

SOFTWARE TESTING AND QUALITY MEET & RATE



Dosen Pengampu : Laila Ma'rifatul Azizah, S.Kom, M.I.M Disusun
Oleh Kelompok C5:

Davit Putra Hartono	(20210140026)
Wahdini Amri Kompani	(20210140154)
Nanang Puji Sulistiyono	(20210140157)
Muhammad Panji Al-Biruni	(20210140007)
Muhammad Khaidar Royani	(20210140128)
Rania Gisela Pramesty	(20210140133)
Teuku Muhammad Attar Fadhillah	(20210140113)

UNIVERSITAS MUHAMMADIYAH YOGYAKARTA

FAKULTAS TEKNIK

PRODI TEKNOLOGI INFORMASI TAHUN

2022/2023

Daftar Isi

Bab I. Pendahuluan	3
1.1 Latar Belakang	3
1.2 Tujuan.....	3
Bab II. Proses Produksi	5
2.1.User Requirement	5
2.2.Test Plan	6
2.3.RTM	10
Bab III. Hasil dan Pembahasan	17
3.1 Unit testing.....	17
3.2 Integration Testing.....	20
3.3 Code Documentation	24
3.4 Bug report	39
3.5 Summary Report	43
Bab IV. Kesimpulan	47

Bab I. Pendahuluan

1.1 Latar Belakang

Dalam era perkembangan teknologi digital, aplikasi telah menjadi alat yang kuat untuk memfasilitasi kehidupan sehari-hari. Aplikasi-aplikasi tersebut telah memberikan kemudahan akses informasi, pemesanan layanan, dan interaksi sosial. Sebagai konsekuensi, permintaan akan aplikasi yang membantu dalam mengelola dan meningkatkan kualitas hidup di kota-kota pun semakin meningkat.

Kualitas hidup di suatu kota dapat diukur melalui berbagai aspek, termasuk infrastruktur, transportasi, keamanan, lingkungan, ketersediaan layanan publik, serta kesempatan ekonomi dan pendidikan. Namun, tidak semua kota mampu memberikan standar hidup yang setara dan berkualitas bagi penduduknya.

Banyak kota yang menghadapi tantangan dalam mengidentifikasi dan mengatasi kekurangan yang ada. Pengambilan keputusan yang tidak efisien dan kurangnya pemahaman yang komprehensif tentang masalah yang dihadapi oleh kota dapat menyebabkan ketimpangan dan ketidakpuasan masyarakat.

Dalam rangka meningkatkan kualitas hidup di suatu kota, kami memperkenalkan aplikasi "Meet and Rate". Aplikasi ini bertujuan untuk memberikan pendekatan holistik dalam mengevaluasi kota dan memberikan informasi yang relevan kepada pengguna.

Aplikasi "Meet and Rate" akan memberikan kesempatan bagi penduduk kota, pengunjung, dan pemerintah untuk berpartisipasi dalam menilai berbagai aspek kualitas hidup. Pengguna dapat memberikan rating dan ulasan terhadap berbagai elemen seperti transportasi, keamanan, fasilitas umum, kebersihan, kualitas lingkungan, serta ketersediaan fasilitas ekonomi dan pendidikan.

1.2 Tujuan

1. Memastikan Keandalan dan Kinerja Aplikasi: Tujuan utama dari testing aplikasi "Meet and Rate" adalah untuk memastikan bahwa aplikasi berjalan dengan baik dan memberikan kinerja yang handal. Pengujian ini melibatkan identifikasi dan perbaikan bug atau kesalahan yang mungkin terjadi dalam aplikasi, serta memastikan bahwa aplikasi dapat berfungsi dengan baik di berbagai perangkat dan sistem operasi.
2. Memverifikasi Fungsionalitas Fitur Aplikasi: Laporan ini bertujuan untuk menguji dan memverifikasi fungsionalitas setiap fitur yang disediakan oleh aplikasi "Meet and Rate". Hal ini meliputi pengujian fitur daftar akun, fitur login, fitur penilaian dan ulasan kualitas suatu kota, fitur history, dan fitur logout. Tujuan dari pengujian ini adalah untuk memastikan bahwa semua fitur

berfungsi sebagaimana mestinya dan memberikan pengalaman pengguna yang memuaskan.

3. Mengevaluasi Pengalaman Pengguna: Salah satu tujuan penting dari pengujian adalah untuk mengevaluasi pengalaman pengguna saat menggunakan aplikasi "Meet and Rate". Pengujian ini melibatkan penilaian keseluruhan antarmuka pengguna, kemudahan penggunaan, responsivitas, dan kepuasan pengguna. Dengan memahami pengalaman pengguna, perbaikan dan pengembangan dapat dilakukan untuk meningkatkan usability dan memberikan pengalaman yang lebih baik bagi pengguna aplikasi.

Bab II. Proses Produksi

2.1. User Requirement

Fitur:

- a. Login
 - b. Sign Up
 - c. Rate Kota
 - d. History
 - e. Log Out
- Pertanyaan:

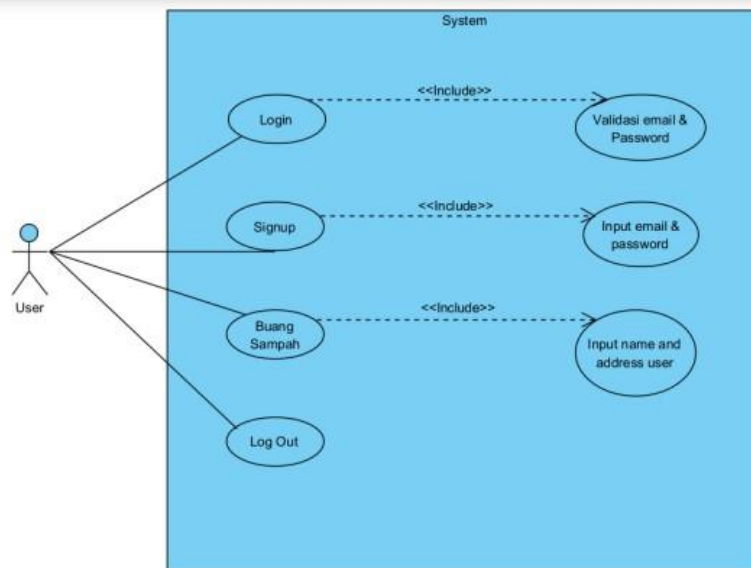
- a. Bagaimana cara melakukan login ke aplikasi?
- b. Bagaimana cara mendaftar atau membuat akun baru di aplikasi?
- c. Bagaimana cara melakukan rating kota pada aplikasi ini?
- d. Bagaimana cara melihat history rating kita pada aplikasi ini?
- e. Bagaimana cara logout dari aplikasi?

Jawaban

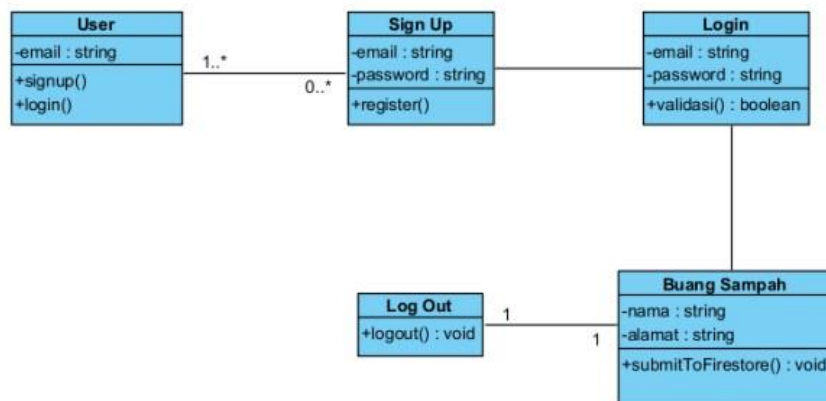
- a. Untuk melakukan login ke aplikasi, pengguna harus memasukkan email dan kata sandi (password) pada halaman login.
- b. Untuk mendaftar atau membuat akun baru di aplikasi, pengguna harus mengisi formulir pendaftaran yang berisi alamat email dan kata sandi.
- c. Untuk merating kota pada aplikasi, pengguna harus mengakses fitur "Rate a City" di dalam aplikasi, kemudian mengisi semua kolom yang sudah tersedia.
- d. Untuk melihat history rating kita, kita dapat mengakses fitur history yang sudah tersedia di dalam aplikasi.
- e. Untuk logout dari aplikasi, pengguna dapat menekan tombol logout pada halaman utama bagian bawah.

Diagram UML

Use Case Diagram



Class Diagram:



2.2.Test Plan

2.2.1 Perkenalan

Test Plan dirancang untuk menentukan ruang lingkup, pendekatan, sumber daya, dan jadwal semua kegiatan pengujian proyek Aplikasi Kebersihan Kota. Rencana tersebut mengidentifikasi item yang akan diuji, fitur yang akan diuji, jenis pengujian yang akan dilakukan, personil yang bertanggung jawab untuk pengujian, sumber daya dan jadwal yang diperlukan untuk menyelesaikan pengujian, dan risiko yang terkait dengan rencana tersebut.

2.2.1.1 Lingkup

1.1.1 Dalam Ruang Lingkup

Semua fitur Aplikasi Kebersihan Kota yang terdefinisi dalam spesifikasi kebutuhan perangkat lunak perlu diunggulkan

Nama Modul	Peran yang Berlaku	Keterangan
Sign Up (Registrasi)	User	User: User dapat melakukan registrasi akun dengan memasukkan email dan password.
Log In	User	User: User dapat melakukan login menggunakan akun yang telah dibuat.
Rating	User	User: User dapat memberikan rating ke suatu kota dengan mengisi form yang terdiri dari nama kota, tanggal, kesan dan rating. Admin: Admin dapat melihat data rating dari form yang telah dimasukan oleh user
	Admin	
Histori	User	User: User dapat melihat histori/riwayat penilaian
Log Out	User	User: User dapat keluar akun melalui fitur logout

1.1.2 Di Luar Cakupan

Fitur ini tidak diuji karena tidak termasuk dalam spesifikasi kebutuhan perangkat lunak

- Database
- Keamanan dan Kinerja Situs Web

1.2 Sasaran Mutu

Tujuan pengujian adalah untuk memverifikasi Fungsi aplikasi BIMESA , proyek harus fokus pada pengujian fungsi sign up, log in, buang sampah dan logout untuk menjamin semua operasi ini dapat bekerja secara normal di lingkungan nyata.

1.3 Peran dan Tanggung Jawab

Proyek harus menggunakan anggota outsourcing sebagai penguji untuk menghemat biaya proyek.

No	Anggota	Tugas
1.	Test Manager (Davit)	Kelola seluruh proyek Tentukan arah proyekDapatkan sumber daya yang sesuai
2.	Test (Nanang) (Khaidar)	Mengidentifikasi dan menjelaskan teknik/alat/arsitektur otomasi pengujian yang sesuai Verifikasi dan nilai Pendekatan Pengujian Jalankan pengujian, Catat hasil, Laporkan kerusakan. Anggota outsourcing
3.	Developer in Test (Attar)	Terapkan kasus uji, program uji, rangkaian uji, dll.
4.	Tes Administrator (AI)	Membangun dan memastikan lingkungan pengujian dan aset dikelola dan dipelihara. Mendukung Penguji untuk menggunakan lingkungan pengujian untuk eksekusi pengujian
5.	SQA Members (Dini) (Rania)	Bertanggung jawab atas jaminan kualitas Periksa untuk memastikan apakah proses pengujian memenuhi persyaratan yang ditentukan

2. Metodologi Uji 2.1

Tingkat Ujian

Di proyek aplikasi Meet And Rate, ada beberapa jenis pengujian yang harus dilakukan.

- Pengujian Unit
- Penerimaan Pengguna

2.2 Kriteria Penanganan dan Persyaratan Dimulainya Kembali

Jika anggota tim melaporkan bahwa ada 40% kasus pengujian yang gagal , tunda pengujian hingga tim pengembangan memperbaiki semua kasus yang gagal.

2.3 Ketuntasan Tes

- Menentukan kriteria yang menunjukkan keberhasilan penyelesaian fase pengujian
- Ibarat wajib 100% kecuali alasan yang jelas diberikan.
- kelulusan adalah 80%, mencapai tingkat kelulusan adalah wajib

2.4 Tugas proyek dan perkiraan serta jadwal

Tugas	Anggota	Perkiraan usaha
Buat spesifikasi pengujian	Desainer Tes	17 jam kerja
Lakukan Eksekusi Tes	Penguji, Administrator Tes	8 jam kerja
Laporan pengujian	Penguji	1 jam kerja
Pengiriman Tes		2 jam kerja
Total		28 jam kerja

Jadwalkan untuk menyelesaikan tugas-tugas ini

3. Hasil Uji

Kiriman tes disediakan seperti di bawah ini

Sebelum tahap pengujian

- Dokumen rencana pengujian.
- Dokumen kasus uji
- Spesifikasi Desain Uji.

Selama pengujian

- Simulator Alat Uji.
- Uji Data
- Test Trace-ability Matrix
- Log kesalahan dan log eksekusi.

Setelah siklus pengujian selesai

- Hasil Uji/laporan
- Laporan Cacat
- Panduan prosedur pemasangan/pengujian
- Catatan rilis

4. Kebutuhan Sumber Daya & Lingkungan

4.1 Alat Pengujian

TID AK	Sumber daya	Deskripsi
1.	Server	Perlu server Database yang menginstal server MySQL Server web yang menginstal Apache Server
2.	Jaringan	Siapkan LAN Gigabit dan 1 jalur internet dengan kecepatan minimal 5 Mb/s
3.	Komputer	Setidaknya 4 komputer menjalankan Windows 7, Ram 2GB, CPU 3.4GHZ

2.3.RTM

BR#	Module Name	Applicable Roles	Description
B1	Sign Up (Registrasi)	User	User: User dapat melakukan registrasi akun dengan memasukkan email dan password
B2	Login	User Admin	User: User dapat melakukan login menggunakan akun yang telah dibuat Admin: Admin dapat melakukan login menggunakan akun admin yang disediakan
B3	Rating	User	User: User dapat memberikan rating ke suatu kota dengan mengisi form yang terdiri dari nama kota, tanggal, mengunggah foto serta kesan dan rating
B4	Histori	User	User: User dapat melihat histori/riwayat penilaian
B5	Log Out	User Admin	User & Admin: User dapat keluar akun melalui fitur logout
B6	Lupa kata sandi	User	User: User dapat mereset kata sandi yang telah dibuat
B7	Hapus akun	User	User: User dapat menghapus akun yang telah dibuat
B8	Edit Profil	User	User & Admin: User dapat mengedit profil pada akun
B9	Pencarian (searching)	User	User: User dapat mencari riwayat penilaian kota dengan mengetikkan keyword pada fitur search di menu history
B10	Beranda	User Admin	User: User dapat melihat unggahan postingan penilaian dari user lain Admin: Admin dapat melihat dan menghapus unggahan postingan penilaian yang telah dimasukan oleh user
B11	Reset Kata Sandi	User	User: User dapat reset kata sandi

BR(Business requirements) adalah pernyataan formal tentang apa yang diharapkan dari suatu aplikasi. Business requirements menggambarkan kebutuhan-kebutuhan yang harus dipenuhi oleh solusi atau sistem yang sedang dikembangkan atau diperbaiki.

2. Technical Requirements

Technical Requirements

New Account/User Sign Up

- T1 User Name – Angka tidak diperbolehkan
- T2 User Name – Spesial karakter tidak diperbolehkan
- T3 User Name - Nama customer tidak boleh kosong
- T4 User Name - Karakter pertama tidak boleh ada jarak
- T5 User Email - Email customer tidak boleh kosong
- T6 User Email - Karakter pertama tidak boleh ada jarak
- T7 User Email - Alamat email tidak valid
- T8 User Password - Kata sandi tidak boleh kosong
- T9 User Password - Karakter pertama tidak boleh ada jarak
- T10 User Password – Password minimal harus berisi enam karakter

User Login

- T11 Email atau nama user tidak boleh kosong
- T12 Kata sandi user tidak boleh kosong
- T13 Jika Email/nama user dan kata sandi benar. Login

Reset Kata sandi

- T14 Email atau username tidak boleh kosong
- T15 Kata sandi baru tidak boleh kosong
- T16 Kata sandi harus berisi minimal enam karakter
- T17 Pilih kata sandi yang sulit
- T18 Konfirmasi kata sandi tidak boleh kosong
- T19 Password tidak cocok

Fitur Rate a City

- T20 Kolom nama kota tidak boleh kosong
- T21 Kolom nama kota tidak boleh menggunakan angka
- T22 Kolom nama kota tidak boleh karakter pertamanya berupa jarak (spasi)
- T23 Kolom nama kota tidak boleh menggunakan spesial karakter
- T24 Kolom tanggal kunjungan tidak boleh kosong
- T25 Kolom tanggal kunjungan tidak boleh menggunakan huruf
- T26 Kolom tanggal kunjungan tidak boleh karakter pertamanya berupa jarak (spasi)
- T27 Kolom tanggal kunjungan tidak boleh menggunakan spesial karakter
- T28 Kolom pendapat user tidak boleh kosong
- T29 Kolom pendapat user tidak boleh karakter pertamanya berupa jarak (spasi)
- T30 Kolom rating kota tidak boleh kosong
- T31 Kolom rating kota tidak boleh menggunakan huruf
- T32 Kolom rating kota tidak boleh menggunakan spesial karakter
- T33 Kolom rating kota tidak boleh karakter pertamanya berupa jarak (spasi)
- T34 Kolom rating kota angka penilaian tidak boleh lebih dari 10
- T35 Jika kolom yang diisi sesuai/benar, tombol rate dapat ditekan (sukses)

Fitur History

- T36 User tidak dapat mengedit ulasan (rate) yang ada di history

User LogOut

- T37 User harus menekan tombol "iya/tidak" untuk keluar dari aplikasi

Technical requirements adalah spesifikasi teknis yang menjelaskan persyaratan dan spesifikasi yang harus dipenuhi oleh suatu sistem, produk, atau solusi teknologi. Technical requirements digunakan untuk merancang, mengembangkan, dan menguji solusi teknis guna memastikan bahwa solusi tersebut memenuhi standar teknis yang diharapkan.

3. Functional Validation

Sign Up					
User					
F1	User tidak dapat menggunakan email yang sudah terdaftar				
F2	Jika nama user yang di isi terdapat spesial karakter, maka sistem menampilkan erorr				
F3	Kata sandi yang user isi terdapat minimal enam karakter, dapat berupa huruf, angka maupun simbol				
Login					
User					
F4	User hanya bisa login menggunakan email dan password yang telah terdaftar				
F5	User hanya bisa login menggunakan kata sandi yang telah di daftarkan				
F6	Jika email atau kata sandi salah, maka sistem akan menampilkan pesan erorr				
Reset Kata Sandi					
User					
F7	Jika kata sandi lama tidak valid, sistem akan menampilkan pesan error				
F8	Kata sandi dapat berupa huruf, angka maupun simbol				
Rate Kebersihan					
User					
F9	User dapat menginput nama kota, tanggal kejadian, tanggapan tentang kota, dan rating terhadap kota				
Admin					
F10	Admin dapat melihat hasil rating para user				
History					
User					
F11	User dapat melihat history dari rating sampah yang telah dilakukan				
Admin					
F12	Sistem akan menampilkan history rating, saat user selesai memberikan rating				
Logout					
User					
F13	Saat user logout, sistem akan mengeluarkan user dari laman aplikasi				
F14	Jika ingin menggunakan aplikasi user harus login kembali				

functional validation adalah proses menguji dan memastikan bahwa sebuah aplikasi, sistem, atau produk memenuhi persyaratan fungsional yang telah ditentukan. Functional validation dilakukan untuk memverifikasi bahwa sistem berperilaku sesuai dengan harapan dan mampu menjalankan fungsi-fungsi yang diperlukan untuk mencapai tujuan bisnis yang ditetapkan.

4. Test Scenario

Test Scenario #	Requirement ID	Test Scenario Description	Test Cases
1	K1.1	mengecek fungsi Sign Up	1. ujcoba memasukkan email dan password yang benar 2. ujcoba membuat password dengan angka 3. ujcoba membuat password dengan huruf 4. ujcoba membuat password dengan karakter 5. ujcoba membuat password dengan huruf kapital
2	K1.2	mengecek fungsi login	1. ujcoba memasukkan email dan password yang telah terdaftar 2. ujcoba memasukkan email yang salah dan password yang benar 3. ujcoba memasukkan email yang benar dan password yang salah 4. ujcoba memasukkan email dan password yang salah 5. ujcoba login dengan tidak memasukkan email dan password 6. ujcoba membuat email yang terdaftar dengan memasukkan pasword yang salah
3	K1.3	mengecek fungsi rating	1. ujcoba memasukkan rating dengan memasukkan kota 2. ujcoba memasukkan rating dengan memasukkan tanggal 3. ujcoba memasukkan rating dengan memasukkan kesan dan rating
4	K1.4	mengecek fungsi histori	1. ujcoba menampilkan histori/riwayat kota 2. ujcoba menampilkan histori/riwayat tanggal 3. ujcoba menampilkan histori/riwayat kesan 4. ujcoba menampilkan histori/riwayat rating
5	K1.5	mengecek fungsi log out	1. ujcoba mengeluarkan akun dari aplikasi 2. ujcoba menampilkan kembali menu login

Test scenario adalah rangkaian tujuan untuk menguji fitur, fungsi, atau perilaku tertentu dari suatu aplikasi, sistem, atau produk. Skenario pengujian menggambarkan situasi tau kondisi tertentu yang akan diuji untuk memverifikasi apakah sistem sesuai dengan apa yang diharapkan dan memenuhi persyaratan yang telah ditentukan.

5. Test Case

Test Case ID	CV_301	Test Case Description	Test fungsi sign up ke aplikasi
Created By	David	Reviewed By	David
QA Tester's Log	login beta version	Version	Beta
Tester's Name	David PH	Date Tested	26-Maret-2023
		Test Case (Pass/Fail/Not Executed)	Berhasil
S #	Prerequisites:	S #	Test Data Requirement
1	Akses ke aplikasi	1	email = salman@gmail.com
2		2	password = salman123
3		3	
4		4	
Test Conditions	Mendaftarkan akun dengan mengisi email dan kata sandi yang akan digunakan di menu login		
Step #	Step Details	Expected Results	Actual Results
1	Membuka Aplikasi	Aplikasi dapat dibuka	Seperti yang diharapkan
2	Menekan tombol sign up here	Tombol dapat ditekan dan berpindah halaman	Seperti yang diharapkan
3	Memasukan email & password	User dapat mengisi email & password	Seperti yang diharapkan
4	Menekan tombol sign up	User mengkonfirmasi email & password yang akan digunakan di halaman login	Terjadi Kesalahan/Error
			Gagal
Test Case ID	CV_302	Test Case Description	Test fungsi login ke aplikasi
Created By	David	Reviewed By	David
QA Tester's Log	login beta version	Version	Beta
Tester's Name	David PH	Date Tested	26-Maret-2023
		Test Case (Pass/Fail/Not Executed)	Berhasil
S #	Prerequisites:	S #	Test Data Requirement
1	Akses ke aplikasi	1	email = salman@gmail.com
2		2	password = salman123
3		3	
4		4	
Test Conditions	User memberikan ulasan kota yang sedang dikunjungi		
Step #	Step Details	Expected Results	Actual Results
1	Membuka Aplikasi	Aplikasi dapat dibuka	Seperti yang diharapkan
2	Memasukan email & password	Kolom dapat diisi	Seperti yang diharapkan
3	Menekan tombol login	User dapat login ke aplikasi dan berpindah ke halaman utama	Seperti yang diharapkan
4			Berhasil
Test Case ID	CV_003	Test Case Description	Test fungsi rate a city ke aplikasi
Created By	David	Reviewed By	David
QA Tester's Log	login beta version	Version	Beta
Tester's Name	David PH	Date Tested	26-Maret-2023
		Test Case (Pass/Fail/Not Executed)	Berhasil
Tester's Name	David PH	Date Tested	26-Maret-2023
		Test Case (Pass/Fail/Not Executed)	Berhasil
S #	Prerequisites:	S #	Test Data Requirement
1	Akses ke aplikasi	1	email = salman@gmail.com
2		2	password = salman123
3		3	
4		4	
Test Conditions	Verifikasi saat memasukkan id pengguna dan kata sandi yang valid, user dapat login		
Step #	Step Details	Expected Results	Actual Results
1	Membuka Aplikasi	Aplikasi dapat dibuka	Seperti yang diharapkan
2	Menekan fitur rate a city	Halaman utama berpindah ke halaman rate a city	Seperti yang diharapkan
3	Mengisikan nama kota	Kolom nama kota dapat diisi	Seperti yang diharapkan
4	Mengisikan tanggal	Kolom tanggal dapat diisi	Seperti yang diharapkan
5	Mengisikan pendapat	Kolom pendapat dapat diisi	Seperti yang diharapkan
			Berhasil

Tester's Name		David P.H.		Date Tested		26-Maret-2023		Test Case (Pass/Fail/Not Executed)		Berhasil	
S #		Prerequisites:		S #		Test Data Requirement					
1		Akses ke aplikasi		1		email = salman@gmail.com					
2				2		password = salman123					
3				3							
4				4							
Test Conditions		User memberikan ulasan kota yang sedang dikunjungi									
Step #	Step Details		Expected Results		Actual Results		Pass / Fail / Not executed / Suspended				
1	Membuka Aplikasi		Aplikasi dapat dibuka		Seperti yang diharapkan		Berhasil				
2	Menekan fitur history		Halaman utama berpindah ke halaman history		Seperti yang diharapkan		Berhasil				
Test Case ID	CV_005		Test Case Description		Test fungsi log out ke aplikasi						
Created By	David		Reviewed By		David		Wason		Beta		
QA Tester's Log	login beta version										
Tester's Name		David P.H.		Date Tested		26-Maret-2023		Test Case (Pass/Fail/Not Executed)		Berhasil	
S #		Prerequisites:		S #		Test Data Requirement					
1		Akses ke aplikasi		1		email = salman@gmail.com					
2				2		password = salman123					
3				3							
4				4							
Test Conditions		User memberikan ulasan kota yang sedang dikunjungi									
Step #	Step Details		Expected Results		Actual Results		Pass / Fail / Not executed / Suspended				
1	Menekan tombol/icon log out		Halaman utama berpindah ke halaman login		Seperti yang diharapkan		Berhasil				

Test Case adalah deskripsi rinci tentang langkah-langkah yang harus diikuti dan kondisi yang harus terpenuhi untuk menguji, fitur, fungsi atau perilaku dari sebuah sistem, aplikasi, atau solusi. Test case menjelaskan bagaimana sebuah pengujian harus dilakukan, input yang harus digunakan, dan hasil yang diharapkan.

6. Matrix Grid

REQUIREMENTS TRACEABILITY MATR				
Project Name:	Next & Easy			
Reviewer/Approver:				
Traceability #	Requirement ID	Technical Requirement ID	Functional Validation	Test Case ID
1	B1	T1	F1	CV_001
		T2	F2	
		T3	F3	
		T4		
		T5		
		T6		
		T7		
		T8		
		T9		
		T10		
2	B2	T11	F4	CV_002
		T12	F5	
		T13	F6	
3	B3	T20	F9	CV_003
		T21	F10	
		T22		
		T23		
		T24		
		T25		
		T26		
		T27		
		T28		
		T29		
		T30		
		T31		
		T32		
		T33		
		T34		
		T35		
4	B4	T36	F11	CV_004
			F12	
5	B5	T37	F13	CV_005
			F14	

Matrix grid digunakan untuk menganalisis data dengan menggabungkan dua atau lebih dimensi. Ini membantu dalam menyajikan informasi kompleks dengan cara terstruktur dan mudah dibaca.

Bab III. Hasil dan Pembahasan

3.1 Unit testing

- code login test

```

test > login_test.dart > main > group("Login Page Widget Test") > testWidgets("Login Page UI Test")
1 import 'package:flutter/material.dart';
2 import 'package:flutter_test/flutter_test.dart';
3 import 'package:get/get.dart';
4 import 'package:final_project_2023/app/modules/home/views/login_page.dart';
5 import 'package:final_project_2023/app/modules/home/views/signup_page.dart';
6
Run | Debug
7 void main() {
Run | Debug
8   group("Login Page Widget Test", () {
Run | Debug
9     testWidgets("Login Page UI Test", (WidgetTester tester) async {
10       await tester.pumpWidget(const GetMaterialApp(
11         home: LoginPage(),
12       )); // GetMaterialApp
13
14       expect(find.text('Bijak Mengolah Sampah'), findsOneWidget);
15       expect(find.text('Login to Your App'), findsOneWidget);
16       expect(find.byType(TextFormField), findsNWidgets(2));
17       expect(find.byType(RawMaterialButton), findsOneWidget);
18       expect(find.text('Don\'t Remember Your Password'), findsOneWidget);
19       expect(find.text('Don\'t Have an Account? '), findsOneWidget);
20       expect(find.text('Sign Up'), findsOneWidget);
21     });
22
Run | Debug
23   testWidgets("Login Page User Input Test", (WidgetTester tester) async {
24     await tester.pumpWidget(const GetMaterialApp(
25       home: LoginPage(),
26     )); // GetMaterialApp
27
28     final emailField = find.byType(TextFormField).first;
29     final passwordField = find.byType(TextFormField).last;
30     final email = 'login@gmail.com'; Use 'const' for final variables initialized to a constant value. Try replacing 'final' with 'const'.
31     final password = '123456'; Use 'const' for final variables initialized to a constant value. Try replacing 'final' with 'const'.
32
33     await tester.enterText(emailField, email);
34     await tester.enterText(passwordField, password);
35
36     await tester.enterText(emailField, email);
37     await tester.enterText(passwordField, password);
38     expect(find.text(email), findsOneWidget);
39     expect(find.text(password), findsOneWidget);
40   });
41
Run | Debug
42   testWidgets("Login Page Navigation Test", (WidgetTester tester) async {
43     await tester.pumpWidget(const GetMaterialApp(
44       home: LoginPage(),
45     )); // GetMaterialApp
46
47     await tester.tap(find.text('Sign Up'));
48     await tester.pumpAndSettle();
49     expect(find.byType(SignUp), findsOneWidget);
50   });

```

(1) Menu Login

Terdapat tampilan login untuk masuk ke dalam aplikasi Bimesa dengan menggunakan akun user Email dan Password yang telah dibuat di Sign Up contoh yang sudah dibuat:

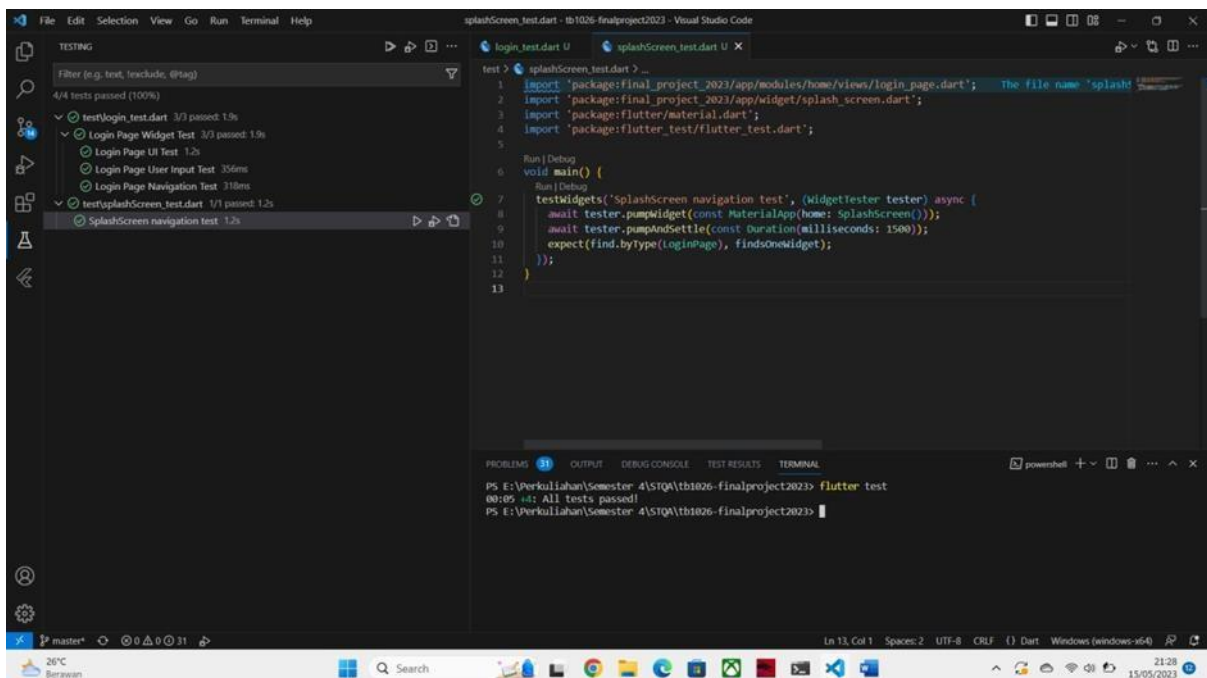
(Penting) contoh email yang telah dibuat:

email : login@gmail.com password
: 123456

- Code splashScreen_test

```
test > splashScreen_test.dart > ...
1  import 'package:final_project_2023/app/modules/home/views/login_page.dart';
2  import 'package:final_project_2023/app/widget/splash_screen.dart';
3  import 'package:flutter/material.dart';
4  import 'package:flutter_test/flutter_test.dart';
5
6  Run | Debug
void main() {
7      Run | Debug
testWidgets('SplashScreen navigation test', (WidgetTester tester) async {
8          await tester.pumpWidget(const MaterialApp(home: SplashScreen()));
9          await tester.pumpAndSettle(const Duration(milliseconds: 1500));
10         expect(find.byType(LoginPage), findsOneWidget);
11     });
12 }
13
```

- Hasil running test



(2) Menu Tampilan

Terdapat Menu dengan cara menekan garis tiga di pojok kiri atas, di Menu terdapat fitur
Buang Sampah

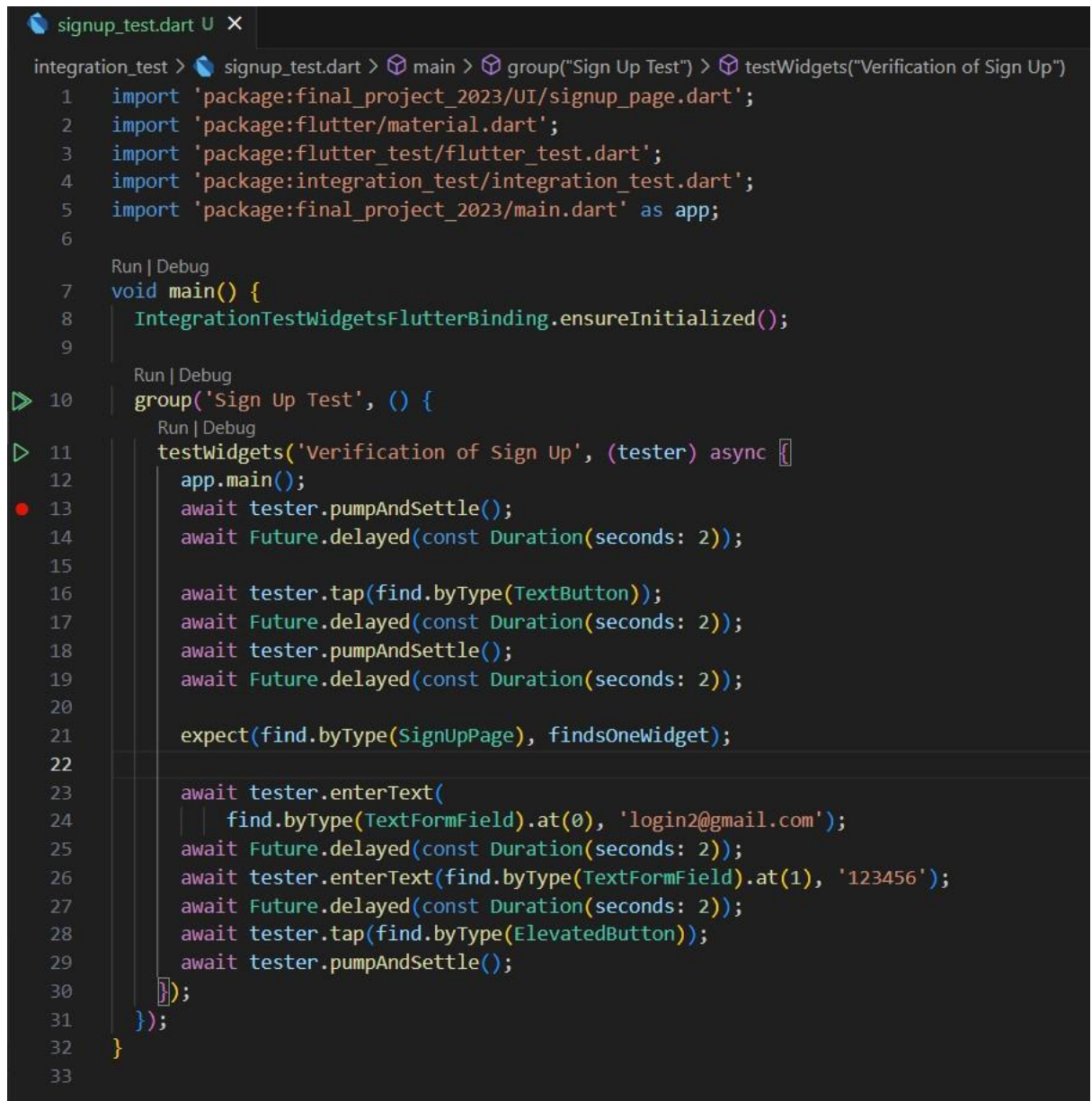
Fitur Buang Sampah digunakan untuk memanggil petugas kebersihan untuk mengambil sampah. User mengisi alamat dan nama terlebih dahulu kemudian menekan tombol call us. Data akan tersimpan di fire store.

Kemudian

User akan kembali ke menu home page secara otomatis

3.2 Integration Testing

a. Sign Up



```
integration_test > signup_test.dart > main > group("Sign Up Test") > testWidgets("Verification of Sign Up")
1  import 'package:final_project_2023/UI/signup_page.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_test/flutter_test.dart';
4  import 'package:integration_test/integration_test.dart';
5  import 'package:final_project_2023/main.dart' as app;
6
7  Run | Debug
8  void main() {
9      IntegrationTestWidgetsFlutterBinding.ensureInitialized();
10
11  Run | Debug
12  group('Sign Up Test', () {
13      Run | Debug
14      testWidgets('Verification of Sign Up', (tester) async {
15          app.main();
16          await tester.pumpAndSettle();
17          await Future.delayed(const Duration(seconds: 2));
18
19          await tester.tap(find.byType(TextButton));
20          await Future.delayed(const Duration(seconds: 2));
21          await tester.pumpAndSettle();
22          await Future.delayed(const Duration(seconds: 2));
23
24          expect(find.byType(SignUpPage), findsOneWidget);
25
26          await tester.enterText(
27              find.byType(TextFormField).at(0), 'login2@gmail.com');
28          await Future.delayed(const Duration(seconds: 2));
29          await tester.enterText(find.byType(TextFormField).at(1), '123456');
30          await Future.delayed(const Duration(seconds: 2));
31          await tester.tap(find.byType(ElevatedButton));
32          await tester.pumpAndSettle();
33      });
34  });
35  }
```

Pengujian ini menguji fitur pendaftaran (Sign Up) dalam aplikasi tersebut. Langkah-langkah pengujian yang dilakukan adalah sebagai berikut:

1. Melakukan inisialisasi pengujian integrasi.
2. Membuka aplikasi dan menunggu beberapa saat untuk memastikan bahwa antarmuka pengguna telah selesai memuat.

3. Mengetuk tombol (widget) yang ditemukan di aplikasi.
4. Menunggu beberapa saat lagi untuk memastikan bahwa halaman pendaftaran telah ditampilkan.
5. Memasukkan teks pada kolom teks pertama (TextFormField) dengan alamat email 'login2@gmail.com'.
6. Memasukkan teks pada kolom teks kedua (TextFormField) dengan kata sandi '123456'.
7. Mengetuk tombol (widget) yang ditemukan di aplikasi.
8. Menunggu beberapa saat lagi untuk memastikan bahwa aksi pendaftaran selesai.

b. Login


```
login_test.dart U X
integration_test > login_test.dart > main > group("Login Test")
1  import 'package:flutter/material.dart';
2  import 'package:flutter_test/flutter_test.dart';
3  import 'package:integration_test/integration_test.dart';
4  import 'package:final_project_2023/main.dart' as app;
5  import 'package:final_project_2023/UI/home_page.dart';
6
Run | Debug
7  void main() {
8      IntegrationTestWidgetsFlutterBinding.ensureInitialized();
9
Run | Debug
10 group('Login Test', () {
Run | Debug
11     testWidgets('Verifikasi valid Username dan password di Login Screen',
12         (tester) async {
13         app.main();
14         await tester.pumpAndSettle();
15         await Future.delayed(const Duration(seconds: 2));
16         await tester.enterText(
17             find.byType(TextField).at(0), 'login@gmail.com');
18         await Future.delayed(const Duration(seconds: 2));
19         await tester.enterText(find.byType(TextField).at(1), '123456');
20         await Future.delayed(const Duration(seconds: 2));
21         await tester.tap(find.byType(ElevatedButton));
22         await Future.delayed(const Duration(seconds: 2));
23         await tester.pumpAndSettle();
24
25         await Future.delayed(const Duration(seconds: 2));
26         expect(find.byType(HomePage), findsOneWidget);
27     });
28 });
29
30
```

Pengujian ini menguji fitur login dengan memasukkan username dan password yang valid. Langkah-langkah pengujian yang dilakukan adalah sebagai berikut:

1. Melakukan inisialisasi pengujian integrasi.
2. Membuka aplikasi dan menunggu beberapa saat untuk memastikan tampilan stabil.
3. Memasukkan teks 'login@gmail.com' pada kolom username.
4. Memasukkan teks '123456' pada kolom password.
5. Mengetuk tombol (ElevatedButton) yang ditemukan dalam tampilan aplikasi.
6. Menunggu beberapa saat untuk memastikan tampilan stabil setelah mengetuk tombol.
7. Memastikan bahwa halaman HomePage ditampilkan dalam tampilan aplikasi.

c. Rate a City

```
rate_test.dart U X
integration_test > rate_test.dart > main > group("Rate Page Integration Test")
1 import 'package:flutter/material.dart';
2 import 'package:flutter_test/flutter_test.dart';
3 import 'package:integration_test/integration_test.dart';
4 import 'package:final_project_2023/main.dart' as app;
5
6 void main() {
7   IntegrationTestWidgetsFlutterBinding.ensureInitialized();
8
9   group('Rate Page Integration Test', () {
10     testWidgets('Rate a City', (WidgetTester tester) async {
11       app.main();
12
13       await tester.pumpAndSettle();
14
15       await tester.enterText(find.byType(TextField).at(0), 'Jakarta');
16       await tester.pumpAndSettle();
17
18       await tester.enterText(find.byType(TextField).at(1), '01-01-2023');
19       await tester.pumpAndSettle();
20
21       await tester.enterText(find.byType(TextField).at(2), 'Great City!');
22       await tester.pumpAndSettle();
23
24       await tester.enterText(find.byType(TextField).at(3), '8');
25       await tester.pumpAndSettle();
26
27       await tester.tap(find.widgetWithText(ElevatedButton, 'Rate!'));
28       await tester.pumpAndSettle(
29         Duration(seconds: 5)); // Add some additional delay for saving data Use 'const' with
30
31       // Add assertions here to verify the expected results after rating
32       expect(find.text('Thank you!'), findsOneWidget);
33       expect(find.byType(TextField),
34         findsNothing); // Check that form fields are cleared
35
36       // You can also add assertions to check if the app navigates back to the home page
37       // Example: expect(find.byType(HomePage), findsOneWidget);
38     });
39   });
40 }
41
42
```

Pengujian ini menguji fitur penilaian (rating) pada halaman Rate Page. Langkah-langkah pengujian yang dilakukan adalah sebagai berikut:

1. Membuka aplikasi dan menunggu hingga tampilan stabil.
2. Memasukkan teks 'Jakarta' pada kolom pertama (TextField).
3. Memasukkan teks '01-01-2023' pada kolom kedua (TextField).
4. Memasukkan teks 'Great City!' pada kolom ketiga (TextField).
5. Memasukkan teks '8' pada kolom keempat (TextField).
6. Mengetuk tombol 'Rate!' (ElevatedButton).
7. Menunggu beberapa saat setelah mengetuk tombol.
8. Memastikan bahwa teks 'Thank you!' ditemukan dalam tampilan.

9. Memastikan bahwa kolom-kolom input (TextFormField) telah dikosongkan.

3.3 Code Documentation

Main

```
main.dart M X
lib > main.dart > MyApp > build
1 import 'package:final_project_2023/controller/token_adapter.dart';
2 import 'package:firedart/firedart.dart';
3 import 'package:flutter/material.dart';
4 import 'package:get/get.dart';
5 import 'package:hive_flutter/hive_flutter.dart';
6 import 'UI/login_page.dart';
7 import 'controller/hive_store.dart';
8
9 // API key Firebase
10 const apiKey = "AIzaSyBf_ks0i3Llj795vao5FStSFaBuydpQqhQ";
11 // ID project Firebase
12 const projectId = "kelompok-c5";
13
14 Run | Debug | Profile
15 void main() async {
16   WidgetsFlutterBinding.ensureInitialized();
17
18   // Inisialisasi Hive
19   await Hive.initFlutter();
20   // Registrasi adapter Hive untuk kelas TokenAdapter
21   Hive.registerAdapter(TokenAdapter());
22
23   // Inisialisasi FirebaseAuth dengan API key dan HiveStore
24   FirebaseAuth.initialize(apiKey, await HiveStore.create());
25   // Inisialisasi Firestore dengan ID project
26   Firestore.initialize(projectId);
27
28   // Menjalankan aplikasi Flutter
29   runApp(const MyApp());
30 }
31
32 class MyApp extends StatelessWidget {
33   const MyApp({Key? key}) : super(key: key);
34
35   @override
36   Widget build(BuildContext context) {
37     return GetMaterialApp(
38       debugShowCheckedModeBanner: false,
39       theme: ThemeData(
40         primarySwatch: Colors.green,
41       ), // ThemeData
42       home: const LoginPage(),
43     ); // GetMaterialApp
44   }
45 }
```


Login

```
login_page.dart M X
lib > UI > login_page.dart > LoginPage
1 import 'package:final_project_2023/controller/data_cotroller.dart';
2 import 'package:flutter/material.dart';
3 import 'package:get/get.dart';
4 import 'signup_page.dart';
5
6 // Import modul dan pustaka yang diperlukan
7
8 class LoginPage extends StatefulWidget {
9   const LoginPage({super.key});
10
11   @override
12   State<LoginPage> createState() => _LoginPageState();
13
14   class _LoginPageState extends State<LoginPage> {
15     final _formKey = GlobalKey<FormState>(); // Kunci global untuk form
16     DataController dc = Get.put(DataController()); // Instance dari DataController
17
18     @override
19     void initState() {
20       super.initState();
21       // Metode yang dipanggil ketika State diinisialisasi
22     }
23
24     @override
25     Widget build(BuildContext context) {
26       return Scaffold(
27         body: Container(
28           height: double.infinity,
29           width: double.infinity,
30           decoration: const BoxDecoration(
31             image: DecorationImage(
32               image: AssetImage('images/gambar1.jpeg'), fit: BoxFit.cover), // DecorationImage
33             ), // BoxDecoration
34         child: Column(
35           children: [
36             Padding(
37               padding: const EdgeInsets.only(left: 20, right: 20, top: 180),
38               child: Card(
39                 color: Colors.green.shade50,
40                 elevation: 50,
41                 child: Form(
42                   key: _formKey,
43                   child: Column(
44                     // Tampilan UI LoginPage menggunakan widget Scaffold, Container, Column, dan Card
45                     children: [
46                       const SizedBox(
47                         height: 30,
48                       ), // SizedBox
49                       const Text(
50                         "Welcome!",
51                         style: TextStyle(fontSize: 25),
52                       ), // Text
53                       const SizedBox(
54                         height: 20,
55                       ),
56                     ],
57                   ),
58                 ),
59             ],
60           ),
61         ),
62       );
63     }
64   }
65 }
```

```

login_page.dart M X
lib > UI > login_page.dart > LoginPage
49   ), // SizedBox
50   const Text(
51     "Welcome!",
52     style: TextStyle(fontSize: 25),
53   ), // Text
54   const SizedBox(
55     height: 20,
56   ), // SizedBox
57   Padding(
58     padding:
59       const EdgeInsets.only(left: 20, right: 20, top: 10),
60     child: TextFormField(
61       key: const Key('emailField'),
62       decoration: const InputDecoration(
63         icon: Icon(Icons.person),
64         labelText: 'Email',
65       ), // InputDecoration
66       style: const TextStyle(fontSize: 12),
67       onSave: (newValue) {
68         if (newValue != null) {
69           dc.email = newValue;
70         }
71       },
72       validator: (value) {
73         if (value == null || value.isEmpty) {
74           return 'Fill this first!';
75         }
76         return null;
77       },
78     ), // TextFormField
79   ), // Padding
80   Padding(
81     padding:
82       const EdgeInsets.only(left: 20, right: 20, top: 10),
83     child: TextFormField(
84       key: const Key('passwordField'),
85       decoration: const InputDecoration(
86         icon: Icon(Icons.lock),
87         labelText: 'Password',
88       ), // InputDecoration
89       style: const TextStyle(fontSize: 12),
90       onSave: (newValue) {
91         if (newValue != null) {
92           dc.pass = newValue;
93         }
94       },
95       validator: (value) {
96         if (value == null || value.isEmpty) {
97           return 'Fill this first!';
98         }
99         return null;
100      },
101    ), // TextFormField
102  ), // Padding
103  const SizedBox(
104    height: 20,
105  ), // SizedBox

```

```

login_page.dart M X
lib > UI > login_page.dart > _LoginPageState > build
102     ), // Padding
103     const SizedBox(
104       height: 40,
105     ), // SizedBox
106     ElevatedButton(
107       onPressed: () {
108         if (_formKey.currentState!.validate()) {
109           _formKey.currentState?.save();
110         }
111         dc.signIn(dc.email, dc.pass);
112       },
113       child: const Text(
114         'Login',
115         style: TextStyle(fontSize: 12),
116       ), // Text
117     ), // ElevatedButton
118     const SizedBox(
119       height: 50,
120     ), // SizedBox
121     Padding(
122       padding: const EdgeInsets.only(bottom: 20),
123       child: Row(
124         mainAxisAlignment: MainAxisAlignment.center,
125         children: [
126           const Text(
127             "Don't have account?",
128             style: TextStyle(
129               fontSize: 12, fontWeight: FontWeight.w400), // TextStyle
130           ), // Text
131           Padding(
132             padding: const EdgeInsets.all(8.0),
133             child: TextButton(
134               onPressed: () {
135                 Navigator.push(
136                   context,
137                   MaterialPageRoute(
138                     builder: (context) => const SignUpPage(),
139                   ), // MaterialPageRoute
140                 );
141               },
142               child: const Text(
143                 'Sign up here!',
144                 style: TextStyle(fontSize: 12),
145               ), // Text
146             ), // TextButton
147           ), // Padding
148         ],
149       ), // Row
150     ), // Padding
151   ],
152 ), // Column
153 ), // Form
154 ), // Card
155 ), // Padding
156 ],
157 ), // Column
158 ), // Container
159 ); // Scaffold
160 }
161 }
162

```

Sign Up

```

signup_page.dart M X
lib > UI > signup_page.dart >...
1 import 'package:final_project_2023/controller/data_cotroller.dart';
2 import 'package:flutter/material.dart';
3 import 'package:get/get.dart';
4 import 'login_page.dart';
5
6 class SignUpPage extends StatefulWidget {
7   const SignUpPage({super.key});
8
9   @override
10  State<SignUpPage> createState() => _SignUpPageState();
11 }
12
13 class _SignUpPageState extends State<SignUpPage> {
14   final _formKey = GlobalKey<FormState>(); // Kunci global untuk form
15   DataController dc = Get.put(DataController()); // Instance dari DataController
16
17   @override
18   void initState() {
19     super.initState();
20     // Metode yang dipanggil ketika State diinisialisasi
21   }
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       body: Container(
27         height: double.infinity,
28         width: double.infinity,
29         decoration: const BoxDecoration(
30           image: DecorationImage(
31             image: AssetImage('images/gambar1.jpeg'), fit: BoxFit.cover), // DecorationImage
32         ), // BoxDecoration
33       child: Column(
34         children: [
35           Padding(
36             padding: const EdgeInsets.only(left: 20, right: 20, top: 180),
37             child: Card(
38               color: Colors.green.shade50,
39               elevation: 50,
40               child: Form(
41                 key: _formKey,
42                 child: Column(
43                   // Tampilan UI SignUpPage menggunakan widget Scaffold, Container, Column, dan Card
44                   children: [
45                     const Text(
46                       "Please fill the form!",
47                       style: TextStyle(fontSize: 25),
48                     ), // Text
49                     const SizedBox(
50                       height: 20,
51                     ), // SizedBox
52                     Padding(
53                       padding:
54                         const EdgeInsets.only(left: 20, right: 20, top: 10),
55                       child: TextFormField(
56                         decoration: const InputDecoration(
57                           icon: Icon(Icons.person),
58                           labelText: 'Email',
59                         ), // InputDecoration
60                         style: const TextStyle(fontSize: 12),
61                         onSave: (newValue) {
62                           if (newValue != null) {
63                             dc.email = newValue;
64                           }
65                         },
66                       ), // TextFormField
67                     ), // Padding
68                   ],
69                 ),
70               ),
71             ),
72           ),
73         ],
74       ),
75     );
76   }
77 }

```



```
signup_page.dart M X
lib > UI > signup_page.dart > ...
68      Padding(
69        padding:
70          const EdgeInsets.only(left: 20, right: 20, top: 10),
71        child: TextFormField(
72          decoration: const InputDecoration(
73            icon: Icon(Icons.lock),
74            labelText: 'Password',
75          ), // InputDecoration
76          style: const TextStyle(fontSize: 12),
77          onSave: (newValue) {
78            if (newValue != null) {
79              dc.pass = newValue;
80            }
81          },
82        ), // TextFormField
83      ), // Padding
84      const SizedBox(
85        height: 40,
86      ), // SizedBox
87      ElevatedButton(
88        onPressed: () {
89          _formKey.currentState?.save();
90          dc.signUp(dc.email, dc.pass);
91
92          // Navigasi ke halaman login setelah sign up selesai
93          Navigator.pushReplacement(
94            context,
95            MaterialPageRoute(
96              builder: (context) => LoginPage(), // MaterialPageRoute
97            );
98          },
99        child: const Text(
100          'Sign Up!',
101          style: TextStyle(fontSize: 12),
102        ), // Text
103      ), // ElevatedButton
104      const SizedBox(
105        height: 50,
106      ), // SizedBox
107    ],
108  ), // Column
109 ), // Form
110 ), // Card
111 ) // Padding
112 ],
113 ), // Column
114 ), // Container
115 ); // Scaffold
116 }
117 }
118
```

HomePage

```

home_page.dart M X
lib > UI > home_page.dart > ...
1 import 'package:final_project_2023/controller/data_cotroller.dart';
2 import 'package:flutter/material.dart';
3 import 'package:get/get.dart';
4 import 'login_page.dart'; Unused import: 'login_page.dart'. Try removing the import directive.
5
6 class HomePage extends StatefulWidget {
7   const HomePage({super.key});
8
9   @override
10  State<HomePage> createState() => _HomePageState();
11 }
12
13 class _HomePageState extends State<HomePage> {
14   DataController dc = Get.put(DataController()); // Instance dari DataController
15
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       appBar: AppBar(
20         automaticallyImplyLeading: false), // AppBar tanpa tombol kembali // AppBar
21       backgroundColor: Colors.green.shade50, // Warna latar belakang
22       body: Column(
23         children: [
24           OutlinedButton(
25             onPressed: () {
26               dc.toRate(); // Panggil metode toRate() pada DataController
27             },
28             style: OutlinedButton.styleFrom(
29               fixedSize: const Size(double.maxFinite, 200), // Ukuran tombol
30             ),
31             child: Row(
32               children: [
33                 Container(
34                   height: 180,
35                   width: 180,
36                   decoration: const BoxDecoration(
37                     image: DecorationImage(
38                       image: AssetImage('images/gambar2.png'),
39                       fit: BoxFit.fill), // DecorationImage
40                   ), // BoxDecoration
41                 ), // Container
42                 const SizedBox(
43                   width: 20,
44                 ), // SizedBox
45                 const Text('Rate a City'), // Teks pada tombol
46               ],
47             ), // Row
48           ), // OutlinedButton
49           OutlinedButton(
50             onPressed: () {
51               dc.toHistory(); // Panggil metode toHistory() pada DataController
52             },
53             style: OutlinedButton.styleFrom(
54               fixedSize: const Size(double.maxFinite, 200), // Ukuran tombol
55           ),

```

```
home_page.dart M X
lib > UI > home_page.dart > ...
56 child: Row(
57   children: [
58     Container(
59       height: 180,
60       width: 180,
61       decoration: const BoxDecoration(
62         image: DecorationImage(
63           image: AssetImage('images/gambar3.png'),
64           fit: BoxFit.fill), // DecorationImage
65       ), // BoxDecoration
66     ), // Container
67     const SizedBox(
68       width: 20,
69     ), // SizedBox
70     const Text('Rate History'), // Teks pada tombol
71   ],
72 ), // Row
73 ), // OutlinedButton
74 OutlinedButton(
75   onPressed: () {
76     dc.signOut(); // Panggil metode signOut() pada DataController
77   },
78   style: OutlinedButton.styleFrom(
79     fixedSize: const Size(double.maxFinite, 200), // Ukuran tombol
80   ),
81   child: Row(
82     children: [
83       Container(
84         height: 190,
85         width: 190,
86         child: const Icon(Icons.logout_rounded,
87           size: 50), // Icon pada tombol // Icon
88       ), // Container
89       const SizedBox(
90         width: 10,
91       ), // SizedBox
92       const Text('Log out'), // Teks pada tombol
93     ],
94   ), // Row
95 ), // OutlinedButton
96 ],
97 ), // Column
98 ); // Scaffold
99 }
100 }
101
```

Rate A City

```
rate_page.dart M X
lib > UI > rate_page.dart > ...
1 import 'package:final_project_2023/controller/data_controller.dart';
2 import 'package:firebase/firebase.dart';
3 import 'package:flutter/material.dart';
4 import 'package:get/get.dart';
5
6 class RatePage extends StatefulWidget {
7   const RatePage({Key? key}) : super(key: key);
8
9   @override
10  State<RatePage> createState() => _RatePageState();
11 }
12
13 class _RatePageState extends State<RatePage> {
14   CollectionReference dataCollection = Firestore.instance.collection('Data');
15   final _formKey = GlobalKey<FormState>(); // Kunci global untuk form
16   DataController dc = Get.put(DataController()); // Instance dari DataController
17
18   late String kota = ''; // Variabel penyimpanan nama kota
19   late String tgl = ''; // Variabel penyimpanan tanggal
20   late String kesan = ''; // Variabel penyimpanan kesan tentang kota
21   late int rating = 0; // Variabel penyimpanan rating
22
23   @override
24   Widget build(BuildContext context) {
25     return Scaffold(
26       backgroundColor: Colors.green.shade50, // Warna latar belakang
27       appBar: AppBar(
28         title: const Text('Rate a City'), // Judul AppBar
29       ), // AppBar
30       body: Column(
31         children: [
32           const SizedBox(
33             height: 30,
34           ), // SizedBox
35           Card(
36             child: Column(
37               children: [
38                 Padding(
39                   padding: const EdgeInsets.only(left: 12, right: 12, top: 15),
40                   child: Form(
41                     key: _formKey, // Menggunakan _formKey pada form
42                     child: Column(
43                       children: [
44                         TextFormField(
45                           controller: dc.kota,
46                           decoration: const InputDecoration(
47                             icon: Icon(Icons.location_city),
48                             labelText: 'Name of the city',
49                             hintText:
50                               'Jakarta'), // Label dan hint pada TextFormField // InputDecoration
51                           style: const TextStyle(fontSize: 13),
52                           onSave: (newValue) {
53                             if (newValue != null) {
54                               dc.kota.text =
55                                 newValue; // Menyimpan nilai inputan pada variabel dc.kota
56                             }
57                           },
58                         ),
59                       ],
60                     ),
61                   ),
62                 ],
63               ),
64             ),
65           ],
66         ),
67       ),
68     );
69   }
70 }
```



```

rate_page.dart M X
lib > UI > rate_page.dart > ...
58     ), // TextFormField
59     TextFormField(
60       decoration: const InputDecoration(
61         icon: Icon(Icons.date_range),
62         labelText: 'Date of visit',
63         hintText:
64           '01-01-2023'), // Label dan hint pada TextFormField // InputDecoration
65       style: const TextStyle(fontSize: 13),
66       onSave: (newValue) {
67         if (newValue != null) {
68           dc.tgl.text =
69             newValue; // Menyimpan nilai inputan pada variabel dc.tgl
70         }
71       },
72     ), // TextFormField
73     TextFormField(
74       decoration: const InputDecoration(
75         icon: Icon(Icons.label_important_rounded),
76         labelText:
77           'What do you think about the city?'), // Label dan hint pada TextFormField // InputDecoration
78       maxLines:
79         4, // Mengatur jumlah baris pada TextFormField
80       style: const TextStyle(fontSize: 13),
81       onSave: (newValue) {
82         if (newValue != null) {
83           dc.kesan.text =
84             newValue; // Menyimpan nilai inputan pada variabel dc.kesan
85         }
86       },
87     ), // TextFormField
88     TextFormField(
89       decoration: const InputDecoration(
90         icon: Icon(Icons.star_rate),
91         labelText: 'Rate the sanitation',
92         hintText:
93           'Scale from 1 to 10'), // Label dan hint pada TextFormField // InputDecoration
94       style: const TextStyle(fontSize: 13),
95       onSave: (newValue) {
96         if (newValue != null) {
97           dc.rating.text =
98             newValue; // Menyimpan nilai inputan pada variabel dc.rating
99         }
100      },
101      validator: (value) {
102        if (value == null || value.isEmpty) {
103          return 'Fill this first!'; // Validasi jika input kosong
104        }
105        bool apakahAngka = int.tryParse(value) !=
106          null; // Validasi jika input bukan angka
107        if (!apakahAngka) {
108          return 'Only numbers are allowed!'; // Pesan kesalahan jika input bukan angka
109        }
110        return null;
111      },
112     ), // TextFormField

```

```

lib > UI > rate_page.dart > ...
111         ],
112       ), // TextFormField
113     ],
114   ), // Column
115 ), // Form
116 ), // Padding
117 const SizedBox(
118   height: 80,
119 ), // SizedBox
120 Padding(
121   padding: const EdgeInsets.all(12),
122   child: Row(
123     mainAxisAlignment: MainAxisAlignment.end,
124     children: [
125       ElevatedButton(
126         onPressed: () async {
127           if (_formKey.currentState!.validate()) {
128             _formKey.currentState!
129               .save(); // Validasi form dan menyimpan nilai inputan
130           }
131           await dc.saveData(
132             dc.kota.text,
133             dc.kesan.text,
134             dc.tgl.text,
135             dc.rating
136               .text); // Menyimpan data dengan memanggil metode saveData pada DataController
137         },
138         child: const Text('Rate!'), // Teks pada tombol
139       ), // ElevatedButton
140     ],
141   ), // Row
142 ), // Padding
143 ],
144 ), // Column
145 ), // Card
146 ],
147 ), // Column
148 ); // Scaffold
149 }
150 }
151

```

History

```

history_page.dart M X
lib > UI > history_page.dart > HistoryPage > build
1 // ignore_for_file: prefer_interpolation_to_compose_strings
2
3 import 'package:firebase/firebase.dart';
4 import 'package:flutter/material.dart';
5
6 class HistoryPage extends StatelessWidget { This class (or a class that this class inherits from) i
7   HistoryPage({super.key});
8
9   CollectionReference dataCollection = Firestore.instance.collection('Data');
10
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         title: const Text('Rating History'), // Judul AppBar
16       ), // AppBar
17       body: StreamBuilder(
18         stream: dataCollection.stream, // Menggunakan stream dari dataCollection
19         builder: (context, snapshot) {
20           if (snapshot.hasData) {
21             return ListView.builder(
22               itemBuilder: (context, index) {
23                 return Card(
24                   elevation: 5,
25                   child: ListTile(
26                     leading: const Icon(Icons.location_city_rounded,
27                       size: 70), // Icon pada leading ListTile // Icon
28                     title: Center(
29                       child: Text(
30                         snapshot.data![index]
31                           ['Kota'], // Mengambil data kota dari snapshot
32                         style: const TextStyle(fontWeight: FontWeight.bold),
33                       ), // Text // Center
34                     subtitle: Column(
35                       children: [
36                         Text('Tanggal: ' +
37                           snapshot.data![index][
38                             'Tanggal']), // Mengambil data tanggal dari snapshot // Text
39                         Text('Kesan: ' +
40                           snapshot.data![index][
41                             'Kesan']), // Mengambil data kesan dari snapshot // Text
42                         Text('Rating Kebersihan: ' +
43                           snapshot.data![index][
44                             'Rating']), // Mengambil data rating dari snapshot // Text
45                       ],
46                     ), // Column
47                   ), // ListTile
48                 ); // Card
49             },
50             itemCount: snapshot.data?.length, // Jumlah item dalam ListView
51           ); // ListView.builder
52         } else {
53           return const Text(
54             'Oops, No data has been recorded..'); // Teks jika tidak ada data // Text
55         }
56       ), // StreamBuilder
57     ); // Scaffold
58   }
59 }
60
61

```

Data Controller

```

data_cotroller.dart M X
lib > controller > data_cotroller.dart > ...
1 import 'package:final_project_2023/UI/history_page.dart';
2 import 'package:final_project_2023/UI/home_page.dart';
3 import 'package:final_project_2023/UI/login_page.dart';
4 import 'package:final_project_2023/UI/rate_page.dart';
5 import 'package:firebase_auth/firebase_auth.dart';
6 import 'package:flutter/material.dart';
7 import 'package:get/get.dart';
8
9 class DataController extends GetxController {
10   static DataController get to => Get.find<DataController>();
11
12   String email = '';
13   String pass = '';
14
15   // Fungsi untuk mendaftar akun baru
16   void signUp(String email, String pass) async {
17     await FirebaseAuth.instance.signUp(email, pass);
18     Get.back();
19     Get.showSnackBar(
20       const GetSnackBar(
21         title: 'Success',
22         message: 'Please login',
23         duration: Duration(seconds: 2),
24       ), // GetSnackBar
25     );
26   }
27
28   // Fungsi untuk masuk ke akun
29   void signIn(String email, String pass) async {
30     try {
31       await FirebaseAuth.instance.signIn(email, pass);
32       var user = await FirebaseAuth.instance.getUser();
33       Get.to() => const HomePage();
34     } catch (e) {
35       // Menangani kesalahan yang mungkin terjadi saat masuk
36       if (e.toString() == 'AuthException: INVALID_PASSWORD') {
37         Get.showSnackBar(
38           const GetSnackBar(
39             title: 'Wrong Password',
40             duration: Duration(seconds: 2),
41           ), // GetSnackBar
42         );
43       } else if (e.toString() == 'AuthException: EMAIL_NOT_FOUND') {
44         Get.showSnackBar(
45           const GetSnackBar(
46             title: 'Email not Found',
47             message: 'Please sign up!',
48             duration: Duration(seconds: 2),
49           ), // GetSnackBar
50         );
51       } else if (e.toString() == 'AuthException: INVALID_EMAIL') {
52         Get.showSnackBar(
53           const GetSnackBar(
54             title: 'Invalid email',
55             message: 'Please re-enter your email!',
56             duration: Duration(seconds: 2),
57           ), // GetSnackBar
58         );
59       }
60     }
61   }
62 }

```

```

62
63 // Fungsi untuk keluar dari akun
64 void signOut() {
65     FirebaseAuth.instance.signOut();
66     Get.offAll(() => const LoginPage());
67 }
68
69 late TextEditingController kota, kesan, tgl, rating;
70
71 @override
72 void onInit() {
73     super.onInit();
74     // Menginisialisasi TextEditingController
75     kota = TextEditingController();
76     kesan = TextEditingController();
77     tgl = TextEditingController();
78     rating = TextEditingController();
79 }
80
81 // Fungsi untuk menyimpan data rating ke Firestore
82 Future saveData(String kota, String kesan, String tgl, String rating) async {
83     Firestore.initialize("kelompok-c5"); // Inisialisasi Firestore
84     CollectionReference dataCollection = Firestore.instance.collection('Data');
85     await dataCollection.add({
86         'Kota': kota,
87         'Kesan': kesan,
88         'Tanggal': tgl,
89         'Rating': rating,
90     });
91     Get.back();
92     Get.showSnackBar(
93         const GetSnackBar(
94             title: 'Rate Success',
95             message: 'Thank you!',
96             duration: Duration(seconds: 2),
97         ), // GetSnackBar
98     );
99 }
100
101 // Fungsi untuk pindah ke halaman penilaian
102 void toRate() {
103     Get.to(() => const RatePage());
104 }
105
106 // Fungsi untuk pindah ke halaman riwayat
107 void toHistory() {
108     Get.to(() => const HistoryPage());
109 }
110 }
111

```

Hive Store


```

hive_store.dart M X
lib > controller > hive_store.dart > ...
1  import 'package:firebase_auth/firebase_auth.dart';
2  import 'package:hive/hive.dart';
3
4  class HiveStore extends TokenStore {
5      static const keyToken = "auth_token";
6      final Box _box;
7      HiveStore._internal(this._box);
8
9      static Future<HiveStore> create() async {
10         // Membuat instansi HiveStore
11         // Pastikan untuk memanggil kedua metode berikut:
12         // Hive.init(storePath);
13         // Hive.registerAdapter(TokenAdapter(), adapterId);
14
15         // Membuka Box dengan nama "auth_store" dan menggunakan strategi kompaksi
16         var box = await Hive.openBox("auth_store",
17             compactionStrategy: (entries, deletedEntries) => deletedEntries > 50);
18         return HiveStore._internal(box);
19     }
20
21     @override
22     void delete() {
23         // Menghapus token dari HiveStore
24         _box.delete(keyToken);
25     }
26
27     @override
28     Token? read() {
29         // Membaca token dari HiveStore
30         return _box.get(keyToken);
31     }
32
33     @override
34     void write(Token? token) {
35         // Menulis atau menyimpan token ke dalam HiveStore
36         _box.put(keyToken, token);
37     }
38 }
39

```

Token Adapter

```

token_adapter.dart M X
lib > controller > token_adapter.dart > ...
1 import 'package:firedart/firedart.dart';
2 import 'package:hive_flutter/hive_flutter.dart';
3
4 class TokenAdapter extends TypeAdapter<Token> {
5   @override
6   final typeId = 42; // ID unik untuk adapter ini
7
8   @override
9   void write(BinaryWriter writer, Token token) => The parameter name 'token'
10      writer.writeMap(token.toMap()); // Menulis objek Token ke dalam format
11
12   @override
13   Token read(BinaryReader reader) => Token.fromMap(reader
14      .readMap()
15      .map<String, dynamic>((key, value) => MapEntry<String, dynamic>(
16         key, value))); // Membaca objek Token dari format Map // MapEntry
17 }
18

```

3.4 Bug report

1 Defect Header:

Date	24/06/2023
Defect #	Berhasil masuk kedalam homepage tanpa memasukkan username dan password
Application Name-Screen Name	MeetAndRate-Login Page
Screenshots (Y/N)	Y
Submitted by	Elwins
Reproducible (Y/N)	Y
Reference	Screenshoot
Date	24/06/2003

Software Version	1
Severity	Backlog
Priority	Medium

2 Defect Description:

2.1 Module name - Login

2.2 Steps followed Ø

Buka Aplikasi

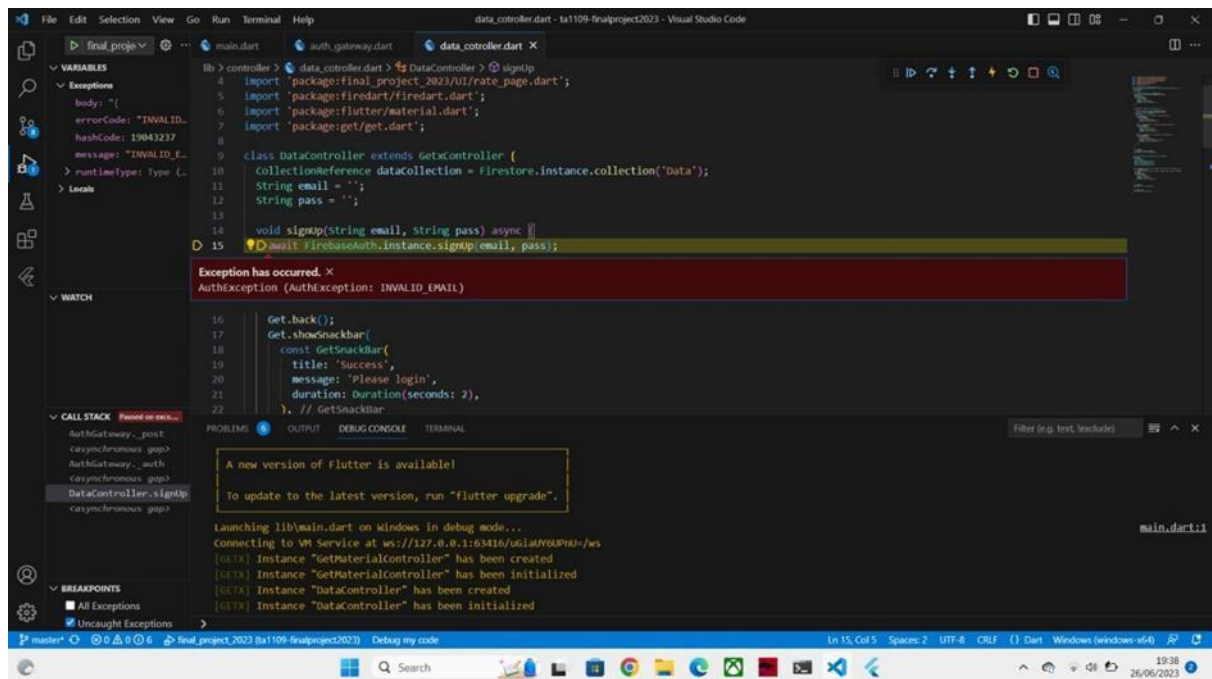
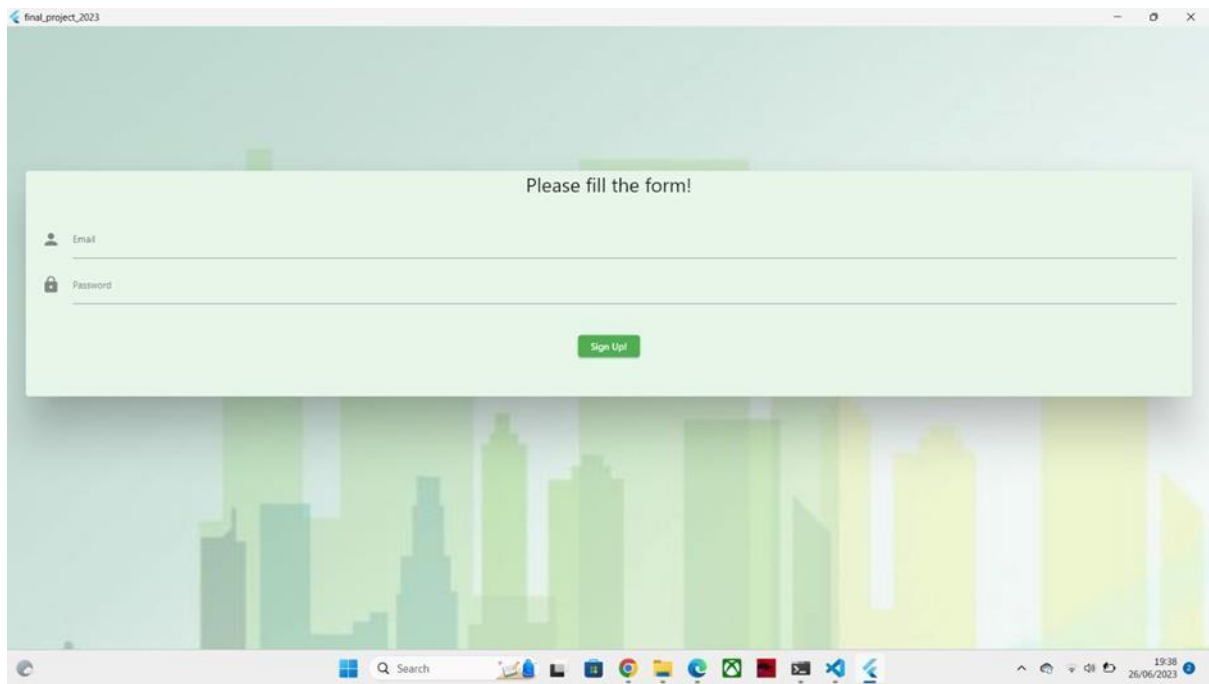
Ø Masuk ke Halaman Login

Ø Tekan button login tanpa mengisi username dan password

2.3 Desired Results

Saat user masuk kedalam halaman login, user menekan login tanpa memasukkan username dan password.

3 Screen shots:



4 Defect Header:

Date	24/06/2023
Defect #	Berhasil masuk kedalam homepage tanpa memasukkan username dan password
Application Name-Screen Name	MeetAndRate-Login Page
Screenshots (Y/N)	Y
Submitted by	Elwins
Reproducible (Y/N)	Y
Reference	Screenshoot
Date	24/06/2003
Software Version	1
Severity	Backlog

Priority	Medium
----------	--------

5 Defect Description:

5.1 Module name - Login

5.2 Steps followed Ø

Buka Aplikasi

Ø Masuk ke Halaman Login

Ø Tekan button login tanpa mengisi username dan password

5.3 Desired Results

Saat user masuk kedalam halaman login, user menekan login tanpa memasukkan username dan password.

3.5 Summary Report

Usability Analysis

Preparation or usability testing plan

The Purpose of Usability

Pengujian kegunaan bertujuan untuk mengevaluasi sejauh mana aplikasi "Meet and rate" memenuhi kebutuhan dan harapan pengguna.

Tujuan pengujian kegunaan meliputi:

a. Mengidentifikasi masalah usability:

Pengujian kegunaan membantu mengidentifikasi masalah atau hambatan yang dialami pengguna dalam menggunakan aplikasi. Dengan melakukan pengujian secara langsung, tim pengembang dapat mengetahui aspek mana yang sulit dipahami, fitur yang membingungkan, atau interaksi yang tidak intuitif. Masalah-masalah ini kemudian dapat dicatat dan dianalisis untuk diperbaiki agar pengalaman pengguna menjadi lebih baik.

b. Mengukur kepuasan pengguna:

Pengujian kegunaan juga bertujuan untuk mengukur tingkat kepuasan pengguna terhadap aplikasi. Ini dapat dilakukan dengan mengumpulkan umpan balik pengguna melalui survei, wawancara, atau pengamatan. Pertanyaan tentang kepuasan pengguna dapat mencakup aspek-aspek seperti kejelasan antarmuka, keterbacaan teks, efisiensi tugas, dan responsivitas aplikasi. Dengan memahami kepuasan pengguna, tim pengembang dapat mengidentifikasi area yang perlu ditingkatkan untuk meningkatkan kepuasan secara keseluruhan.

c. Meningkatkan pengalaman pengguna:

Tujuan utama pengujian kegunaan adalah meningkatkan pengalaman pengguna secara keseluruhan. Dengan mengidentifikasi masalah usability dan mendapatkan umpan balik dari pengguna, tim pengembang dapat melakukan perbaikan dan penyesuaian yang diperlukan. Hal ini dapat meliputi penyederhanaan antarmuka, peningkatan navigasi, penambahan bantuan atau panduan, atau perbaikan kinerja aplikasi. Melalui pengujian kegunaan secara teratur, pengalaman pengguna dapat terus diperbaiki dan dioptimalkan.

Metrics

Metrik yang akan digunakan untuk mengukur kegunaan aplikasi "Meet and Rate" meliputi:

a. Waktu penyelesaian tugas:

Tujuan pengujian kegunaan adalah untuk mengukur seberapa cepat pengguna dapat menyelesaikan tugas-tugas yang diberikan dalam aplikasi. Pengujian ini dapat dilakukan dengan memberikan tugas-tugas yang representatif kepada pengguna dan mencatat waktu yang dibutuhkan untuk menyelesaikan setiap tugas. Dengan demikian, tim pengembang dapat mengevaluasi efektivitas desain antarmuka dan mengidentifikasi area yang memerlukan perbaikan untuk meningkatkan efisiensi penggunaan.

b. Kesalahan pengguna:

Pengujian kegunaan juga melibatkan pengamatan terhadap kesalahan-kesalahan yang dilakukan oleh pengguna saat menggunakan aplikasi. Hal ini mencakup kesalahan dalam navigasi, pemahaman instruksi, penggunaan fitur, atau interaksi dengan elemen antarmuka. Dengan mengidentifikasi jenis kesalahan yang sering terjadi, tim pengembang dapat membuat perbaikan yang diperlukan untuk mengurangi kesalahan dan meningkatkan pengalaman pengguna.

c. Efisiensi penggunaan: Pengujian kegunaan juga bertujuan untuk mengukur sejauh mana aplikasi "Meet and rate" dapat digunakan secara efisien oleh pengguna. Hal ini melibatkan pengamatan terhadap bagaimana pengguna berinteraksi dengan antarmuka, sejauh mana aplikasi mempermudah tugas-tugas pengguna, dan apakah ada langkah-langkah yang dapat disederhanakan atau dihilangkan untuk meningkatkan efisiensi. Dengan memperhatikan efisiensi penggunaan, tim pengembang dapat mengoptimalkan desain dan alur kerja aplikasi agar lebih efisien dan meminimalkan beban kerja pengguna.

Prepared Tasks Based on Selected Metrics

USABILITY TESTING	
Intention	Mengukur kemudahan penggunaan aplikasi Meet&Rate
Metric	Time on Task
	Task Success
	Efesiensi
People	Mahasiswa
Activity	Task 1 → Melakukan Sign Up
	Task 2 → Melakukan Login
	Task 3 → Rate kota yang dikunjungi
	Task 4 → Melihat Histori
	Task 5 → end
Partisipant	Nama
1	Siddiq Fitriansyah TI'21
2	Zebadiah Vashti Agroteknologi'21
3	Purba agroteknologi'17
4	Bagus Satrio'21
5	Elwins'22

Usability testing results

Characteristics and Criteria of Invited Participants

- Mahasiswa UMY: Peserta yang diundang adalah mahasiswa dari Universitas Muhammadiyah Yogyakarta (UMY). Hal ini diharapkan agar bisa mendapatkan hasil yang berbeda.
- Tingkat Kemahiran Teknis yang Beragam: Peserta sebaiknya mewakili berbagai tingkat kemahiran teknis, termasuk pengguna pemula dan pengguna yang lebih mahir. Hal ini memungkinkan evaluasi yang komprehensif terhadap kegunaan aplikasi di berbagai tingkat keterampilan pengguna.
- Familiaritas dengan Aplikasi Seluler: Peserta sebaiknya memiliki pengalaman menggunakan aplikasi seluler di smartphone atau tablet. Familiaritas ini memastikan bahwa mereka nyaman berinteraksi dengan aplikasi dalam konteks seluler.

Pengujian

- Memberikan pengantar kepada peserta tentang aplikasi "Meet and Rate" dan tujuan pengujian.
- Menginstruksikan peserta untuk menyelesaikan tugas yang telah disiapkan satu per satu.
- Mengamati dan mencatat interaksi peserta dengan aplikasi, termasuk kesulitan yang dihadapi, komentar, dan reaksi mereka.
- Meminta peserta untuk berbicara secara verbal saat menggunakan aplikasi untuk mendapatkan wawasan lebih lanjut tentang pemikiran mereka.

Usability Testing Data

Task Success						
X	Task 1	Task 2	Task 3	Task 4	Task 5	Rata
Partisipan 1	1	1	1	1	1	100,00%
Partisipan 2	1	1	1	1	1	100,00%
Partisipan 3	1	1	1	1	1	100,00%
Partisipan 4	1	1	1	1	1	100,00%
Partisipan 5	1	0	0	1	1	60,00%
Time on Task						
X	Task 1	Task 2	Task 3	Task 4	Task 5	
Partisipan 1	20	30	49	17	3	
Partisipan 2	17	15	48	4	5	
Partisipan 3	19	17	36	10	1	
Partisipan 4	17	20	42	14	1	
Partisipan 5	33	5	41	11	2	
rata detik	21,2	17,4	43,2	11,2	2,4	
menit	0,3533333333	0,29	0,72	0,1866666667	0,04	
Efesiensi						
Task	Task Complation Rate	Task Time (min)	Efesiensi (%)			
Task 1	100%	0,35	35,00%			
Task 2	80%	0,29	23,20%			
Task 3	80%	0,72	57,60%			
Task 4	100%	0,18	18,00%			
Task 5	100%	0,04	4,00%			

Bab IV. Kesimpulan

Dalam laporan ini, kami berhasil mengembangkan dan mengimplementasikan aplikasi Meet and Rate yang bertujuan untuk menyediakan wadah interaktif bagi para profesional, mahasiswa, dan praktisi dalam bidang Software Testing and Quality Assurance (STQA). Aplikasi ini menjadi sarana penting bagi para ahli dan akademisi STQA untuk berbagi pengetahuan, pengalaman, serta memberikan ulasan dan penilaian terhadap berbagai aspek terkait. Pentingnya aplikasi Meet and Rate terbukti dalam menyediakan platform yang khusus untuk pengguna dari berbagai latar belakang profesional terkait STQA. Fitur "Rating" pada aplikasi memungkinkan para pengguna memberikan penilaian dan ulasan yang berguna terhadap alat, metode, dan praktik STQA. Dengan demikian, pengguna lain dapat memahami secara lebih mendalam mengenai kualitas dan efektivitas teknik atau alat dalam pengujian perangkat lunak.

Selain itu, fitur "Rating" juga membawa manfaat besar bagi para pengguna yang berencana untuk berpergian atau tinggal di suatu kota. Dengan adanya ulasan dan penilaian terkait kota tertentu, pengguna dapat membuat keputusan yang lebih bijaksana dan akurat terkait pilihan destinasi atau tempat tinggal mereka. Proses pengembangan aplikasi ini tidak luput dari beberapa tantangan dan hambatan, namun tim pengembang berhasil mengatasi mereka dan menghasilkan sebuah aplikasi yang andal dan bermanfaat. Kesuksesan aplikasi Meet and Rate dalam menciptakan lingkungan interaktif dan kolaboratif bagi para pengguna terkait STQA menegaskan nilai tambahnya dalam mendukung pertumbuhan dan pengembangan dalam bidang tersebut.

Melalui kesimpulan ini, kami berharap bahwa aplikasi Meet and Rate dapat terus menjadi sarana yang efektif bagi pertukaran pengetahuan dan kolaborasi di antara para profesional, mahasiswa, dan praktisi STQA. Selain itu, aplikasi ini juga diharapkan dapat memberikan dampak positif dalam membantu para pengguna membuat keputusan yang lebih baik terkait teknik pengujian perangkat lunak dan pilihan destinasi yang tepat dalam berbagai kota.