

Modul 2

Listening to a Stream of Data

Using `shared_preferences` to read and write app data.

Module Overview

Mengenal metode penyimpanan data `SharedPreferences` pada aplikasi pada flutter dan bagaimana cara menerapkannya.

Module Objectives

Setelah mempelajari dan mempraktikkan modul ini, mahasiswa diharapkan dapat:

- Mengetahui apa yang dimaksud penyimpanan data sementara
- Menerapkan `SharedPreferences` sesuai dengan penggunaannya

Ada beberapa cara untuk menyimpan data ke perangkat mobile Anda, seperti penyimpanan data ke dalam file, atau menggunakan database lokal, seperti `SQLite`, atau menggunakan `SharedPreferences`. `Shared_preferences` merupakan media penyimpanan data yang paling sederhana dan dapat menampung data dalam

pasangan key/value. Akan tetapi, SharedPreferences tidak boleh digunakan untuk data penting karena data yang disimpan tidak dienkripsi, dan penulisan tidak selalu dijamin tersedia.

Saat menggunakan Flutter, Anda dapat memanfaatkan pustaka ***shared_preferences*** sehingga Anda dapat menyimpan data sederhana baik di iOS dan Android tanpa berurusan dengan spesifikasi kedua sistem operasi. Data akan selalu disimpan ke disk secara asinkron saat Anda menggunakan `shared_preferences`, sehingga anda memerlukan fungsi ***Future***, untuk menangani SharedPreferences. Tetapi, anda hanya dapat menyimpan data primitif: `int`, `double`, `bool`, `String`, `listString` dan `char`. SharedPreferences tidak dirancang untuk menyimpan banyak data, tetapi, dapat digunakan untuk memberikan pengalaman lebih di dalam aplikasi yang anda bangun.

Mengapa menggunakan SharedPreferences di Flutter?

Misalkan Anda ingin menyimpan nilai (seperti sebuah nilai ***flag***) yang ingin Anda gunakan nantinya ketika pengguna meluncurkan aplikasi, untuk menandakan status dari pengguna terakhir. Maka anda dapat menggunakan SharedPreferences. Data SharedPreferences disimpan di dalam aplikasi, jadi, saat pengguna mencopot (*uninstall*) aplikasi Anda, data juga akan secara otomatis dihapus dari perangkat anda. Kami juga akan melihat metode lain yang berbeda untuk menangani data di pertemuan-pertemuan selanjutnya.

SharedPreferences

Pada contoh ini, Kita akan mencoba menyertakan SharedPreferences dalam proyek kita. Jadi, di file pubspec.yaml / pubspec assist anda, tambahkan dependensi, sebagai berikut:

```
shared_preferences: ^2.0.15
```

kita juga akan menyertakan widget untuk menampilkan angka dengan menggunakan flutter_spinbox. SpinBox pada Flutter adalah widget input numerik untuk memasukkan nilai tertentu, dan menyediakan tombol untuk penyesuaian nilai dengan cepat, nyaman, dan akurat. Oleh karena itu, tambahkan dependensi di file pubspec.yaml / pubspec assist anda, sebagai berikut:

```
flutter_spinbox: ^0.10.0
```

Kita juga akan mencoba penyimpanan gambar menggunakan SharedPreferences. Untuk memudahkan anda dalam mengambil gambar, gunakanlah image_picker dengan menambahkan dependensi di file pubspec.yaml / pubspec assist anda, sebagai berikut:

```
image_picker: ^0.8.5+3
```

Sebagai contoh awal, kita akan merancang tampilan untuk menampilkan gambar dan spinbox, tanpa menggunakan SharedPreferences. Rancanglah tampilan Aplikasi anda dengan tampilan Front-End berikut:

Bagian Front-End

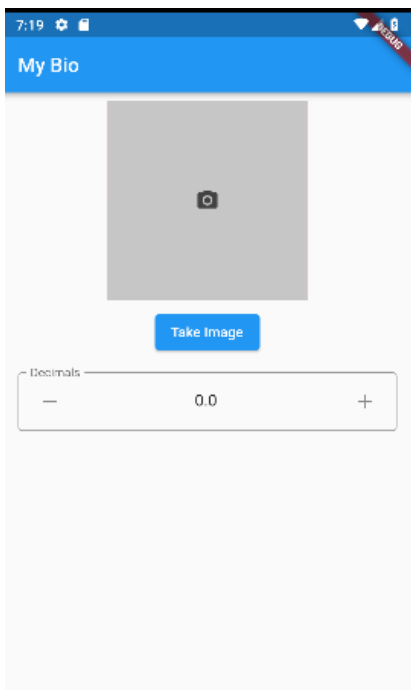
```
1 import 'dart:io';
2 import 'package:flutter/material.dart';
3 import 'package:flutter_spinbox/flutter_spinbox.dart';
4 import 'package:image_picker/image_picker.dart';
5
6 class MyBio extends StatefulWidget {
7   const MyBio({Key? key}) : super(key: key);
8
9   @override
10  State<MyBio> createState() => _MyBioState();
11 }
12
13 class _MyBioState extends State<MyBio> {
14   String? _image;
15   double _score = 0;
16   final ImagePicker _picker = ImagePicker();
17   @override
18   Widget build(BuildContext context) {
19     return Scaffold(
20       appBar: AppBar(title: const Text("My Bio")),
21       body: Padding(
22         padding: const EdgeInsets.all(8.0),
23         child: Center(
24           child: Column(children: [
25             Container(
26               width: 200,
27               height: 200,
28               decoration: BoxDecoration(color: Colors.red[200]),
29               child: _image != null
30                 ? Image.file(
31                     File(_image!),
32                     width: 200.0,
33                     height: 200.0,
34                     fit: BoxFit.fitHeight,
35                 )
36             : Container(
37               decoration: BoxDecoration(
38                 color: Color.fromARGB(255, 198, 198, 198)),
39               width: 200,
40               height: 200,
41               child: Icon(
42                 Icons.camera_alt,
43                 color: Colors.grey[800],
44               ),
45             ),
46           ],
```

```

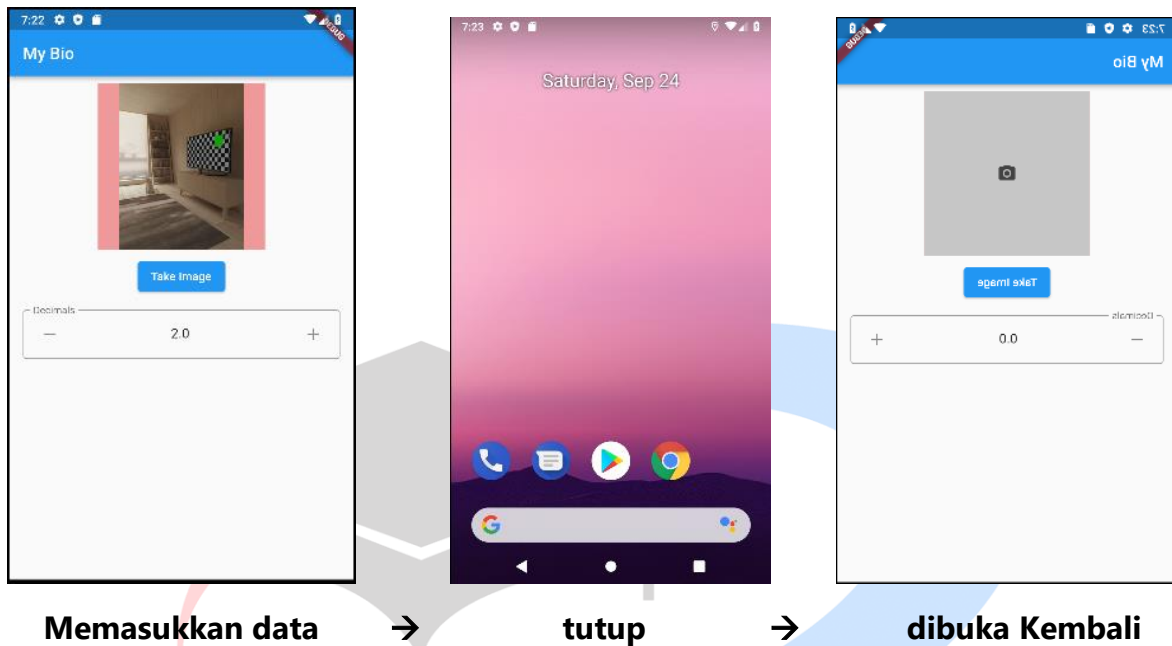
47         Padding(
48           padding: const EdgeInsets.all(8.0),
49           child: ElevatedButton(
50             onPressed: () async {
51               XFile? image =
52                 await _picker.pickImage(source: ImageSource.gallery);
53               setState(() {
54                 if (image != null) {
55                   _image = image.path;
56                 }
57               });
58             },
59             child: Text("Take Image")),
60         ),
61         Padding(
62           padding: const EdgeInsets.all(8.0),
63           child: SpinBox(
64             max: 10.0,
65             min: 0.0,
66             value: _score,
67             decimals: 1,
68             step: 0.1,
69             decoration: InputDecoration(labelText: 'Decimals'),
70           ),
71       ),
72     ),
73   ),
74 ),
75 );
76 }
77 }
78

```

Hasil :



Adapun hasil saat ini adalah, ketika anda menutup aplikasi, maka setiap data yang telah anda masukkan akan menghilang dan harus dimasukkan kembali.



Untuk menyelesaikan masalah tersebut, kita dapat menggunakan SharedPreferences. Tahapan dalam memanfaatkan SharedPreferences adalah:

1. Menginisialisasi SharedPreferences. Pastikan anda menginisialisasi SharedPreferences pada Bagian initState() siklus pertama dari aplikasi.

```
@override
void initState() {
  super.initState();
  loadData();
}
```

Pada contoh sebelumnya, kita memerlukan 2 data yang harus disimpan ke dalam SharedPreferences. Oleh karena itu, pastikan anda menyediakan kunci unik yang digunakan untuk menyimpan kedua data tersebut. Adapun proses untuk inisialisasi SharedPreferences adalah sebagai berikut:

```
15 class _MyBioState extends State<MyBio> {
16   String? _image;
17   double _score = 0;
18   final ImagePicker _picker = ImagePicker();
19   final String _keyScore = 'score';
20   final String _keyImage = 'image';
21   late SharedPreferences prefs;
22
23   void loadData() async {
24     prefs = await SharedPreferences.getInstance();
25     setState(() {
26       _score = (prefs.getDouble(_keyScore) ?? 0);
27       _image = prefs.getString(_keyImage);
28     });
29   }
30
31   @override
32   void initState() {
33     super.initState();
34     loadData();
35   }
```

Proses dalam loadData adalah menginisialisasi SharedPreferences, dan langsung membaca data di dalam SharedPreferences ketika ditemukan data.

2. Melakukan proses baca dan tulis ke dalam SharedPreferences untuk data score

```
95 Future<void> _setScore(double value) async {
96   prefs = await SharedPreferences.getInstance();
97   setState(() {
98     prefs.setDouble(_keyScore, value);
99     _score = ((prefs.getDouble(_keyScore) ?? 0));
100   });
101 }
```

Pastikan saat bekerja dengan SharedPreferences, harus dilakukan dengan cara asinkron, dengan menggunakan Future. Fungsi ini akan mengakses SharedPreferences kembali, melakukan penulisan data baru menggunakan fungsi prefs.setDouble sesuai tipe data dari score, dan memperbaharui nilai

di dalam aplikasi dengan mengubah nilai `_score` dengan data yang baru di simpan ke dalam `SharedPreferences`.

3. Melakukan proses baca dan tulis ke dalam `SharedPreferences` untuk data image

```

103 Future<void> _setImage(String? value) async {
104     prefs = await SharedPreferences.getInstance();
105     if (value != null)
106     {
107         setState(() {
108             prefs.setString(keyImage, value);
109             image = prefs.getString(keyImage);
110         });
111     }

```

Sama dengan proses menyimpan score. Fungsi ini akan mengakses `SharedPreferences` kembali, melakukan penulisan data baru menggunakan fungsi `prefs.setString` sesuai tipe data dari `_image`, dan memperbaharui nilai di dalam aplikasi dengan mengubah nilai `_image` dengan data yang baru di simpan ke dalam `SharedPreferences`.

4. Untuk menjalankan fungsi `_setScore`, anda dapat mengatur fungsi `onChange` pada widget `SpinBox`, dengan fungsi `_setScore`

```

77 Padding(
78     padding: const EdgeInsets.all(8.0),
79     child: SpinBox(
80         max: 10.0,
81         min: 0.0,
82         value: _score,
83         decimals: 1,
84         step: 0.1,
85         decoration: InputDecoration(labelText: 'Decimals'),
86         onChanged: _setScore,

```

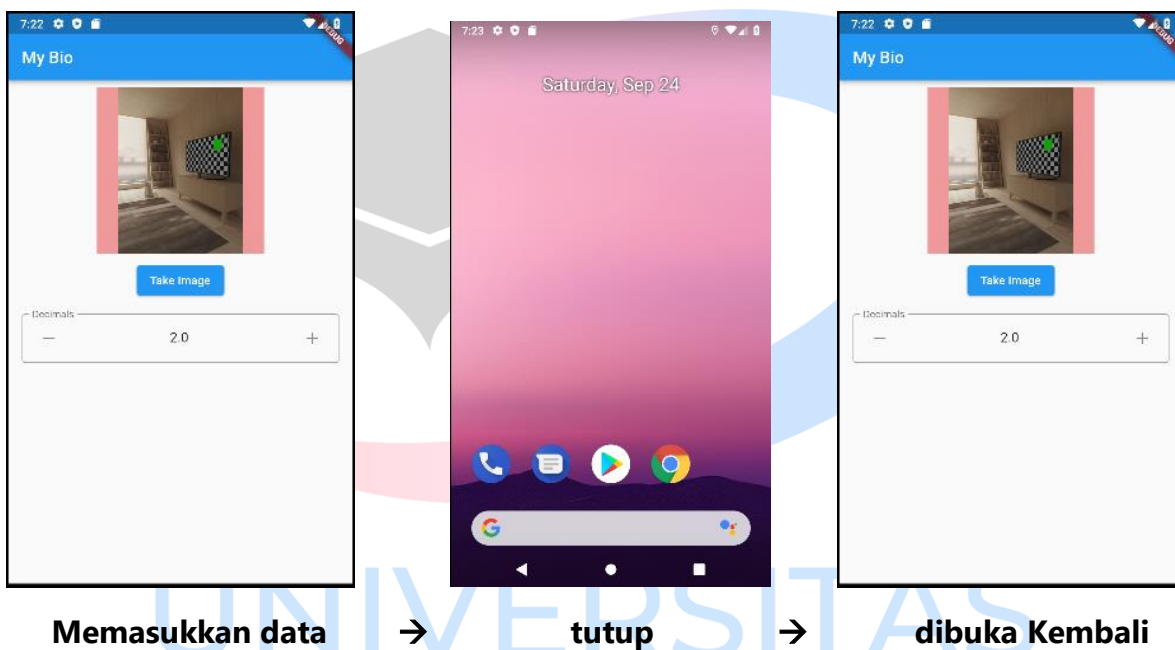
5. Untuk menjalankan fungsi `setImage`, anda dapat menambahkan fungsi `_setImage` pada fungsi `onPressed` pada widget `Button`


```

67   Padding(
68     padding: const EdgeInsets.all(8.0),
69     child: ElevatedButton(
70       onPressed: () async {
71         XFile? image =
72           await _picker.pickImage(source: ImageSource.gallery);
73         _setImage(image?.path);
74       },
75       child: Text("Take Image"), // ElevatedButton
76     ), // Padding

```

Hasil :



Setelah mengimplementasikan fungsi SharedPreferences, maka saat ini aplikasi anda dapat menyimpan data sementara yang membuat aplikasi anda dapat mengingat data terakhir yang dimasukkan ke dalam aplikasi.

Latihan

1. Dari contoh SharedPreferences, dapatkan anda menambahkan proses dalam mengisikan tanggal untuk di simpan pada SharedPreferences?
2. Dari contoh SharedPreferences, dapatkan anda mengubah proses setState, menjadi Provide?



UNIVERSITAS
MIKROSKIL