Nama : Nissa 'Uzzulfa
Kelas : QE B

Penjelasan fitur mobile testing pada aplikasi Login dan Register.

Berikut adalah script Login dan Register feature.

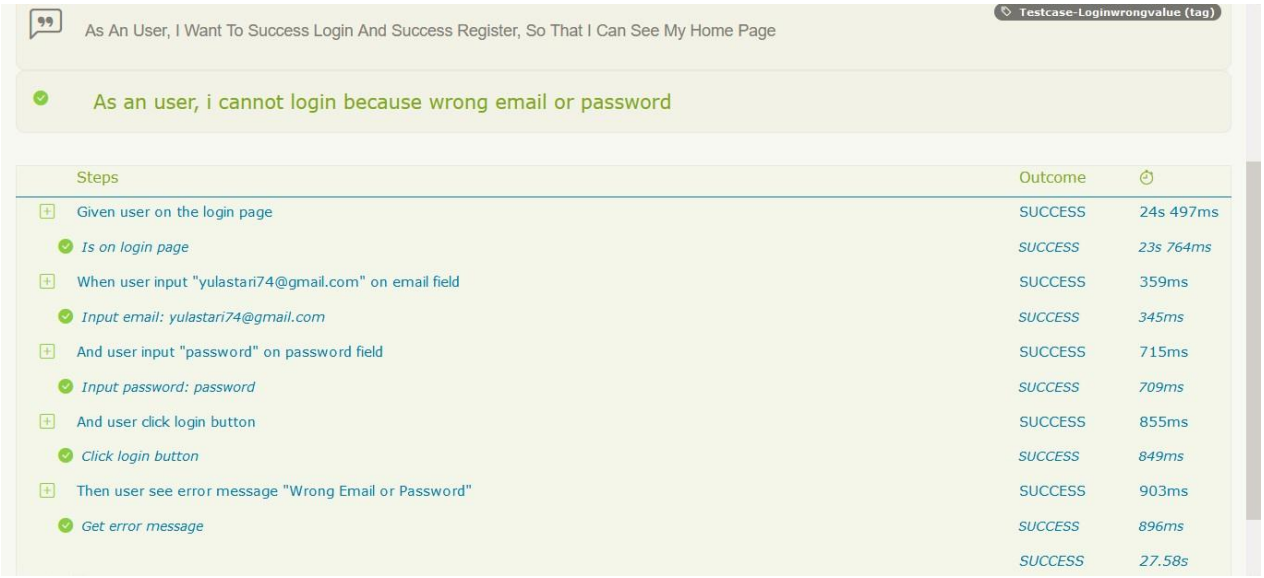
Feature: As an user, I want to success login, So that i can see my home page

#TESTCASE-1

```
@TestCase-LoginWrongValue
Scenario: As an user, i cannot login because wrong email or password
  Given user on the login page
  When user input "yulastari74@gmail.com" on email field
  And user input "password" on password field
  And user click login button
  Then user see error message "Wrong Email or Password"
```

Pada test case ini menjelaskan bahwa user tidak dapat login jika terdapat pengisian email atau password yang salah.

Berikut adalah hasil report pengujian pada test case tersebut :



Berikut adalah LoginSteps yang diimplementasikan pada feature diatas :

```
public class LoginSteps {

    @Steps
    LoginScreen loginScreen;
```

```
@Given("user on the login page")
public void userOnTheLoginPage() {
    Assert.assertTrue(loginScreen.isOnLoginPage());
}
@When("user input {string} on email field")
public void inputEmail(String email) {
    loginScreen.inputEmail(email);
}

@And("user input {string} on password field")
public void inputPassword(String password) {
    loginScreen.inputPassword(password);
}

@And("user click login button")
public void clickLoginButton() {
    loginScreen.clickLoginButton();
}

@Then("user see error message {string}")
public void userSeeErrorMessage(String message) {
    Assert.assertEquals(message, loginScreen.getErrorMessage());
}
```

Kelas ini terdiri dari beberapa metode yang masing-masing merepresentasikan satu langkah dalam skenario uji, seperti berikut:

- `userOnTheLoginPage()`: Metode ini mengecek apakah pengguna berada di halaman login atau tidak.
- `inputEmail(String email)`: Metode ini mengisi kolom email dengan nilai email yang diberikan.
- `inputPassword(String password)`: Metode ini mengisi kolom password dengan nilai password yang diberikan.
- `clickLoginButton()`: Metode ini mengeklik tombol login.
- `userSeeErrorMessage(String message)`: Metode ini memastikan bahwa pesan error yang muncul di layar sama dengan pesan error yang diharapkan.

#TESTCASE-2

```
@TestCase-LoginValidValue
Scenario:  As an user, i can login with a valid email and valid password
  Given user on the login page
  When user input "nissa@gmail.com" on email field
  And user input "Ridwan123" on password field
  And user click login button
  Then user see home page
```

Berikut adalah LoginSteps yang diimplementasikan pada feature diatas :

```
public class LoginSteps {

    @Steps
    LoginScreen loginScreen;

    @Given("user on the login page")
    public void userOnTheLoginPage() {
        Assert.assertTrue(loginScreen.isOnLoginPage());
    }
    @When("user input {string} on email field")
    public void inputEmail(String email) {
        loginScreen.inputEmail(email);
    }

    @And("user input {string} on password field")
```

```
public void inputPassword(String password) {
    loginScreen.inputPassword(password);
}

@And("user click login button")
public void clickLoginButton() {
    loginScreen.clickLoginButton();
}

   @Then("user see home page")
public void userSeeHomePage() {
     Assert.assertTrue(loginScreen.homePage());
}
```

Kelas ini terdiri dari beberapa metode yang masing-masing merepresentasikan satu langkah dalam skenario uji, seperti berikut:

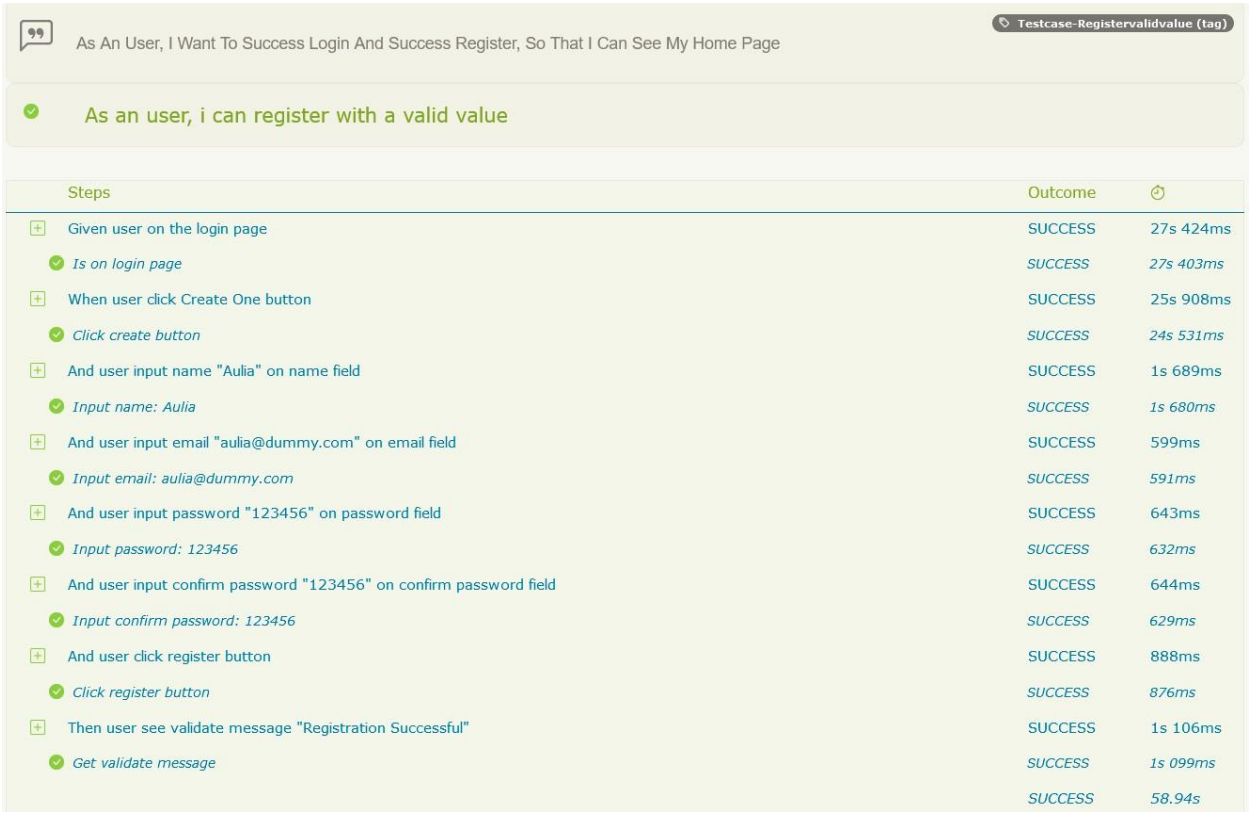
- `userOnTheLoginPage()`: Metode ini mengecek apakah pengguna berada di halaman login atau tidak.
- `inputEmail(String email)`: Metode ini mengisi kolom email dengan nilai email yang diberikan.
- `inputPassword(String password)`: Metode ini mengisi kolom password dengan nilai password yang diberikan.
- `clickLoginButton()`: Metode ini mengeklik tombol login.
- `userSeeHomePage(String message)`: Metode ini memastikan dan memvalidasi jika pengisian email dan password valid maka akan menampilkan ke halaman home

#TESTCASE-3

```
@TestCase-RegisterValidValue
Scenario:  As an user, i can register with a valid value
  Given user on the login page
  When user click Create One button
  And user input name "Nisa" on name field
  And user input email "nisa@gmail.com" on email field
  And user input password "123456" on password field
  And user input confirm password "123456" on confirm password field
  And user click register button
  Then user see validate message "Registration Successful"
```

Pada test case ini menjelaskan bahwa user dapat register akun dengan melengkapi pengisian semua field

Berikut adalah hasil report pengujian pada test case tersebut :



Berikut adalah RegisterSteps yang diimplementasikan pada feature diatas :

```
public class RegisterSteps {

    @Steps
    RegisterScreen registerScreen;
```

```
@When("user click Create One button")
public void ClickCreateButton() {
    registerScreen.clickCreateButton();
}

@And("user input name {string} on name field")
public void InputName(String Name) {
    registerScreen.InputName(Name);
}

@And("user input email {string} on email field")
public void InputEmail(String Email) {
    registerScreen.InputEmail(Email);
}

@And("user input password {string} on password field")
public void InputPassword(String Password) {
    registerScreen.InputPassword(Password);
}

@And("user input confirm password {string} on confirm password field")
public void InputConfirmPassword(String ConfirmPassword) {
    registerScreen.InputConfirmPassword(ConfirmPassword);
}

@And("user click register button")
public void ClickRegisterButton() {
    registerScreen.ClickRegisterButton();
}

@Then("user see validate message {string}")
public void userSeeValidateMessage(String validate) {
    Assert.assertEquals(validate, registerScreen.getValidateMessage());
}
```

Kelas ini terdiri dari beberapa metode yang masing-masing merepresentasikan satu langkah dalam skenario uji, seperti berikut:

- `ClickCreateButton()`: Metode ini mengeklik tombol "Create One" untuk membuka halaman registrasi.
- `InputName(String Name)`: Metode ini mengisi kolom "name" dengan nilai nama yang diberikan.
- `InputEmail(String Email)`: Metode ini mengisi kolom "email" dengan nilai email yang diberikan.
- `InputPassword(String Password)`: Metode ini mengisi kolom "password" dengan nilai password yang diberikan.
- `InputConfirmPassword(String ConfirmPassword)`: Metode ini mengisi kolom "confirm password" dengan nilai password konfirmasi yang diberikan.
- `ClickRegisterButton()`: Metode ini mengeklik tombol "register" untuk menyelesaikan proses registrasi.

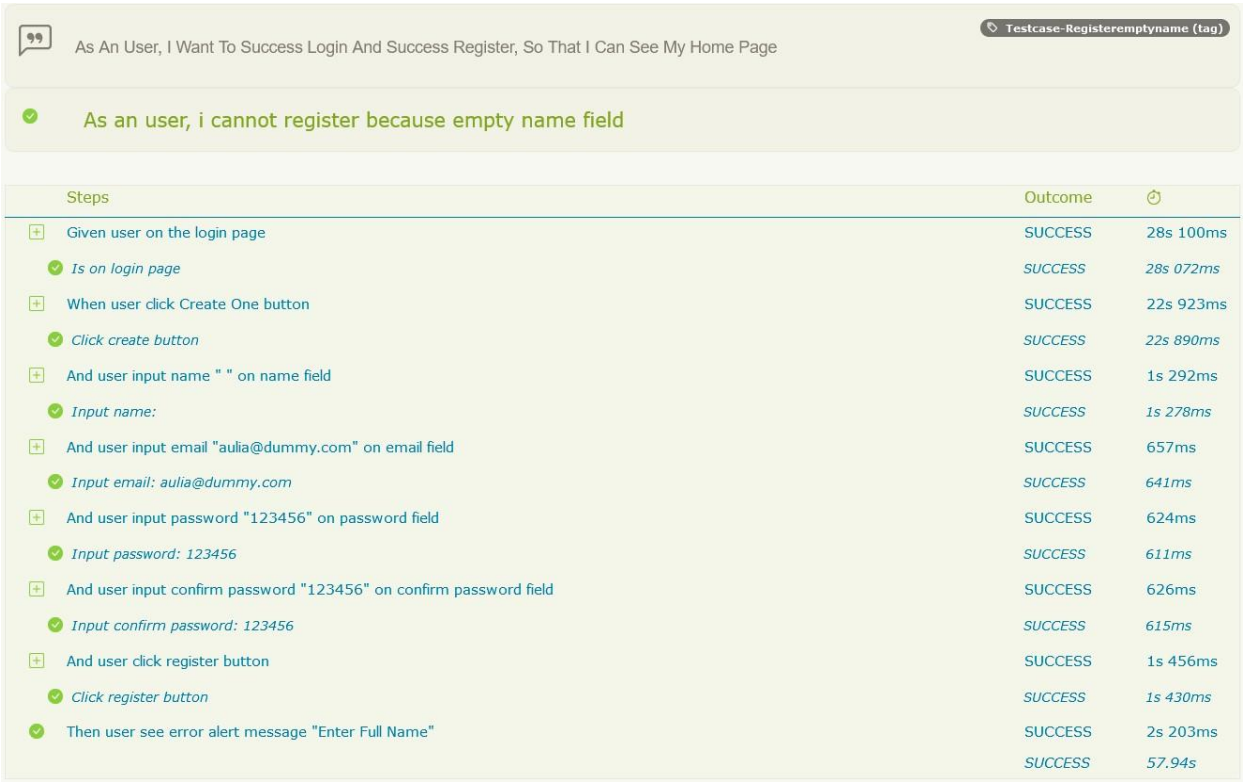
- `userSeeValidateMessage(String validate)`: Metode ini memastikan bahwa pesan validasi yang muncul di layar sama dengan pesan validasi yang diharapkan.

#TESTCASE-4

```
@TestCase-RegisterEmptyName
 Scenario:  As an user, i cannot register because empty name field
   Given user on the login page
   When user click Create One button
   And user input name " " on name field
   And user input email "nisa@dummy.com" on email field
   And user input password "123456" on password field
   And user input confirm password "123456" on confirm password field
   And user click register button
   Then user see error alert message "Enter Full Name"
```

Pada test case ini menjelaskan bahwa user dapat register akun dengan mengosongkan pengisian field Name

Berikut adalah hasil report pengujian pada test case tersebut :



Berikut adalah RegisterSteps yang diimplementasikan pada feature alert message “Enter Valid Full Name” diatas :

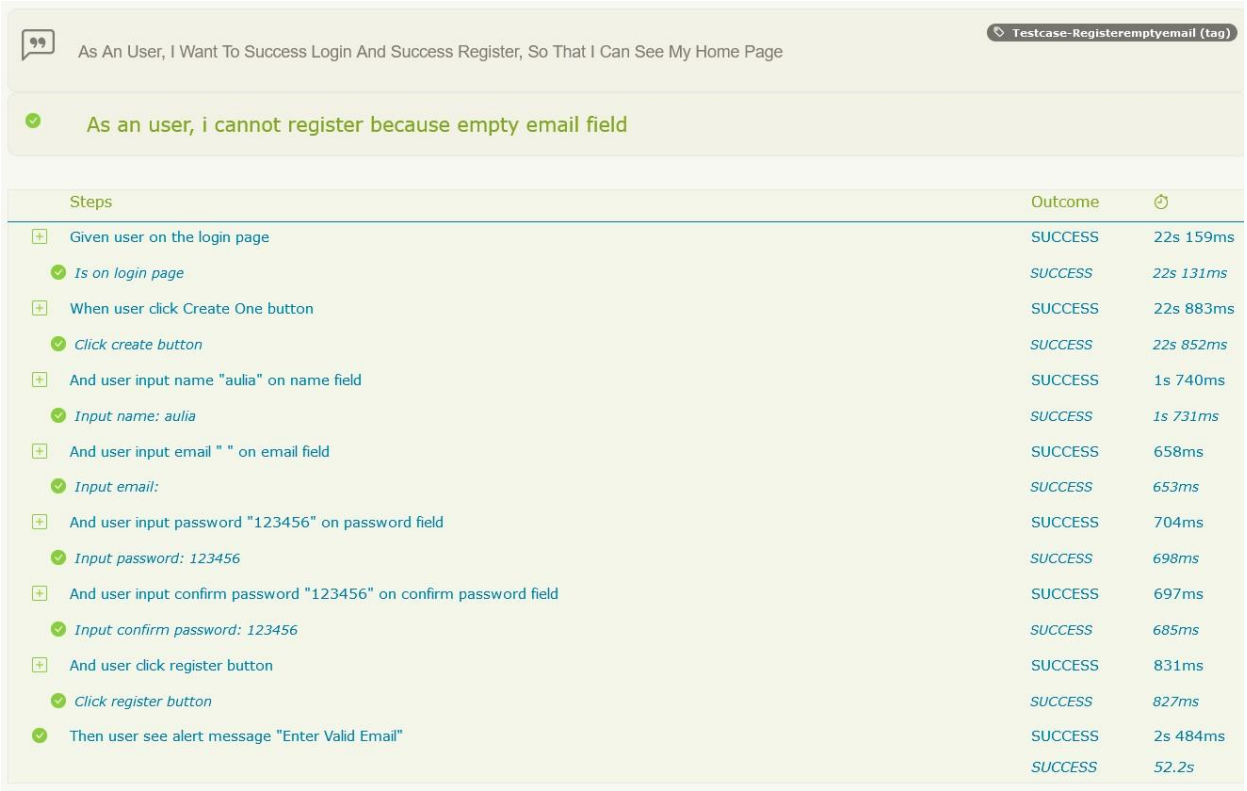
```
@Then("user see error alert message {string}")
    public void userSeeErrorAlertMessage(String alert) {
        Assert.assertEquals(alert, registerScreen.getAlertMessage());
    }
```

#TESTCASE-5

```
@TestCase-RegisterEmptyEmail
Scenario:  As an user, i cannot register because empty email field
  Given user on the login page
  When user click Create One button
  And user input name "nisa" on name field
  And user input email " " on email field
  And user input password "123456" on password field
  And user input confirm password "123456" on confirm password field
  And user click register button
  Then user see alert message "Enter Valid Email"
```

Pada test case ini menjelaskan bahwa user dapat register akun dengan mengosongkan pengisian field Email

Berikut adalah hasil report pengujian pada test case tersebut :



Berikut adalah RegisterSteps yang diimplementasikan pada feature alert message “Enter Valid Email” diatas :

```
@Then("user see alert message {string}")
    public void userSeeAlertMessage(String alertMessage) {
        Assert.assertEquals(alertMessage, registerScreen.getAlert());
    }
```