Pengujian BDD Dengan Selenium Webdriver

Mata Kuliah : Pembangunan Perangkat Lunak (I1)
Kelompok : 3
Anggota Kelompok :

- Abhinaya Zaim (187231022)
- Balya Ananta Wicaksana Annahid (187231033)
- Raihan Arif Mustofa (187231084)
- Amir Gymnastiar (187231093)
- Ahmad Alvan Nurdin (187231043)
- Nala I'lma Kamila (187231003)

1. Sign Up

User Story

As a new user, I want to sign up by providing my username, email, and password, so that I can create an account.

Scenario 1 – Successful registration

GIVEN I am on the registration page
WHEN I fill in "username" with "newteacher"
AND I fill in "email" with "teacher@example.com"
AND I fill in "password" with "SecurePass123"
AND I press "Sign Up"
THEN I should be redirected to the login page
AND I should see "Registration successful" message

Scenario 2 – Registration fails (email already registered)

GIVEN I am on the registration page
AND there is an existing user with email "teacher@example.com"
WHEN I fill in "username" with "anotheruser"
AND I fill in "email" with "teacher@example.com"
AND I fill in "password" with "AnotherPass123"
AND I press "Sign Up"
THEN I should see "Email already registered" error

Implementasi:

```
class TestSignUpForm(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.base_url = "https://demoqa.com/automation-practice-form"

    def test_signup_form(self):
        driver = self.driver
        driver.get(self.base_url)

        wait = WebDriverWait(driver, 10)

        #  🔹 Isi form simulasi "Sign Up"
        driver.find_element(By.ID, "firstName").send_keys("Abhinaya")
        driver.find_element(By.ID, "lastName").send_keys("Zaim")
        driver.find_element(By.ID, "userEmail").send_keys("teacher@example.com")

        # Klik gender radio button
        gender_label = driver.find_element(By.XPATH, "//label[@for='gender-radio-1']")
        driver.execute_script("arguments[0].click();", gender_label)

        driver.find_element(By.ID, "userNumber").send_keys("08123456789")

        # Scroll ke bawah agar tombol terlihat
        driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")

        # Klik tombol submit
        submit_btn = driver.find_element(By.ID, "submit")
        driver.execute_script("arguments[0].click();", submit_btn)
```

Output:

```
Testing started at 22:59 ...
Launching unittests with arguments python -m unittest D:\pythonProject1\Sign up ppl.py in D:\pythonProject1


Process finished with exit code 0
DEBUG: Hasil Sign Up -> Thanks for submitting the form


Ran 1 test in 13.405s

OK
```

2. Login

User Story

As a registered teacher, I want to log in with my email and password so that I can access the dashboard.

Scenario 1 – Login with valid credentials

GIVEN I am on the login page
AND a user exists with email "x@gmail.com" and password "123"
WHEN I fill in "email" with "x@gmail.com"
AND I fill in "password" with "123"
AND I press "Login"
THEN I should be redirected to the main dashboard
AND I should see "Welcome" message

Scenario 2 – Login with invalid credentials

GIVEN I am on the login page
WHEN I fill in "email" with "teacher@example.com"
AND I fill in "password" with "WrongPassword"
AND I press "Login"
THEN I should see "Invalid credentials" error
AND I should remain on the login page

Implementasi:

```python
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

class TestDemoWebsite(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = "https://practicetestautomation.com/practice-test-login/"
        self.driver.maximize_window()

    def test_login_valid(self):
        driver = self.driver
        driver.get(self.base_url)
        driver.find_element(By.ID, "username").send_keys("student")
        driver.find_element(By.ID, "password").send_keys("Password123")
        driver.find_element(By.ID, "submit").click()
        success_message = driver.find_element(By.TAG_NAME, "h1").text
        self.assertIn("Logged In Successfully", success_message)

    def test_login_invalid(self):
        driver = self.driver
        driver.get(self.base_url)
        driver.find_element(By.ID, "username").send_keys("wronguser")
        driver.find_element(By.ID, "password").send_keys("wrongpass")
        driver.find_element(By.ID, "submit").click()
```

Output:

```
Testing started at 22:45 ...
Launching unittests with arguments python -m unittest D:\pythonProject1\ppl prak.py in D:\pythonProject1


Process finished with exit code 0
DEBUG: Pesan error yang terbaca -> Your username is invalid!


Ran 2 tests in 13.027s

OK
```

3. Upload Question File

User Story

As a teacher, I want to upload a valid file so that the system can classify my questions.

Scenario 1 – Upload valid file

GIVEN I am logged in as "teacher@example.com"
AND I am on the "Upload Questions" page
WHEN I attach the file "questions.csv" to "file_upload"
AND I press "Upload"
THEN the system should accept the file
AND I should see "File validated successfully" message

Scenario 2 – Unsupported file format

GIVEN I am logged in as "teacher@example.com"
AND I am on the "Upload Questions" page
WHEN I attach the file "questions.exe" to "file_upload"
AND I press "Upload"
THEN I should see "Unsupported file format" error
Implementasi:

```python
class TestUploadFile(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.base_url = "https://the-internet.herokuapp.com/upload"
        self.driver.maximize_window()

    def test_upload_valid_file(self):
        driver = self.driver
        driver.get(self.base_url)

        # 🔹 Path file dummy
        file_path = os.path.abspath("questions.csv")

        # 🔹 Upload file
        upload_input = driver.find_element(By.ID, "file-upload")
        upload_input.send_keys(file_path)
        driver.find_element(By.ID, "file-submit").click()

        wait = WebDriverWait(driver, 10)
        success_text = wait.until(
            EC.text_to_be_present_in_element((By.TAG_NAME, "h3"), "File Uploaded!")
        )
```

Output:

```
Testing started at 22:53 ...
Launching unittests with arguments python -m unittest D:\pythonProject1\upload file ppl.py in D:\pythonProject1


Process finished with exit code 0
DEBUG: Hasil upload -> File Uploaded!
DEBUG: Isi halaman tanpa file -> Internal Server Error


Ran 2 tests in 40.043s

OK
```

4. Regenerate Question

User Story

As a teacher, I want to regenerate a question to a new cognitive level so that I can adjust question difficulty.

Scenario 1 – Successfully regenerate question

GIVEN I have classification results displayed
AND one question is labeled as "C3"
WHEN I choose that question and select new level "C5"
AND I press "Regenerate"
THEN I should see a regenerated version of the question
AND I should see both original and new versions

Scenario 2 – Regeneration fails

GIVEN I have classification results displayed
WHEN I choose a question and select new level "C6"
AND the system is unable to generate a new phrasing
THEN I should see "Regeneration failed, try again" message
Implementasi:

```python
class TestRegenerateQuestion(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.base_url = "https://the-internet.herokuapp.com/inputs"

    def test_regenerate_question_success(self):
        driver = self.driver
        wait = WebDriverWait(driver, 10)
        driver.get(self.base_url)

        # 🔹 Simulasi mengisi pertanyaan
        input_field = driver.find_element(By.TAG_NAME, "input")
        input_field.send_keys("Explain Artificial Intelligence")
        time.sleep(1)

        # 🔹 Simulasi memilih level kognitif baru (disimulasikan dengan menambah teks)
        input_field.send_keys(Keys.TAB)
        driver.execute_script("alert('New question generated successfully!');")
        time.sleep(1)

        # 🔹 Tunggu dan tangkap alert hasil regenerasi
        alert = wait.until(EC.alert_is_present())
        alert_text = alert.text
        print("DEBUG: Pesan regenerasi ->", alert_text)
```

Output:

```
Testing started at 23:12 ...
Launching unittests with arguments python -m unittest D:\pythonProject1\regenerate soal ppl.py in D:\pythonProject1


Ran 2 tests in 30.157s

OK

Process finished with exit code 0
DEBUG: Pesan regenerasi gagal -> Regeneration failed: empty input!
DEBUG: Pesan regenerasi -> New question generated successfully!
```

5. Download Report

User Story

As a teacher, I want to download the classification results in PDF so that I can save them for records.

Scenario 1 – Successful download

GIVEN classification results are available
WHEN I press "Download Report"
THEN the system should generate a PDF
AND the PDF should include summary table and charts

Scenario 2 – Report generation fails

GIVEN classification results are available
WHEN I press "Download Report"
AND an internal error occurs
THEN I should see "Error generating report" message
AND I should be able to retry

Implementasi

```python
class TestDownloadReport(unittest.TestCase):
    def setUp(self):
        download_dir = os.path.abspath("downloads")

        if not os.path.exists(download_dir):
            os.makedirs(download_dir)

        options = webdriver.ChromeOptions()
        prefs = {
            "download.default_directory": download_dir,
            "download.prompt_for_download": False,
            "safebrowsing.enabled": True
        }
        options.add_experimental_option("prefs", prefs)

        self.driver = webdriver.Chrome(options=options)
        self.driver.maximize_window()
        self.base_url = "https://the-internet.herokuapp.com/download"
        self.download_dir = download_dir

    def test_download_report(self):
        driver = self.driver
        driver.get(self.base_url)

        # 🔹 Klik file pertama dari daftar
        file_link = driver.find_element(By.CSS_SELECTOR, "div.example a")
        file_name = file_link.text
        file_link.click()
```

Output:

```
Testing started at 23:03 ...
Launching unittests with arguments python -m unittest D:\pythonProject1\download report ppl.py in D:\pythonProject1

OK

Process finished with exit code 0
DEBUG: File 'temp_upload.txt' diklik untuk diunduh...
DEBUG: Isi folder download -> ['temp_upload.txt']


Ran 1 test in 19.862s
```

6.Logout

User Story

As a teacher, I want to log out from my account so that I can securely end my session.

Scenario 1 – Manual logout

GIVEN I am logged in
WHEN I press the "Logout" button
THEN my session should terminate
AND I should be redirected to the login page

Scenario 2 – Session timeout

GIVEN I have been inactive for a defined timeout
WHEN I attempt to perform an action
THEN I should be redirected to the login page
AND see "Session expired, please log in again" message

Implementasi:

```python
class TestLogout(unittest.TestCase):
    def setUp(self):
        self.driver = webdriver.Chrome()
        self.driver.maximize_window()
        self.base_url = "https://practicetestautomation.com/practice-test-login/"
        self.logged_in_url = "https://practicetestautomation.com/logged-in-successfully/"

    def test_logout_success(self):
        driver = self.driver
        wait = WebDriverWait(driver, 10)
        driver.get(self.base_url)

        # 🔹 Login dulu
        driver.find_element(By.ID, "username").send_keys("student")
        driver.find_element(By.ID, "password").send_keys("Password123")
        driver.find_element(By.ID, "submit").click()

        # 🔹 Pastikan login sukses
        wait.until(EC.text_to_be_present_in_element((By.TAG_NAME, "h1"), "Logged In Successfully"))
        print("DEBUG: Login sukses, siap logout")

        # 🔹 Klik tombol logout
        logout_button = driver.find_element(By.LINK_TEXT, "Log out")
        logout_button.click()
```

Output:

```
Testing started at 23:08 ...
Launching unittests with arguments python -m unittest D:\pythonProject1\logout ppl.py in D:\pythonProject1


Process finished with exit code 0
DEBUG: Isi halaman tanpa login -> HOME PRACTICE COURSES BLOG CONTACT
Logged In Successfully
Congratulations student. You successfully logged in!
Log out
© Copyright 2020 Practice Test Automation. All rights reserved | Privacy Policy
DEBUG: Login sukses, siap logout
DEBUG: Setelah logout, teks halaman -> Test login


Ran 2 tests in 21.162s

OK
```