**Ketentuan;**

Simpanlah file lembar jawaban ini dengan format; **Kelas\_Nama Lengkap**

Contoh; **TI 3C\_Rifki Fakhrudin**

**Upload tugas jobsheet ini dengan batas maksimum sesuai jadwal perkuliahan masing masing kelas**

Upload file tugas jobsheet di website **lms.polinema**

|  |  |  |
| --- | --- | --- |
| **Nama** | : | Rifki Fakhrudin |
| **Nim** | : | 2241720218 |
| **Kelas** | : | 3C |

Tulislah Jawaban Pada Kolom Yang tersedia di bawah ini;

**LEMBAR JAWABAN JOBSHEET-13**

|  |  |
| --- | --- |
| **Soal No** | **Jawaban** |
|  | Di editor favorit Anda, buat proyek Flutter baru dan beri nama store\_data |
|  | Pada file main.dart, hapus kode yang ada dan tambahkan kode awal untuk aplikasi dengan kode berikut:  import 'package:flutter/material.dart';  void main() {    runApp(const MyApp());  }  class MyApp extends StatelessWidget {    const MyApp({super.key});    @override    Widget build(BuildContext context) {      return MaterialApp(        title: 'Flutter JSON Demo',        theme: ThemeData(          primarySwatch: Colors.blue,        ),        home: const MyHomePage(),      );    }  }  class MyHomePage extends StatefulWidget {    const MyHomePage({super.key});    @override    State<MyHomePage> createState() => \_MyHomePageState();  }  class \_MyHomePageState extends State<MyHomePage> {    @override    Widget build(BuildContext context) {      return Scaffold(        appBar: AppBar(          title: const Text('JSON'),        ),      );    }  } |
|  | Tambahkan folder baru ke root proyek Anda dengan nama assets. |
|  | Di dalam folder aset, buat file baru bernama pizzalist.json  [      {        "id": 1,        "pizzaName": "Margherita",        "description": "Pizza with tomato, fresh mozzarella and basil",        "price": 8.75,        "imageUrl": "images/margherita.png"      },      {        "id": 2,        "pizzaName": "Marinara",        "description": "Pizza with tomato, garlic and oregano",        "price": 7.50,        "imageUrl": "images/marinara.png"      },      {        "id": 3,        "pizzaName": "Napoli",        "description": "Pizza with tomato, garlic and anchovies",        "price": 9.50,        "imageUrl": "images/marinara.png"      },      {        "id": 4,        "pizzaName": "Carciofi",        "description": "Pizza with tomato, fresh mozzarella and artichokes",        "price": 8.80,        "imageUrl": "images/marinara.png"      },      {        "id": 5,        "pizzaName": "Bufala",        "description": "Pizza with tomato, buffalo mozzarella and basil",        "price": 12.50,        "imageUrl": "images/marinara.png"      }     ] |
|  | Di file pubspec.yaml, tambahkan referensi ke folder aset baru, seperti yang ditunjukkan di sini: |
|  | Pada kelas \_MyHomePageState, di main.dart, tambahkan sebuah variabel state bernama pizzaString: |
|  | Untuk membaca isi file pizzalist.json, di bagian bawah kelas \_MyHomePageState di main.dart, tambahkan metode asinkron baru yang disebut readJsonFile, yang akan mengatur nilai pizzaString, seperti yang ditunjukkan di sini: |
|  | Pada kelas \_MyHomePageState, timpa metode initState dan, di dalamnya, panggil metode readJsonFile: |
|  | Sekarang, kita ingin menampilkan JSON yang diambil di properti dalam Scaffold. Untuk  melakukannya, tambahkan widget Teks sebagai child dari Container kita: |
|  | Mari kita jalankan aplikasinya. Jika semuanya berjalan seperti yang diharapkan, Anda akan melihat konten file JSON di layar |
|  | Kita ingin mengubah String ini menjadi sebuah List of Objects. Kita akan mulai dengan membuat kelas baru. Dalam folder lib aplikasi kita, buat file baru bernama pizza.dart. |
|  | Di dalam file tersebut, tentukan properti kelas Pizza: |
|  | Di dalam kelas Pizza, tentukan konstruktor bernama fromJson, yang akan mengambil sebuah Map sebagai parameter dan mengubah Map menjadi sebuah instance dari Pizza: |
|  | Refaktor metode readJsonFile() pada kelas \_MyHomePageState. Langkah pertama adalah mengubah String menjadi Map dengan memanggil metode jsonDecode. Pada method readJsonFile, tambahkan kode yang di cetak tebal berikut ini: |
|  | Pastikan editor Anda secara otomatis menambahkan pernyataan impor untuk pustaka "dart:convert" di bagian atas file main.dart; jika tidak, tambahkan saja secara manual. Tambahkan juga pernyataan impor untuk kelas pizza: |
|  | . Langkah terakhir adalah mengonversi string JSON kita menjadi List of native Dart objects. Kita dapat melakukan ini dengan mengulang pizzaMapList dan mengubahnya menjadi objek Pizza. Di dalam metode readJsonFile, di bawah metode jsonDecode, tambahkan kode berikut: |
|  | Hapus atau beri komentar pada metode setState yang mengatur String pizzaString dan kembalikan daftar objek Pizza sebagai gantinya: |
|  | Ubah signature metode sehingga Anda dapat menampilkan nilai balik secara eksplisit: |
|  | Sekarang kita memiliki objek List of Pizza. Daripada hanya menampilkan sebuah Teks kepada pengguna, kita dapat menampilkan sebuah ListView yang berisi sekumpulan widget ListTile. Di bagian atas kelas \_MyHomePageState, buat List bernama myPizzas: |
|  | Dalam metode initState, pastikan Anda mengatur myPizzas dengan hasil panggilan ke readJsonFile: |
|  | Tambahkan kode berikut ini di dalam Scaffold, di dalam metode build(): |
|  | Jalankan aplikasi. Antarmuka pengguna sekarang seharusnya jauh lebih ramah dan terlihat seperti yang ditunjukkan pada |
|  | **Praktikum 2: Reading the JSON file**  Tambahkan metode baru ke kelas Pizza, di file pizza.dart, yang disebut toJson. Ini akan mengembalikan sebuah Map dari objek: |
|  | Setelah Anda memiliki sebuah Map, Anda dapat menserialisasikannya kembali ke dalam string JSON. Tambahkan metode baru di di bagian bawah kelas \_MyHomePageState, di dalam file main.dart, yang disebut convertToJSON: |
|  | Metode ini mengubah objek List of Pizza kembali menjadi string Json dengan memanggil metode jsonEncode lagi di pustaka dart\_convert. |
|  | Terakhir, mari panggil metode tersebut dan cetak string JSON di Debug Console. Tambahkan kode berikut ke metode readJsonFile, tepat sebelum mengembalikan List myPizzas: |
|  | Jalankan aplikasi. Anda akan melihat string JSON dicetak, seperti yang ditunjukkan pada gambar berikut: |
|  | Praktikum 3: Saving data simply with SharedPreferences |
|  | Untuk memperbarui dependensi dalam proyek Anda, jalankan perintah flutter pub get dari jendela Terminal |
|  | Di bagian atas file main.dart, impor shared\_preferences: |
|  | Di bagian atas kelas \_MyHomePageState, buat variabel status integer baru bernama appCounter |
|  | Dalam kelas \_MyHomePageState, buat metode asinkron baru yang disebut readAndWritePreferences(): |
|  | Di dalam metode readAndWritePreference, buatlah sebuah instance dari SharedPreferences: |
|  | Setelah membuat instance preferensi, kita membuat kode yang mencoba baca nilai kunci appCounter. Jika nilainya nol, setel ke 0; lalu naikkan nilainya: |
|  | Setelah itu, atur nilai kunci appCounter di preferensi ke nilai baru: |
|  | Memperbarui nilai status appCounter: |
|  | Pada metode initState di kelas \_MyHomePageState, panggil metode readAndWritePreference() dengan kode yang dicetak tebal: |
|  | Dalam metode build, tambahkan kode berikut ini di dalam widget Container: |
|  | Tambahkan metode baru ke kelas \_MyHomePageState yang disebut deletePreference(), yang akan menghapus nilai yang disimpan: |
|  | Dari properti onPressed dari widget ElevatedButton di metode build(), memanggil metode deletePreference(), dengan kode di cetak tebal: |
|  | Jalankan aplikasi lagi. Sekarang, saat Anda menekan tombol Reset penghitung, nilai appCounter akan dihapus |
|  | **Praktikum 4: Accessing the filesystem, part 1: path\_provider**  menambahkan dependency yang relevan ke file pubspec.yaml. Tambahkan path\_provider dengan mengetikkan perintah ini dari Terminal Anda: |
|  | Di bagian atas file main.dart, tambahkan impor path\_provider: |
|  | Di bagian atas kelas \_MyHomePageState, tambahkan variabel State yang akan kita gunakan untuk memperbarui antarmuka pengguna: |
|  | Masih dalam kelas \_MyHomePageState, tambahkan metode untuk mengambil direktori temporary dan dokumen: |
|  | Pada metode initState dari kelas \_MyHomePageState, panggil metode getPaths: |
|  | Pada metode build \_MyHomePageState, buat UI dengan dua widget Teks yang menunjukkan path yang diambil: |
|  | Jalankan aplikasi. Anda akan melihat layar yang terlihat seperti berikut ini:  terjadi eror saya tidak tau mengapa, saya sudah melakukan langkah2 yang ditentukan, akan tetapi pathnya tidak terpanggil. Ditemukan eror binding.dart |
|  | **Praktikum 5: Accessing the filesystem, part 2: Working with directories**  Di bagian atas berkas main.dart, impor pustaka dart:io: |
|  | Di bagian atas kelas \_MyHomePageState, di file main.dart, buat dua variabel State baru untuk file dan isinya: |
|  | Masih dalam kelas MyHomePageState, buat metode baru bernama writeFile dan gunakan kelas File dari pustaka dart:io untuk membuat file baru: |
|  | Dalam metode initState, setelah memanggil metode getPaths, dalam metode then, buat sebuah file dan panggil metode writeFile: |
|  | Buat metode untuk membaca file: |
|  | Dalam metode build, di widget Column, perbarui antarmuka pengguna dengan ElevatedButton. Ketika pengguna menekan tombol, tombol akan mencoba membaca konten file dan menampilkannya di layar, cek kode cetak tebal: |
|  | Jalankan aplikasi dan tekan tombol Baca File. Di bawah tombol tersebut, Anda akan melihat teks Margherita, Capricciosa, Napoli, seperti yang ditunjukkan pada tangkapan layar berikut:    Tetap tidak tampil saya menyimpulkan mungkin ini dikarenakan versi flutter saya yang tidak bisa mengakses despendencies path\_provider. |
|  | **Praktikum 6: Using secure storage to store data** Tambahkan flutter\_secure\_storage ke proyek Anda, dengan mengetik |
|  | Di file main.dart, salin kode berikut: import 'dart:convert';  import 'pizza.dart';  import 'package:flutter/material.dart';  import 'package:flutter\_secure\_storage/flutter\_secure\_storage.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 JSON 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> {    // String pizzaString = '';    // List<Pizza> myPizzas = [];    final pwdController = TextEditingController();    String myPass ='';    final storage = const FlutterSecureStorage();    final myKey = '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: (){writeToSecureStorage();                  }),                ElevatedButton(                  child: const Text('Read Value'),                  onPressed: (){readFromSecureStorage().then((value){                    setState(() {                      myPass = value;                    });                  });                }),                Text(myPass),              ],            ),          ),        ),        // body: Container(),      );    }    Future writeToSecureStorage() async {      await storage.write(key: myKey, value: pwdController.text);    }    Future<String> readFromSecureStorage() async{      String secret = await storage.read(key: myKey)?? '';      return secret;    }  } |
|  | Di bagian atas file main.dart, tambahkan impor yang diperlukan: |
|  | Di bagian atas kelas \_myHomePageState, buat penyimpanan yang aman: |
|  | Di kelas \_myHomePageState, tambahkan metode untuk menulis data ke penyimpanan aman: |
|  | Pada metode build() dari kelas \_myHomePageState, tambahkan kode yang akan menulis ke penyimpanan ketika pengguna menekan tombol Save Value, cek kode cetak tebal: |
|  | Di kelas \_myHomePageState, tambahkan metode untuk membaca data dari penyimpanan aman: |
|  | Pada metode build() dari kelas \_myHomePageState, tambahkan kode untuk membaca dari penyimpanan ketika pengguna menekan tombol Read Value dan memperbarui variabel myPass State: |
|  | Jalankan aplikasi dan tulis beberapa teks pilihan Anda di bidang teks. Kemudian, tekan tombol Save Value. Setelah itu, tekan tombol Read Value. Anda akan melihat teks yang Anda ketik di kolom teks, seperti yang ditunjukkan pada tangkapan layar berikut: |
|  |  |