

**TUGAS**  
**WORKSHOP APLIKASI MOBILE**  
**SHARE PREFERENCE DAN SQLITE**



**DISUSUN OLEH:**  
**LIGA BAYU HERDIANTO**  
**E41190139**

**TEKNIK INFORMATIKA**  
**TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI JEMBER**  
**TAHUN 2021**

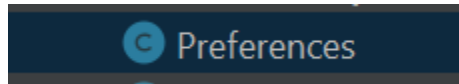
## Dasar Teori :

### a. Shared preference

Pada Android, terdapat beberapa opsi penyimpanan (*storage*). Salah satunya adalah *Shared Preferences*. 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.

## Kegiatan Praktikum :

- Buka project minggu 7 tentang SQLite yang sudah dibuat. Kemudian membuat file java untuk Share Preference nantinya. Tambahkan code sintak seperti dibawah ini.



File yang barusan dibuat dengan ekstensi .java

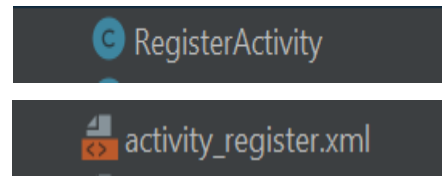
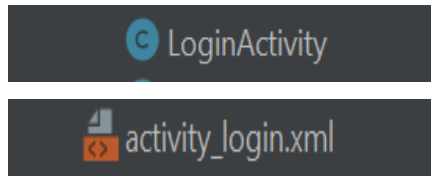
```
Preferences.java x
1 package com.example.minggu7_sqlite;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.preference.PreferenceManager;
6
7 public class Preferences {
8
9     /** Pendeklarasian key-data berupa String, untuk sebagai wadah penyimpanan data.
10      * Jadi setiap data mempunyai key yang berbeda satu sama lain */
11     static final String KEY_USER_TEREGISTER = "user", KEY_PASS_TEREGISTER = "pass";
12     static final String KEY_USERNAME_SEDANG_LOGIN = "Username_logged_in";
13     static final String KEY_STATUS_SEDANG_LOGIN = "Status_logged_in";
14
15     /** Pendeklarasian Shared Preferences yang berdasarkan paramater context */
16     private static SharedPreferences getSharedPreference(Context context){
17         return PreferenceManager.getDefaultSharedPreferences(context);
18     }
19
20     /** Deklarasi Edit Preferences dan mengubah data
21      * yang memiliki key isi KEY_USER_TEREGISTER dengan parameter username */
22     public static void setRegisteredUser(Context context, String username){
23         SharedPreferences.Editor editor = getSharedPreference(context).edit();
24         editor.putString(KEY_USER_TEREGISTER, username);
25         editor.apply();
26     }
27 }
```

```

27  /** Mengembalikan nilai dari key KEY_USER_TEREGISTER berupa String */
28  public static String getRegisteredUser(Context context){
29      return getSharedPreferences(context).getString(KEY_USER_TEREGISTER, defValue: "");
30  }
31
32  /** Deklarasi Edit Preferences dan mengubah data
33   * yang memiliki key KEY_PASS_TEREGISTER dengan parameter password */
34  public static void setRegisteredPass(Context context, String password){
35      SharedPreferences.Editor editor = getSharedPreferences(context).edit();
36      editor.putString(KEY_PASS_TEREGISTER, password);
37      editor.apply();
38  }
39
40  /** Mengembalikan nilai dari key KEY_PASS_TEREGISTER berupa String */
41  public static String getRegisteredPass(Context context){
42      return getSharedPreferences(context).getString(KEY_PASS_TEREGISTER, defValue: "");
43  }
44
45  /** Deklarasi Edit Preferences dan mengubah data
46   * yang memiliki key KEY_USERNAME_SEDANG_LOGIN dengan parameter username */
47  public static void setLoggedInUser(Context context, String username){
48      SharedPreferences.Editor editor = getSharedPreferences(context).edit();
49      editor.putString(KEY_USERNAME_SEDANG_LOGIN, username);
50      editor.apply();
51  }
52
53  /** Mengembalikan nilai dari key KEY_USERNAME_SEDANG_LOGIN berupa String */
54  public static String getLoggedInUser(Context context){
55      return getSharedPreferences(context).getString(KEY_USERNAME_SEDANG_LOGIN, defValue: "");
56  }
57
58  /** Deklarasi Edit Preferences dan mengubah data
59   * yang memiliki key KEY_STATUS_SEDANG_LOGIN dengan parameter status */
60  public static void setLoggedInStatus(Context context, boolean status){
61      SharedPreferences.Editor editor = getSharedPreferences(context).edit();
62      editor.putBoolean(KEY_STATUS_SEDANG_LOGIN, status);
63      editor.apply();
64  }
65
66  /** Mengembalikan nilai dari key KEY_STATUS_SEDANG_LOGIN berupa boolean */
67  public static boolean getLoggedInStatus(Context context){
68      return getSharedPreferences(context).getBoolean(KEY_STATUS_SEDANG_LOGIN, defValue: false);
69  }
70
71  /** Deklarasi Edit Preferences dan menghapus data, sehingga menjadikannya bernilai default
72   * khusus data yang memiliki key KEY_USERNAME_SEDANG_LOGIN dan KEY_STATUS_SEDANG_LOGIN */
73  public static void clearLoggedInUser (Context context){
74      SharedPreferences.Editor editor = getSharedPreferences(context).edit();
75      editor.remove(KEY_USERNAME_SEDANG_LOGIN);
76      editor.remove(KEY_STATUS_SEDANG_LOGIN);
77      editor.apply();
78  }

```

- Selanjutnya kita akan membuat form untuk melakukan login dan register. Buat new Activity dengan nama “ LoginActivity” dan “ RegisterActivity” jika sudah tekan finish.



- Jika sudah dibuat untuk activity login dan register maka selanjutnya menambahkan code sintak. Perhatikan gambar berikut.

➔ LoginActivity.java :

```

LoginActivity.java x
1  package com.example.minggu7_sqlite;
2
3  import androidx.appcompat.app.AppCompatActivity;
4
5  import android.content.Intent;
6  import android.os.Bundle;
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 public class LoginActivity extends AppCompatActivity {
15
16     private EditText mViewUser, mViewPassword;
17
18     @Override
19     protected void onCreate(Bundle savedInstanceState) {
20         super.onCreate(savedInstanceState);
21         setContentView(R.layout.activity_login);
22         /* Menginisialisasi variable dengan Form User dan Form Password dari Layout LoginActivity */
23         mViewUser=findViewById(R.id.et_emailSignin);
24         mViewPassword =findViewById(R.id.et_passwordSignin);
25         /* Menjalankan Method razia() jika tombol SignIn di keyboard di sentuh */
26         mViewPassword.setOnEditorActionListener(new TextView.OnEditorActionListener() {
27             @Override

```

```

28 public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {
29     if (actionId == EditorInfo.IME_ACTION_DONE || actionId == EditorInfo.IME_NULL) {
30         razia();
31         return true;
32     }
33     return false;
34 }
35 }
36
37 /* Menjalankan Method razia() jika merasakan tombol SignIn disentuh */
38 findViewById(R.id.button_signinSignin).setOnClickListener(new View.OnClickListener() {
39     @Override
40     public void onClick(View v) { razia(); }
41 });
42
43 /* Ke RegisterActivity jika merasakan tombol SignUp disentuh */
44 findViewById(R.id.button_signupSignin).setOnClickListener(new View.OnClickListener() {
45     @Override
46     public void onClick(View v) {
47         startActivity(new Intent(getBaseContext(), RegisterActivity.class));
48     }
49 });
50 }
51
52

```

```

52
53 /** ke MainActivity jika data Status Login dari Data Preferences bernilai true */
54 @Override
55 protected void onStart() {
56     super.onStart();
57     if (Preferences.getLoggedInStatus(getBaseContext())){
58         startActivity(new Intent(getBaseContext(), MainActivity.class));
59         finish();
60     }
61 }
62
63 /** Men-check inputan User dan Password dan Memberikan akses ke MainActivity */
64 private void razia(){
65     /* Mereset semua Error dan fokus menjadi default */
66     mViewUser.setError(null);
67     mViewPassword.setError(null);
68     View fokus = null;
69     boolean cancel = false;
70
71     /* Mengambil text dari form User dan form Password dengan variable baru bertipe String*/
72     String user = mViewUser.getText().toString();
73     String password = mViewPassword.getText().toString();
74
75     /* Jika form user kosong atau TIDAK memenuhi kriteria di Method cekUser() maka, Set error
76      * di Form User dengan menset variable fokus dan error di Viewnya juga cancel menjadi true*/
77     if (TextUtils.isEmpty(user)){

```

```

77     if (TextUtils.isEmpty(user)){
78         mViewUser.setError("This field is required");
79         fokus = mViewUser;
80         cancel = true;
81     }else if(!cekUser(user)){
82         mViewUser.setError("This Username is not found");
83         fokus = mViewUser;
84         cancel = true;
85     }
86
87     /* Sama syarat percabangannya dengan User seperti di atas. Bedanya ini untuk Form Password*/
88     if (TextUtils.isEmpty(password)){
89         mViewPassword.setError("This field is required");
90         fokus = mViewPassword;
91         cancel = true;
92     }else if(!cekPassword(password)){
93         mViewPassword.setError("This password is incorrect");
94         fokus = mViewPassword;
95         cancel = true;
96     }
97
98     /* Jika cancel true, variable fokus mendapatkan fokus */
99     if (cancel) fokus.requestFocus();
100    else masuk();
101 }

```

```

102
103 /** Menuju ke MainActivity dan Set User dan Status sedang login, di Preferences */
104 private void masuk(){
105     Preferences.setLoggedInUser(getBaseContext(),Preferences.getRegisteredUser(getBaseContext()));
106     Preferences.setLoggedInStatus(getBaseContext(), status: true);
107     startActivity(new Intent(getBaseContext(),MainActivity.class));finish();
108 }
109
110 /** True jika parameter password sama dengan data password yang terdaftar dari Preferences */
111 @ private boolean cekPassword(String password){
112     return password.equals(Preferences.getRegisteredPass(getBaseContext()));
113 }
114
115 /** True jika parameter user sama dengan data user yang terdaftar dari Preferences */
116 @ private boolean cekUser(String user){
117     return user.equals(Preferences.getRegisteredUser(getBaseContext()));
118 }
119 }

```

## ➔ RegisterActivity.java :

```
RegisterActivity.java x
1 package com.example.minggu7_sqlite;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.text.TextUtils;
7 import android.view.KeyEvent;
8 import android.view.View;
9 import android.view.inputmethod.EditorInfo;
10 import android.widget.EditText;
11 import android.widget.TextView;
12
13 public class RegisterActivity extends AppCompatActivity {
14     private EditText mViewUser, mViewPassword, mViewRepassword;
15
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_register);
21
22         /* Menginisialisasi variable dengan Form User, Form Password, dan Form Repassword
23         dari Layout RegisterActivity */
24         mViewUser = findViewById(R.id.et_emailSignup);
25         mViewPassword = findViewById(R.id.et_passwordSignup);
26         mViewRepassword = findViewById(R.id.et_passwordSignup2);
27
28         /* Menjalankan Method razia() jika merasakan tombol SignUp di keyboard disentuh */
29         mViewRepassword.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         /* Menjalankan Method razia() jika merasakan tombol SignUp disentuh */
40         findViewById(R.id.button_signupSignup).setOnClickListener(new View.OnClickListener() {
41             @Override
42             public void onClick(View v) { razia(); }
43         });
44     }
45
46
47     /** Men-check inputan User dan Password dan Memberikan akses ke MainActivity */
48     private void razia(){
49         /* Mereset semua Error dan fokus menjadi default */
50         mViewUser.setError(null);
51         mViewPassword.setError(null);
52         mViewRepassword.setError(null);
53     }
```



```

53     mViewRepassword.setError(null);
54     View fokus = null;
55     boolean cancel = false;
56
57     /* Mengambil text dari Form User, Password, Repassword dengan variable baru bertipe String*/
58     String repassword = mViewRepassword.getText().toString();
59     String user = mViewUser.getText().toString();
60     String password = mViewPassword.getText().toString();
61
62     /* Jika form user kosong atau MEMENUHI kriteria di Method cekUser() maka, Set error di Form-
63     * User dengan menset variable fokus dan error di Viewnya juga cancel menjadi true*/
64     if (TextUtils.isEmpty(user)){
65         mViewUser.setError("This field is required");
66         fokus = mViewUser;
67         cancel = true;
68     }else if(cekUser(user)){
69         mViewUser.setError("This Username is already exist");
70         fokus = mViewUser;
71         cancel = true;
72     }
73
74     /* Jika form password kosong dan MEMENUHI kriteria di Method cekPassword maka,
75     * Reaksinya sama dengan percabangan User di atas. Bedanya untuk Password dan Repassword*/
76     if (TextUtils.isEmpty(password)){
77         mViewPassword.setError("This field is required");

```

```

77         mViewPassword.setError("This field is required");
78         fokus = mViewPassword;
79         cancel = true;
80     }else if (!cekPassword(password, repassword)){
81         mViewRepassword.setError("This password is incorrect");
82         fokus = mViewRepassword;
83         cancel = true;
84     }
85
86     /** Jika cancel true, variable fokus mendapatkan fokus. Jika false, maka
87     * Kembali ke LoginActivity dan Set User dan Password untuk data yang terdaftar */
88     if (cancel){
89         fokus.requestFocus();
90     }else{
91         Preferences.setRegisteredUser(getBaseContext(), user);
92         Preferences.setRegisteredPass(getBaseContext(), password);
93         finish();
94     }
95 }
96
97 /** True jika parameter password sama dengan parameter repassword */
98 @ private boolean cekPassword(String password, String repassword){
99     return password.equals(repassword);
100 }
101

```

```

101
102 /** True jika parameter user sama dengan data user yang terdaftar dari Preferences */
103 @ private boolean cekUser(String user){
104     return user.equals(Preferences.getRegisteredUser(getBaseContext()));
105 }
106 }

```

- Sekarang kita akan menambahkan code sintak pada form layoutnya. Tambahkan pada activity\_login.xml dan juga activity\_register.xml.

➔ activity\_login.xml :

```
activity_login.xml x
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>
```

➔ activity\_register.xml :

```
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>
```

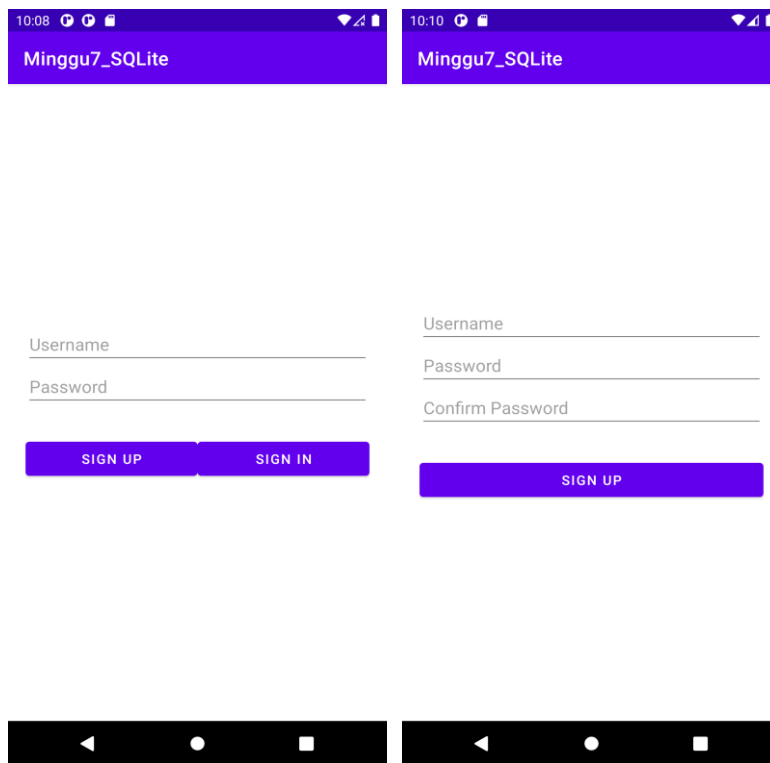
- Terakhir, jangan lupa untuk mengubah code pada file 'AndroidManifest.xml' untuk mendeklarasikan activity baru yang telah dibuat sebelumnya, dengan 'LoginActivity' sebagai activity utamanya.

```

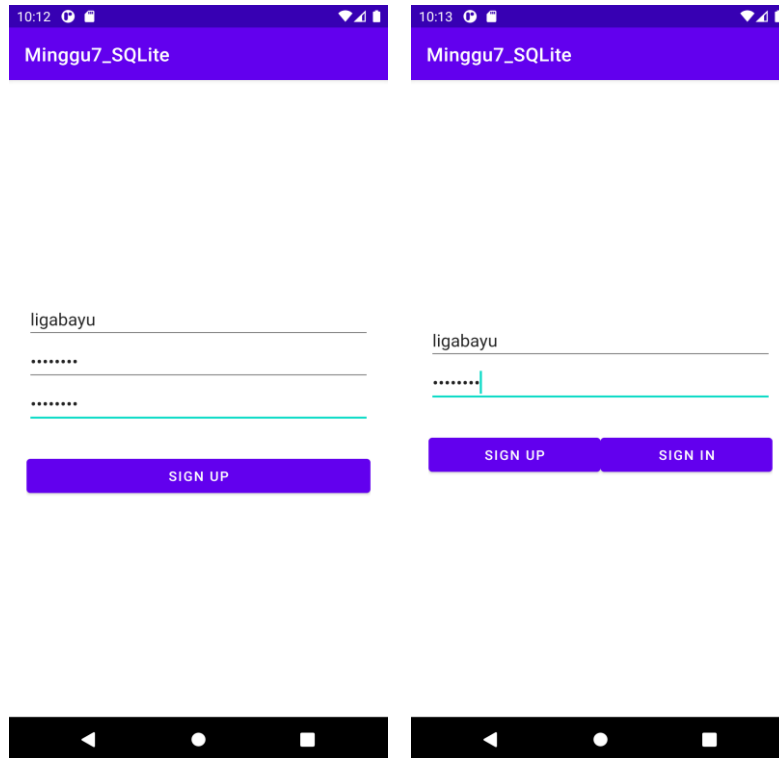
14      <activity android:name=".LoginActivity">
15          <intent-filter>
16              <action android:name="android.intent.action.MAIN" />
17
18              <category android:name="android.intent.category.LAUNCHER" />
19          </intent-filter>
20      </activity>

```

- Berikut tampilan hasil ketika di Run :



- Tampilan untuk form login. Ditampilkan bahwa jika kita sudah memiliki username dan password maka kita bisa langsung melakukan Sign In dengan akun yang sudah kita daftarkan sebelumnya. Tetapi jika kita belum pernah mendaftar maka kita bisa masuk form registrasi terlebih dahulu untuk membuat akun dengan memasukkan username dan password yang akan kita gunakan.



- Tampilan diatas merupakan contoh ketika kita akan melakukan registrasi dan melakukan login. Jika true atau berhasil maka akan masuk ke tampilan CRUD yang project sebelumnya.



- Pada menu CRUD ini kita bisa menambahkan data input baru seperti yang telah dicontohkan pada project sebelumnya.

The image displays three screenshots of an Android application interface. The first two screenshots show the 'Update Biodata' form with fields for Nomor, Nama, Tanggal Lahir, Jenis Kelamin, and Alamat. The third screenshot shows the 'Minggu7\_SQLite' screen with a list of data and a 'LOG OUT' button.

**Update Biodata (10:18):**

- Nomor:
- Nama:
- Tanggal Lahir:
- Jenis Kelamin:
- Alamat:

**Update Biodata (10:19):**

- Nomor: 2
- Nama: masligabayu
- Tanggal Lahir: 11-11-1996
- Jenis Kelamin: L
- Alamat: Jember

**Minggu7\_SQLite (10:20):**

- Darsiwani
- masligabayu

**Buttons:**

- ✓ SIMPAN
- < KEMBALI
- + BUAT BIODATA BARU
- LOG OUT

- Ketika data input sudah berhasil tersimpan maka kita bisa mengedit hasil input tersebut.

The image shows a screenshot of the 'Minggu7\_SQLite' screen. A modal menu titled 'Pilihan' is displayed, containing options: Lihat Biodata, Update Biodata, and Hapus Biodata. The background shows a list of data and a 'LOG OUT' button.

**Minggu7\_SQLite (10:21):**

- Darsiwani
- masligabayu

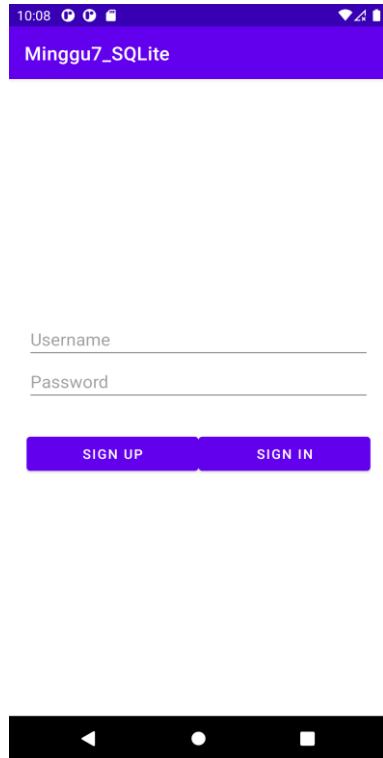
**Pilihan:**

- Lihat Biodata
- Update Biodata
- Hapus Biodata

**Buttons:**

- + BUAT BIODATA BARU
- LOG OUT

- Button Log Out digunakan untuk kembali ke menu Share Preference yaitu ke menu Login In tadi diawal.



The screenshot shows a mobile application interface with a white background. At the top, there is a status bar with the time '10:08' and various icons. Below the status bar is a blue header bar with the text 'Minggu7\_SQLite' in white. The main content area contains two text input fields: 'Username' and 'Password', both with light gray borders. Below these fields are two blue buttons with white text: 'SIGN UP' and 'SIGN IN'. At the bottom of the screen is a black navigation bar with three white icons: a back arrow, a home circle, and a recent apps square.