Laporan Proyek Pengujian dan Kualitas Perangkat Lunak
BDD - Automation Testing Using Cucumber
ATDD –Automation Testing Using Robot Framework

Team	:	14
NIM/Name	:	1. Horas MP Saragih Sidabalok – 11322016 2. Mario Andreas Manurung -11322030 3. Nania Avantika Oligiviana Pangaribuan - 11322045 4. Vanessa Siahaan - 11322060

Pengujian Website Foodpedia - Numero Sada (CUCUMBER)

1. Fungsi Login Admin
   a. Scenario

```
Scenario: Haven't logged in yet

    Given I am on the Login Page

    Then I fill my login info

    Then I click the login button

    Then  There the dashboard should be there
```

   b. Step Definition

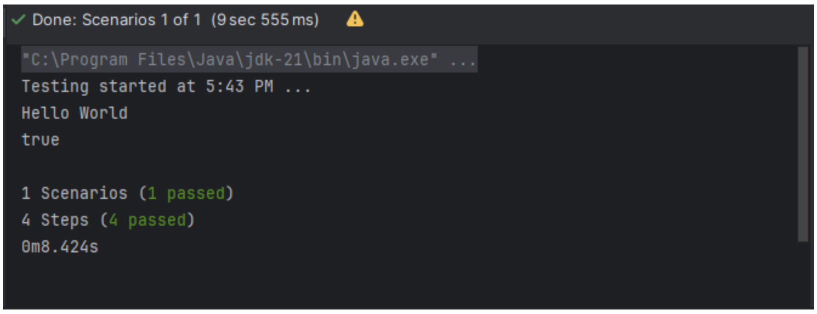
```
public class WebLogin {

    WebDriver browser = new ChromeDriver();

    @Given("I am on the Login Page")
    public void visitThePage() {
        browser.get("http://127.0.0.1:8000/login");
        System.out.println("Hello World");
    }

    @Then("I fill my login info")
    public void iFillMyLoginInfo() {

browser.findElement(By.id("email")).sendKeys("numerosada@gmail.com");

browser.findElement(By.id("password")).sendKeys("numerosada123");
    }

    @Then("I click the login button")
    public void iClickTheLoginButton() {
        var button =
browser.findElement(By.cssSelector("button[type=submit].btn.btn-
primary"));
        button.click();
    }

    @Then("There the dashboard should be there")
    public void thereTheDashboardShouldBeThere() {
        var dashboardContentContainer =
browser.findElement(By.cssSelector("section.content"));
        if(dashboardContentContainer.isDisplayed()){
```

```
System.out.println(dashboardContentContainer.isDisplayed());
            }
      }
}
```

c. Test Runner

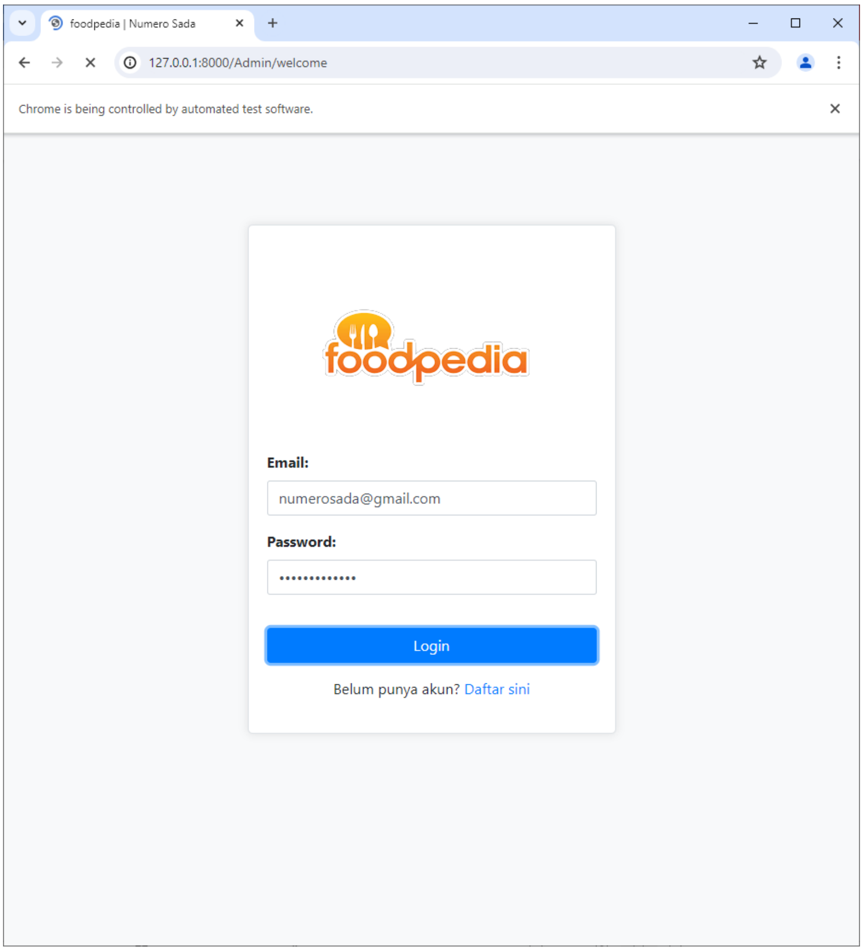
```
public class TestRunner {
}
```

d. *Console* setelah Running



```
✓ Done: Scenarios 1 of 1  (9 sec 555 ms)

"C:\Program Files\Java\jdk-21\bin\java.exe" ...
Testing started at 5:43 PM ...
Hello World
true

1 Scenarios (1 passed)
4 Steps (4 passed)
0m8.424s
```

e. *Snippet* Browser saat Testing



127.0.0.1:8000/Admin/welcome

Chrome is being controlled by automated test software.

Email: numerosada@gmail.com

Password: •••••••••••••

Belum punya akun? Daftar sini

2. Fungsi Login Registrasi
    a. Scenario

```
Feature: Haven't logged in yet
  As a user
  I want to register account
  To be able to access features

  Scenario:
    Given I am on the register page
    Then I fill my informations with this details
      | field            | value                |
      | nama             | John Doe             |
      | email            | john@doe.com         |
      | password         | johndoeloveindonesia |
      | confirm_password | johndoeloveindonesia |
    Then I click the signup button
    Then I Log In to the website
      | field            | value                |
      | nama             | John Doe             |
      | email            | john@doe.com         |
      | password         | johndoeloveindonesia |
      | confirm_password | johndoeloveindonesia |
    Then the user will open the homepage
```

    b. Step Definition

```
import io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import io.cucumber.java.en.When;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import javax.xml.crypto.Data;

public class WebRegister {

    WebDriver browser = new ChromeDriver();

    @Given("I am on the register page")
    public void IamOnTheSignUpPage() throws Throwable {
        browser.get("http://localhost:8000/register");
    }

    @Then("I fill my informations with this details")
    public void iFillMyInformationsWithThisDetails(DataTable
signUpInfo) {
        var fetchedSignupInfo = signUpInfo.asMap(String.class,
String.class);

        var getUsernameField =
browser.findElement(By.cssSelector("input#name.form-control"));
        getUsernameField.sendKeys(fetchedSignupInfo.get("nama"));

        var getEmailField =
browser.findElement(By.cssSelector("input#email.form-control"));
        getEmailField.sendKeys(fetchedSignupInfo.get("email"));
        var                              getPasswordField =
browser.findElement(By.cssSelector("input#password.form-control"));
        getPasswordField.sendKeys(fetchedSignupInfo.get("password"));

        var getConfirmPasswordField =
browser.findElement(By.cssSelector("input#password_confirmation.form-
control"));

getConfirmPasswordField.sendKeys(fetchedSignupInfo.get("confirm_passwor
d"));
    }

    @Then("I click the signup button")
    public void iClickTheSignupButton() {
        var button =
browser.findElement(By.cssSelector("button[type=submit]"));
        button.click();
    }


    @Then("I Log In to the website")
    public void iLogInToTheWebsite(DataTable signUpInfo) {
        var fetchedSignupInfo = signUpInfo.asMap(String.class,
String.class);

        var getEmailField = browser.findElement(By.id("email"));
        getEmailField.sendKeys(fetchedSignupInfo.get("email"));
        var getPasswordField = browser.findElement(By.id("password"));
        getPasswordField.sendKeys(fetchedSignupInfo.get("password"));

        var button =
browser.findElement(By.cssSelector("button[type=submit]"));
        button.click();
    }

    @Then("the user will open the homepage")
    public void theUserWillOpenTheHomepage() throws Exception {
        browser.findElement(By.cssSelector("h1.display-3.font-weight-
bold.text-white")).isDisplayed();
    }
}
```

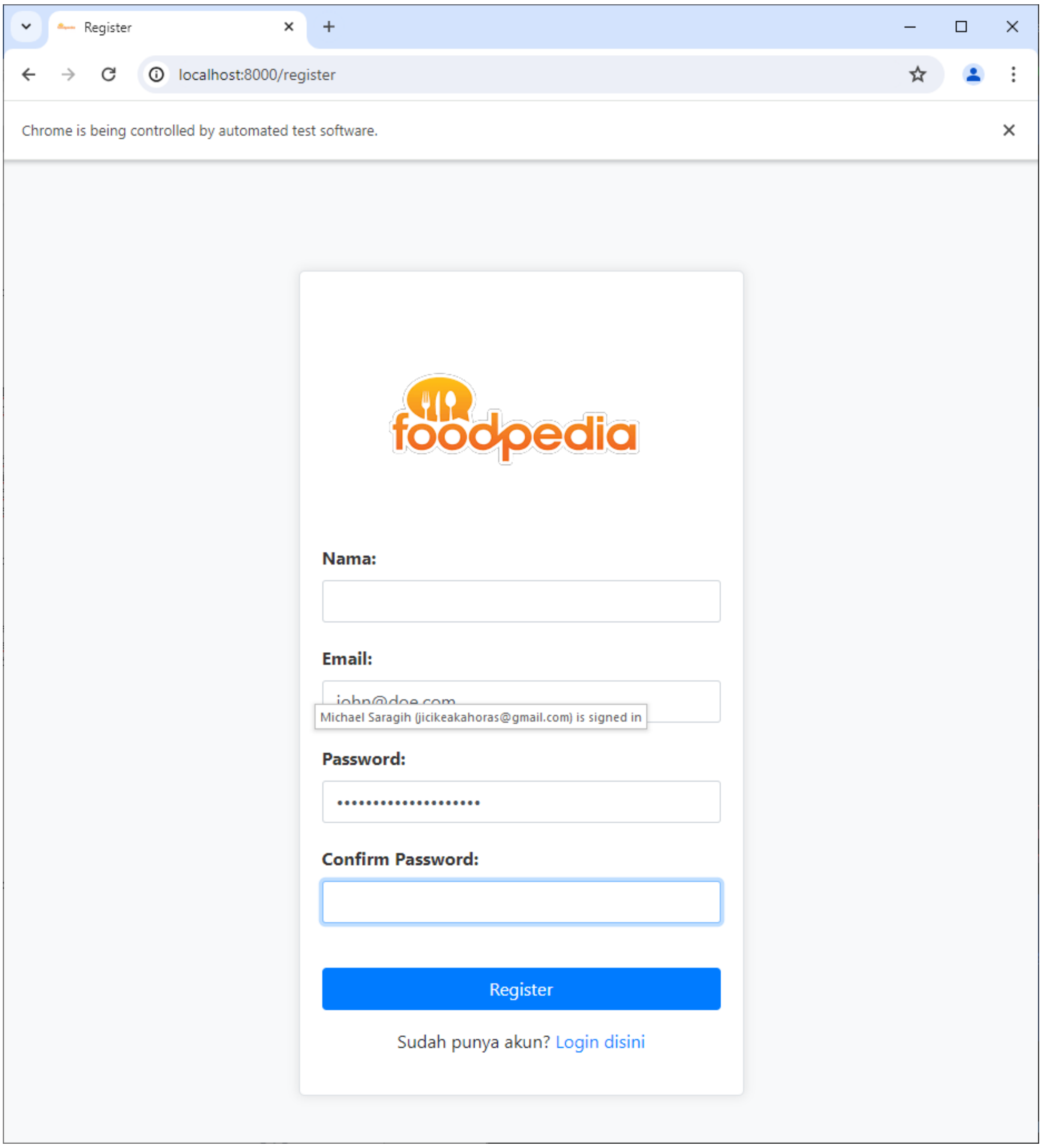
c. Test Runner

```
public class TestRunner {
}
```

d. *Console* setelah Running

```
Failed scenarios:
file:///C:/Users/neko/IdeaProjects/pkpltest/src/test/resources/featu
res/register.feature:6 #

1 Scenarios (1 failed)
5 Steps (1 failed, 4 passed)
0m6.717s
```

e. *Snippet* Browser saat Testing



Register

localhost:8000/register

Chrome is being controlled by automated test software.

Nama:

Email:

john@doe.com

Michael Saragih (jicikeakahoras@gmail.com) is signed in

Password:

•••••••••••••••••••

Confirm Password:

Sudah punya akun? Login disini

3. Fungsi Login Customer

a. Scenario

```
Feature: Haven't logged in yet
    As a registered user
    I want to login to the web app
    So I manage my website

    Scenario: Successful user login
        Given I am on the Login Page
        Then I fill my login info
            | field    | value                 |
            | email    | john@doe.com          |
            | password | johndoeloveindonesia |
        Then I click the login button
```

b. Step Definition

```
import                                io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class WebLogin {

    WebDriver browser = new ChromeDriver();

    @Given("I am on the Login Page")
    public void visitThePage() {
        browser.get("http://127.0.0.1:8000/login");
        System.out.println("Hello World");
    }

    @Then("I fill my login info")
    public void iFillMyLoginInfo(DataTable inputs) {
        var UserLoginInfo = inputs.asMap(String.class, String.class);


browser.findElement(By.id("email")).sendKeys(UserLoginInfo.get("email")
);

browser.findElement(By.id("password")).sendKeys(UserLoginInfo.get("pass
word"));
    }

    @Then("I click the login button")
    public void iClickTheLoginButton() {
        var button =
browser.findElement(By.cssSelector("button[type=submit].btn.btn-
primary"));
        button.click();
    }

    @Then("the dashboard should be there")
    public void thereTheDashboardShouldBeThere() {
        var dashboardContentContainer =
browser.findElement(By.cssSelector("section.content"));
        if(dashboardContentContainer.isDisplayed()){
           System.out.println(dashboardContentContainer.isDisplayed());
        }
    }
}
```

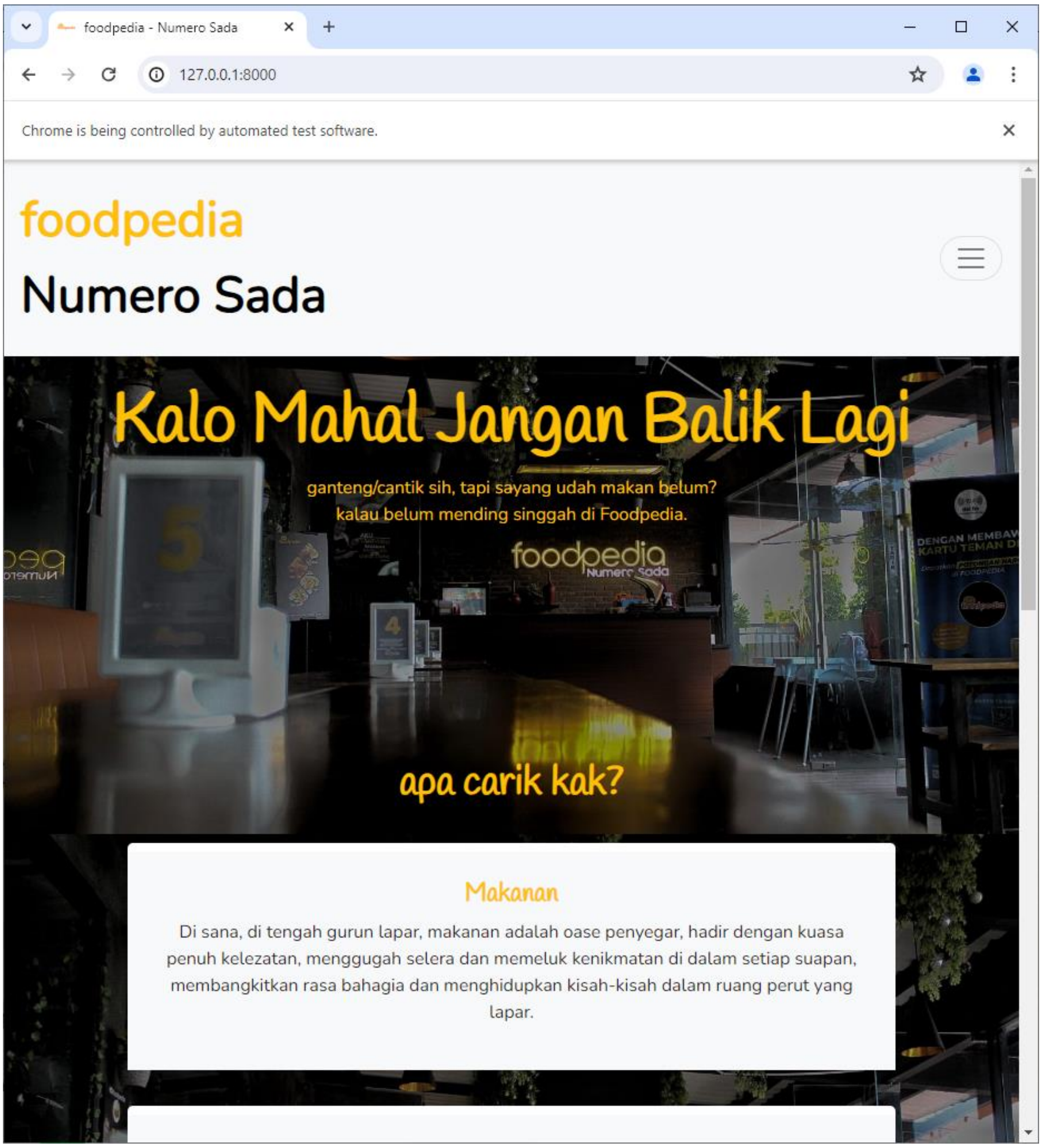
c. Test Runners

```
public class TestRunner {
}
```

d. *Console* setelah Running

```
Testing started at 9:49 PM ...
Hello World

1 Scenarios (1 passed)
3 Steps (3 passed)
0m9.580s
```

e. *Snippet* Browser saat Testing



Chrome is being controlled by automated test software.

foodpedia

Numero Sada

Kalo Mahal Jangan Balik Lagi

ganteng/cantik sih, tapi sayang udah makan belum?
kalau belum mending singgah di Foodpedia.

apa carik kak?

Makanan

Di sana, di tengah gurun lapar, makanan adalah oase penyegar, hadir dengan kuasa penuh kelezatan, menggugah selera dan memeluk kenikmatan di dalam setiap suapan, membangkitkan rasa bahagia dan menghidupkan kisah-kisah dalam ruang perut yang lapar.

4. Fungsi menambah menu

a. Scenario

```
Feature: Haven't logged in yet
  As an admin, I want to log to the system
  and adding a new menu item

  Scenario: Successful user login
    Given I open the Login Page
    Then Fill my login info
      | field    | value               |
      | email    | numerosada@gmail.com |
      | password | numerosada123       |
    Then Click at the login button
    Then Open the menu page
    Then Open add menu page
    Then fill the new menu informations
```

b. Step Definition

```
import                                io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class AddMenu {

    WebDriver browser = new ChromeDriver();

    @Given("I open the Login Page")
    public void IOpenTheLoginPage() {
        browser.get("http://127.0.0.1:8000/login");
        System.out.println("Hello World");
    }

    @Then("Fill my login info")
    public void FillMyLoginInfo(DataTable inputs) {
        var UserLoginInfo = inputs.asMap(String.class, String.class);


browser.findElement(By.id("email")).sendKeys(UserLoginInfo.get("email")
);

browser.findElement(By.id("password")).sendKeys(UserLoginInfo.get("pass
word"));
    }

    @Given("Open the menu page")
    public void visitThePage() {
        browser.get("http://127.0.0.1:8000/post");
    }

    @Then("Click at the login button")
    public void clickAtTheLoginButton() {
        var button =
browser.findElement(By.cssSelector("button[type=submit].btn.btn-
primary"));
        button.click();
    }

    @Then("Open add menu page")
    public void openAddMenuPage() {
        var button = browser.findElement(By.cssSelector("body > div >
div.content-wrapper > section.content > div > div.card-body > a"));
        button.click();
    }

    @Then("fill the new menu informations")
    public void fillTheNewMenuInformations() {
        var nameForm =
browser.findElement(By.cssSelector("input[name=\"nama\"].form-
control"));
        nameForm.sendKeys("Batu Goreng");
        var deskripsiForm =
browser.findElement(By.cssSelector("input[name=\"nama\"].form-
control"));
        deskripsiForm.sendKeys("Batu Goreng sangat nikmat");
        var hargaForm =
browser.findElement(By.cssSelector("input[name=\"price\"].form-
control"));
hargaForm.sendKeys("45000");     }
}
```

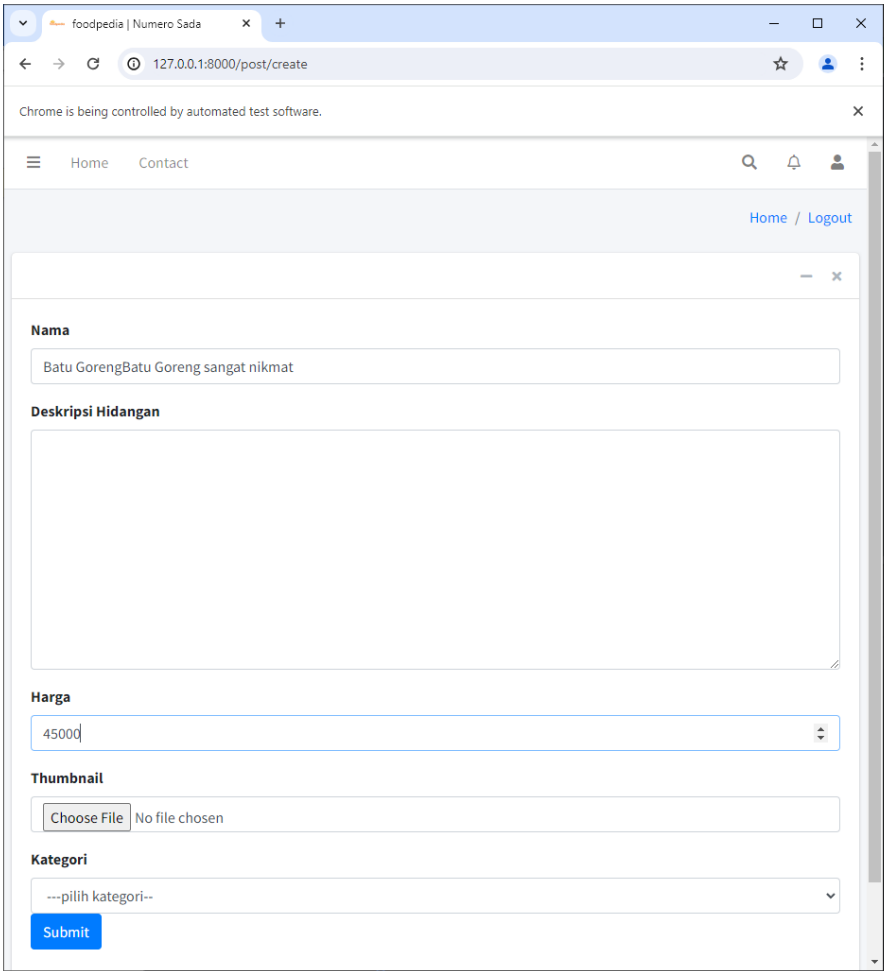
c. Test Runner

```
public class TestRunner {
}
```

d. *Console* setelah Running

```
Testing started at 11:20 PM ...
Hello World

1 Scenarios (1 passed)
6 Steps (6 passed)
0m7.256s
```

e. *Snippet* Browser saat Testing



Chrome is being controlled by automated test software.

Nama: Batu GorengBatu Goreng sangat nikmat

Deskripsi Hidangan:

Harga: 45000

Thumbnail: No file chosen

Kategori: ---pilih kategori--

5. Fungsi melihat reservasi

a. Scenario

```
Feature: Add Menu
  As an admin, I want to log to the system
  and adding a new menu item
```

```
Scenario: Haven't logged            in yet
  Given I visit the Login Page
  Then Do The Login
    | field    | value                  |
    | email    | numerosada@gmail.com |
    | password | numerosada123          |
  Then Open the reservation page
```

b. Step Definition

```
import io.cucumber.datatable.DataTable;
import io.cucumber.java.en.Given;
import io.cucumber.java.en.Then;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class SeeReservation {

    WebDriver browser = new ChromeDriver();

    @Given("I visit the Login Page")
    public void VisitTheLoginPage() {
        browser.get("http://127.0.0.1:8000/login");
        System.out.println("Hello World");
    }

    @Then("Do The Login")
    public void DoTheLogin(DataTable inputs) {
        var UserLoginInfo = inputs.asMap(String.class, String.class);


browser.findElement(By.id("email")).sendKeys(UserLoginInfo.get("email")
);

browser.findElement(By.id("password")).sendKeys(UserLoginInfo.get("pass
word"));

        var button =
browser.findElement(By.cssSelector("button[type=submit].btn.btn-
primary"));
        button.click();
    }

    @Given("Open the reservation page")
    public void openTheReservationPage() {

browser.findElement(By.cssSelector("a[href='/reservations']")).click();
    }
}
```

c. Test Runners

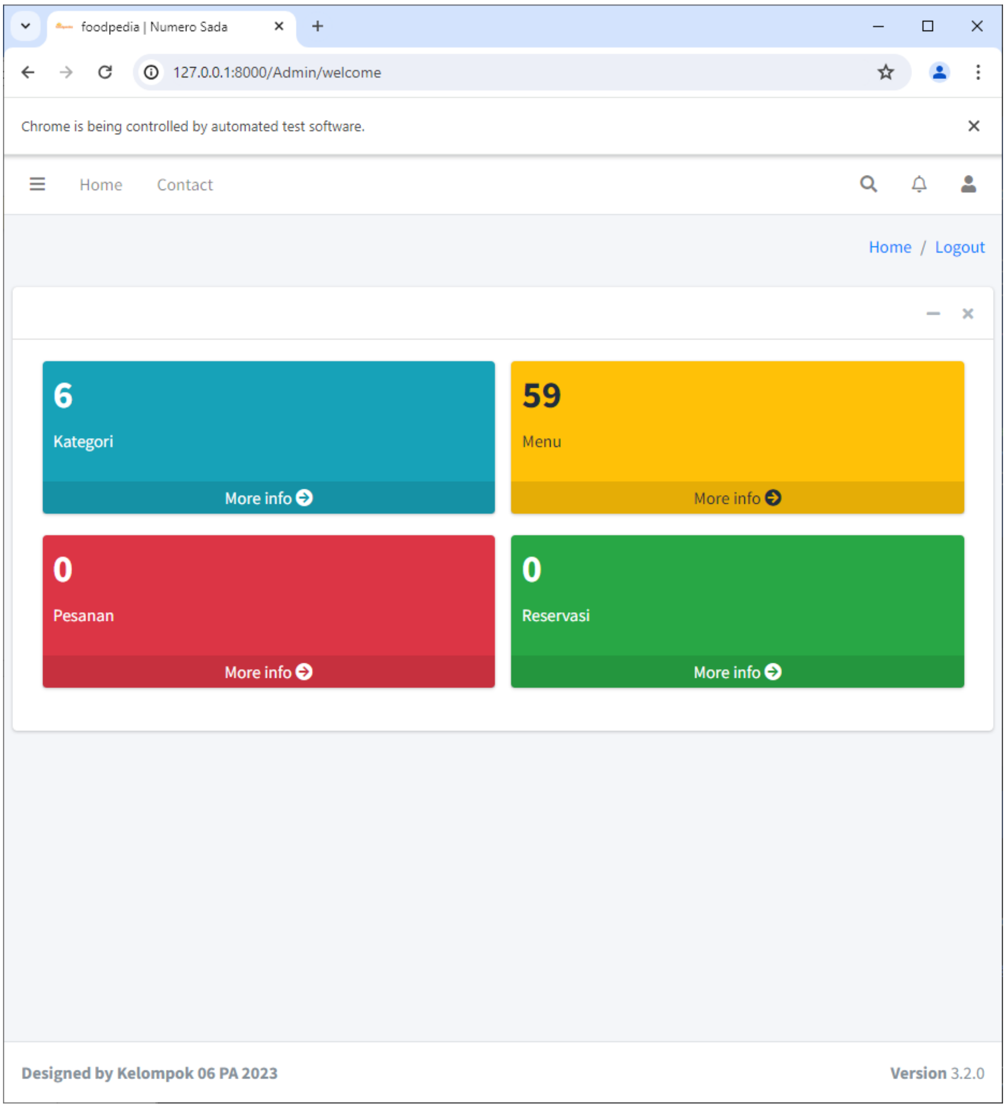
```
public class TestRunner {
}
```

d. *Console* setelah Running

```
1 Scenarios (1 failed)
```

```
3   Steps   (1   failed,   2            passed)
0m8.833s
```

e. *Snippet* Browser saat Testing



foodpedia | Numero Sada

127.0.0.1:8000/Admin/welcome

Chrome is being controlled by automated test software.

6 Kategori

59 Menu

0 Pesanan

0 Reservasi

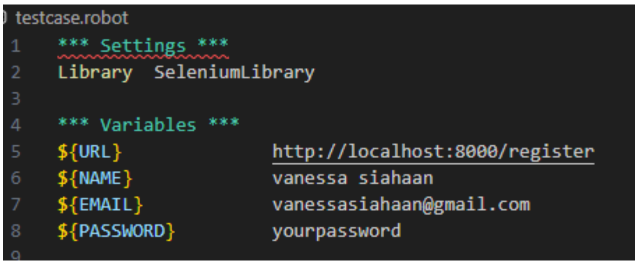
Designed by Kelompok 06 PA 2023

Version 3.2.0

Pengujian Website Foodpedia - Numero Sada (Robot Framework)

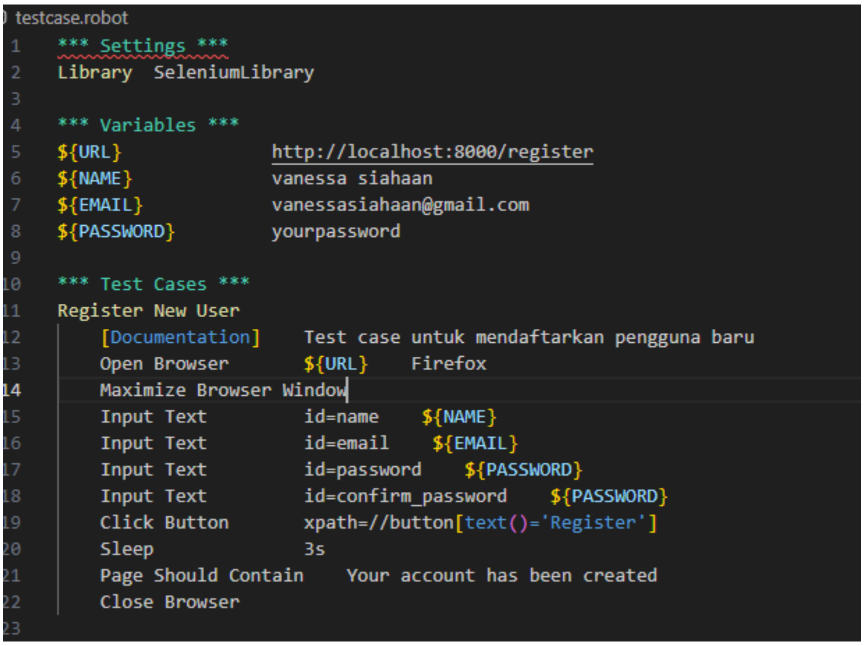
1. Fungsi Registrasi Customer

a. Scenario



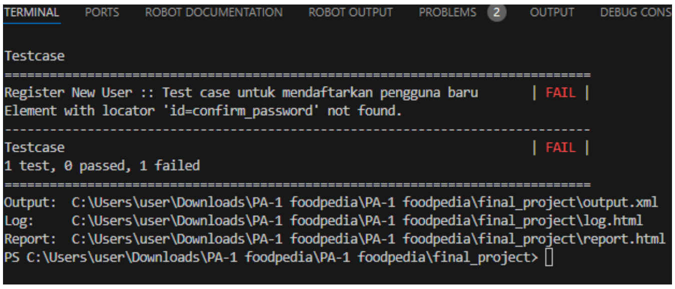
```
*** Settings ***
Library  SeleniumLibrary

*** Variables ***
${URL}              http://localhost:8000/register
${NAME}             vanessa siahaan
${EMAIL}            vanessasiahaan@gmail.com
${PASSWORD}         yourpassword
```

b. Step Definition



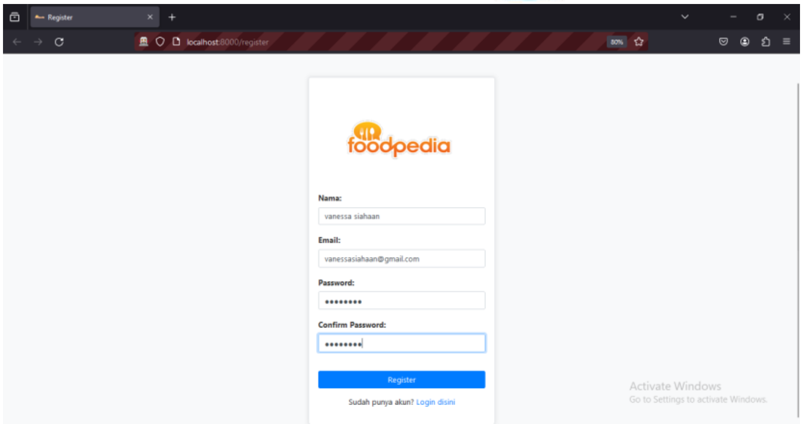
```
*** Settings ***
Library  SeleniumLibrary

*** Variables ***
${URL}              http://localhost:8000/register
${NAME}             vanessa siahaan
${EMAIL}            vanessasiahaan@gmail.com
${PASSWORD}         yourpassword

*** Test Cases ***
Register New User
    [Documentation]    Test case untuk mendaftarkan pengguna baru
    Open Browser       ${URL}    Firefox
    Maximize Browser Window
    Input Text         id=name     ${NAME}
    Input Text         id=email     ${EMAIL}
    Input Text         id=password     ${PASSWORD}
    Input Text         id=confirm_password     ${PASSWORD}
    Click Button       xpath=//button[text()='Register']
    Sleep              3s
    Page Should Contain    Your account has been created
    Close Browser
```

c. *Console* setelah Running



```
Testcase
==============================================================================
Register New User :: Test case untuk mendaftarkan pengguna baru       | FAIL |
Element with locator 'id=confirm_password' not found.
------------------------------------------------------------------------------
Testcase                                                              | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\output.xml
Log:     C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\log.html
Report:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\report.html
PS C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project> 
```

d. *Snippet* Browser saat Testing



Nama: vanessa siahaan

Email: vanessasiahaan@gmail.com

Password: ••••••••

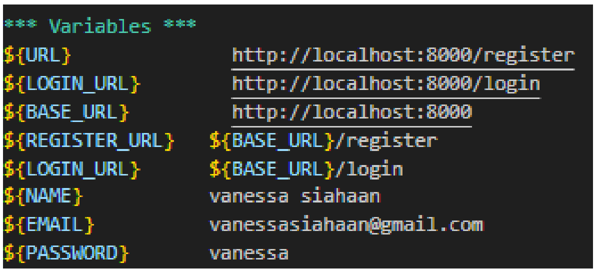
Confirm Password: ••••••••

Register
Sudah punya akun? Login disini
Activate Windows
Go to Settings to activate Windows.

Penjelasan : Kode Robot Framework ini mengotomatisasi registrasi pengguna baru di halaman web. Menggunakan SeleniumLibrary untuk kontrol browser, skrip ini mengisi form registrasi dengan variabel yang ditentukan (URL, nama, email, password), lalu mengklik tombol "Register". Setelah menunggu 3 detik, skrip memverifikasi bahwa halaman menampilkan pesan sukses registrasi. Akhirnya, browser ditutup. Pastikan locator sesuai dengan elemen HTML halaman.

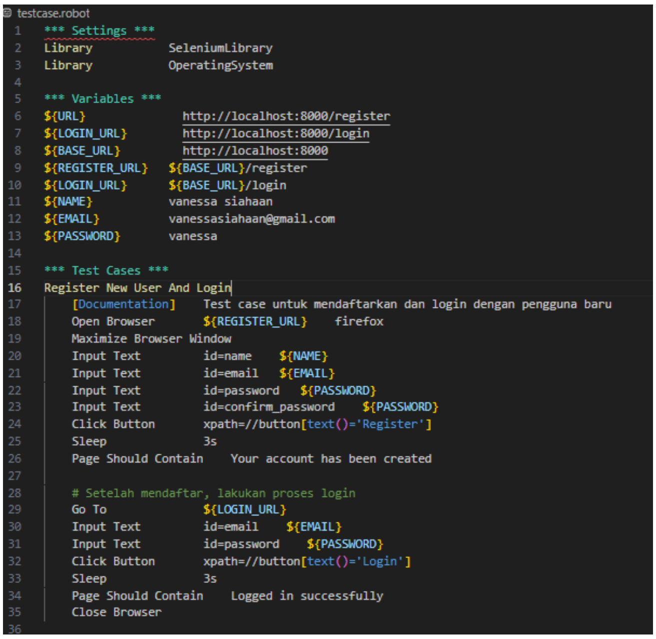
2. Fungsi Login Registrasi

a. Scenario



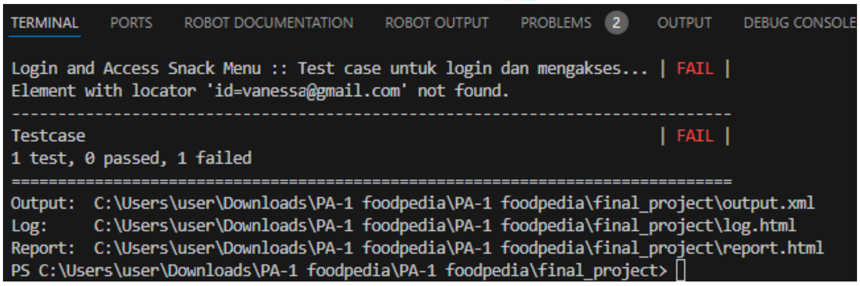
```
*** Variables ***
${URL}               http://localhost:8000/register
${LOGIN_URL}         http://localhost:8000/login
${BASE_URL}          http://localhost:8000
${REGISTER_URL}    ${BASE_URL}/register
${LOGIN_URL}       ${BASE_URL}/login
${NAME}            vanessa siahaan
${EMAIL}           vanessasiahaan@gmail.com
${PASSWORD}        vanessa
```

b. Step Definition



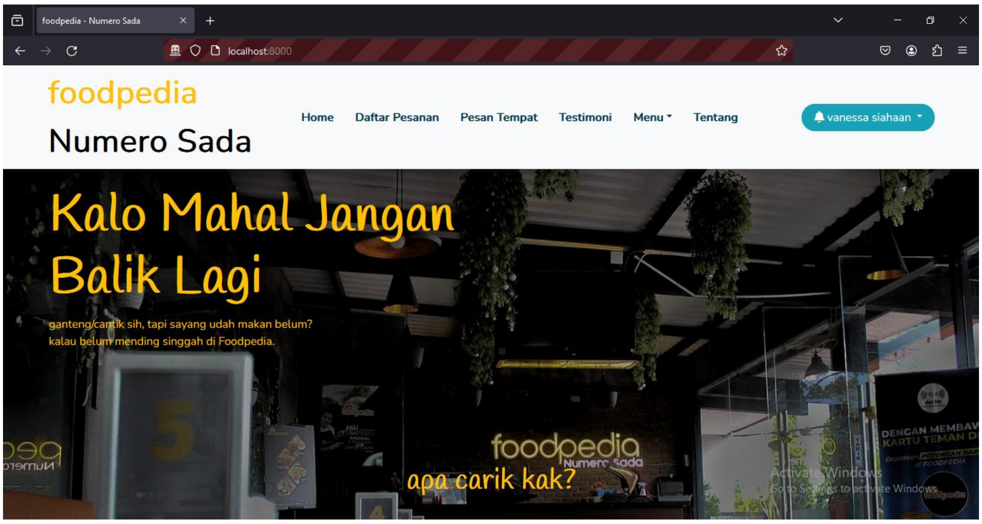
```
*** Settings ***
Library           SeleniumLibrary
Library           OperatingSystem

*** Variables ***
${URL}              http://localhost:8000/register
${LOGIN_URL}        http://localhost:8000/login
${BASE_URL}         http://localhost:8000
${REGISTER_URL}   ${BASE_URL}/register
${LOGIN_URL}      ${BASE_URL}/login
${NAME}           vanessa siahaan
${EMAIL}          vanessasiahaan@gmail.com
${PASSWORD}       vanessa

*** Test Cases ***
Register New User And Login
    [Documentation]    Test case untuk mendaftarkan dan login dengan pengguna baru
    Open Browser       ${REGISTER_URL}    firefox
    Maximize Browser Window
    Input Text         id=name    ${NAME}
    Input Text         id=email   ${EMAIL}
    Input Text         id=password   ${PASSWORD}
    Input Text         id=confirm_password    ${PASSWORD}
    Click Button       xpath=//button[text()='Register']
    Sleep              3s
    Page Should Contain    Your account has been created

    # Setelah mendaftar, lakukan proses login
    Go To              ${LOGIN_URL}
    Input Text         id=email    ${EMAIL}
    Input Text         id=password    ${PASSWORD}
    Click Button       xpath=//button[text()='Login']
    Sleep              3s
    Page Should Contain    Logged in successfully
    Close Browser

```

c. *Console* setelah Running



```
Login and Access Snack Menu :: Test case untuk login dan mengakses... | FAIL |
Element with locator 'id=vanessa@gmail.com' not found.
------------------------------------------------------------------------------
Testcase                                                              | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\output.xml
Log:     C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\log.html
Report:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\report.html
PS C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project>
```

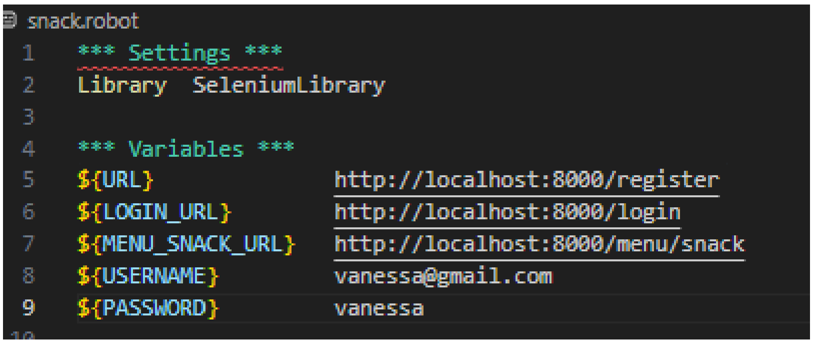
d. *Snippet* Browser saat Testing



Penjelasan : untuk mengotomatisasi Registrasi customer,masuk ke login dan akses ke halaman "Dashboard" dalam aplikasi web lokal. Pertama, browser dibuka dan diarahkan ke halaman Registrasi kemudian mengisis form lalu lanjut ke form login. Kemudian, kredensial login dimasukkan ke dalam form, dan tombol login ditekan. Setelah berhasil login, halaman diarahkan ke "halaman utama". Pada akhirnya, kode memeriksa apakah halaman "snack menu" telah dimuat dengan sukses.

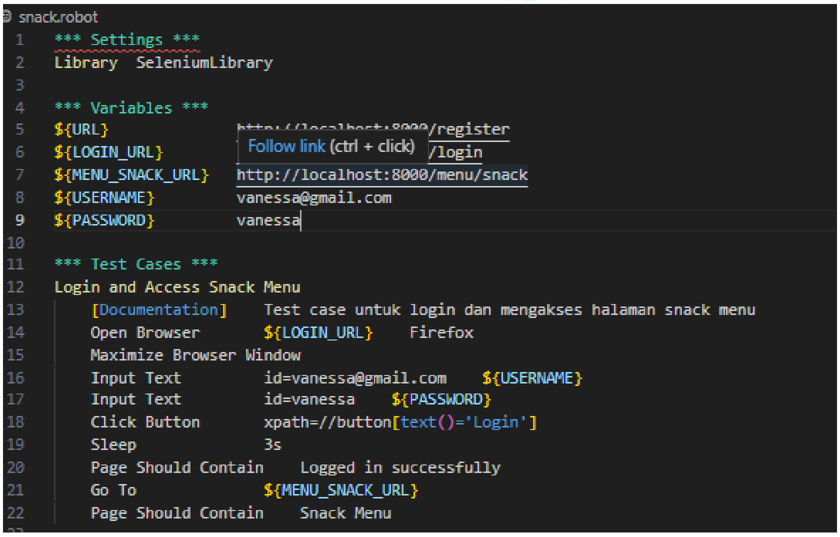
3. Fungsi melihat menu snak

a. Scenario


snack.robot

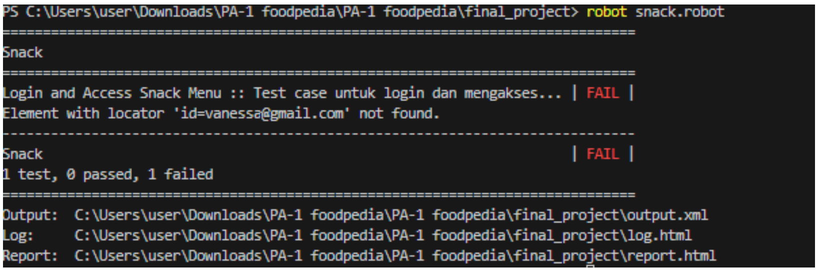
```
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${URL}               http://localhost:8000/register
${LOGIN_URL}         http://localhost:8000/login
${MENU_SNACK_URL}    http://localhost:8000/menu/snack
${USERNAME}          vanessa@gmail.com
${PASSWORD}          vanessa
```

b. Step Definition



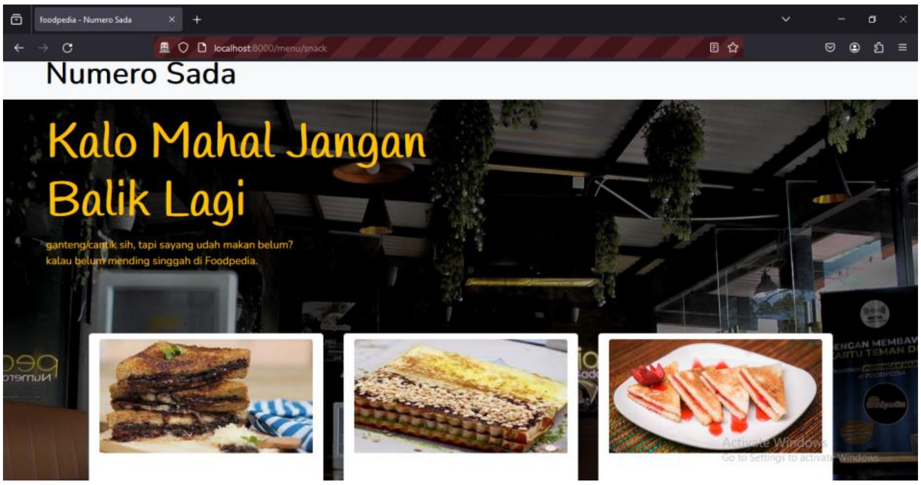
```
*** Settings ***
Library  SeleniumLibrary

*** Variables ***
${URL}                http://localhost:8000/register
${LOGIN_URL}                                   /login
${MENU_SNACK_URL}     http://localhost:8000/menu/snack
${USERNAME}           vanessa@gmail.com
${PASSWORD}           vanessa

*** Test Cases ***
Login and Access Snack Menu
    [Documentation]    Test case untuk login dan mengakses halaman snack menu
    Open Browser       ${LOGIN_URL}    Firefox
    Maximize Browser Window
    Input Text         id=vanessa@gmail.com    ${USERNAME}
    Input Text         id=vanessa    ${PASSWORD}
    Click Button       xpath=//button[text()='Login']
    Sleep              3s
    Page Should Contain    Logged in successfully
    Go To              ${MENU_SNACK_URL}
    Page Should Contain    Snack Menu
```

c. *Console* setelah Running



```
PS C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project> robot snack.robot
==============================================================================
Snack
==============================================================================
Login and Access Snack Menu :: Test case untuk login dan mengakses... | FAIL |
Element with locator 'id=vanessa@gmail.com' not found.
------------------------------------------------------------------------------
Snack                                                                 | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\output.xml
Log:     C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\log.html
Report:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\report.html
```

d. *Snippet* Browser saat Testing

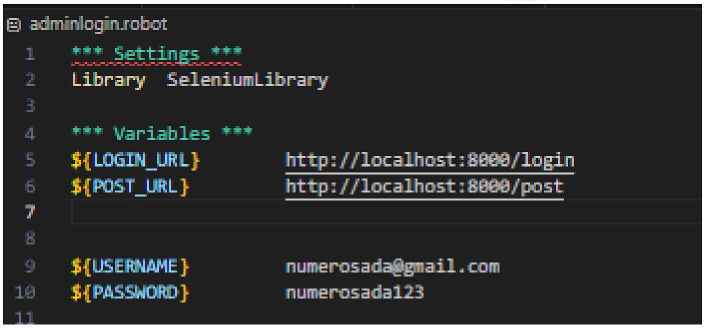


foodpedia - Numero Sada
localhost:8000/menu/snack
Numero Sada
Kalo Mahal Jangan Balik Lagi
ganteng/cantik sih, tapi sayang udah makan belum?
kalau belum mending singgah di Foodpedia.
Activate Windows
Go to Settings to activate Windows.

Penjelasan : Kode ini mengotomatisasi login dan akses ke halaman menu snack menggunakan Robot Framework. Pertama, SeleniumLibrary diimpor untuk mengontrol browser. URL login, URL menu snack, username, dan password didefinisikan sebagai variabel.Test case "Login and Access Snack Menu" membuka browser, mengakses halaman login, mengisi form login dengan username dan password, dan mengklik tombol login. Setelah login berhasil, browser diarahkan ke halaman menu snack dan memastikan halaman tersebut berisi teks "Snack Menu".

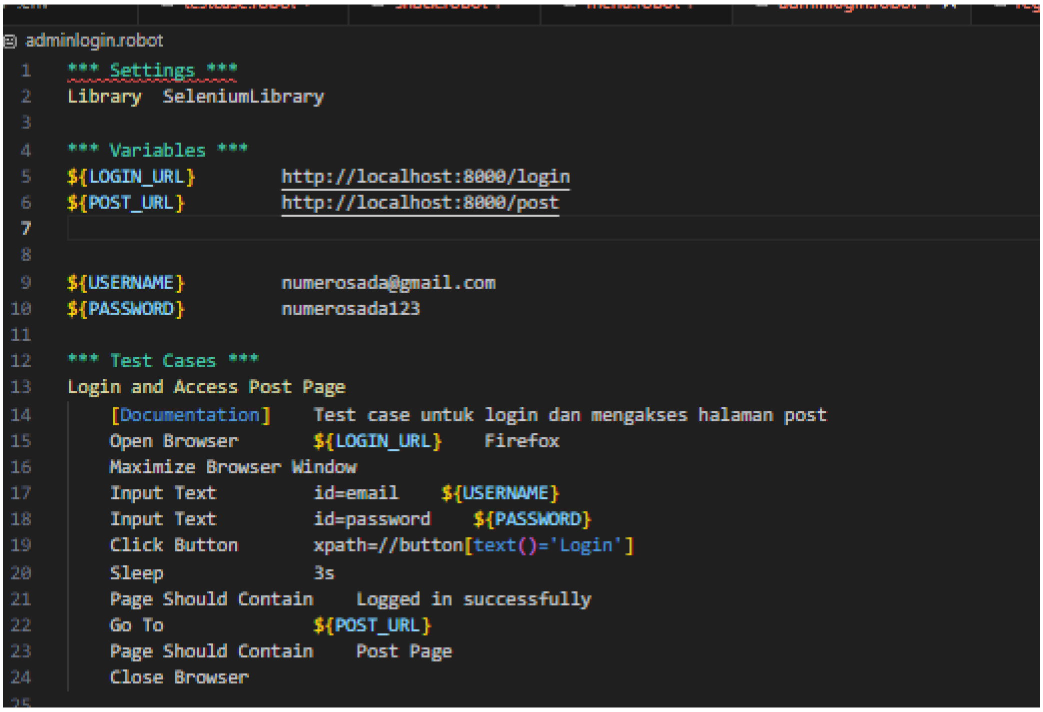
4. Fungsi login akun admin

   a. Scenario



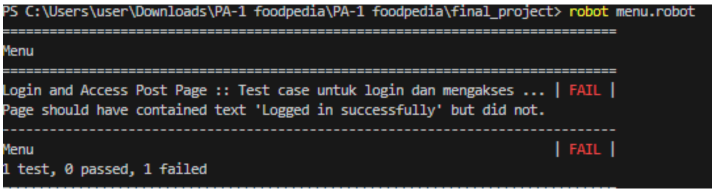
```
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${LOGIN_URL}          http://localhost:8000/login
${POST_URL}           http://localhost:8000/post


${USERNAME}           numerosada@gmail.com
${PASSWORD}           numerosada123
```

b. Step Definition



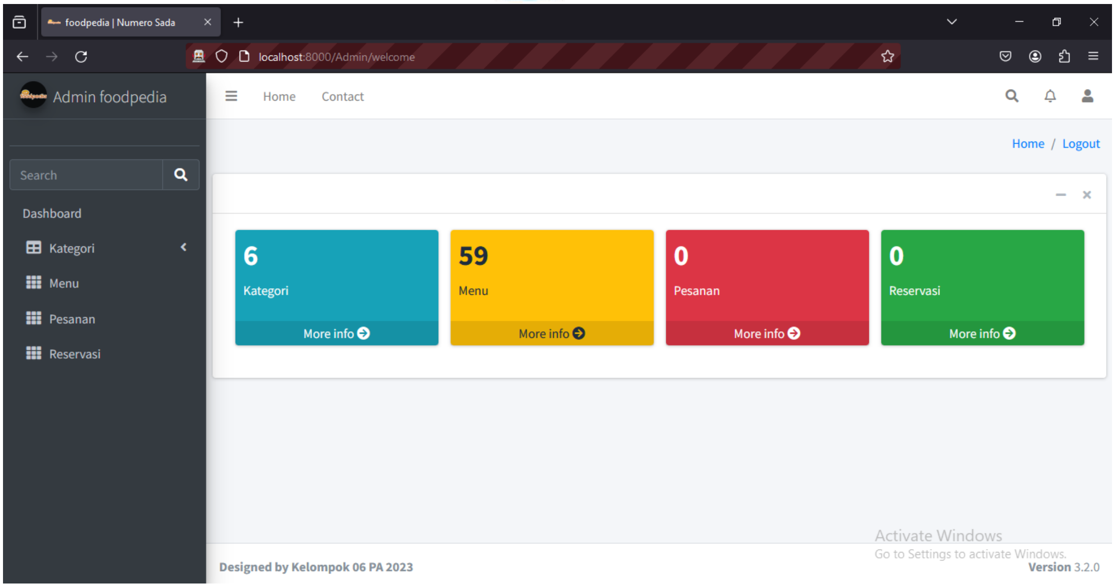
```
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${LOGIN_URL}          http://localhost:8000/login
${POST_URL}           http://localhost:8000/post


${USERNAME}           numerosada@gmail.com
${PASSWORD}           numerosada123

*** Test Cases ***
Login and Access Post Page
    [Documentation]    Test case untuk login dan mengakses halaman post
    Open Browser         ${LOGIN_URL}     Firefox
    Maximize Browser Window
    Input Text           id=email     ${USERNAME}
    Input Text           id=password     ${PASSWORD}
    Click Button         xpath=//button[text()='Login']
    Sleep                3s
    Page Should Contain     Logged in successfully
    Go To                ${POST_URL}
    Page Should Contain     Post Page
    Close Browser
```

c. *Console* setelah Running



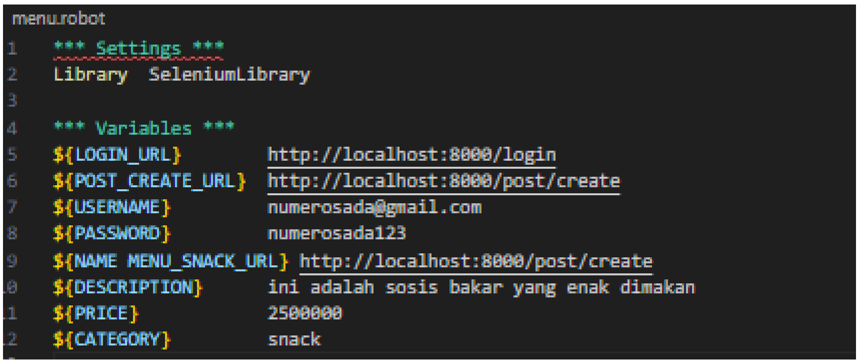
```
PS C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project> robot menu.robot
==============================================================================
Menu
==============================================================================
Login and Access Post Page :: Test case untuk login dan mengakses ... | FAIL |
Page should have contained text 'Logged in successfully' but did not.
------------------------------------------------------------------------------
Menu                                                                  | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
```

d. *Snippet* Browser saat Testing



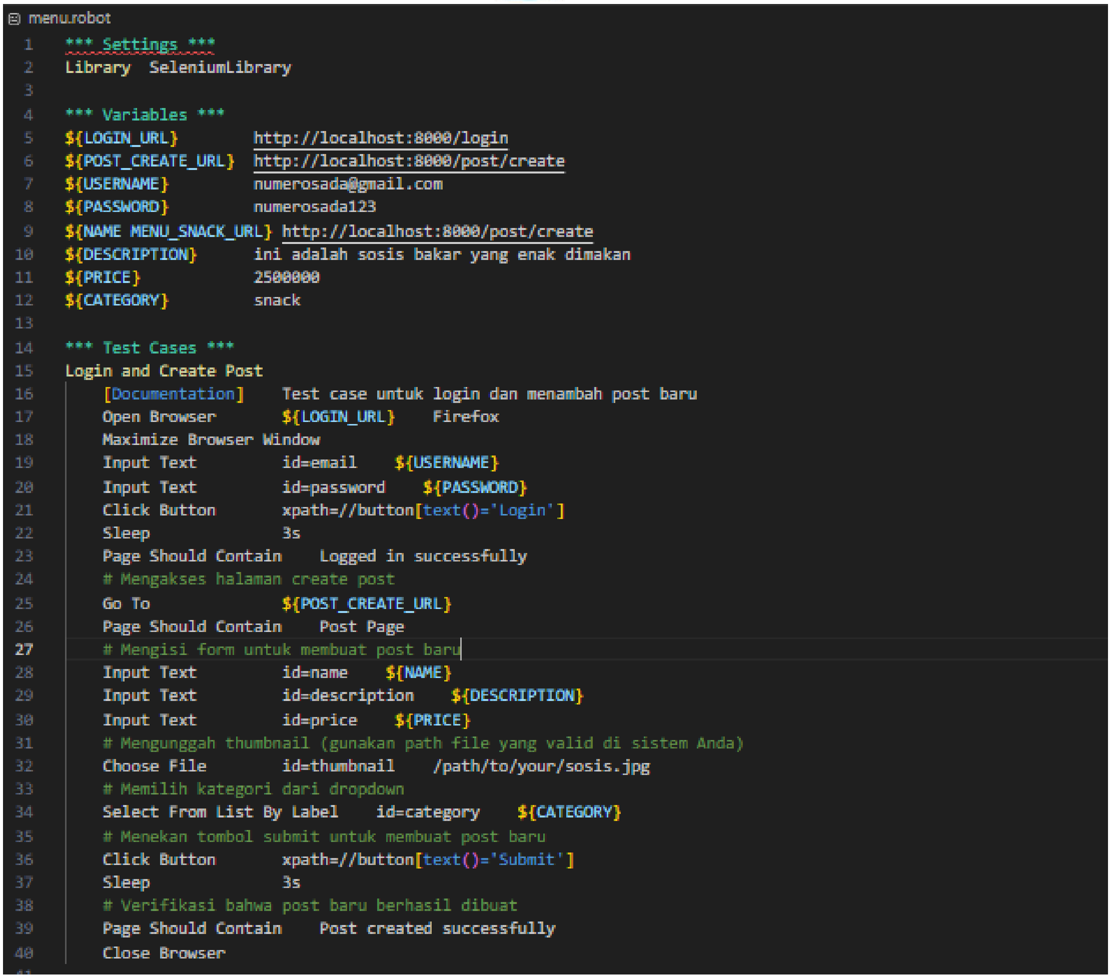
Penjelasan: skrip ini menggunakan SeleniumLibrary untuk mengotomatisasi pengujian login ke sebuah aplikasi web dan mengakses halaman post. Skrip ini mendefinisikan URL login dan URL halaman post, serta kredensial login (username dan password). Pada bagian test case, skrip ini membuka browser Firefox, memaksimalkan jendela browser, memasukkan username dan password pada form login, kemudian menekan tombol login. Setelah itu, skrip memverifikasi bahwa login berhasil dengan mencari teks "Logged in successfully" di halaman. Selanjutnya, skrip mengakses URL halaman post dan memverifikasi keberadaan teks "Post Page" untuk memastikan halaman tersebut terbuka dengan benar. Akhirnya, skrip menutup browser.

5. Fungsi menambah form menu
   a. Scenario



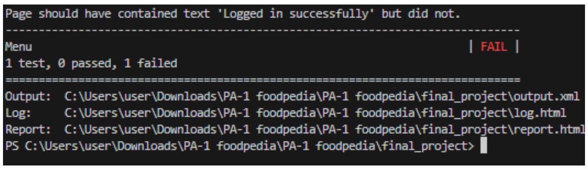
```
menu.robot
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${LOGIN_URL}           http://localhost:8000/login
${POST_CREATE_URL}     http://localhost:8000/post/create
${USERNAME}            numerosada@gmail.com
${PASSWORD}            numerosada123
${NAME MENU_SNACK_URL} http://localhost:8000/post/create
${DESCRIPTION}         ini adalah sosis bakar yang enak dimakan
${PRICE}               2500000
${CATEGORY}            snack
```

   b. Step Definition


menu.robot

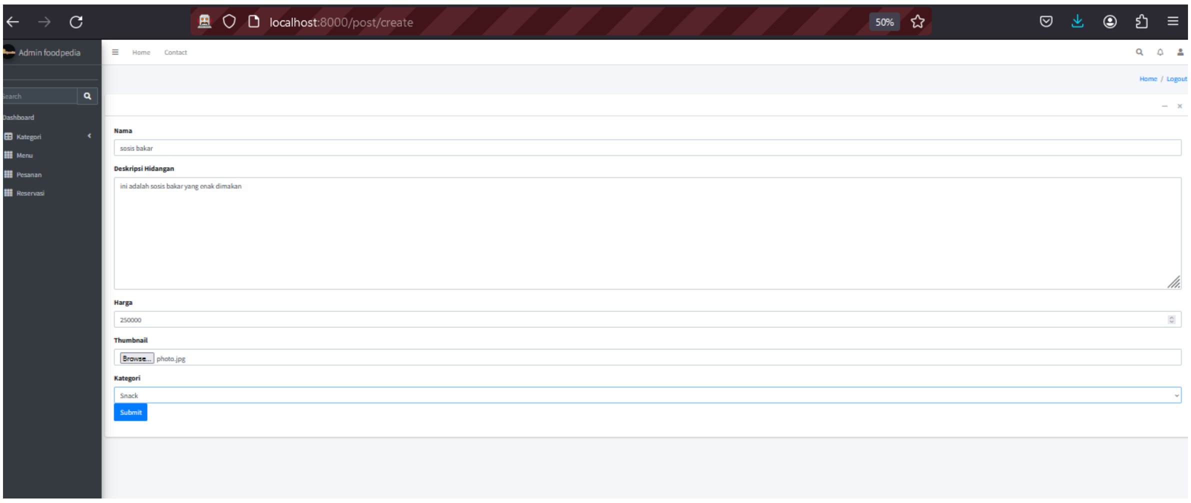
```
*** Settings ***
Library  SeleniumLibrary

*** Variables ***
${LOGIN_URL}        http://localhost:8000/login
${POST_CREATE_URL}  http://localhost:8000/post/create
${USERNAME}         numerosada@gmail.com
${PASSWORD}         numerosada123
${NAME MENU_SNACK_URL} http://localhost:8000/post/create
${DESCRIPTION}      ini adalah sosis bakar yang enak dimakan
${PRICE}            2500000
${CATEGORY}         snack

*** Test Cases ***
Login and Create Post
    [Documentation]    Test case untuk login dan menambah post baru
    Open Browser        ${LOGIN_URL}    Firefox
    Maximize Browser Window
    Input Text          id=email    ${USERNAME}
    Input Text          id=password    ${PASSWORD}
    Click Button        xpath=//button[text()='Login']
    Sleep               3s
    Page Should Contain    Logged in successfully
    # Mengakses halaman create post
    Go To               ${POST_CREATE_URL}
    Page Should Contain    Post Page
    # Mengisi form untuk membuat post baru
    Input Text          id=name    ${NAME}
    Input Text          id=description    ${DESCRIPTION}
    Input Text          id=price    ${PRICE}
    # Mengunggah thumbnail (gunakan path file yang valid di sistem Anda)
    Choose File         id=thumbnail    /path/to/your/sosis.jpg
    # Memilih kategori dari dropdown
    Select From List By Label    id=category    ${CATEGORY}
    # Menekan tombol submit untuk membuat post baru
    Click Button        xpath=//button[text()='Submit']
    Sleep               3s
    # Verifikasi bahwa post baru berhasil dibuat
    Page Should Contain    Post created successfully
    Close Browser
```

c. *Console* setelah Running



```
Page should have contained text 'Logged in successfully' but did not.
------------------------------------------------------------------------------
Menu                                                                  | FAIL |
1 test, 0 passed, 1 failed
==============================================================================
Output:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\output.xml
Log:     C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\log.html
Report:  C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project\report.html
PS C:\Users\user\Downloads\PA-1 foodpedia\PA-1 foodpedia\final_project>
```

d. *Snippet* Browser saat Testing



localhost:8000/post/create

Nama: sosis bakar

Deskripsi Hidangan: ini adalah sosis bakar yang enak dimakan

Harga: 250000

Thumbnail: Browse... photo.jpg

Kategori: Snack