LAPORAN PRAKTIKUM 3 MOBILE PROGAMMING CODING PROGRAM HALAMAN AWAL DAN PEMBUATAN REPOSITORY GITHUB

Dosen Pengampu:

Ghea Chandra Surawan, M.Pd.



KELOMPOK 1 DI SUSUN OLEH

NAMA : SHAFIRA AUREL AZZAHRA

NPM: 2213025005

PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN UNIVERSITAS LAMPUNG

2024

DAFTAR ISI

DAFTAR ISI	1
I. Waktu Praktikum	2
II. Alat dan Bahan	
III. Pembahasan Praktikum	2
IV. Logbook Hasil Praktikum	2
Pembuatan Repository Github	2
Gambar Desain Yang Akan Dibuat	3
Pembuatan Code Program Halaman Spalsh Screen	4
Pembuatan Code Program Halaman Sign In	7
Hasil Running Program Dengan Elmulator	8
V. Kesimpulan	11

I. Waktu Praktikum

Hari/Tanggal : Rabu/11 Agustus 2024

Waktu : 12.00 WIB

II. Alat dan Bahan

Alat : Laptop

Bahan : Visual Studio Code, Android Studio, Github

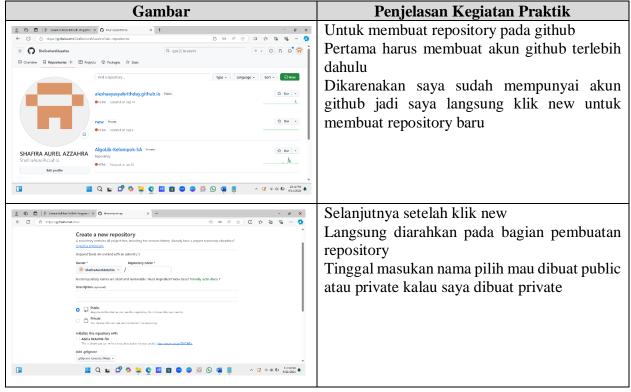
III. Pembahasan Praktikum

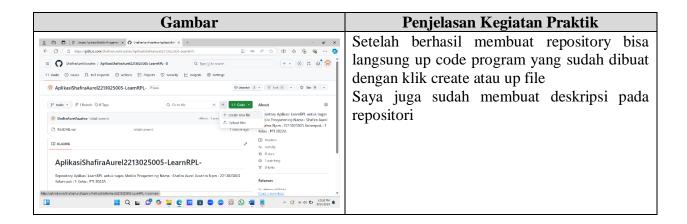
1. Pembuatan repository github

- 2. Gambar desain yang akan dibuat
- 3. Pembuatan code program Halaman Splash Screen
- 4. Pembuatan code program Halaman Sign In
- 5. Hasil running program dengan elmulator

IV. Logbook Hasil Praktikum

Pembuatan Repository Github



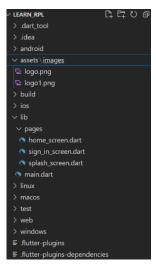


Gambar Desain Yang Akan Dibuat

Gambar Penjelasan Kegiatan Praktik		
LearnRPL Get Strated	Pada praktikum 3 saya akan membuat program untuk halaman aplikasi saya yaitu halaman awal yang berisi logo dari aplikasi saya yang terdapat tombol yang akan langsung mengarahkan user ke halaman sign in Spalsh Screen	
Sign in (Paul) LoamitPL Final Piscoverd Sign in Or Sign in With G Google	Selanjutnya saya juga akan membuat code program untuk halaman sign in user harus input email dan password Setelah user berhasil sign in nanti akan diarahkan langsung pada halaman home page Sign In Screen	

Pembuatan Code Program Halaman Spalsh Screen

Gambar



Penjelasan Kegiatan Praktik

Untuk membuat project pada fluterr

- 1. Buka vscode
- 2. Tekan ctrl+shift+p
- 3. Ketik flutter pilih new project pilih aplikasi
- 4. Setelah itu tempatin sesuai folder yang udh di buat
- 5. Selanjutnya buat folder dalama folder lib buat folder page untuk nanti isinya code program setiap halaman aplikasi
- 6. Selanjutnya buat folder assets untuk isinya gambar dan logo untuk aplikasi

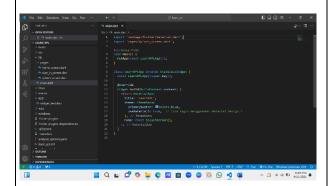
Setting Pada Bagian File pubspec.yaml

- 1. Nama Proyek: learn_rpl proyek Flutter baru.
- 2. Versi SDK Dart: >=2.18.0 <3.0.0.
- 3. Dependensi Utama:
 - 1) flutter: SDK Flutter.
 - 2) firebase_auth: Versi ^5.2.1 untuk autentikasi.
 - 3) firebase_auth_web: Versi ^5.13.0 untuk autentikasi web.
 - 4) firebase_core: Versi ^3.4.1 untuk inisialisasi Firebase.
- 4. Dependensi Pengembangan:
 - 1) flutter_test: Untuk pengujian.
 - 2) flutter lints: Untuk linter kode.
- 5. Konfigurasi Flutter:
 - 1) Menggunakan desain material.
 - 2) Menyertakan aset gambar logo.png dan logo1.png di assets/images/.

Gambar

Penjelasan Kegiatan Praktik

Code Program Pada Bagian File main.dart



1. Impor Paket:

- 'import
 'package:flutter/material.dart';':
 Mengimpor pustaka material design
 dari Flutter.
- 2) `import 'pages/splash_screen.dart';`: Mengimpor widget untuk layar splash dari file `splash_screen.dart`.

2. Fungsi Utama:

1) `void main() { runApp(const LearnRPLApp()); }`: Fungsi utama yang memulai aplikasi dengan `LearnRPLApp` sebagai widget utama.

3. Kelas `LearnRPLApp`:

- 1) `class LearnRPLApp extends StatelessWidget`: Kelas utama aplikasi yang tidak memiliki state (stateless).
- 2) `const LearnRPLApp({super.key});`: Konstruktor dengan key opsional.
- 3) `Widget build(BuildContext context)`: Metode untuk membangun antarmuka pengguna.

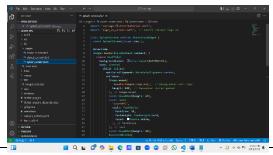
`MaterialApp`: Widget dasar aplikasi yang menyediakan tema dan rute.

`title: 'LearnRPL'`: Judul aplikasi.

`theme: ThemeData(...)`: Mengatur tema aplikasi dengan warna primer biru dan Material Design 3.

`home: const SplashScreen()`: Menetapkan `SplashScreen` sebagai tampilan utama saat aplikasi dimulai.

Code Program Pada Bagian File Halaman Awal (Splash Screen)



1. Impor Paket:

- 'import
 'package:flutter/material.dart';`:
 Mengimpor pustaka material design
 dari Flutter.
- 2) `import 'sign_in_screen.dart';`:

 Mengimpor halaman Sign In dari file
 `sign_in_screen.dart`.

Gambar Codingan: import 'package:flutter/material.dart'; import 'sign_in_screen.dart'; // Import halaman Sign In class SplashScreen extends StatelessWidget { const SplashScreen({super.key}); @override Widget build(BuildContext context) { return Scaffold(backgroundColor: const Color(0xFF093791), body: Center(child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [Image.assett 'assets/images/logo.png', // Menggunakan path logo height: 100, // Sesuaikan ukuran gambar const SizedBox(height: 20), const Text('LearnRPL', style: TextStyle(fontSize: 30, fontWeight: FontWeight.bold, color: Colors.white,), const SizedBox(height: 20), ElevatedButton(onPressed: () { // Pindah ke Sign In Screen Navigator.push(context, MaterialPageRoute(builder: (context) => const SignInScreen()), child: const Text("Get Started", style: TextStyle(color: Color(0xFF093791), // Mengubah warna teks jadi biru fontWeight: FontWeight.bold, // Membuat teks menjadi bold),),

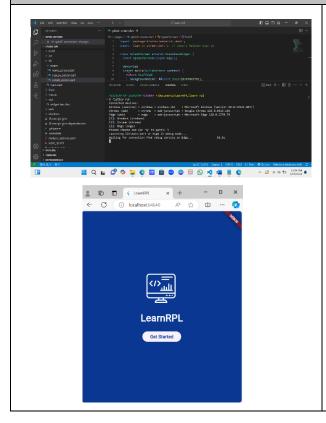
Penjelasan Kegiatan Praktik

- 2. Kelas `SplashScreen`:
 - 1) `class SplashScreen extends StatelessWidget`: Kelas yang merupakan widget stateless untuk layar splash.
 - 2) `const SplashScreen({super.key});`: Konstruktor dengan key opsional.

3. Metode `build`:

- 1) `Widget build(BuildContext context)`: Metode untuk membangun antarmuka pengguna layar splash.
- 2) `Scaffold`: Widget dasar yang menyediakan struktur layout.
- 3) `backgroundColor: const Color(0xFF093791)`: Mengatur warna latar belakang menjadi biru gelap.
- 4) `body: Center(...): Menempatkan konten di tengah layar.
- 5) 'Column': Menyusun widget secara vertikal.
- 6) `mainAxisAlignment: MainAxisAlignment.center`: Mengatur widget di dalam kolom agar berada di tengah secara vertikal.
- 7) `Image.asset(...)`: Menampilkan gambar logo dengan ukuran tinggi 100 piksel.
- 8) `SizedBox(height: Menambahkan jarak vertikal antara widget.
- 9) `Text(...)`: Menampilkan teks "LearnRPL" dengan ukuran font 30, tebal, dan berwarna putih.
- 10) `ElevatedButton(...)`: Tombol dengan aksi untuk berpindah ke layar Sign In.
- 11) `onPressed: () { Navigator.push(...) }`: Menetapkan aksi saat tombol ditekan untuk membuka `SignInScreen`.
- 12) child: const Text(...): Teks pada tombol dengan warna biru dan teks tebal.

Gambar Penjelasan Kegiatan Praktik Hasil Running Menggunakan Edge



Setelah saya membuat code program untuk bagian halaman awal splash screen saya mencoba untuk running dengan cara ketikan di terminal:

Flutter run

Pembuatan Code Program Halaman Sign In

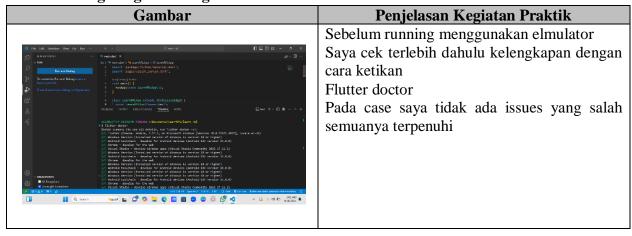
The fact delicates there for the service states of the service sta

Penjelasan Kegiatan Praktik

- 1. Impor Paket:
 - 1) Mengimpor pustaka material Flutter dan halaman `home_screen.dart`.
- 2. Kelas `SignInScreen`:
 - 1) Widget stateless untuk layar Sign In dengan konstruktor opsional.
- 3. Metode `build`:
 - 1) `Scaffold` dengan dua bagian utama:
 - 2) Bagian Atas:
 - `Container` putih dengan logo dan teks "LearnRPL" di tengah.
 - 3) Bagian Bawah:
 - 'Container' biru dengan sudut membulat.

Gambar Penjelasan Kegiatan Praktik Formulir login dengan dua `TextField` untuk email dan password. - Tombol "Sign In" untuk berpindah ke `HomeScreen` saat ditekan. Program membuat tampilan layar Sign In dengan dua bagian, bagian atas untuk logo dan teks, dan bagian bawah untuk formulir login dan tombol untuk berpindah ke layar Home. Hasil Running Menggunakan Edge Setelah saya membuat code program untuk bagian halaman awal splash screen saya mencoba untuk running dengan cara ketikan di terminal: Flutter run LearnRPL

Hasil Running Program Dengan Elmulator



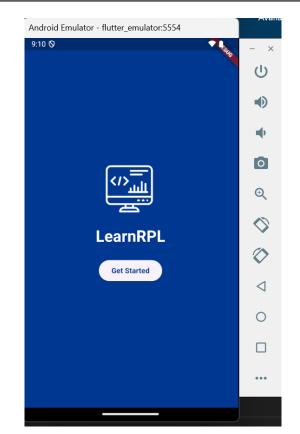


Cambar | International Continues | Internat

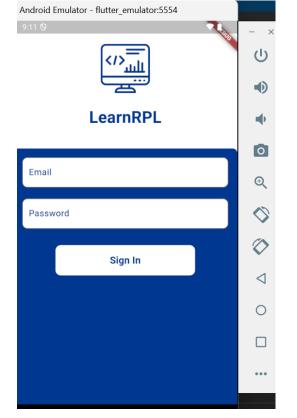
Penjelasan Kegiatan Praktik

Hasil Tampilan





Tampilan Halaman Awal Splash Screen



Tampilan Halaman Sign In

V. Kesimpulan

Dalam praktikum Mobile Programming yang saya lakukan secara individu, saya mempraktikkan pembuatan repository GitHub dan pengembangan aplikasi menggunakan Flutter. Meskipun beberapa kendala muncul, seperti sinkronisasi repository dengan GitHub dan pengaturan dependency pada proyek Flutter, saya berhasil menemukan solusi melalui panduan dan pencarian mandiri. Praktikum ini memberikan saya pemahaman yang lebih mendalam tentang pentingnya penggunaan alat pengelolaan versi dan kemampuan troubleshooting dalam proses pengembangan aplikasi mobile.