext(), status: true);
        startActivity(new Intent(getBaseContext(), MainActivity.class)); finish();
    }

    /* True jika parameter password sama dengan data password yang terdaftar dari Preferences */
    private boolean cekPassword(String password) {
        return password.equals(Preferences.getRegisteredPass(getBaseContext()));
    }

    /* True jika parameter user sama dengan data user yang terdaftar dari Preferences */
    private boolean cekUser(String user) {
        return user.equals(Preferences.getRegisteredUser(getBaseContext()));
    }
}

/* Melakukan pengecekan inputan Username dan Password dan Memberikan akses ke MainActivity */
private void razia() {
    /* Mereset semua Error dan fokus menjadi default */
    mViewUser.setError(null);
    mViewPassword.setError(null);
    View fokus = null;
    boolean cancel = false;

    /* Mengambil nilai dari UI dengan variable baru bertipe String */
    String user = mViewUser.getText().toString();
    String password = mViewPassword.getText().toString();

    /* Pengecekan jika UI komponen username tidak diisi user */
    if (TextUtils.isEmpty(user)) {
        mViewUser.setError("This field is required");
        fokus = mViewUser;
        cancel = true;
    } else if (!cekUser(user)) {
        mViewUser.setError("This Username is not found");
        fokus = mViewUser;
        cancel = true;
    }

    /* Pengecekan jika UI komponen password tidak diisi user */
    if (TextUtils.isEmpty(password)) {
        mViewPassword.setError("This field is required");
        fokus = mViewPassword;
        cancel = true;
    } else if (!cekPassword(password)) {
        mViewPassword.setError("This password is incorrect");
        fokus = mViewPassword;
        cancel = true;
    }
}

```

ActivityRegister.java

```
public class ActivityRegister extends AppCompatActivity {
    private EditText mViewUser, mViewPassword, mViewRepassword;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);

        /* Melakukan Inisialisasi variable dari Layout RegisterActivity */
        mViewUser = findViewById(R.id.et_emailSignup);
        mViewPassword = findViewById(R.id.et_passwordSignup);
        mViewRepassword = findViewById(R.id.et_passwordSignup2);

        /* Menjalankan fungsi razia() jika tombol SignUp di disentuh */
        mViewRepassword.setOnEditorActionListener((v, actionId, event) → {
            if (actionId == EditorInfo.IME_ACTION_DONE || actionId == EditorInfo.IME_NULL) {
                razia();
                return true;
            }
            return false;
        });

        /* Menjalankan fungsi razia() jika tombol SignUp disentuh */
        findViewById(R.id.button_signupSignup).setOnClickListener((v) → {
            razia();
        });

        /** Melakukan pengecekan inputan User dan Password dan Memberikan akses ke MainActivity */
        private void razia() {
            /* Mereset semua Error dan fokus menjadi default */
            mViewUser.setError(null);
            mViewPassword.setError(null);
            mViewRepassword.setError(null);
            View fokus = null;
            boolean cancel = false;
        }
    }
}
```

```

/* Mengambil nilai dari UI komponen dengan variable baru bertipe String*/
String repassword = mViewRepassword.getText().toString();
String user = mViewUser.getText().toString();
String password = mViewPassword.getText().toString();

/* Pengecekan requirement username */
if (TextUtils.isEmpty(user)) {
    mViewUser.setError("This field is required");
    fokus = mViewUser;
    cancel = true;
}else if (cekUser(user)) {
    mViewUser.setError("This Username is already exist");
    fokus = mViewUser;
    cancel = true;
}

/* Pengecekan requirement password */
if (TextUtils.isEmpty(password)) {
    mViewPassword.setError("This field is required");
    fokus = mViewPassword;
    cancel = true;
}else if (!cekPassword(password, repassword)) {
    mViewRepassword.setError("This password is incorrect");
    fokus = mViewRepassword;
    cancel = true;
}

/* Jika cancel true, variable fokus mendapatkan fokus. Jika false, kembali ke LoginActivity
dan Set User dan Password untuk data yang terdaftar */
if (cancel) {
    fokus.requestFocus();
}else{
    Preferences.setRegisteredUser(getBaseContext(), user);
    Preferences.setRegisteredPass(getBaseContext(), password);
    finish();
}

```

```

    }
}

/* True jika parameter password sama dengan parameter repassword */
private boolean cekPassword(String password, String repassword) {
    return password.equals(repassword);
}

/* True jika parameter user sama dengan data user yang terdaftar dari Preferences */
private boolean cekUser(String user) {
    return user.equals(Preferences.getRegisteredUser(getBaseContext()));
}
}

```

Berikutnya buatlah dua buah layout dengan nama **activity_login.xml** dan **activity_register.xml** dengan baris kodingan berikut:

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:padding="18dp"
    android:orientation="vertical"
    tools:context=".ActivityLogin">
    <EditText
        android:id="@+id/et_emailSignin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="textEmailAddress"
        android:maxLines="1"
        android:singleLine="true"
        android:imeOptions="actionNext"/>
    <EditText
        android:id="@+id/et_passwordSignin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="30dp"
        android:hint="Password"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true"
        android:imeActionId="6"
        android:imeActionLabel="Sign In"
        android:imeOptions="actionDone" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <Button
            android:id="@+id/button_signupSignin"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Sign Up"
            android:layout_weight="1"/>
        <Button
            android:id="@+id/button_signinSignin"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:text="Sign In"
            android:layout_weight="1"/>
    </LinearLayout>
</LinearLayout>
```

activity_register.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:padding="18dp"
    android:orientation="vertical"
    tools:context=".ActivityRegister">
    <EditText
        android:id="@+id/et_emailSignup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username"
        android:inputType="textEmailAddress"
        android:maxLines="1"
        android:singleLine="true"
        android:imeOptions="actionNext"/>
    <EditText
        android:id="@+id/et_passwordSignup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true"
        android:imeOptions="actionNext"/>
    <EditText
        android:id="@+id/et_passwordSignup2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="30dp"
        android:hint="Confirm Password"
        android:inputType="textPassword"
        android:maxLines="1"
        android:singleLine="true"
        android:imeActionLabel="Sign Up"
        android:imeOptions="actionDone" />
    <Button
        android:id="@+id/button_signupSignup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sign Up" />
</LinearLayout>
```

Pada Class **ActivityMain.java**, lakukan modifikasi baris program sebagai berikut:


```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        /* Mendeklarasikan dan Melakukan Inisialisasi variable nama dengan Label Nama dari Layout MainActivity */
        TextView nama = findViewById(R.id.tv_namaMain);

        /* Mengatur nilai Label Nama dengan data User sedang login dari Preferences */
        nama.setText(Preferences.getLoggedInUser(getBaseContext()));

        /* Mengatur Status dan User yang sedang login menjadi default stay kosong di data Preferences.
        Kemudian menuju ke LoginActivity*/
        findViewById(R.id.button_logoutMain).setOnClickListener((v) -> {
            //Menghapus Status login dan kembali ke Login Activity
            Preferences.clearLoggedInUser(getBaseContext());
            startActivity(new Intent(getBaseContext(), ActivityLogin.class));
            finish();
        });
    }
}
```

Setelahnya pada layout **activity_main.xml** rubahlah baris programnya menjadi berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="30dp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:text="Masuk sebagai"/>
        <TextView
            android:id="@+id/tv_namaMain"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="30sp"
            tools:text="Namaku adalah"/>
    </LinearLayout>
    <Button
        android:id="@+id/button_logoutMain"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_marginTop="10dp"
        android:text="Log Out" />
</RelativeLayout>
```

Terakhir kita daftarkan Activity untuk login dan register ke file manifest seperti berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tutorial4">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Tutorial4"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity" />
        <activity android:name=".ActivityLogin">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".ActivityRegister"/>
    </application>
</manifest>
```

Jalankan program!

