

LAPORAN PROGRESS PRAKTIKUM



NIM : 2003071

Nama : Hakim Asrori

Kelas : D3TI.2C

Mata Kuliah : [Pemrograman Perangkat Bergerak \(TIU3403\)](#)

Praktikum ke / Judul : 12 / SHARED PREFERENCES

Tanggal Praktikum : 12 Mei 2022

Dosen Pengampu : Fachrul Pralienka Bani Muhamad, S.ST.,
M.Kom

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI INDRAMAYU
2022**



A. Tujuan Praktikum

Tujuan Umum

Mahasiswa dapat mengimplementasikan shared preferences pada flutter untuk menyimpan state tertentu

Tujuan Khusus

Mahasiswa mampu:

1. Menjelaskan pengertian shared preferences
2. Melakukan pengaturan lingkungan (dependency) shared preferences
3. Menyimpan state suatu nilai dan tema dengan shared preferences
4. Menyimpan state login dengan shared preferences

B. Rangkuman Teori Singkat

SharedPreferences Pada Android

Shared Preferences merupakan salah satu opsi penyimpanan storage local pada android. Karakter penyimpanan ini adalah key-value storage, sehingga hanya bisa menyimpan tipe data primitif.

NuserDefault pada iOS

Dengan kelas NSUserDefaults dapat menyimpan pengaturan dan properti yang terkait dengan aplikasi atau data pengguna. Misalnya, Anda bisa menyimpan gambar profil yang ditetapkan oleh pengguna atau skema warna default untuk aplikasi. Objek akan disimpan dalam apa yang dikenal sebagai "sistem default" iOS. Sistem iOS default tersedia di seluruh kode di aplikasi Anda, dan semua data yang disimpan ke sistem default akan bertahan melalui sesi aplikasi.

Shared_preferences pada Flutter

Tidak jauh berbeda dengan pengertian SharedPreferences pada Android. Shared_preferences pada flutter merupakan salah satu package pendukung yang berfungsi sebagai penyimpanan storage local pada platform tersebut.



C. Langkah dan Hasil Pelaksanaan Praktikum

<https://www.youtube.com/watch?v=j7J9Kq0sqQI&t=15s>

Langkah-langkah praktikum Menyimpan Nilai dan Tema dengan Shared Preferences

Link Github = <https://github.com/hakim-asrori/Pemrograman-Perangkat-Bergerak/blob/main/Tugas%20Praktikum%2012%20Mei%202022/sharedpreferences/lib/main.dart>

1. Install shared_preferences dengan cara menambahkan baris berikut pada file pubspec.yaml

```
dependencies:  
  flutter:  
    sdk: flutter  
  
# The following adds the Cupertino Icons font to your application.  
# Use with the CupertinoIcons class for iOS style icons.  
cupertino_icons: ^1.0.2  
shared_preferences: ^2.0.15
```

2. Buat kode program seperti yang ada di video youtube

```
import 'dart:convert';  
  
import 'package:flutter/material.dart';  
import 'package:shared_preferences/shared_preferences.dart';  
  
void main() {  
  runApp(const MyApp());  
}  
  
class MyApp extends StatefulWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  State<MyApp> createState() => _MyAppState();  
}
```



```
class _MyAppState extends State<MyApp> {  
  int counter = 0;  
  bool isDark = false;  
  
  Future<void> setPreference() async {  
    final jembatan = await SharedPreferences.getInstance();  
  
    if (jembatan.containsKey('myData')) {  
      jembatan.clear();  
    }  
  
    final myData = json  
      .encode({'counter': counter.toString(), 'isDark': isDark.toString()});  
  
    jembatan.setString('myData', myData);  
    setState(() {});  
  }  
}
```

```
Future<void> getPreference() async {  
  final jembatan = await SharedPreferences.getInstance();  
  
  if (jembatan.containsKey('myData')) {  
    final myData =  
      json.decode(jembatan.getString('myData')!) as Map<String, dynamic>;  
  
    counter = int.parse(myData['counter']);  
    isDark = myData['isDark'] == "true" ? true : false;  
  }  
}  
  
void minus() {  
  counter = counter - 1;  
  
  setPreference();  
}
```



```
void add() {
    counter = counter + 1;

    setPreference();
}

void changeTheme() {
    isDark = !isDark;
    setPreference();
}

void refresh() {
    counter = 0;
    isDark = false;
    setPreference();
}

ThemeData dark = ThemeData(
    brightness: Brightness.dark,
    primaryColor: Colors.amber,
    primarySwatch: Colors.amber,
    accentColor: Colors.amber,
    outlinedButtonTheme: OutlinedButtonThemeData(
        style: OutlinedButton.styleFrom(
            primary: Colors.white,
            side: BorderSide(color: Colors.white, width: 2),
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(50))));

ThemeData light = ThemeData(
    brightness: Brightness.light,
    primaryColor: Colors.amber,
    primarySwatch: Colors.amber,
    accentColor: Colors.amber,
    outlinedButtonTheme: OutlinedButtonThemeData(
        style: OutlinedButton.styleFrom(
            primary: Colors.black,
            side: BorderSide(color: Colors.black, width: 2),
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(50))));
```



```
@override
Widget build(BuildContext context) {
  return FutureBuilder(
    future: getPreference(),
    builder: (context, _) => MaterialApp(
      debugShowCheckedModeBanner: false,
      theme: isDark ? dark : light,
      home: Scaffold(
        appBar: AppBar(
          title: Text("Counter Apps"),
          actions: [
            IconButton(onPressed: refresh, icon: Icon(Icons.refresh))
          ],
        ),
        body: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                "Angka saat ini adalah : $counter",
                style: TextStyle(fontSize: 25),
              ),
              SizedBox(
                height: 50,
              ),
              Row(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: [
                  OutlinedButton(
                    onPressed: minus, child: Icon(Icons.remove)),
                  OutlinedButton(onPressed: add, child: Icon(Icons.add))
                ],
              ),
            ],
          ),
        ),
        floatingActionButton: FloatingActionButton(
          onPressed: changeTheme,
          child: Icon(Icons.color_lens),
        ),
      ),
    ));
}
```

3. Jalankan program yang telah dibuat, maka hasilnya seperti berikut

3.1. Tema Terang



3.2. Tema Gelap





D. Langkah dan Hasil Latihan

Buatlah implementasi penggunaan shared preferences baik dengan login yang tidak menggunakan basis data maupun menggunakan basis data (pilih salah satu)

Link Github = https://github.com/hakim-asrori/Pemrograman-Perangkat-Bergerak/tree/main/Tugas%20Praktikum%2012%20Mei%202022/latihan_shared_preferences

1. Buat kode program pada main.dart

```
import 'package:flutter/material.dart';
import 'package:latihan_shared_preferences/views/admin/v_dashboard.dart';
import 'package:latihan_shared_preferences/views/auth/v_login.dart';
import 'package:shared_preferences/shared_preferences.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  bool? loggedIn;

  void getPrefs() async {
    var prefs = await SharedPreferences.getInstance();

    loggedIn = prefs.getBool('isLoggedIn');
  }

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(fontFamily: 'Lato'),
      debugShowCheckedModeBanner: false,
      home: (loggedIn == true) ? DashboardPage() : LoginPage(),
    );
  }
}
```

2. Buat kode program pada v_login.dart



```
import 'dart:ui';

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:latihan_shared_preferences/models/user.dart';
import 'package:latihan_shared_preferences/views/admin/v_dashboard.dart';
import 'package:latihan_shared_preferences/views/constants/custom_checkbox.dart';
import 'package:shared_preferences/shared_preferences.dart';

class LoginPage extends StatefulWidget {
  const LoginPage({Key? key}) : super(key: key);

  @override
  State<LoginPage> createState() => _LoginPageState();
}
```

```
class _LoginPageState extends State<LoginPage> {
  bool passwordVisible = false;
  final _formKey = GlobalKey<FormState>();
  TextEditingController _teleponController = TextEditingController();
  TextEditingController _passwordController = TextEditingController();

  void togglePassword() {
    setState(() {
      passwordVisible = !passwordVisible;
    });
  }

  Future<bool> _loginProses() async {
    final prefs = await SharedPreferences.getInstance();
    var response = await http
      .post(Uri.parse('http://api.rtq-freelance.my.id/api-v1/login'), body: {
        'no_hp': _teleponController.text,
        'password': _passwordController.text,
      });

    if (response.statusCode == 200) {
      var user = userFromJson(response.body);

      prefs.setString('nama', user.nama.toString());
      prefs.setBool('isLoggedIn', true);

      return true;
    } else {
      return false;
    }
  }
}
```



```
@override
Widget build(BuildContext context) {
  return Scaffold(
    resizeToAvoidBottomInset: false,
    backgroundColor: Colors.white,
    body: SafeArea(
      child: Padding(
        padding: EdgeInsets.fromLTRB(24.0, 24.0, 24.0, 0),
        child: Column(
          crossAxisAlignment: CrossAxisAlignment.start,
          children: [
            Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                Text(
                  'Login to your \naccount',
                  style: TextStyle(
                    fontSize: 24,
                    fontWeight: FontWeight.w700,
                    color: Color(0xff222222)),
                ),
                SizedBox(
                  height: 20,
                ),
                Image.asset(
                  'assets/images/accent.png',
                  width: 99,
                  height: 4,
                )
              ],
            ),
            SizedBox(
              height: 48,
            ),
          ],
        ),
      ),
    ),
  );
}
```



```
Form(  
  key: _formKey,  
  child: Column(  
    children: [  
      Container(  
        decoration: BoxDecoration(  
          color: Color(0xffff1f1f5),  
          borderRadius: BorderRadius.circular(14.0)),  
        child: TextFormField(  
          controller: _teleponController,  
          decoration: InputDecoration(  
            hintText: 'Telepon',  
            hintStyle: TextStyle(  
              fontSize: 16,  
              fontWeight: FontWeight.w600,  
              color: Color(0xff94959b)),  
            border: OutlineInputBorder(  
              borderSide: BorderSide.none)),  
          validator: (value) {  
            if (value == null || value.isEmpty) {  
              return "Harap masukan telepon!";  
            }  
            return null;  
          },  
        ),  
      ),  
      SizedBox(  
        height: 32,  
      ),  
    ],  
  ),  
)
```



Nama : Hakim Asrori

```
Container(
    decoration: BoxDecoration(
        color: Color(0xfffff1f5),
        borderRadius: BorderRadius.circular(14.0)),
    child: TextFormField(
        controller: _passwordController,
        obscureText: !passwordVisible,
        decoration: InputDecoration(
            hintText: 'Password',
            hintStyle: TextStyle(
                fontSize: 16,
                fontWeight: FontWeight.w600,
                color: Color(0xff94959b),
            ),
            suffixIcon: IconButton(
                onPressed: togglePassword,
                icon: Icon(passwordVisible
                    ? Icons.visibility_outlined
                    : Icons.visibility_off_outlined)),
            border:
                OutlineInputBorder(borderSide: BorderSide.none),
        ),
        validator: (value) {
            if (value == null || value.isEmpty) {
                return "Harap masukan password!";
            }
            return null;
        },
    ),
),
],
)),
 SizedBox(
    height: 32,
),
```



Nama : Hakim Asrori

3. Buat kode program pada v_dashboard.dart



```
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;
import 'package:latihan_shared_preferences/models/detail_user.dart';
import 'package:latihan_shared_preferences/models/user.dart';
import 'package:latihan_shared_preferences/views/auth/v_login.dart';
import 'package:shared_preferences/shared_preferences.dart';

class DashboardPage extends StatefulWidget {
  const DashboardPage({Key? key}) : super(key: key);

  @override
  State<DashboardPage> createState() => _DashboardPageState();
}
```



```
class _DashboardPageState extends State<DashboardPage> {
  String? _nama;

  void getPrefs() async {
    var prefs = await SharedPreferences.getInstance();
    setState(() {
      _nama = prefs.getString('nama');
    });
  }

  void logout() async {
    var prefs = await SharedPreferences.getInstance();

    setState(() {
      prefs.remove('nama');
    });

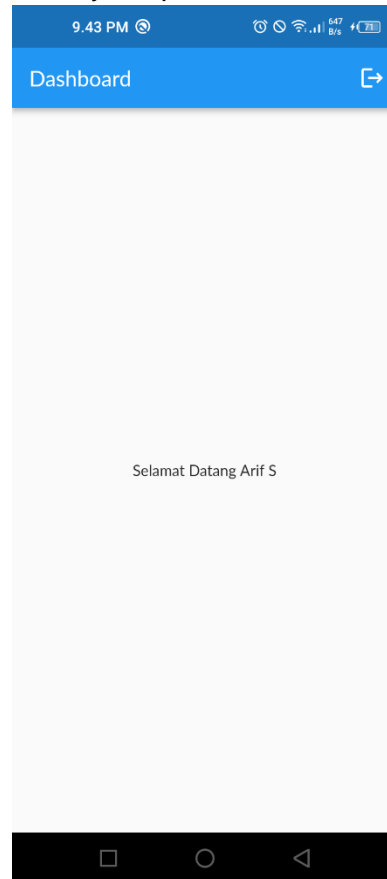
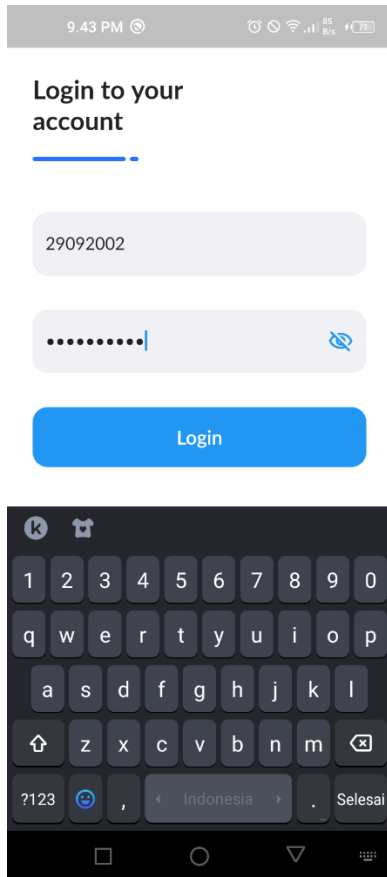
    Navigator.pushAndRemoveUntil(
      context,
      MaterialPageRoute(
        builder: (BuildContext context) => LoginPage(),
      ),
      (route) => false,
    );
  }

  @override
  void initState() {
    getPrefs();
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Dashboard"),
        automaticallyImplyLeading: false,
        actions: [IconButton(onPressed: logout, icon: Icon(Icons.logout))],
      ),
      body: Center(
        child: Text("Selamat Datang ${_nama}"),
      ),
    );
  }
}
```



4. Running kode program, maka hasil nya seperti berikut





Program Studi D3 Teknik Informatika
Jurusan Teknik Informatika
Politeknik Negeri Indramayu

NIM : 2003071

Nama : Hakim Asrori

E. Hasil dan Penjelasan Tugas



F. Kesimpulan

Prinsip SOLID ini sangat direkomendasikan karena dengan prinsip ini kita dapat merawat kode lebih muda, dapat dimengerti dan fleksibel. Dapat membantu programmer dalam menghindari bad code, membantu dalam refactoring kode program serta mengembangkan aplikasi secara Agile atau Adaptive.

G. Referensi

- Alberto Miola. "Flutter Complete Reference Create Beautiful, Fast and Native Apps for Any Device". Independently Published. 2020.
- Simone Alessandria, Brian Kayfirz. "Flutter Cookbook: Over 100 proven techniques and solutions for app development with Flutter 2.2 and Dart". Packt Publishing. Birmingham - Mumbai. 2021.
- Dieter Meiller. "Modern App Development with Dart and Flutter 2: A Comprehensive Introduction to Flutter". Walter de Gruyter GmbH. Berlin - Boston. 2021.
- Priyanka Tyagi. "Pragmatic Flutter: Building Cross-Platform Mobile Apps for Android, iOS, Web & Desktop". CRC Press Taylor & Francis Group, LLC. London - New York. 2022.
- Fikry Farenza. "Tutorial Penggunaan Shared Preferences pada Android". <https://blog.dot.co.id/tutorial-penggunaan-shared-preferences-pada-android-eddc300d7509>. Diakses pada 18 Mei 2022.
- Jeroen Van Rijn. "iOS SDK: Bekerja dengan NSUserDefaults". <https://code.tutsplus.com/id/tutorials/ios-sdk-working-with-nsuserdefaults--mobile-6039>. Diakses pada 18 Mei 2022.