**LAPORAN TUGAS AKHIR**

**PEMROGRAMAN BERIORIENTASI OBJEK**

**PEMBUATAN APLIKASI PENGELOLAAN JADWAL DOKTER**



Disusun Oleh:

1. Muhamad Ikbal Asmin (065122125)
2. Revanza Permana (065122128)
3. Muhamad Azril Hakim (065122129)
4. Geza Muhamad Firdaus (065122143)
5. Rafi Muhamad Nabil (065121144)

Asisten Praktikum:

1. Fakhriza Sidhqi Wafiq Fauzi (065121131)
2. Muhammad Fahdin (065121154)

**PROGRAM STUDI ILMU KOMPUTER**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PAKUAN**

**BOGOR**

**2023**

# KATA PENGANTAR

Assalamu’alaikum Wr. Wb

Puji syukur atas kehadirat Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan Laporan Tugas Akhir Mata Kuliah Pemrograman Beriorientasi Objek yang berjudul *"*Aplikasi Pengelolaan Jadwal Dokter*"* ini dapat diselesaikan dengan tepat waktu. Tak lupa pula, kami mengucapkan terima kasih kepada Dosen Mata Kuliah Pemrograman Beriorientasi Objek yaitu Ibu Dinar Munggaran Akhmad, M.Kom. dan juga semua pihak yang terlibat dalam proses penyelesaian Laporan Tugas Akhir Mata Kuliah Pemrograman Beriorientasi Objek ini.

Penulis menyadari bahwa dalam proses penulisan Laporan ini masih jauh dari kesempurnaan baik materi maupun cara penulisannya. Namun demikian, penulis telah berusaha dengan segala kemampuan dan pengetahuan yang dimiliki sehingga dapat selesai dengan baik. Penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan laporan-laporan penulis selanjutnya.

Wassalamu’alaikum Wr. Wb

Bogor, 24 Oktober 2023

Penyusun

# DAFTAR ISI

[**KATA PENGANTAR ii**](#_Toc149509654)

[**DAFTAR ISI iii**](#_Toc149509655)

[**BAB 1 5**](#_Toc149509656)

[**PENDAHULUAN 5**](#_Toc149509657)

[**1.1. Latar Belakang 5**](#_Toc149509658)

[**1.2. Tujuan 6**](#_Toc149509659)

[**1.3. Ruang Lingkup 6**](#_Toc149509660)

[**1.4. Manfaat 6**](#_Toc149509661)

[**BAB 2 7**](#_Toc149509662)

[**DESKRIPSI TEMA 7**](#_Toc149509663)

[**2.1. Tema Proyek 7**](#_Toc149509664)

[**2.2. Fitur Utama 7**](#_Toc149509665)

[**BAB 3 8**](#_Toc149509666)

[**CLASS DIAGRAM 8**](#_Toc149509667)

[**3.1. Struktur Class 8**](#_Toc149509668)

[**3.2. Hubungan Antar Class 9**](#_Toc149509669)

[**BAB 4 10**](#_Toc149509670)

[**PENGEMBANGAN DART PBO 10**](#_Toc149509671)

[**4.1. Implementasi Konsep PBO 10**](#_Toc149509672)

[**4.1.1. Inheritance 10**](#_Toc149509673)

[**4.1.2. Encapsulation 11**](#_Toc149509674)

[**4.1.3. Abstraction 11**](#_Toc149509675)

[**4.2. Struktur Aplikasi 12**](#_Toc149509677)

[**4.3. Hasil 17**](#_Toc149509678)

[**BAB 5 17**](#_Toc149509679)

[**PENGEMBANGAN APLIKASI FLUTTER 17**](#_Toc149509680)

[**5.1. Fitur 17**](#_Toc149509681)

[**5.2. Hasil 17**](#_Toc149509682)

[**BAB 6 18**](#_Toc149509683)

[**KESIMPULAN 18**](#_Toc149509684)

[**6.1. Ringkasan 18**](#_Toc149509685)

[**6.2. Saran 19**](#_Toc149509686)

[**DAFTAR PUSTAKA 19**](#_Toc149509687)

[**LAMPIRAN 20**](#_Toc149509688)

# DAFTAR GAMBAR

[Gambar 1. Struktur Class 8](#_Toc155143211)

[Gambar 2. Hubungan Antar CLass 9](#_Toc155143212)

[Gambar 3. Flowchart atau Struktur Aplikasi 12](#_Toc155143213)

[Gambar 4. Hasil Running nya 13](#_Toc155143214)

[Gambar 5. Halaman Homepage 15](https://d.docs.live.net/f8f0ca34996aefa3/Dokumen/LAPORAN%20TUGAS%20AKHIR%20PRAKTIKUM%20PBO_unfix.docx#_Toc155143215)

[Gambar 6. Halaman Login Page 15](#_Toc155143216)

[Gambar 7. Halaman Edit Page 16](#_Toc155143217)

[Gambar 8. Halaman tampilan data yang sudah diperbarui 16](https://d.docs.live.net/f8f0ca34996aefa3/Dokumen/LAPORAN%20TUGAS%20AKHIR%20PRAKTIKUM%20PBO_unfix.docx#_Toc155143218)

[Gambar 9. Halaman menghapus data 17](#_Toc155143219)

[Gambar 10. Halaman tampilan data yang sudah dihapus 17](#_Toc155143220)

[Gambar 11. Halaman masukkan data baru 18](#_Toc155143221)

[Gambar 12. Halaman tampilan data yang sudah diperbarui 18](#_Toc155143222)

# BAB 1

# PENDAHULUAN

## 1.1. Latar Belakang

Penerapan informasi jenis pelayanan medis dan jadwal dinas dokter spesialis berbasis program dimana dalam sistem ini akan menyajikan informasi tentang jenis pelayanan dan jadwal dinas dokter, dengan adanya aplikasi ini dapat membantu masyarakat dengan mudah memperoleh informasi seputar kesehatan, jadwal dokter spesialis.

Rumah sakit dan klinik perlu menjadwalkan jadwal dokter dengan efisien untuk memastikan bahwa pasien mendapatkan perawatan yang tepat pada waktu yang tepat. Penjadwalan yang buruk dapat mengakibatkan peningkatan waktu tunggu pasien dan gangguan dalam pengobatan. Dokter sering memiliki jadwal yang padat, termasuk jadwal di berbagai lokasi atau fasilitas medis.

Pengelolaan jadwal yang efisien dapat membantu rumah sakit dan klinik meningkatkan reputasi mereka dan menjaga pasien tetap puas. Ini dapat berkontribusi pada pertumbuhan bisnis dan daya saing. Aplikasi semacam ini membantu mengatasi berbagai tantangan terkait manajemen jadwal dalam lingkungan medis yang sering kali kompleks dan berubah-ubah.

Rumah sakit juga membutuhkan jadwal dokter yang akurat. Teknologi Informasi pada bidang pelayanan kesehatan sangat diperlukan untuk meningkatkan mutu pelayanan kesehatan. Saat ini hampir semua orang menuntut pelayanan publik untuk semakin efektif dan efisien, mengingat pelayanan kesehatan tidak hanya sebatas pengobatannya saja akan tetapi bagaimana suatu klinik memberikan pelayanan yang prima karena berkaitan erat dengan masalah kesehatan.

Jadwal dokter sangatlah penting diantaranya untuk optimalisasi sumber daya, rumah sakit dapat memanfaatkan sumber daya manusia dengan lebih efisien. Penjadwalan dokter yang tepat waktu dan sesuai dengan kompetensi mereka akan membantu rumah sakit memaksimalkan potensi dokter yang ada. Hal ini akan mengurangi waktu tunggu pasien dan meningkatkan produktivitas staf medis.

Pelayanan pasien yang lebih cepat, manajemen jadwal dokter yang efisien akan memastikan bahwa pasien mendapatkan perawatan yang cepat dan tepat waktu. pasien dapat dengan mudah membuat janji temu dengan dokter yang tersedia sesuai dengan kebutuhan mereka. Ini akan mengurangi waktu tunggu pasien di rumah sakit dan meningkatkan kepuasan pasien.mengurangi risiko kesalahan dalam penjadwalan dokter. ini dapat menghindari konflik jadwal yang dapat mengganggu operasional rumah sakit. dokter dapat melihat jadwal mereka dengan jelas, menghindari tumpang tindih, dan mengatur jadwal libur dengan lebih baik.

## 1.2. Tujuan

1. Untuk mencegah konflik jadwal, double-booking, dan kesalahan penjadwalan.
2. Memudahkan pasien untuk membuat janji dengan dokter secara online.
3. Membantu mencegah kesalahan medis dan meningkatkan keamanan pasien.

## 1.3. Ruang Lingkup

Ruang lingkupnya :

1. Penjadwalan Dokter
2. Manajemen Jadwal
3. Pemberitahuan dan Pengingat

## 1.4. Manfaat

1. Aplikasi ini membantu meningkatkan efisiensi operasional dengan penjadwalan yang lebih baik dan otomatis
2. Menghindari konflik jadwal
3. Membantu mencegah kesalahan medis dan meningkatkan keamanan pasien
4. Memberikan perawatan yang lebih baik kepada pasien

# BAB 2

# DESKRIPSI TEMA

## 2.1. Tema Proyek

Aplikasi Pengelolaan Jadwal Dokter

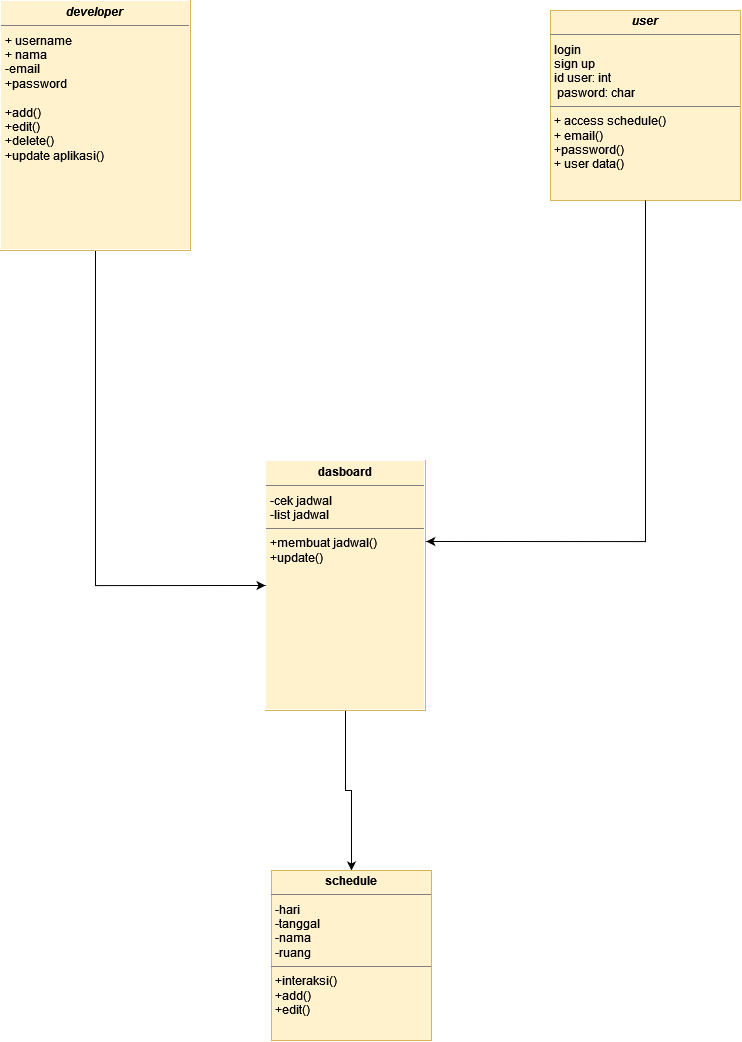
## 2.2. Fitur Utama

* Login
* Register
* Tambah jadwal
* Pengingat
* Cek jadwal
* Dashboard

# BAB 3

# CLASS DIAGRAM

## 3.1. Struktur Class



Gambar . Struktur Class

## 3.2. Hubungan Antar Class

### 

Gambar . Hubungan Antar CLass

### 

# BAB 4

# PENGEMBANGAN DART PBO

## Implementasi Konsep PBO

### Inheritance

1. class JadwalDokter extends Jadwal {
2. String \_spesialis;
3. JadwalDokter(String nama, *this*.\_spesialis, String ruangan, DateTime tanggal)
4. : *super*(nama, tanggal, ruangan);
5. *// Getter untuk mendapatkan nilai atribut spesialis*
6. String get spesialis => \_spesialis;
7. @override
8. void display() {
9. print(
10. "Nama Dokter: $nama, Spesialis: $\_spesialis, Ruangan: $\_ruangan, Tanggal: ${tanggal.toLocal()}");
11. }
12. }
13. class JadwalPerawat extends Jadwal {
14. String \_shift;
15. JadwalPerawat(String nama, *this*.\_shift, String ruangan, DateTime tanggal)
16. : *super*(nama, tanggal, ruangan);
17. *// Getter untuk mendapatkan nilai atribut shift*
18. String get shift => \_shift;
19. @override
20. void display() {
21. print(
22. "Nama Perawat: $nama, Shift: $\_shift, Ruangan: $\_ruangan ,Tanggal: ${tanggal.toLocal()}");
23. }
24. }
25. class JadwalPasien extends Jadwal {
26. String \_penyakit;
27. JadwalPasien(String nama, *this*.\_penyakit, String ruangan, DateTime tanggal)
28. : *super*(nama, tanggal, ruangan);
29. *// Getter untuk mendapatkan nilai atribut penyakit*
30. String get penyakit => \_penyakit;
31. @override
32. void display() {
33. print(
34. "Nama Pasien: $nama, Penyakit: $\_penyakit, Ruangan: $\_ruangan ,Tanggal: ${tanggal.toLocal()}");
35. }
36. }

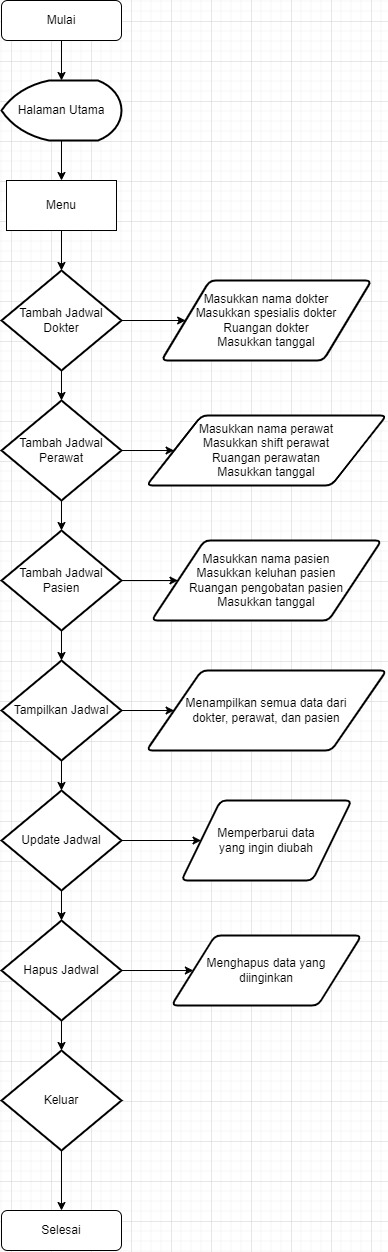
### 4.1.2. Encapsulation

1. class SistemJadwal {
2. List<Jadwal> \_jadwalList = [];
3. *// Getter untuk mendapatkan nilai atribut jadwalList*
4. List<Jadwal> get jadwalList => \_jadwalList;
5. *// Pilar PBO: Enkapsulasi*

### 4.1.3. Abstraction

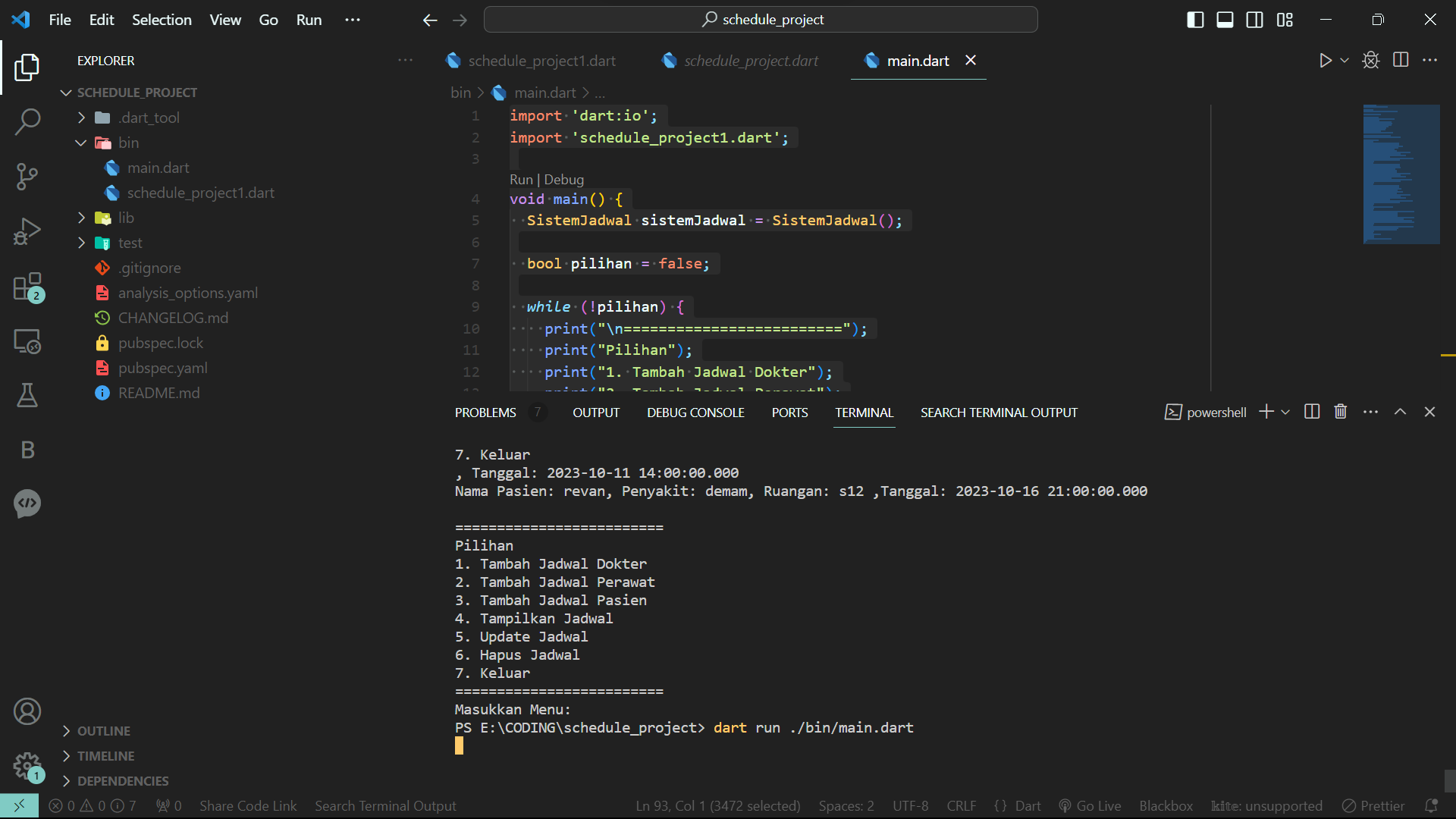
1. abstract class Jadwal {
2. String \_nama;
3. String \_ruangan;
4. DateTime \_tanggal;
5. Jadwal(*this*.\_nama, *this*.\_tanggal, *this*.\_ruangan);
6. *// Getter untuk mendapatkan nilai atribut nama*
7. String get nama => \_nama;
8. *//Getter untuk mendapatkan nilai atribut ruangan*
9. String ruangan() => \_ruangan;
10. *// Getter untuk mendapatkan nilai atribut tanggal*
11. DateTime get tanggal => \_tanggal;
12. *// Pilar PBO: Abstraksi*
13. void display();

## 4.2. Struktur Aplikasi



Gambar . Flowchart atau Struktur Aplikasi

## 4.3. Hasil



Gambar . Hasil Running nya

## 

# BAB 5

# PENGEMBANGAN APLIKASI FLUTTER

## 

## 5.1 Fitur

### 5.1.1. Halaman Homepage

Halaman homepage ini dibuat sebagai tampilan awal ketika pengguna mengunjungi Aplikasi Pengelolaan Jadwal Dokter dapat dibagian hasil.

### 5.1.2 Halaman Login Page

Halaman Form login ini dibuat sebagai tampilan form input untuk admin atau pengguna yaitu dengan mengisi form login dan memilih fitur yang akan di ikuti yang ada di Aplikasi Tools ini. Halaman Form Login dapat dilihat pada gambar dibagian hasil.

### 5.1.3. Halaman Edit page

Pada halaman ini anda dapat mengedit page sesuai dengan apa yang anda butuhkan untuk pengelolaan jadwal dokter.

### 5.1.4. Halaman tampilan data yang sudah di perbarui

Halaman tampilan ini adalah data yang sudah di perbarui oleh admin.

### 5.1.5. Halaman Menghapus Data

Halaman ini bisa menghapus data sesuai dengan apa yang anda butuhkan pada aplikasi tools ini dan bisa dilihat dibagian hasil.

### 5.1.6. Halaman Tampilan Data yang sudah dihapus

Pada halaman ini anda dapat menampilkan data yang sudah dihapus pada halaman menghapus data dan dilihat dibagian hasil.

### 5.1.7. Halaman Masukkan Data Baru

Pada halaman ini anda bisa memasukkan data yang baru atau data yang ingin anda tambahkan dari data sebelumnya dan bisa dilihat dibagian hasil.

### 5.1.8. Halaman Tampilan Data Yang Sudah Diperbarui

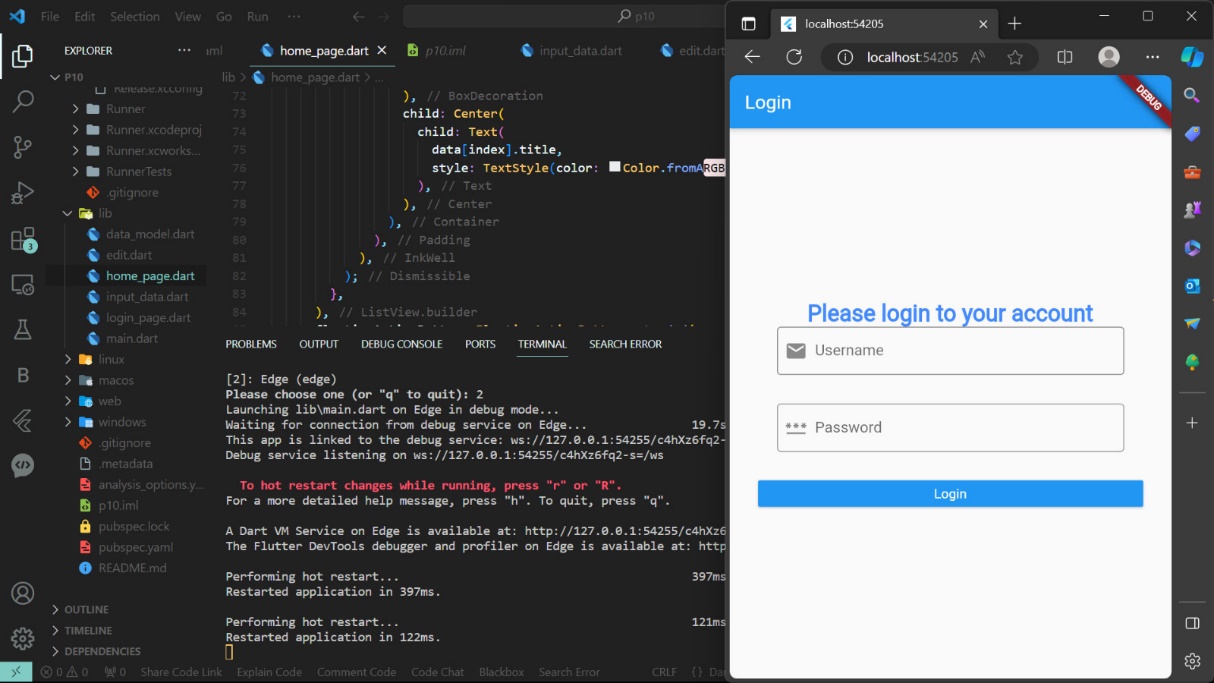
Halaman ini bisa melihat data yang sudah anda perbarui atau data yang sudah anda tambahkan sebelumnya dan bisa dilihat dibagian hasil.

## 5.2. Hasil

### 5.2.1. Halaman Homepage

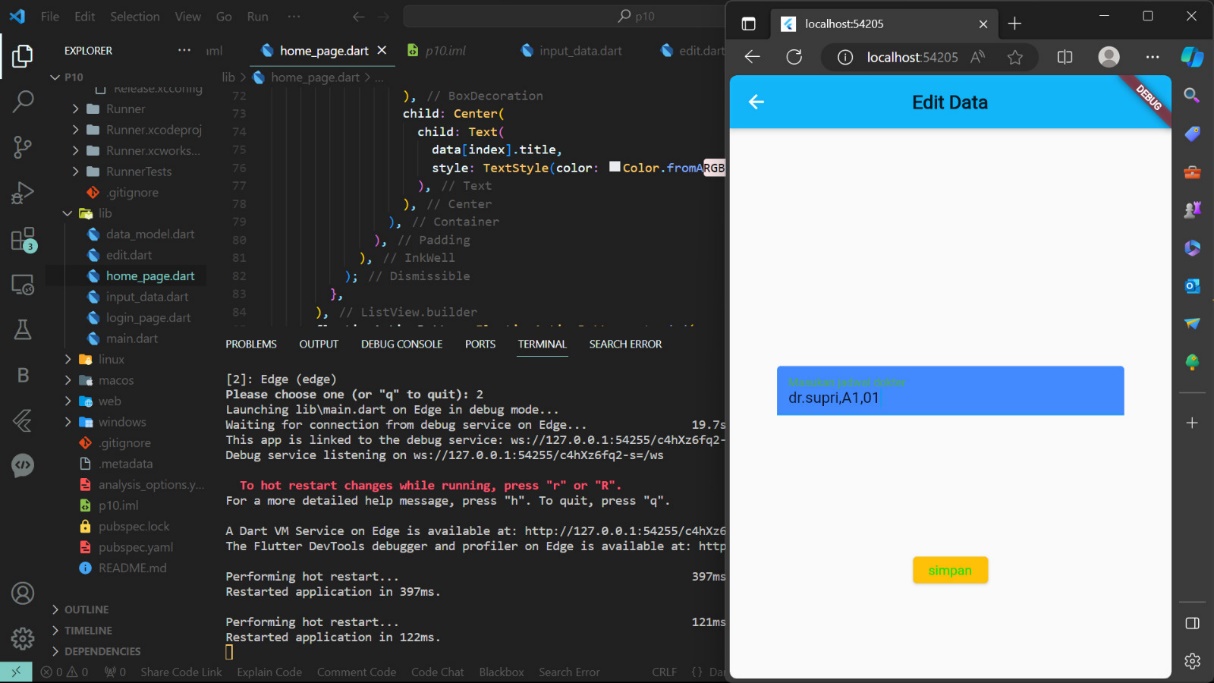
Gambar . Halaman Homepage

### 5.2.2. Halaman Login Page



Gambar . Halaman Login Page

### 5.2.3. Halaman Edit Page



Gambar . Halaman Edit Page

### 5.2.4. Halaman Tampilan Data Yang Sudah Diperbarui

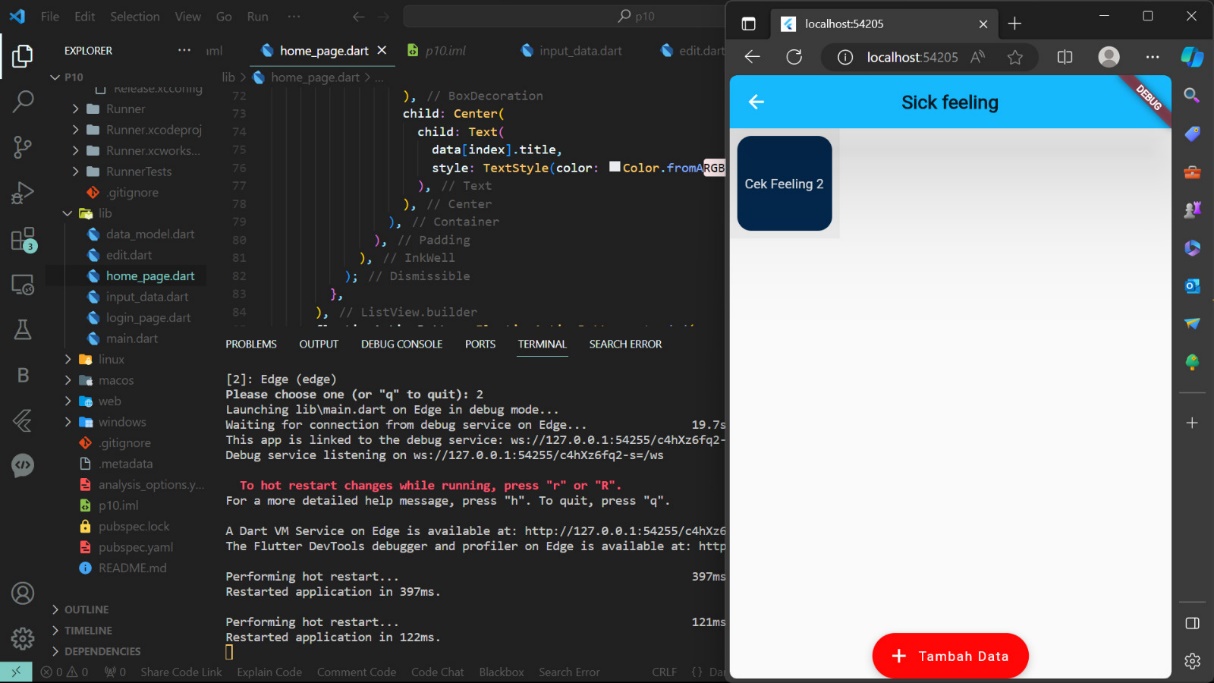
Gambar . Halaman tampilan data yang sudah diperbarui

### 5.2.5. Halaman Menghapus Data

### 

Gambar . Halaman menghapus data

### 5.2.6. Halaman Tampilan Data Yang Sudah Dihapus



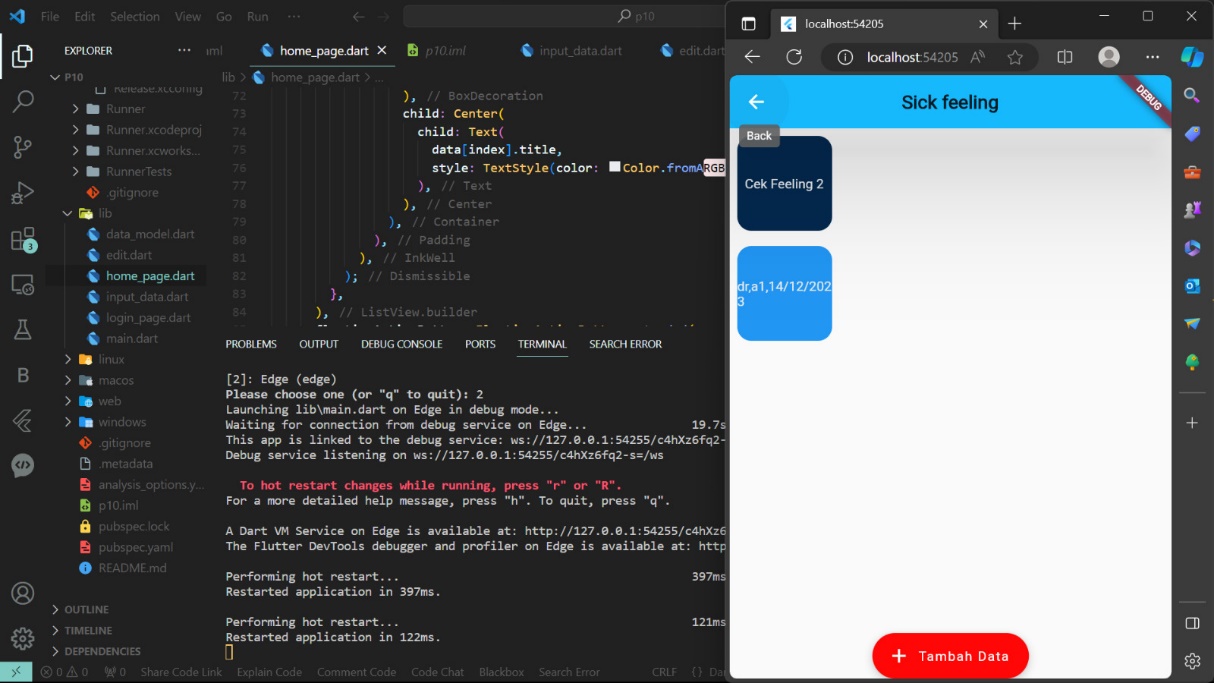
Gambar . Halaman tampilan data yang sudah dihapus

### 5.2.7. Halaman Masukkan Data Baru

### 

Gambar . Halaman masukkan data baru

### 5.1.8. HalamanTampilan Data Yang Sudah Diperbarui



Gambar . Halaman tampilan data yang sudah diperbarui

# BAB 6

# KESIMPULAN

## 6.1. Ringkasan

Secara umum aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Para ahli pun memiliki pengertian tersendiri mengenai apa itu aplikasi.

Metode dalam pembuatan Tugas Akhir ini menggunakan tahapan analisa, desain, pengkodean, pengujian, penerapan, dan pemeliharaan serta dapat memberikan solusi bagi para pengguna dalam menggunakan Aplikasi Tools ini.

## 6.2. Saran

Kami perlu banyak belajar agar lebih memahami bahasa pemrograman Dart dengan menggunakan Flutter supaya dapat membuat program yang lebih baik lagi kedepannya, karena kami rasa masih terdapat banyak kekurangan dalam hal fitur dan juga bug / error dalam program yang kami buat saat ini.

Kami sangat mengharapkan kritik dan saran yang membangun demi perbaikan laporan dan juga program kami selanjutnya.

# DAFTAR PUSTAKA

* Adiwijaya, Mohammad ; S, Kodrat Iman; Christyono, Yuli;. (2015).

PERANCANGAN GAME EDUKASI PLATFORM BELAJAR MATEMATIKA BERBASIS ANDROID MENGGUNAKAN

CONSTRUCT 2. *Transient*, 4.

* Aji, S., Pratmanto, D., & Saifudin. (2021). Implementasi Framework Laravel Dalam Perancangan Sistem Informasi Desa. *ejournal.bsi*, 7.
* Lararenjana, E. (2022, June 19). Aplikasi Adalah Program dengan Fungsi Tertentu, Ini Pengertian dan Jenisnya. *Merdeka*, p. 1.
* Norsapani, Cici;. (2021). PERANCANGAN APLIKASI MEDIA PEMBELAJARAN FLUTTER BERBASIS ANDROID. *UNIVERSITAS POTENSI UTAMA*, 20.
* Setiadi, A., Handayani, I., & Fadhilah, F. (2021). Perancangan Aplikasi Fit Your Weight Untuk Menghitung Berat Badan Ideal Berbasis Android.

*Technomedia Journal*, 5.

# LAMPIRAN

abstract class Jadwal {

  String \_nama;

  String \_ruangan;

  DateTime \_tanggal;

  Jadwal(*this*.\_nama, *this*.\_tanggal, *this*.\_ruangan);

*// Getter untuk mendapatkan nilai atribut nama*

  String get nama => \_nama;

*//Getter untuk mendapatkan nilai atribut ruangan*

  String ruangan() => \_ruangan;

*// Getter untuk mendapatkan nilai atribut tanggal*

  DateTime get tanggal => \_tanggal;

*// Pilar PBO: Abstraksi*

  void display();

}

class JadwalDokter extends Jadwal {

  String \_spesialis;

  JadwalDokter(String nama, *this*.\_spesialis, String ruangan, DateTime tanggal)

      : *super*(nama, tanggal, ruangan);

*// Getter untuk mendapatkan nilai atribut spesialis*

  String get spesialis => \_spesialis;

  @override

  void display() {

    print(

        "Nama Dokter: $nama, Spesialis: $\_spesialis, Ruangan: $\_ruangan, Tanggal: ${tanggal.toLocal()}");

  }

}

class JadwalPerawat extends Jadwal {

  String \_shift;

  JadwalPerawat(String nama, *this*.\_shift, String ruangan, DateTime tanggal)

      : *super*(nama, tanggal, ruangan);

*// Getter untuk mendapatkan nilai atribut shift*

  String get shift => \_shift;

  @override

  void display() {

    print(

        "Nama Perawat: $nama, Shift: $\_shift, Ruangan: $\_ruangan ,Tanggal: ${tanggal.toLocal()}");

  }

}

class JadwalPasien extends Jadwal {

  String \_penyakit;

  JadwalPasien(String nama, *this*.\_penyakit, String ruangan, DateTime tanggal)

      : *super*(nama, tanggal, ruangan);

*// Getter untuk mendapatkan nilai atribut penyakit*

  String get penyakit => \_penyakit;

  @override

  void display() {

    print(

        "Nama Pasien: $nama, Penyakit: $\_penyakit, Ruangan: $\_ruangan ,Tanggal: ${tanggal.toLocal()}");

  }

}

class SistemJadwal {

  List<Jadwal> \_jadwalList = [];

*// Getter untuk mendapatkan nilai atribut jadwalList*

  List<Jadwal> get jadwalList => \_jadwalList;

*// Pilar PBO: Enkapsulasi*

  void tambahJadwal(Jadwal jadwal) {

    \_jadwalList.add(jadwal);

  }

  void tampilkanJadwal() {

*if* (\_jadwalList.isEmpty) {

      print("Jadwal Kosong");

    } *else* {

*for* (var jadwal *in* \_jadwalList) {

        jadwal.display();

      }

    }

  }

  void updateJadwal(String namaLama, String namaBaru, DateTime tanggalBaru) {

*for* (var jadwal *in* \_jadwalList) {

*if* (jadwal.nama == namaLama) {

*// Pilar PBO: Enkapsulasi*

*// Menggunakan setter untuk mengubah nilai atribut yang bersifat privat*

        (jadwal as dynamic).\_nama = namaBaru;

        (jadwal as dynamic).\_tanggal = tanggalBaru;

      }

    }

  }

  void hapusJadwal(String nama) {

    \_jadwalList.removeWhere((jadwal) => jadwal.nama == nama);

  }

}

import 'dart:io';

import 'schedule\_project1.dart';

void main() {

  SistemJadwal sistemJadwal = SistemJadwal();

  bool pilihan = false;

*while* (!pilihan) {

    print("\n=========================");

    print("Pilihan");

    print("1. Tambah Jadwal Dokter");

    print("2. Tambah Jadwal Perawat");

    print("3. Tambah Jadwal Pasien");

    print("4. Tampilkan Jadwal");

    print("5. Update Jadwal");

    print("6. Hapus Jadwal");

    print("7. Keluar");

    print("=========================");

    stdout.write("Masukkan Menu: ");

    int pilih = int.parse(stdin.readLineSync()!);

*switch* (pilih) {

*case* 1:

        stdout.write("Masukkan nama dokter: ");

        var namaDokter = stdin.readLineSync()!;

        stdout.write("Masukkan spesialis dokter: ");

        var spesialis = stdin.readLineSync()!;

        stdout.write("Ruangan dokter: ");

        var ruangan = stdin.readLineSync()!;

        stdout.write("Masukkan tanggal (YYYY-MM-DD HH:mm): ");

        var inputTanggal = stdin.readLineSync()!;

        DateTime tanggal = DateTime.parse(inputTanggal);

        JadwalDokter jadwalDokter =

            JadwalDokter(namaDokter, spesialis, ruangan, tanggal);

        sistemJadwal.tambahJadwal(jadwalDokter);

*break*;

*case* 2:

        stdout.write("Masukkan nama perawat: ");

        var namaPerawat = stdin.readLineSync()!;

        stdout.write("Masukkan shift perawat: ");

        var shift = stdin.readLineSync()!;

        stdout.write("Ruangan perawatan: ");

        var ruangan = stdin.readLineSync()!;

        stdout.write("Masukkan tanggal (YYYY-MM-DD HH:mm): ");

        var inputTanggal = stdin.readLineSync()!;

        DateTime tanggal = DateTime.parse(inputTanggal);

        JadwalPerawat jadwalPerawat =

            JadwalPerawat(namaPerawat, shift, ruangan, tanggal);

        sistemJadwal.tambahJadwal(jadwalPerawat);

*break*;

*case* 3:

        stdout.write("Masukkan nama pasien: ");

        var namaPasien = stdin.readLineSync()!;

        stdout.write("Masukkan keluhan pasien: ");

        var penyakit = stdin.readLineSync()!;

        stdout.write("Ruangan pengobatan pasien: ");

        var ruangan = stdin.readLineSync()!;

        stdout.write("Masukkan tanggal (YYYY-MM-DD HH:mm): ");

        var inputTanggal = stdin.readLineSync()!;

        DateTime tanggal = DateTime.parse(inputTanggal);

        JadwalPasien jadwalPasien =

            JadwalPasien(namaPasien, penyakit, ruangan, tanggal);

        sistemJadwal.tambahJadwal(jadwalPasien);

*break*;

*case* 4:

        sistemJadwal.tampilkanJadwal();

*break*;

*case* 5:

        stdout.write("Masukkan nama jadwal yang ingin diupdate: ");

        var namaLama = stdin.readLineSync()!;

        stdout.write("Masukkan nama baru: ");

        var namaBaru = stdin.readLineSync()!;

        stdout.write("Masukkan tanggal baru (YYYY-MM-DD HH:mm): ");

        var inputTanggalBaru = stdin.readLineSync()!;

        DateTime tanggalBaru = DateTime.parse(inputTanggalBaru);

        sistemJadwal.updateJadwal(namaLama, namaBaru, tanggalBaru);

*break*;

*case* 6:

        stdout.write("Masukkan nama jadwal yang ingin dihapus: ");

        var namaHapus = stdin.readLineSync()!;

        sistemJadwal.hapusJadwal(namaHapus);

*break*;

*case* 7:

        pilihan = true;

*break*;

*default*:

        print("Pilihan tidak valid");

    }

  }

}

