

Nama : Juwita Stefany Hutapea

NPM : 1214026

Mata Kuliah : Software Quality

## LAPORAN DOKUMENTASI

### A. Aplikasi Yang Digunakan

HealHeroo <https://healhero.my.id/>

### B. Bagian Yang Dilakukan Untuk Testing

Halaman pada pengguna di bagian Cek Kesehatan pada kategori Cek Nutrisi

### C. Pengetesan Menggunakan Selenium Web Driver

1. Membuat file baru dengan nama tes\_cek\_kesehatan.py
2. Melakukan pip install selenium

```
import time
import unittest
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys

class SystemTest(unittest.TestCase):
    def setUp(self):
        # Inisialisasi WebDriver
        self.driver = webdriver.Chrome()

    def tearDown(self):
        # Tambahkan penundaan waktu sebelum menutup WebDriver
        time.sleep(5)

        # Menutup WebDriver
        self.driver.quit()

    def login(self, username, password):
        # Membuka halaman login
        self.driver.get("https://healhero.my.id/signin.html")

        # Mencari elemen input username dan password menggunakan XPath
        email_input = self.driver.find_element(By.XPATH, "//input[@id='email']")
        password_input = self.driver.find_element(By.XPATH,
            "//input[@id='password']")

        # Memasukkan nama pengguna dan kata sandi
        email_input.send_keys(username)
```

```
password_input.send_keys(password)

time.sleep(2)

# Klik tombol Login
button = self.driver.find_element(By.XPATH, "//button[@type='submit']")
button.click()

time.sleep(2)

# Cari dan klik tombol OK pada popup
swal = self.driver.find_element(By.CLASS_NAME, "swal2-confirm")
swal.click()

def cek_home(self):
    # Membuka halaman cek kesehatan
    self.driver.get("https://healhero.my.id/pengguna/index.html")

    time.sleep(5)

    time.sleep(2)

def cek_kesehatan(self):
    # Membuka halaman cek kesehatan
    self.driver.get("https://healhero.my.id/pengguna/cekkesehatan.html")

    time.sleep(2)

    # Lakukan pengujian cek kesehatan
    # Lakukan implementasi pengujian cek kesehatan di sini

def cek_diabetes(self):
    # Membuka halaman cek pola tidur
    self.driver.get("https://healhero.my.id/pengguna/cekdiabetes.html")

    time.sleep(2)

    # Mencari elemen input menggunakan ID
    glucoseLevel_input = self.driver.find_element(By.ID, "glucoseLevel")
    age_input = self.driver.find_element(By.ID, "age")
    gender_input = self.driver.find_element(By.ID, "gender")
    # carbs_input = self.driver.find_element(By.ID, "carbs")
    # fat_input = self.driver.find_element(By.ID, "fat")

    # Memasukkan nilai ke dalam input
```

```

glucoseLevel_input.send_keys("70mg")
age_input.send_keys("30")
gender_input.send_keys(Keys.ARROW_DOWN)
# carbs_input.send_keys(carbs)
# fat_input.send_keys(fat)

# Tunggu hingga tombol submit muncul
time.sleep(2)

# Klik tombol Submit
button = self.driver.find_element(By.XPATH, '//button[@type="button"]')
button.click()

def test_system_flow(self):
    # Jalankan pengujian login
    self.login("putril@gmail.com", "putricantik")

    # Jalankan pengujian cek home
    self.cek_home()

    # Jalankan pengujian cek kesehatan
    self.cek_kesehatan()

    # Jalankan pengujian cek pola tidur
    self.cek_diabetes()

if __name__ == "__main__":
    unittest.main()

```

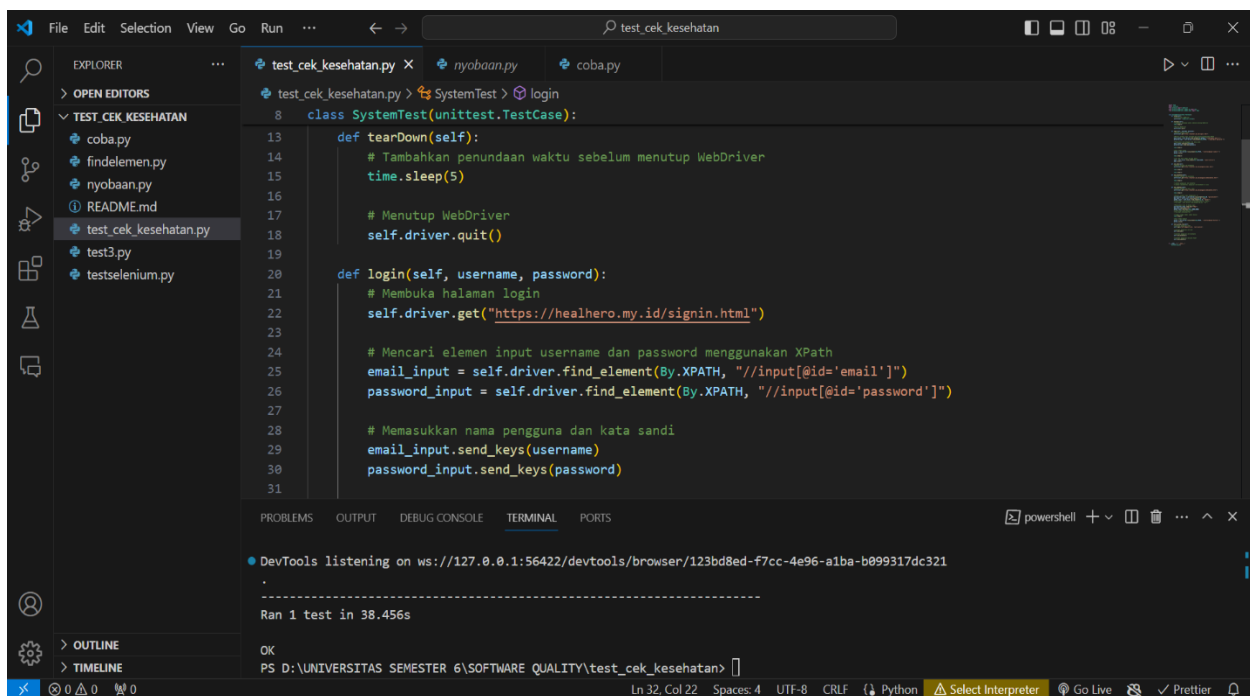
**Berikut penjelasan setiap bagian kode diatas :**

1. Impor Modul
  - time: Digunakan untuk menambahkan penundaan waktu dalam script.
  - unittest: Framework untuk menulis dan menjalankan tes otomatisasi.
  - selenium.webdriver: Modul untuk mengendalikan browser melalui Selenium.
  - selenium.webdriver.common.by: Modul untuk menentukan cara mencari elemen web.
  - selenium.webdriver.common.keys: Modul untuk menyimulasikan penggunaan keyboard.
2. Kelas SystemTest  
 kelas pengujian yang menguji fungsionalitas situs web untuk memeriksa diabetes.
3. Metode setUp()  
 digunakan untuk inisialisasi sebelum setiap pengujian dimulai, dalam hal ini, inisialisasi WebDriver Chrome.

4. Metode `tearDown()`  
digunakan untuk membersihkan setelah setiap pengujian selesai, termasuk menutup `WebDriver` dan menambahkan penundaan waktu.
5. Metode `cek_diabetes()`  
melakukan pengujian fungsi pengujian diabetes di situs web. Ini mengisi formulir dengan data yang diperlukan seperti tingkat glukosa, usia, dan jenis kelamin, kemudian mengklik tombol submit. Metode ini mencari elemen input berdasarkan ID, memasukkan nilai ke dalam input.
6. Metode `test_cek_kesehatan()`  
pengujian utama yang membuka halaman web untuk memeriksa diabetes dan memanggil `cek_diabetes()` untuk menguji fungsionalitas tersebut.
7. Eksekusi Test  
Bagian terakhir script memeriksa apakah script dijalankan sebagai program utama. Jika ya, `unittest.main()` dipanggil untuk menjalankan semua tes dalam kelas `SystemTest`.

## Hasil atau Output dari Testing

### DOKUMENTASI TESTING SELENIUM



The screenshot shows a Visual Studio Code editor with a Python file named `test_cek_kesehatan.py` open. The code defines a `SystemTest` class with two methods: `tearDown` and `login`. The `tearDown` method sleeps for 5 seconds and then quits the WebDriver. The `login` method opens the login page, finds the email and password input fields using XPath, and enters the provided username and password. The terminal at the bottom shows the output of running the test, indicating that the test passed in 38.456 seconds.

```
test_cek_kesehatan.py X nyobaan.py coba.py
test_cek_kesehatan.py > SystemTest > login
8 class SystemTest(unittest.TestCase):
13 def tearDown(self):
14     # Tambahkan penundaan waktu sebelum menutup WebDriver
15     time.sleep(5)
16
17     # Menutup WebDriver
18     self.driver.quit()
19
20 def login(self, username, password):
21     # Membuka halaman login
22     self.driver.get("https://healhero.my.id/signin.html")
23
24     # Mencari elemen input username dan password menggunakan XPath
25     email_input = self.driver.find_element(By.XPATH, "//*[@id='email']")
26     password_input = self.driver.find_element(By.XPATH, "//*[@id='password']")
27
28     # Memasukkan nama pengguna dan kata sandi
29     email_input.send_keys(username)
30     password_input.send_keys(password)
31
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

DevTools listening on ws://127.0.0.1:56422/devtools/browser/123bd8ed-f7cc-4e96-a1ba-b099317dc321

.....

Ran 1 test in 38.456s

OK

PS D:\UNIVERSITAS SEMESTER 6\SOFTWARE QUALITY\test\_cek\_kesehatan>

Ln 32, Col 22 Spaces: 4 UTF-8 CRLF Python Select Interpreter Go Live Prettier