#14 | Persistensi Data

| **Summary** | Mampu mengelola data JSON dan persistensi data di Flutter |
| --- | --- |
| **URL** | 14-data-persistence |
| **Category** | Mobile Flutter |
| **Environment** | web |
| **Status** | Published |
| **Feedback Link** | https://github.com/jti-polinema/flutter-codelab/issues |
| **Author** | Habibie Ed Dien |
| **Author LDAP** | hbb-polinema |
| **Analytics Account** | G-H7507PFEJB |

[Persistensi Data](#_ok7k5uux6)

[Tujuan Praktikum](#_21yzqg98x7h6)

[Sumber Daya yang Dibutuhkan](#_dusjnkkj26ob)

[Pengetahuan yang Anda harus Miliki](#_319h1e1sjzdz)

[Pendahuluan Persistensi Data](#_t0tqoph2ohlh)

[Praktikum 1: Konversi Dart model ke JSON](#_vmpz7ap0erhd)

[Langkah 1: Buat Project Baru](#_spr93e7x3yo2)

[Langkah 2: Buka file main.dart](#_2q85gi9dhaiq)

[Soal 1](#_bb1snip8ruyk)

[Langkah 3: Buat folder baru assets](#_p0t5r1yzb5qp)

[Langkah 4: Buat file baru pizzalist.json](#_xmhut8m5gdv1)

[Langkah 5: Edit pubspec.yaml](#_wtc8ac276zx5)

[Langkah 6: Edit maint.dart](#_rkl6d7ct1s8j)

[Langkah 7: Buka main.dart](#_sf55rqaqaz6r)

[Langkah 8: Tambah variabel](#_wq5i5ww5j4xa)

[Langkah 9:](#_rhi6kwkqddsm)

[Langkah 10: Run](#_kgve4w76y3z3)

[Langkah 11:](#_we6tjrvb0pu2)

[Langkah 12:](#_wpuhk0pub24v)

[Soal 4](#_rarblsa55tt1)

[Langkah 13: Ganti isi method changeColor()](#_q9klfvxqpzx2)

[Soal 5](#_xc5ntuz7143u)

[Langkah 14:](#_z8iykfrhhvie)

[Langkah 15:](#_gbq7eyp2o6to)

[Langkah 16:](#_3gp8ozseqyzo)

[Langkah 17:](#_bi111dsn7x1v)

[Langkah 18:](#_dd3im34kr3ws)

[Langkah 19:](#_cfeqe63mlzyu)

[Langkah 20:](#_iidnmlqf29zh)

[Langkah 21:](#_qx0ge6mtjh3a)

[Langkah 22:](#_crfsxmyaqd0v)

[Langkah 23:](#_8rrslw2a5lrx)

[Langkah 24:](#_yyzc2cx7rj27)

[Langkah 25:](#_y18vs1kulkpq)

[Langkah 26:](#_qwjnvp4m129g)

[Praktikum 2: Handle kompatibilitas data JSON](#_1e543m1dm1c4)

[Langkah 1:](#_u3l46a7g73v0)

[Langkah 2:](#_gfa7d3gyzo4o)

[Langkah 3:](#_wc7bdoy0rvka)

[Langkah 4:](#_dxl4engxhz7o)

[Langkah 5:](#_oi2maeynzj9o)

[Langkah 6:](#_mdpi9i7qep2r)

[Langkah 7:](#_finb1ju2tytk)

[Langkah 8:](#_n1owjg1xgot9)

[Langkah 9: Run](#_wgcvtobmc11h)

[Langkah 10:](#_9ti7kzsslv4y)

[Langkah 11: Run](#_cmbkp9a8xgul)

[Soal 7](#_nmtnybz24m4n)

[Praktikum 3: Menangani error JSON](#_riuqbijaf0zt)

[Langkah 1: Buka pizza.dart](#_uv7foq4euher)

[Langkah 2:](#_di3ldp29lo3b)

[Langkah 3:](#_ma6gko34okso)

[Langkah 4: Run](#_oxnid84td09d)

[Soal 8](#_icte5u7gn468)

[Praktikum 4: SharedPreferences](#_1c5hdgpeva1a)

[Langkah 1:](#_mf0b60ekdgn2)

[Langkah 2:](#_hyt5pupp74tf)

[Langkah 3:](#_45fx8aigkiyt)

[Langkah 4:](#_azsb3vk7tfot)

[Langkah 5:](#_u8zmxi9ax5h1)

[Langkah 6:](#_1uddj4fwvvh9)

[Langkah 7:](#_bbnibecwo5bx)

[Langkah 8:](#_1eglzmupw5yp)

[Langkah 9: Run](#_r0j41tes71rv)

[Langkah 10:](#_12vta3m65u3v)

[Langkah 11:](#_jfku7rr8rcz4)

[Langkah 12:](#_4wpizu87cn4q)

[Langkah 13:](#_hoj51cvqty2v)

[Langkah 14:](#_cum4sm7oyyld)

[Langkah 15: Run](#_jiy412uni7yb)

[Soal 9](#_no5m83l02chs)

[Praktikum 5: Akses filesystem dengan path\_provider](#_3fcxszkylq4h)

[Langkah 1:](#_o8guex945zrp)

[Langkah 2:](#_6z67xj3j5wqh)

[Langkah 3:](#_hu872j19rioc)

[Langkah 4:](#_2fs1ku91ieim)

[Langkah 5:](#_1ajdyynpvb6b)

[Langkah 6:](#_7it4mvvlwwm)

[Langkah 7:](#_fzp90qc1i9q2)

[Soal 11](#_vgspoagmptbp)

[Praktikum 6: Akses filesystem dengan direktori](#_yocqwzbhogdc)

[Langkah 1:](#_fhnygqckvfg)

[Langkah 2:](#_xgvweet1rjak)

[Langkah 3: Tetap di file stream.dart](#_ai3lmohajpie)

[Langkah 4: Edit main.dart](#_gcxu61he253x)

[Langkah 5: Tambah variabel](#_pqxdjfvtezod)

[Langkah 6: Edit initState()](#_8fpvkqkqcuy2)

[Langkah 7: Edit method build()](#_1xpuqms9u3kz)

[Soal 12](#_w2bg8hxbwg58)

[Praktikum 7: Menyimpan data dengan enkripsi/dekripsi](#_z1vcfco4xqur)

[Langkah 1: Buat Project baru](#_j8ii7uwgw1qx)

[Langkah 2: Isi kode random\_bloc.dart](#_jpb7v8sxomdn)

[Langkah 3: Buka main.dart](#_d2p9iyg8m9n5)

[Langkah 4: Buat variabel StreamController](#_ni8oooih0sc8)

[Langkah 5: Buat constructor](#_lmjgavevz3uj)

[Langkah 6: Buat method dispose()](#_axbkqnc24e2v)

[Langkah 7: Edit main.dart](#_frwx0mqs2mlt)

[Langkah 8: Buat file baru random\_screen.dart](#_lqtycge6h8jw)

[Langkah 9: Lakukan impor material dan random\_bloc.dart](#_ar5f5kofgkjk)

[Langkah 10: Buat StatefulWidget RandomScreen](#_qmbx8t4zia5p)

[Soal 13](#_3z4fyksrp32d)

[Selamat!](#_s22qmcj4zu61)

[Apa selanjutnya?](#_9mqifffit2ew)

[Referensi](#_an95hoqxrlhg)

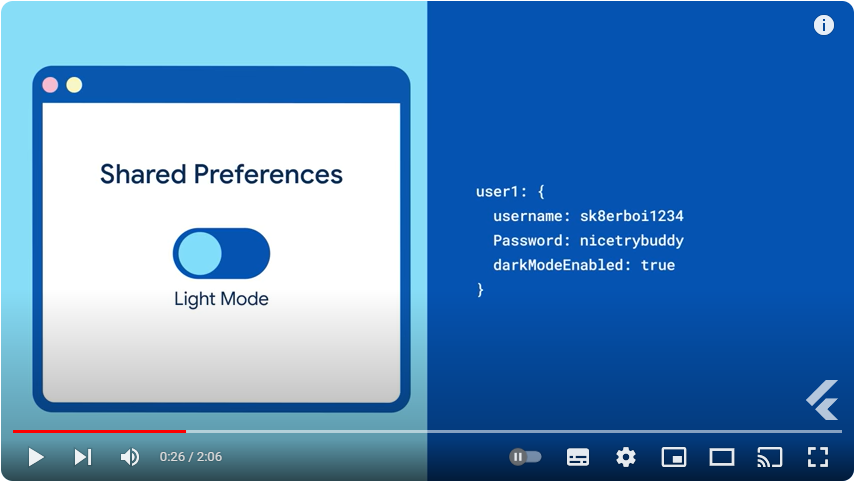
# Persistensi Data

**Terakhir diperbarui:** 27 November 2023

**Penulis:** Habibie Ed Dien

Pada codelab ini, Anda akan mempelajari tentang persistensi data di Flutter beserta contoh penggunaannya. Cara kerja, manfaat, dan cara mengelola data JSON lebih lanjut.

Video berikut menjelaskan tentang persistensi data menggunakan *Shared Preferences*. Silakan simak dan pahami!



## Tujuan Praktikum

Setelah menyelesaikan codelab ini Anda akan mampu untuk:

* Mengonversi model Dart ke JSON
* Menangani skema JSON yang tidak kompatibel dengan model
* Menangani error JSON
* Menyimpan data sederhana dengan SharedPreferences
* Mengakses filesystem dengan plugin path\_provider
* Mengakses filesystem dengan direktori
* Menggunakan secure storage untuk menyimpan data

| **Catatan:** Materi ini diadaptasi dari Buku “Flutter Cookbook, 100+ step-by-step recipes for building cross-platform, professional grade apps with Flutter 3.10.x and Dart 3.x - Second Edition” oleh Simone Alessandria (2023). |
| --- |

## Sumber Daya yang Dibutuhkan

Berikut merupakan sumber daya yang diperlukan untuk menyelesaikan praktikum ini:

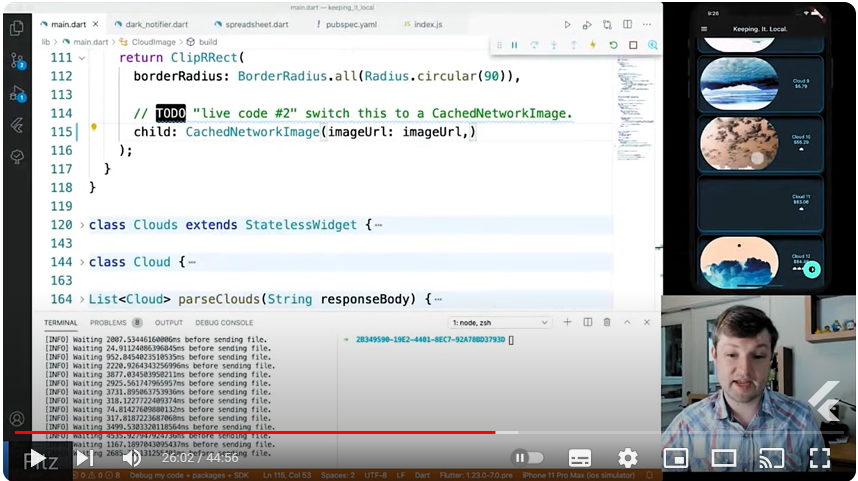
* PC atau Laptop dengan spesifikasi minimum RAM 4GB Processor Core i3
* Koneksi internet minimal kecepatan 1 Mbps
* Chrome Browser
* Perangkat mobile (Android/iOS)

## Pengetahuan yang Anda harus Miliki

* Algoritma dan Pemrograman Dasar
* Pemrograman Berorientasi Objek
* Bahasa Pemrograman Dart Dasar
* Teknik Dasar Pemrograman Asynchronous

# Pendahuluan Persistensi Data

Duration: 50:00



Kebanyakan aplikasi, terutama yang berbasis bisnis, pasti membutuhkan operasi **CRUD: Create, Read, Update, dan Delete** data. Data dapat disimpan secara local atau remote. Tanpa peduli data itu akan disimpan di mana, pada beberapa kasus kita tetap membutuhkan data itu berbentuk JSON sebelum itu disimpan secara permanen. Sehingga kita perlu tau metode apa saja yang diperlukan untuk mengolah JSON di Dart dan Flutter. Hal ini akan sangat membantu pada beberapa teknologi yang mungkin Anda akan gunakan, seperti [SQLite](https://pub.dev/packages/sqflite), [Sembast](https://pub.dev/packages/sembast), dan [database Firebase](https://pub.dev/packages/firebase_database). Semua itu membutuhkan proses pengiriman dan unduh data berupa JSON.

# Praktikum 1: Konversi Dart model ke JSON

Duration: 20:00

Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda (ketik di **README.md**) pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil melakukan setup environment Flutter SDK, VS Code, Flutter Plugin, dan Android SDK pada pertemuan pertama. |
| --- |

## Langkah 1: Buat Project Baru

Buatlah sebuah project flutter baru dengan nama **store\_data\_nama** (beri nama panggilan Anda) di folder **week-14/src/** repository GitHub Anda.

## Langkah 2: Buka file main.dart

Ketiklah kode seperti berikut ini.





| Soal 1  * Tambahkan **nama panggilan Anda** pada title app sebagai identitas hasil pekerjaan Anda. * Gantilah warna tema aplikasi sesuai kesukaan Anda. * Lakukan commit hasil jawaban Soal 1 dengan pesan **“W14: Jawaban Soal 1”** |
| --- |

## Langkah 3: Buat folder baru assets

Buat folder baru assets di root project Anda

## Langkah 4: Buat file baru pizzalist.json

Letakkan file ini di dalam folder assets, lalu salin data JSON berikut ke file tersebut.

|  |
| --- |

Jika Anda ingin menggunakan data JSON yang lain, Anda dapat mengakses salah satu dari daftar API di tautan ini: <https://github.com/public-apis/public-apis>

## Langkah 5: Edit pubspec.yaml

Tambahkan referensi folder assets ke file pubspec.yaml seperti berikut ini.



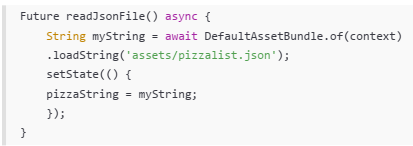
## Langkah 6: Edit maint.dart

Tambahkan kode berikut ini.

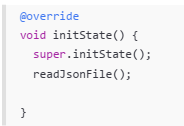


## Langkah 7: Buka main.dart

Ketik kode impor file ini pada file main.dart



## Langkah 8: Tambah variabel

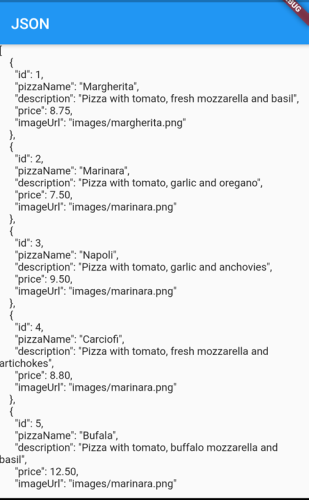


## Langkah 9:

Tetap di file main, Ketik kode seperti berikut



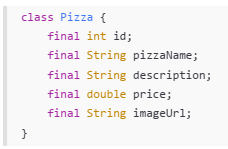
## **Langkah** 10: Run



## **Langkah** 11:

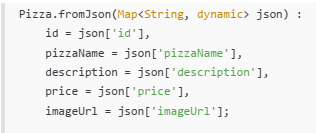
Sesuaikan kode seperti berikut.

## Langkah 12:



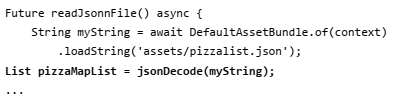
| Soal 4  * Capture hasil praktikum Anda berupa GIF dan lampirkan di README. * Lakukan commit hasil jawaban Soal 4 dengan pesan **“W14: Jawaban Soal 4”** |
| --- |

## Langkah 13: Ganti isi method changeColor()



| Soal 5  * Jelaskan perbedaan menggunakan listen dan await for (langkah 9) ! * Lakukan commit hasil jawaban Soal 5 dengan pesan **“W14: Jawaban Soal 5”** |
| --- |

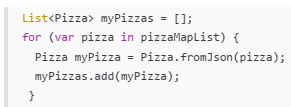
## Langkah 14:



## Langkah 15:



## Langkah16:



## **Langkah** 17:



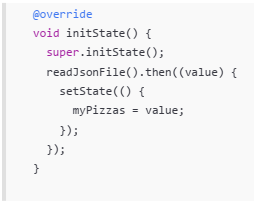
## **Langkah** 18:



## **Langkah** 19:



## **Langkah** 20:



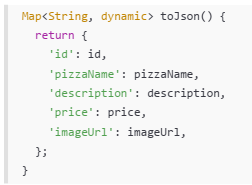
## **Langkah** 21:

| body: ListView.builder(  itemCount: myPizzas.length,  itemBuilder: (context, index) {  return ListTile(  title: Text(myPizzas[index].pizzaName),  subtitle: Text(myPizzas[index].description),  );  },  ));  } |
| --- |

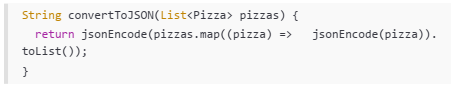
## **Langkah** 22:



## **Langkah** 23:



## **Langkah** 24:



## **Langkah** 25:

| String json = convertToJSON(myPizzas);  print(json);  return myPizzas; |
| --- |

## **Langkah** 26:



# Praktikum 2: Handle kompatibilitas data JSON

Duration: 15:00

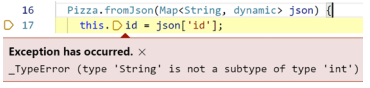
Setelah Anda menyelesaikan praktikum 1, Anda dapat melanjutkan praktikum 2 ini. Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil menyelesaikan Praktikum 1. |
| --- |

Pada codelab ini, kita akan menambah kode dari aplikasi **stream** di praktikum sebelumnya.

## Langkah 1:

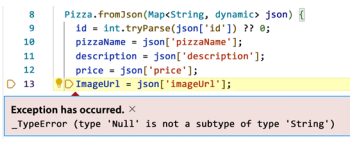
## Langkah 2:



## Langkah 3:

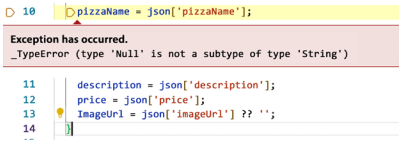


## Langkah 4:

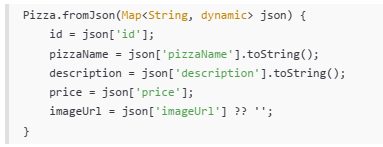


## Langkah 5:

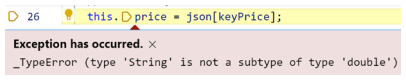




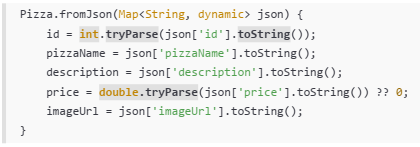
## **Langkah** 6:



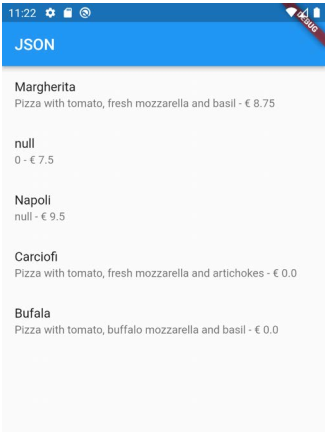
## **Langkah** 7:



## Langkah 8:



## Langkah 9: Run



## Langkah 10:



## Langkah 11: Run

| Soal 7  * Jelaskan maksud kode langkah 13 sampai 15 tersebut! * Kembalikan kode seperti semula pada Langkah 15, comment addError() agar Anda dapat melanjutkan ke praktikum 3 berikutnya. * Lalu lakukan commit dengan pesan “**W14: Jawaban Soal 7**”. |
| --- |

# Praktikum 3: Menangani error JSON

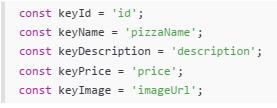
Duration: 15:00

Pada praktikum 3 ini, Anda akan menggunakan StreamTransformers ke dalam stream untuk melakukan map dan filter data.

Setelah Anda menyelesaikan praktikum 2, Anda dapat melanjutkan praktikum 3 ini. Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil menyelesaikan Praktikum 2. |
| --- |

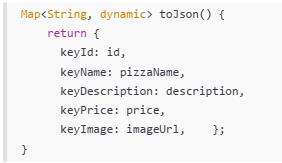
## Langkah 1: Buka pizza.dart



## Langkah 2:



## Langkah 3:



## Langkah 4: Run

| Soal 8  * Jelaskan maksud kode langkah 1-3 tersebut! * Capture hasil praktikum Anda berupa GIF dan lampirkan di README. * Lalu lakukan commit dengan pesan “**W14: Jawaban Soal 8**”. |
| --- |

# Praktikum 4: SharedPreferences

Duration: 15:00

Dari praktikum sebelumnya, Anda telah menggunakan method listen mendapatkan nilai dari stream. Ini akan menghasilkan sebuah Subscription. Subscription berisi method yang dapat digunakan untuk melakukan listen pada suatu event dari stream secara terstruktur.

Pada praktikum 4 ini, kita akan gunakan Subscription untuk menangani event dan error dengan teknik praktik baik (*best practice*), dan menutup Subscription tersebut.

Setelah Anda menyelesaikan praktikum 3, Anda dapat melanjutkan praktikum 4 ini. Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil menyelesaikan Praktikum 3. |
| --- |

## Langkah 1:



## Langkah 2:

## Langkah 3:

Tambahkan kode berikut ini.



## Langkah 4:



## Langkah 5:



## Langkah 6:



## Langkah 7:



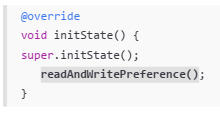
## Langkah 8:



## Langkah 9: Run



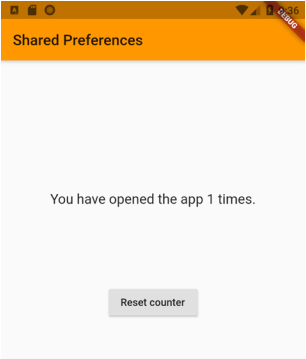
## Langkah 10:



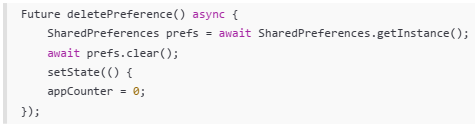
## Langkah 11:



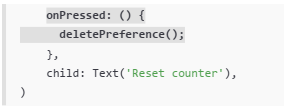
## Langkah 12:



## Langkah 13:



## Langkah 14:



## Langkah 15: Run

| Soal 9  * Jelaskan maksud kode langkah 2, 6 dan 8 tersebut! * Capture hasil praktikum Anda berupa GIF dan lampirkan di README. * Lalu lakukan commit dengan pesan “**W14: Jawaban Soal 9**”. |
| --- |

# Praktikum 5: Akses filesystem dengan path\_provider

Duration: 13:00

Setelah Anda menyelesaikan praktikum 4, Anda dapat melanjutkan praktikum 5 ini. Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil menyelesaikan Praktikum 4. |
| --- |

## Langkah 1:



## Langkah 2:

Ketik kode seperti berikut.



## Langkah 3:



## Langkah 4:



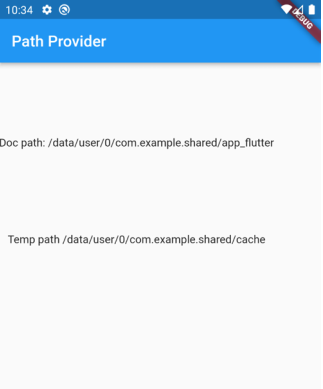
## Langkah 5:

| @override  void initState() {  super.initState();  getPaths();  } |
| --- |

## Langkah 6:



## Langkah 7:



| Soal 11  * Jelaskan mengapa hal itu bisa terjadi ? * Capture hasil praktikum Anda berupa GIF dan lampirkan di README. * Lalu lakukan commit dengan pesan “**W14: Jawaban Soal 10,11**”. |
| --- |

# Praktikum 6: Akses filesystem dengan direktori

Duration: 15:00

Setelah Anda menyelesaikan praktikum 5, Anda dapat melanjutkan praktikum 6 ini. Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil menyelesaikan Praktikum 5. |
| --- |

## Langkah 1:



## Langkah 2:

Ketik kode ini



## Langkah 3: Tetap di file stream.dart

Ketik kode seperti berikut.

| Future<bool> writeFile() async {  try {  await myFile.writeAsString('Margherita, Capricciosa, Napoli');  return true;  } catch (e) {  return false;  }  } |
| --- |

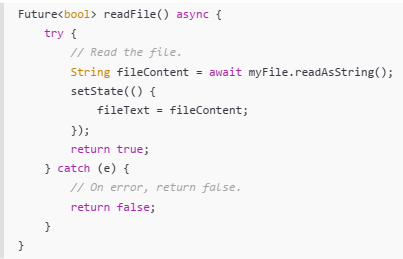
## Langkah 4: Edit main.dart

Ketik kode seperti berikut ini.



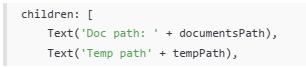
## Langkah 5: Tambah variabel

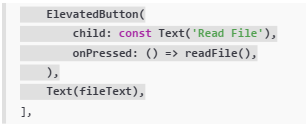
Di dalam class \_StreamHomePageState, ketika variabel ini.



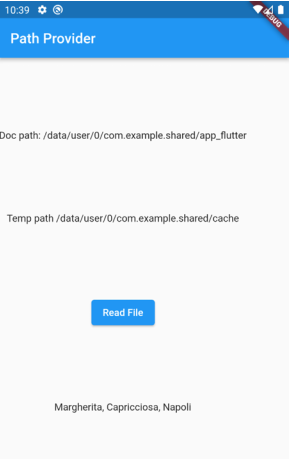
## Langkah 6: Edit initState()

Ketik kode seperti berikut.





## Langkah 7: Edit method build()



| Soal 12  * Jelaskan maksud kode pada langkah 3 dan 7 ! * Capture hasil praktikum Anda berupa GIF dan lampirkan di README. * Lalu lakukan commit dengan pesan “**W14: Jawaban Soal 12**”. |
| --- |

# Praktikum 7: Menyimpan data dengan enkripsi/dekripsi

Duration: 15:00

Setelah Anda menyelesaikan praktikum 6, Anda dapat melanjutkan praktikum 7 ini. Selesaikan langkah-langkah praktikum berikut ini menggunakan editor Visual Studio Code (VS Code) atau Android Studio atau code editor lain kesukaan Anda. Jawablah di laporan praktikum Anda pada setiap soal yang ada di beberapa langkah praktikum ini.

| **Perhatian:** Diasumsikan Anda telah berhasil menyelesaikan Praktikum 6. |
| --- |

## Langkah 1: Buat Project baru

## Langkah 2: Isi kode random\_bloc.dart

Ketik kode impor berikut ini.



## Langkah 3: Buka main.dart

Salin kode berikut.

| import 'package:flutter/material.dart';  void main() {  runApp(const MyApp());  }  class MyApp extends StatelessWidget {  const MyApp({super.key});  // This widget is the root of your application.  @override  Widget build(BuildContext context) {  return MaterialApp(  title: 'Flutter Demo',  theme: ThemeData(  primarySwatch: Colors.deepPurple,  ),  home: const MyHomePage(),  );  }  }  class MyHomePage extends StatefulWidget {    const MyHomePage({super.key});  @override  State<MyHomePage> createState() => \_MyHomePageState();  }  class \_MyHomePageState extends State<MyHomePage> {  final pwdController = TextEditingController();  String myPass = '';  @override  void initState() {  super.initState();  }  @override  Widget build(BuildContext context) {  return Scaffold(  appBar: AppBar(title: const Text('Path Provider')),  body: SingleChildScrollView(  child: Padding(  padding: const EdgeInsets.all(16.0),  child: Column(  children: [  TextField(  controller: pwdController,  ),  ElevatedButton(child: const Text('Save Value'), onPressed: () {}),  ElevatedButton(child: const Text('Read Value'), onPressed: () {}),  Text(myPass),  ],  ),  ),  ),  );  }  } |
| --- |

## Langkah 4: Buat variabel StreamController

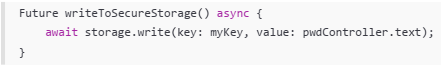
Di dalam class RandomNumberBloc() ketik variabel berikut ini



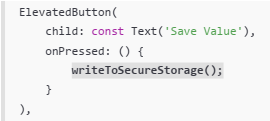
## Langkah 5: Buat constructor



## Langkah 6: Buat method dispose()

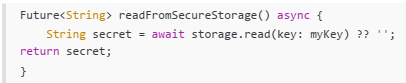


## Langkah 7: Edit main.dart



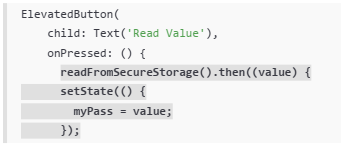
## Langkah 8: Buat file baru random\_screen.dart

Di dalam folder lib project Anda, buatlah file baru ini.



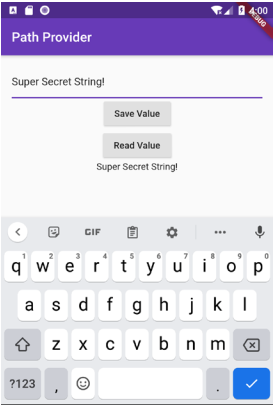
## Langkah 9: Lakukan impor material dan random\_bloc.dart

Ketik kode ini di file baru random\_screen.dart



## Langkah 10: Buat StatefulWidget RandomScreen

Buatlah di dalam file random\_screen.dart



| Soal 13  * Jelaskan maksud praktikum ini ! Dimanakah letak konsep pola BLoC-nya ? * Capture hasil praktikum Anda berupa GIF dan lampirkan di README. * Lalu lakukan commit dengan pesan “**W14: Jawaban Soal 13**”. |
| --- |

# Selamat!

Duration: 0:00

Selamat Anda telah menyelesaikan Codelab ini. Anda telah mempelajari terkait Persistensi Data dan contoh penggunaannya.

Pada codelab berikutnya, Anda akan mempelajari tentang Restful API.

Jangan sungkan jika Anda menemukan kesalahan pada codelab ini untuk merevisi atau sekedar melaporkan issue melalui tautan di pojok kiri bawah ([Report a mistake](https://github.com/jti-polinema/flutter-codelab/issues)).

## Apa selanjutnya?

Silakan cek beberapa sumber belajar lainnya…

* <https://pub.dev/packages/sembast>
* [Flutter Tutorial - Secure Storage & Data Persistence](https://www.youtube.com/watch?v=Oqhrxk_f1RE)
* <https://pub.dev/packages/sqflite>
* <https://pub.dev/packages/firebase_database>
* <https://pub.dev/packages/json_serializable>
* <https://pub.dev/packages/built_value>
* <https://docs.flutter.dev/cookbook/persistence/reading-writing-files>
* <https://api.flutter.dev/flutter/dart-io/File-class.html>
* <https://pub.dev/packages/encrypt>
* <https://pub.dev/packages/crypto>

## Referensi

* [Buku “Flutter Cookbook, 100+ step-by-step recipes for building cross-platform, professional grade apps with Flutter 3.10.x and Dart 3.x - Second Edition” oleh Simone Alessandria (2023)](https://drive.google.com/file/d/1W9uSLC9wlf9fFudmCWFHTx9dan5SXII0/view?usp=sharing)
* [Flutter for Beginners Second Edition](https://drive.google.com/file/d/1hJbctx-_O3B6kUeG70bJD5xfbnsWwBXY/view?usp=sharing)
* <https://www.youtube.com/flutterdev>
* <https://github.com/PacktPublishing/Flutter-for-Beginners-Second-Edition>