

**PENGUJIAN DAN PENJAMINAN MUTU PERANGKAT LUNAK**  
**ASSIGNMENT UNIT TESTING KELOMPOK**



**Nama Kelompok 1:**

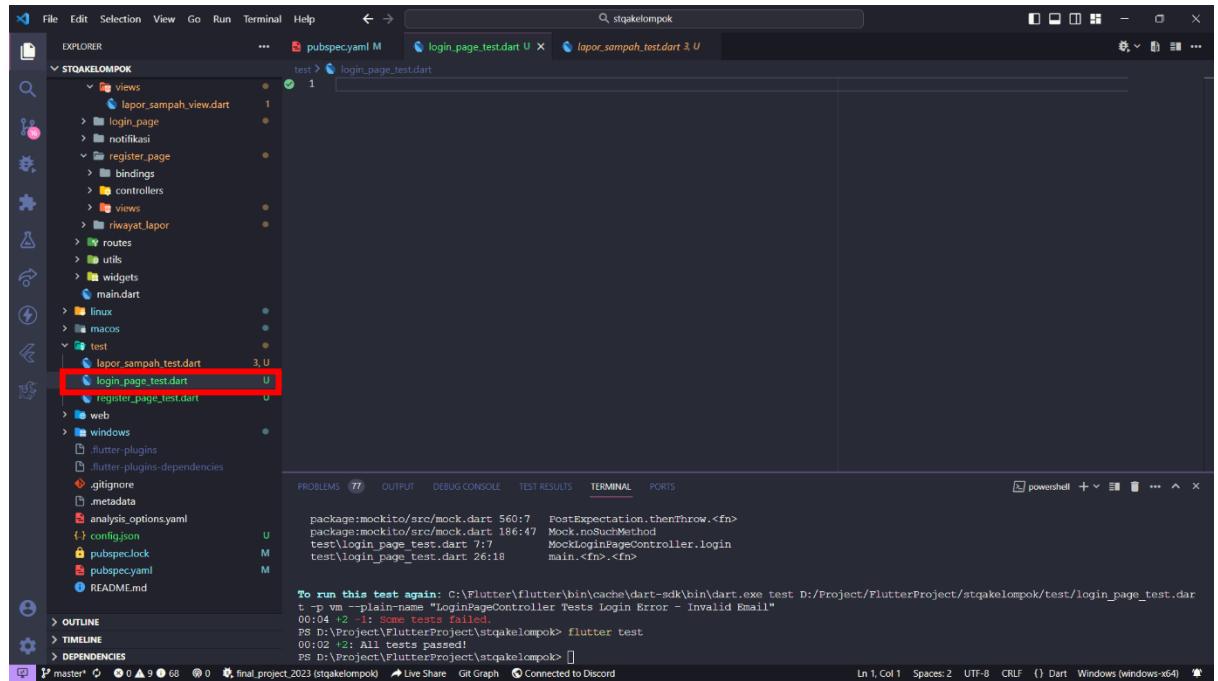
Alam Nurcahaya	20200140086
Aldy Ahmad Fauzi	20200140043
Alfiansah Erik S	20200140025
Naufal Rozan	20200140036
Zulfa Salsabila	20200140013

**PROGRAM STUDI TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MUHAMMADIYAH YOGYAKARTA**  
**(2023/2024)**

## Login Page

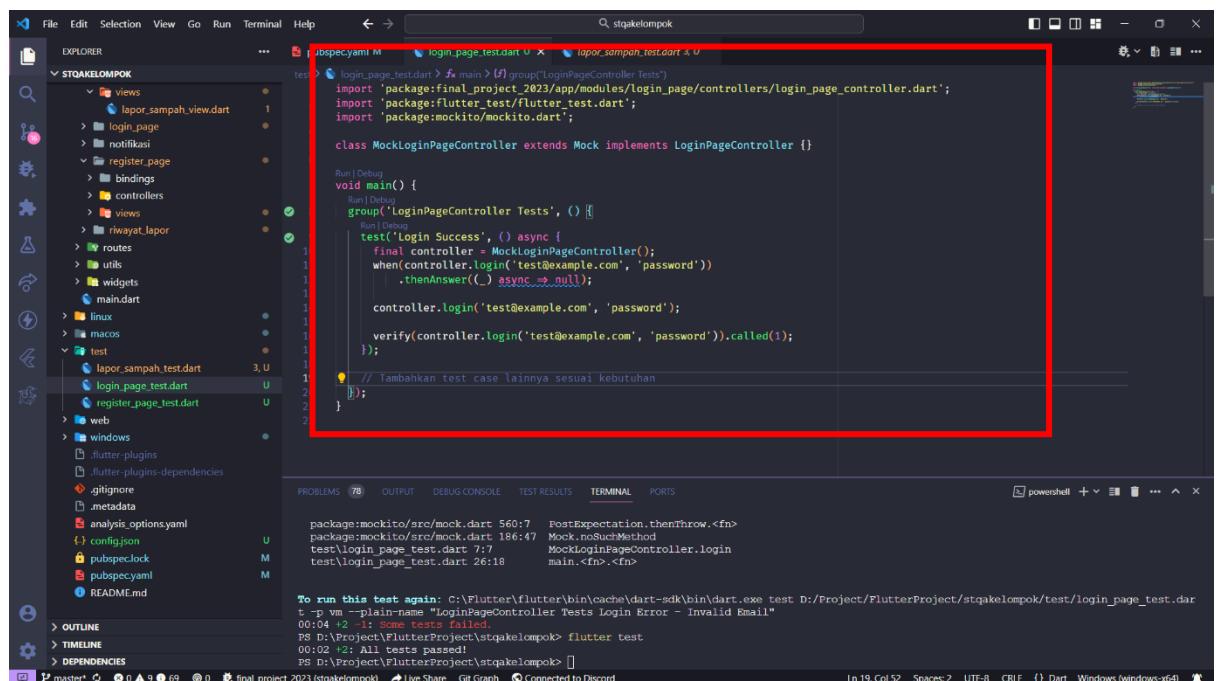
Bagian ini kami melakukan testing pada login controller login

- Buat file login\_page\_test.dart pada folder test



```
To run this test again: C:\Flutter\flutter\bin\cache\dart-sdk\bin\dart.exe test D:/Project/FlutterProject/stqakelompok/test/login_page_test.dart -p vm --plain-name "LoginPageController Tests Login Error - Invalid Email"
00:04 +2 -1: Some tests failed.
PS D:\Project\FlutterProject\stqakelompok> flutter test
00:02 +2: All tests passed!
PS D:\Project\FlutterProject\stqakelompok>
```

- Kemudian kami memasukan code test pada file login\_page\_test.dart. Pada bagian ini kami melakukan testing pada fungsi login berhasil



```
import 'package:flutter_test/flutter_test.dart';
import 'package:mockito/mockito.dart';

class MockLoginPageController extends Mock implements LoginPageController {}

void main() {
  group('LoginPageController Tests', () {
    Run | Debug
    Run | Debug
    test('Login Success', () async {
      final controller = MockLoginPageController();
      when(controller.login('test@example.com', 'password'))
          .thenAnswer((_) async => null);

      controller.login('test@example.com', 'password');

      verify(controller.login('test@example.com', 'password')).called(1);
    });
    // Tambahkan test case lainnya sesuai kebutuhan
  });
}
```

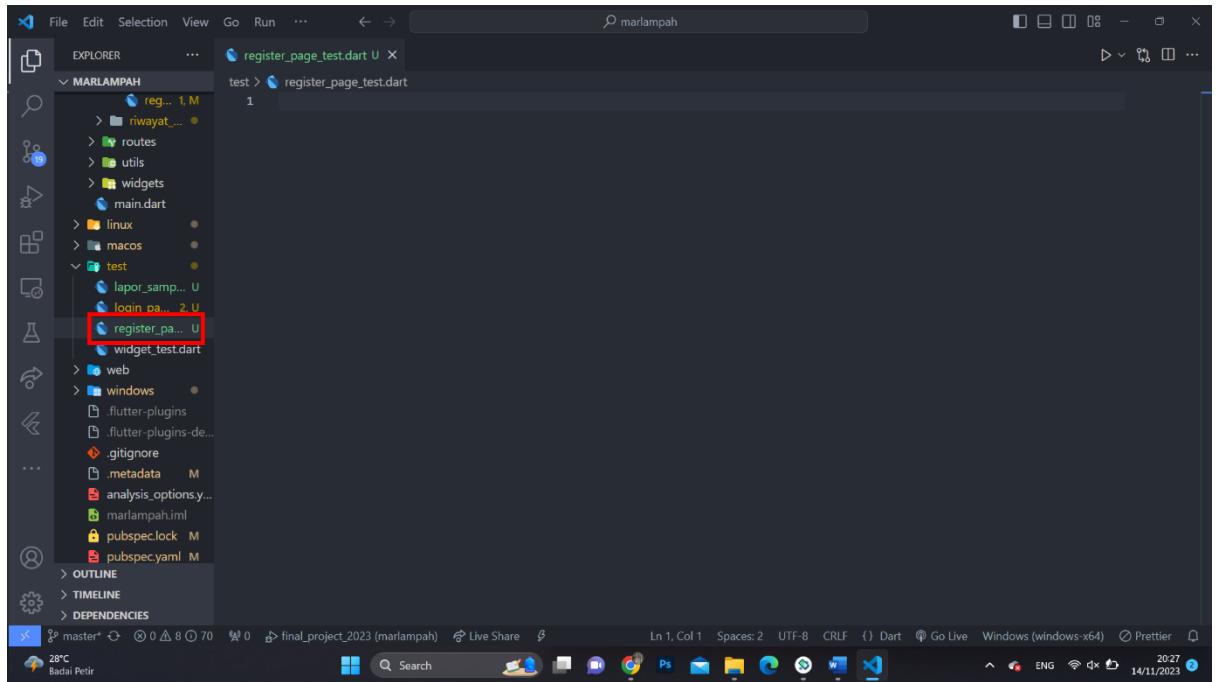
3. Setelah itu tekan button Debug Test. Setelah itu pada bagian pojok kanan visual studio code akan muncul hasil testingnya

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure for "STQAKELOMPOK". The "test" folder contains several test files: "lapor\_sampah\_test.dart", "login\_page\_test.dart", and "register\_page\_test.dart".
- Editor:** The main editor area displays the content of "login\_page\_test.dart". The code is a Dart test for a LoginPageController. It includes imports for packages, defines a MockLoginPageController, and contains a main function with a test group named "LoginPageController Tests". The test itself checks if the login method returns a success message.
- Test Results:** A red box highlights the "TEST RESULTS" tab at the bottom of the interface. This tab shows the output of the last test run. It indicates a successful run at 11/14/2023, 7:56:01 PM, with one test case named "Login Success" marked as passed.
- Bottom Status Bar:** Shows the current branch as "master", commit count as "11", and other status indicators like "Live Share" and "Connected to Discord".

## Register

1. Selanjutnya testing register, pertama kami membuat register\_page\_test.dart pada folder test

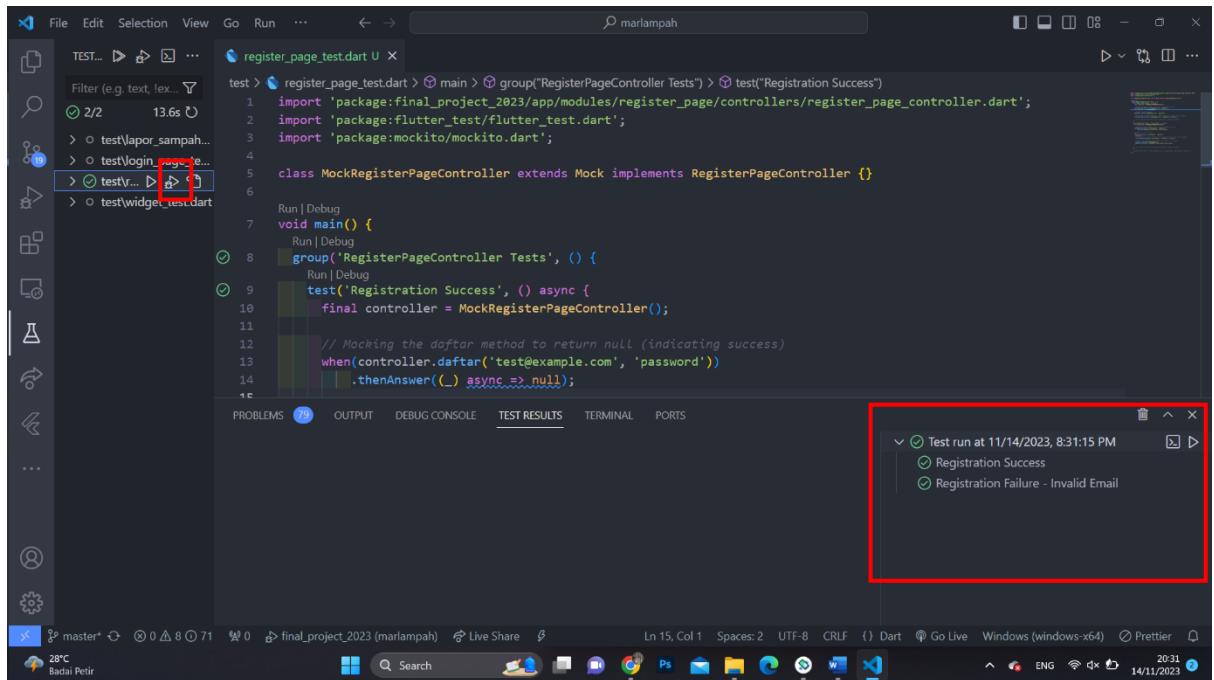


2. Kemudian tulis codenya seperti dibawah ini

```
test@register_page_test.dart:1:1
1 import 'package:flutter_test/flutter_test.dart';
2 import 'package:mockito/mockito.dart';
3
4 class MockRegisterPageController extends Mock implements RegisterPageController {
5
6   Run|Debug void main() {
7     Run|Debug group('RegisterPageController Tests', () {
8       test('Registration Success', () async {
9         final controller = MockRegisterPageController();
10        // Mocking the daftar method to return null (indicating success)
11        when(controller.daftar('test@example.com', 'password'))
12          .thenAnswer(() async => null);
13
14        // Trigger the daftar method
15        controller.daftar('test@example.com', 'password');
16
17        // Verify that the daftar method was called exactly once with the correct arguments
18        verify(controller.daftar('test@example.com', 'password')).called(1);
19
20        // You can add more assertions based on the specific behavior you expect
21      });
22
23    Run|Debug test('Registration Failure - Invalid Email', () async {
24      final controller = MockRegisterPageController();
25
26      // Mocking the daftar method to throw an error (indicating failure)
27      when(controller.daftar('invalidemail', 'password'))
28        .thenThrow(Exception('Format email tidak sesuai'));
29
30      // Trigger the daftar method
31      try {
32        controller.daftar('invalidemail', 'password');
33      } catch (e) {
34        // Verify that the daftar method was called exactly once with the correct arguments
35        verify(controller.daftar('invalidemail', 'password')).called(1);
36
37        expect(e, isInstanceOf<Exception>());
38        expect(e.toString(), 'Exception: Format email tidak sesuai');
39      }
40
41    });
42  });
43}
```

A screenshot of the Visual Studio Code interface. The left sidebar shows the same project structure as the previous screenshot. The main editor area contains the code for 'register\_page\_test.dart'. A large red box highlights the entire content of the editor area. The bottom status bar indicates the file is 13 lines long, has 62 columns, and is in UTF-8 encoding.

3. Kemudian klik Debug test, maka hasilnya akan terlihat pada bagian pojok kanan visual studio code



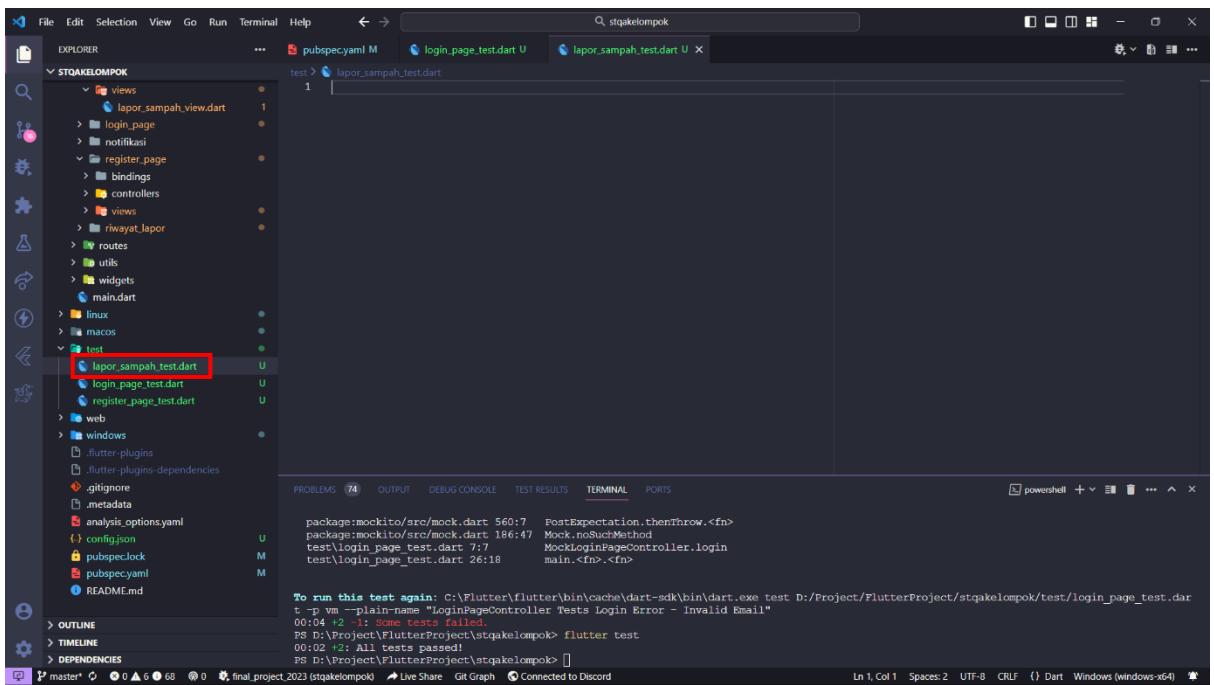
The screenshot shows the Visual Studio Code interface with the 'TEST' tab selected. A red box highlights the 'testy... D' entry in the tree view, which corresponds to the 'testy... D' entry in the code. The code itself is a Dart test for a 'RegisterPageController'. The 'TEST RESULTS' tab is active, showing a summary of the test run:

```

    ✓ Test run at 11/14/2023, 8:31:15 PM
    ✓ Registration Success
    ✓ Registration Failure - Invalid Email
  
```

## Lapor Sampah

- Pada testing laporan sampah, pertama kami membuat file lapor\_sampah\_test.dart pada folder test.



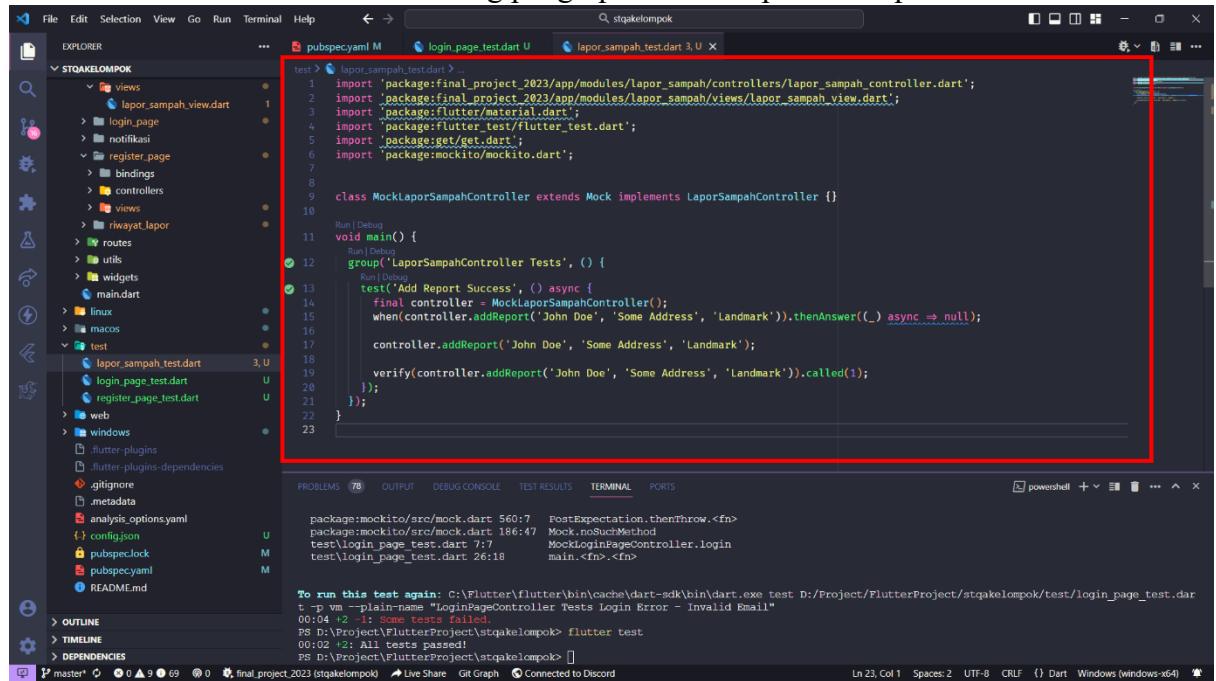
The screenshot shows the Visual Studio Code interface with the 'EXPLORER' tab selected. A red box highlights the 'lapor\_sampah\_test.dart' file in the 'test' folder. The 'TEST RESULTS' tab is active, showing the output of the test run:

```

    package:mockito/src/mock.dart 560:7  PostException.thenThrow.<fn>
    package:mockito/src/mock.dart 186:47  Mock.noSuchMethod
    test\login_page_test.dart 7:7      MockLoginPageController.login
    test\login_page_test.dart 26:18     main.<fn>.<fn>

    To run this test again: C:\Flutter\flutter\bin\cache\dart-sdk\bin\dart.exe test D:/Project/FlutterProject/stqakelompok/test/login_page_test.dart
    00:04 #2 -1: Some tests failed
    00:04 #2 -1: Some tests failed
    PS D:\Project\FlutterProject\stqakelompok> flutter test
    00:02 #2: All tests passed!
    PS D:\Project\FlutterProject\stqakelompok> []
  
```

## 2. Kemudian tulis code untuk mentesting penginputan data laporan sampah



```

test > lapor_sampah_test.dart >
1 import 'package:final_project_2023/app/modules/lor_sampah/controllers/lor_sampah_controller.dart';
2 import 'package:final_project_2023/app/modules/lor_sampah/views/lor_sampah_view.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_test/flutter_test.dart';
5 import 'package:get/get.dart';
6 import 'package:mockito/mockito.dart';

7
8
9 class MockLaporSampahController extends Mock implements LaporSampahController {}
10
11 Run | Debug void main() {
12   Run | Debug
13   group('LaporSampahController Tests', () {
14     Run | Debug
15     test('Add Report Success', () async {
16       final controller = MockLaporSampahController();
17       when(controller.addReport('John Doe', 'Some Address', 'Landmark')).thenAnswer((_) async => null);
18
19       controller.addReport('John Doe', 'Some Address', 'Landmark');
20
21       verify(controller.addReport('John Doe', 'Some Address', 'Landmark')).called(1);
22     });
23   });
}

```

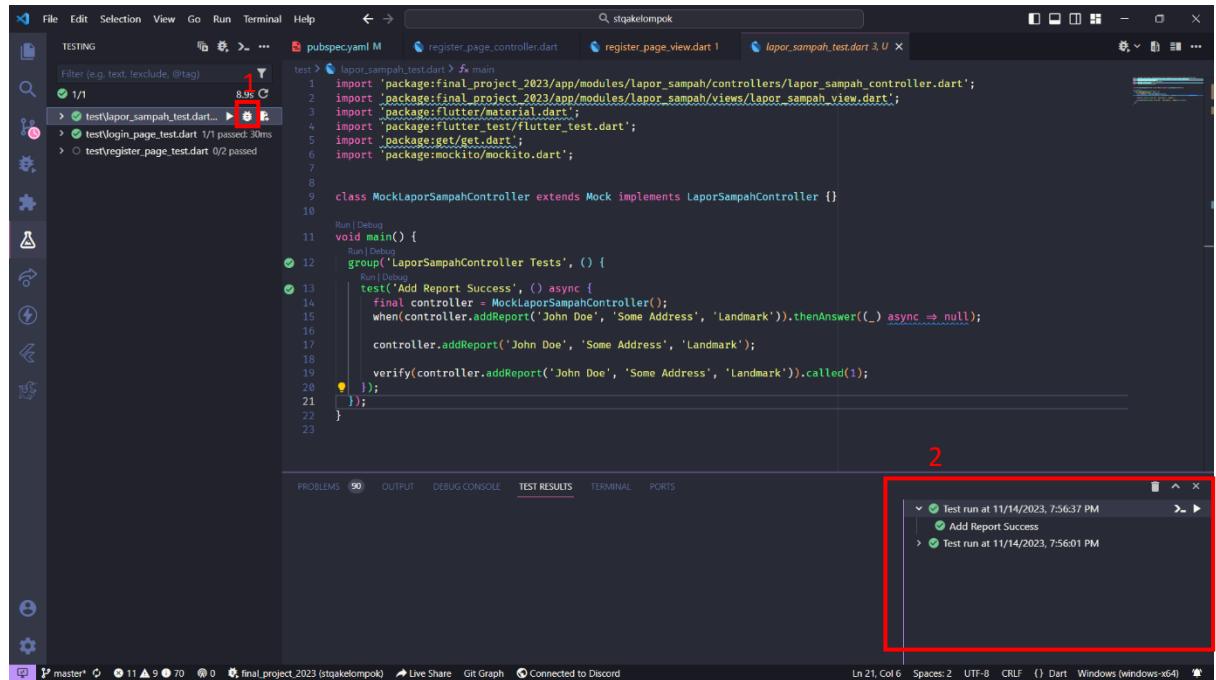
PROBLEMS OUTPUT DEBUG CONSOLE TEST RESULTS TERMINAL PORTS

package:mockito/src/mock.dart 560:7 PostExpectation.thenThrow.<fn>
package:mockito/src/mock.dart 166:47 Mock.noSuchMethod
test\login\_page\_test.dart 7:7 MockLoginPageController.login
test\login\_page\_test.dart 26:18 main.<fn>.<fn>

To run this test again: C:\Flutter\flutter\bin\cache\dart-sdk\bin\dart.exe test D:/Project/FlutterProject/stqakelompok/test/login\_page\_test.dart -p vm --plain-name "LoginPageController Tests Login Error - Invalid Email"
00:04 [2] Some tests failed.
PS D:\Project\FlutterProject\stqakelompok> flutter test
00:02 [2] All tests passed!
PS D:\Project\FlutterProject\stqakelompok> []

Ln 23, Col 6 Spaces: 2 UTF-8 CR/LF {} Dart Windows (Windows-x64)

## 3. Setelah itu klik Debug Test kemudian hasil testing akan muncul pada bagian pojok kanan visual studio code.



TESTING Filter (e.g. text, exclude, @tag) 1

- 1/1 test\lapor\_sampah\_test.dart... 2
- > test\login\_page\_test.dart 1/1 passed: 30ms
- > test\register\_page\_test.dart 0/2 passed

test > lapor\_sampah\_test.dart > main

```

1 import 'package:final_project_2023/app/modules/lor_sampah/controllers/lor_sampah_controller.dart';
2 import 'package:final_project_2023/app/modules/lor_sampah/views/lor_sampah_view.dart';
3 import 'package:flutter/material.dart';
4 import 'package:flutter_test/flutter_test.dart';
5 import 'package:get/get.dart';
6 import 'package:mockito/mockito.dart';

7
8
9 class MockLaporSampahController extends Mock implements LaporSampahController {}
10
11 Run | Debug void main() {
12   Run | Debug
13   group('LaporSampahController Tests', () {
14     Run | Debug
15     test('Add Report Success', () async {
16       final controller = MockLaporSampahController();
17       when(controller.addReport('John Doe', 'Some Address', 'Landmark')).thenAnswer((_) async => null);
18
19       controller.addReport('John Doe', 'Some Address', 'Landmark');
20
21       verify(controller.addReport('John Doe', 'Some Address', 'Landmark')).called(1);
22     });
23   });
}

```

PROBLEMS OUTPUT DEBUG CONSOLE TEST RESULTS TERMINAL PORTS

Test run at 11/14/2023, 7:56:37 PM
 Add Report Success
 Test run at 11/14/2023, 7:56:01 PM

Ln 21, Col 6 Spaces: 2 UTF-8 CR/LF {} Dart Windows (Windows-x64)