

Pengamatan Testing Sistem Informasi Pada Website DemoBlaze

Yehezkiel Arly, Alissa Salim, Rusdiansyah*

Program Studi Sistem Informasi, Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.

*) Surel korespondensi : rusdiansyah@kwikkiangie.ac.id

Abstract. *This project aims to test a dummy e-commerce website to ensure it meets the requirements and functions properly as expected. The testing is conducted using the Python programming language with IntelliJ IDEA as the Integrated Development Environment, and utilizing a tool called Selenium to test the website with Chrome WebDriver as the connector to execute commands on the Google Chrome browser. The testing process includes simulating various user activities such as registration, login, adding items to the cart on the website, and online payment transactions. Each testing step is designed to identify and fix potential issues, ensuring that all main website functions run smoothly and meet user expectations. Furthermore, this testing aims to evaluate students' ability to write effective testing scripts. By conducting this testing, students are expected to gain valuable practical experience in software testing and enhance their understanding of the importance of testing in the software development cycle. The results of this testing are expected to enhance students' understanding of the importance of testing in the software development cycle and ensure that the e-commerce site can provide an optimal user experience. Overall, this project emphasizes the importance of comprehensive and structured testing to seek high-quality software.*

Keywords: Python, Selenium web driver, IntelliJ IDEA



This work is licensed under Attribution-NonCommercial-ShareAlike 4.0 International.
To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Diterbitkan oleh LPPM Institut Bisnis dan Informatika Kwik Kian Gie, Jl. Yos Sudarso Kav 87, Sunter Jakarta 14350, Indonesia.
DOI : <https://doi.org/10.46806/jib.v13i1.1173>

1. Pendahuluan

Dalam proses pembuatan perangkat lunak, tahap pengujian perangkat lunak dilakukan terakhir setelah implementasi atau peng-kode-an. Pengujian perangkat lunak, atau software testing, biasanya dilakukan oleh pekerja Software QA. Proses ini bertujuan untuk menemukan kesalahan awal sebelum program diberikan kepada pengguna. Saat ini, setiap kali perangkat lunak selesai dikembangkan, pengujian selalu dilakukan dengan berbagai teknik yang berbeda (Ginting et al., 2023).

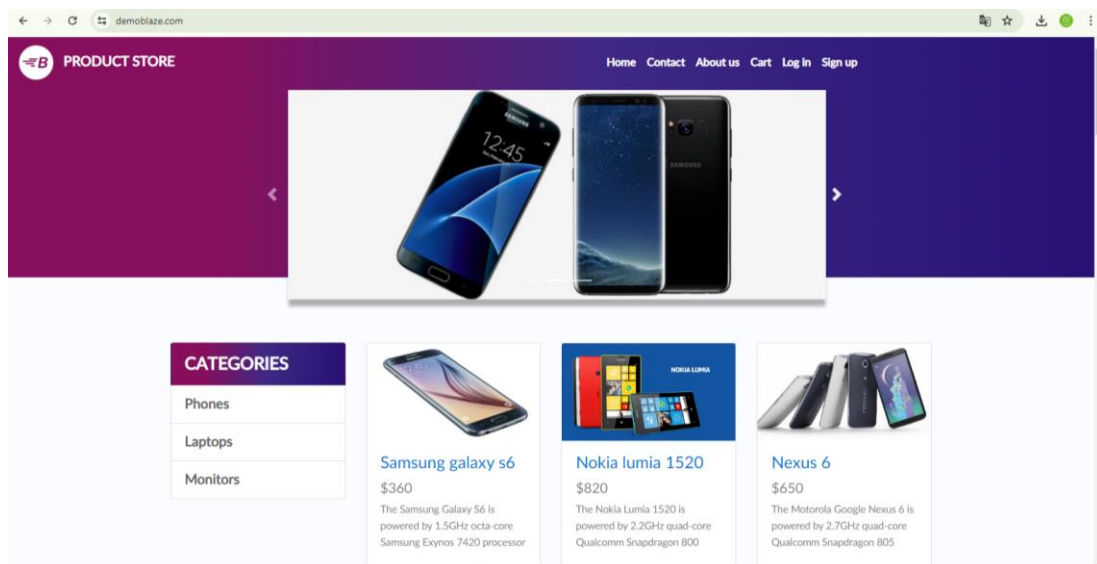
Selenium IDE adalah salah satu software yang dapat digunakan untuk melakukan pengujian otomatis. Selenium IDE merupakan plugin pada web browser yang berfungsi sebagai alat bantu dalam proses pengujian otomatis (Hawari, Kusumo, & Firdaus, 2015). Selenium IDE melakukan pengujian otomatis dengan memeriksa bagian fungsional dari sistem atau perangkat lunak yang sedang dikembangkan (Ginting et al., 2023).

Pengujian ini dirancang untuk mensimulasikan skenario penggunaan nyata yang melibatkan beberapa tahapan kritis dalam interaksi pengguna dengan situs web.

Proses dimulai dengan tindakan registrasi, di mana pengguna baru mendaftar dan membuat akun. Langkah ini penting untuk memastikan bahwa sistem pendaftaran berfungsi dengan benar dan data pengguna ditangani dengan aman. Setelah registrasi, tahap berikutnya adalah login, yang menguji kemampuan sistem dalam mengelola autentikasi pengguna dan memastikan bahwa hanya pengguna yang sah yang dapat mengakses akun mereka.

Setelah berhasil login, pengujian dilanjutkan dengan mensimulasikan kegiatan belanja, di mana pengguna menambahkan produk ke dalam keranjang belanja. Tahap ini menguji fungsionalitas fitur penambahan item dan pengelolaan keranjang belanja. Terakhir, proses pembayaran secara otomatis diuji untuk memastikan bahwa transaksi dapat dilakukan dengan lancar, aman, dan sesuai dengan prosedur yang diharapkan. Pengujian transaksi mencakup validasi metode pembayaran dan pengolahan informasi pembayaran dengan aman.

Website yang dipilih penulis adalah "<https://www.demoblaze.com/>" (Gambar 1).



Gambar 1 Website Demo Blaze Untuk Simulasi Testing

Tulisan ini bertujuan untuk menguji beberapa fungsi pada website DemoBlaze dengan menggunakan Selenium. Fungsi-fungsi yang akan diuji meliputi fungsi pendaftaran, fungsi login, fungsi menambahkan item ke dalam cart, dan fungsi transaksi.

2. Tinjauan Pustaka

2.1 Testing

Testing merupakan suatu operasi pemeriksaan ulang atau pembenaran jalannya suatu sistem atau program sesuai dengan seharusnya. Testing merupakan rangkaian suatu proses yang dilakukan untuk penilaian dari suatu program maupun sistem dan dapat ditetapkan apakah program sudah memenuhi standar yang diinginkan atau belum (Emi et al., 2022).

2.2 Python

Python adalah bahasa pemrograman yang populer saat ini. Bahasa Python dibuat oleh Guido van Rossum, dan dirilis pada tahun 1991. Python dapat digunakan untuk Pengembangan web (sisi server), Pengembangan perangkat lunak atau membuat aplikasi (*software*), Menyelesaikan persamaan Matematika, Pembuatan skrip sistem dan Pemrograman Mikrokontroler (Micro Python) (Ma'arif, 2020).

2.3 Selenium

Selenium adalah alat open-source untuk mengotomatiskan pengujian web. Selenium memungkinkan Anda untuk berinteraksi dengan elemen web seperti tautan, tombol, dan kotak teks (Selenium, 2021).

2.4 WebDriver

WebDriver adalah antarmuka kendali jarak jauh yang memungkinkan introspeksi dan kontrol agen pengguna. Ini menyediakan protokol kabel yang netral platform dan bahasa sebagai cara bagi program di luar proses untuk menginstruksikan perilaku browser web dari jarak jauh (W3C Working Draft, 2024).

2.5 E-Commerce

Menurut Whiston, Choi, & Stahl 1997 E-commerce mengacu pada penggunaan sarana dan teknologi elektronik untuk melakukan perdagangan (penjualan, pembelian, transfer, atau pertukaran produk, layanan, dan atau informasi) termasuk dalam interaksi bisnis, bisnis-ke-bisnis, dan bisnis-ke-konsumen. Pengiriman produk atau layanan dapat terjadi melalui atau di luar internet (Manzoor, 2010).

3. Metode

Pengujian ini menggunakan software IntelliJ IDEA sebagai Integrated Development Environment (IDE) untuk menulis script pengujian, yang mana IDE ini dipilih karena menyediakan fitur-fitur yang lengkap dan memudahkan dalam proses pengembangan kode. Bahasa pemrograman yang digunakan adalah Python, yang dikenal karena sintaksisnya yang sederhana dan mudah dipahami serta kekuatan dan fleksibilitasnya dalam pengembangan perangkat lunak, termasuk untuk tujuan pengujian. Selain itu, alat pengujian yang digunakan adalah Selenium, sebuah alat open-source yang sangat efektif dalam mengotomatisasi pengujian web. Selenium memungkinkan peneliti untuk berinteraksi dengan berbagai elemen web seperti tombol, tautan, dan formulir input secara otomatis, sehingga meningkatkan efisiensi dan konsistensi pengujian.

4. Hasil dan Pembahasan

4.1 Source Code Testing menggunakan Selenium

Source code testing menggunakan Selenium pada tulisan ini ada pada Lampiran 1 (halaman 55). Source code di Lampiran 1 digunakan untuk mengotomatiskan pengujian fungsi-fungsi pada *website* DemoBlaze menggunakan Python dan

Selenium. Isinya ini mencakup langkah-langkah mulai dari mengimpor library yang diperlukan, membuka browser dan *website*, mendefinisikan XPath elemen penting, hingga alur pengujian yang melibatkan proses pendaftaran, *login*, menambahkan barang ke keranjang, dan melakukan pemesanan. Setiap fungsi dan proses akan dijelaskan secara rinci untuk memberikan gambaran lengkap mengenai mekanisme kerja dari *script* ini.

Library random digunakan untuk menghasilkan nilai acak, seperti *username* dan nomor kartu kredit. *Library string* dimanfaatkan untuk memanipulasi string, seperti dalam pembuatan nomor kartu kredit. *Library Selenium* digunakan untuk mengotomatiskan interaksi dengan *browser web*, sementara *library Time* digunakan untuk memberikan jeda waktu dalam proses.

Function `driver = webdriver.Chrome()` digunakan untuk membuat *instance* `WebDriver` untuk `Chrome` guna membuka *browser*. Function `driver.get("https://www.demoblaze.com/")` digunakan untuk mengarahkan *browser* ke *website* `DemoBlaze`. Function `driver.maximize_window()` digunakan untuk memaksimalkan ukuran jendela *browser*. Function `driver.execute_script('document.body.style.zoom = "100%"')` digunakan untuk mengatur skala *zoom* halaman ke 100% guna menghindari kesalahan pada interaksi elemen.

Function `xpath_list` merupakan sebuah *dictionary* yang berisi XPath elemen-elemen penting untuk berinteraksi dengan *website*. Function `scroll_to(elem)` digunakan untuk menggulung layar agar elemen tertentu terlihat. Function `xpath_element(xpath)` digunakan untuk mencari elemen berdasarkan XPath dan menunggu hingga elemen tersebut muncul. Function `linktext_click(text)` digunakan untuk menggulir ke elemen dengan *link text* tertentu dan mengklik elemen tersebut.

Alur pengujian mencakup beberapa langkah. Proses pendaftaran digunakan untuk membuat akun baru dengan *username* acak dan *password* "password". Proses login dilakukan menggunakan akun yang baru dibuat dari proses pendaftaran. Menambahkan barang ke keranjang dilakukan dengan menambahkan "Sony vaio i7" dan "Iphone" ke keranjang. Proses pemesanan meliputi langkah-langkah pergi ke keranjang belanja, menekan tombol "place order", mengisi formulir pemesanan dengan nama 'lisa', negara 'indo', kota 'jakarta', nomor kartu kredit dengan 10 angka acak, bulan *expired* untuk kartu kredit acak antara 1-12, dan tahun *expired* untuk kartu kredit acak antara 2025-2029, kemudian mengklik tombol. *Infinite loop* digunakan agar program terus berjalan tanpa henti.

4.2 Source Code Absensi

Source code absensi ada pada bagian Lampiran 2. *Source code* pada Lampiran 2 menjelaskan cara kerja *source code* yang digunakan untuk mengatur dan mengevaluasi kelompok melalui kelas `GroupEvaluation`. Kelas ini menyimpan informasi dan

mengatur evaluasi kelompok. Atribut kelas ini meliputi `group_name` yang menyimpan nama kelompok (*string*), `members` yang menyimpan daftar nama anggota kelompok (*list*), `attendance_records` yang merupakan dictionary untuk menyimpan kehadiran tiap pertemuan (dengan key sebagai nomor pertemuan dan value sebagai list yang berisi kehadiran tiap anggota - True untuk hadir, False untuk tidak hadir), dan `contributions` yang merupakan dictionary untuk menyimpan kontribusi tiap anggota (dengan key sebagai nama anggota dan value sebagai nilai kontribusi yang awalnya 0).

Metode dalam kelas `GroupEvaluation` mencakup beberapa fungsi penting. Metode `set_group_name(self, group_name)` digunakan untuk memasukkan nama kelompok. Metode `add_member(self, member_name)` menambahkan anggota kelompok ke daftar dan menginisialisasi kontribusi awal menjadi 0. Metode `record_attendance(self, meeting_number, attendance_list)` mencatat kehadiran anggota kelompok pada setiap pertemuan. Metode `generate_attendance_report(self)` mencetak laporan absensi dan kontribusi kelompok. Proses dalam metode ini meliputi menampilkan header laporan dengan nama kelompok dan judul, iterasi melalui `attendance_records` untuk menampilkan kehadiran tiap pertemuan, mencetak rekap kontribusi individual, dan menghitung serta menampilkan metrik seperti Quality Score dan Collaboration.

5. Kesimpulan

Pengujian yang dilakukan pada website DemoBlaze menggunakan bahasa pemrograman Python dan program Selenium dengan WebDriver Chrome dalam IDE IntelliJ IDEA telah berhasil. Hasil pengujian menunjukkan bahwa fungsi-fungsi yang diinginkan dapat berjalan dengan baik. Fungsi pendaftaran, login, penambahan item produk ke dalam keranjang belanja, serta penghitungan harga dan pembayaran semuanya berfungsi dengan baik. Pengujian ini juga menunjukkan bahwa website DemoBlaze berjalan dengan baik, mudah digunakan, stabil, dan aman. Penggunaan Selenium dan Python dalam IDE IntelliJ IDEA terbukti efektif dalam melakukan pengujian terhadap website DemoBlaze. Dengan demikian, pengujian ini dinyatakan selesai dilaksanakan.

Referensi

- Eriana, E. S., Subariah, R., & Farizy, S. (2022). *Testind & Implementasi Sistem*. Tangerang Selatan: Unpam Press.
- Ginting, A.P., Abidin, Z., Asari, A., & Saifudin, A. (2023). Otomatisasi Pengujian Aplikasi Web Toko Sembako Menggunakan Selenium IDE. *Jurnal Ilmu Komputer dan Pendidikan*, 1(2). <https://journal.mediapublikasi.id/index.php/logic>
- Ma'arif, A. (2020). *Buku Ajar Pemrograman Lanjut Bahasa Pemrograman Python* [Universitas Ahmad Dahlan Yogyakarta]. <https://eprints.uad.ac.id/32743/1/buku%20python.pdf>

- Manzoor. (2010). *E-Commerce An Introduction*. LAP LAMBERT Academic Publishing GmbH & Co.KG.
- Selenium. (2021, December 7). *About Selenium*.
<https://www.selenium.dev/documentation/about/>
- W3C Working Draft. (2024, May 21). *WebDriver*. <https://www.w3.org/TR/webdriver/#sotd>

Lampiran 1 Source Code Testing menggunakan Selenium

```
import random
import string

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

import time

# UTS Testing dan Implementasi
# 47210021 Alissa Salim SI Kelas S
# 41210018 Yehezkiel Arly SI Kelas S

driver = webdriver.Chrome()
driver.get("https://www.demoblaze.com/")
driver.maximize_window()
driver.execute_script('document.body.style.zoom = "100%"')

xpath_list = {
    'nav_login': '//*[@id="login2"]',
    'nav_signup': '//*[@id="signin2"]',
    'nav_home': '//*[@id="navbarExample"]/ul/li[1]/a',
    'nav_cart': '//*[@id="cartur"]',
    'login_btn': '//*[@id="loginModal"]/div/div/div[3]/button[2]',
    'signup_btn': '//*[@id="signinModal"]/div/div/div[3]/button[2]',
    'login_username': '//*[@id="loginusername"]',
    'signup_username': '//*[@id="sign-username"]',
    'login_password': '//*[@id="loginpassword"]',
    'signup_password': '//*[@id="sign-password"]',
    'place_order': '//*[@id="page-wrapper"]/div/div[2]/button',
    'order_name': '//*[@id="name"]',
    'order_country': '//*[@id="country"]',
    'order_city': '//*[@id="city"]',
    'order_cc': '//*[@id="card"]',
    'order_month': '//*[@id="month"]',
    'order_year': '//*[@id="year"]',
    'order_btn': '/html/body/div[3]/div/div/div[3]/button[2]'
}

def scroll_to(elem):
    driver.execute_script("arguments[0].scrollIntoView(true);", elem)

def xpath_element(xpath):
    return WebDriverWait(driver,
20).until(ec.presence_of_element_located((By.XPATH, xpath)))

def linktext_click(text):
    elem_to_click = WebDriverWait(driver,
20).until(ec.presence_of_element_located((By.PARTIAL_LINK_TEXT, text)))
    scroll_to(elem_to_click)
    elem_to_click.click()
```

```
username = 'lisalisa' + str(random.randint(1,500000))
password = 'password'

# Sign Up
xpath_element(xpath_list['nav_signup']).click()
time.sleep(2)
xpath_element(xpath_list['signup_username']).send_keys(username)
xpath_element(xpath_list['signup_password']).send_keys(password)
xpath_element(xpath_list['signup_btn']).click()
time.sleep(2)
driver.switch_to.alert.accept()

# Login
xpath_element(xpath_list['nav_login']).click()
time.sleep(2)
xpath_element(xpath_list['login_username']).send_keys(username)
xpath_element(xpath_list['login_password']).send_keys(password)
xpath_element(xpath_list['login_btn']).click()

# Add Sony Vaio to cart
time.sleep(2)
linktext_click("Sony vaio i7")
linktext_click("Add to cart")
time.sleep(2)
driver.switch_to.alert.accept()
xpath_element(xpath_list['nav_home']).click()

# Add iPhone to cart
linktext_click("Iphone")
linktext_click("Add to cart")
time.sleep(2)
driver.switch_to.alert.accept()

# Go to cart
time.sleep(2)
xpath_element(xpath_list['nav_cart']).click()

# Press Order
time.sleep(2)
xpath_element(xpath_list['place_order']).click()

# Fill Order
time.sleep(4)

xpath_element(xpath_list['order_name']).send_keys('lisa')
xpath_element(xpath_list['order_country']).send_keys('indo')
xpath_element(xpath_list['order_city']).send_keys('jakarta')
xpath_element(xpath_list['order_cc']).send_keys(
    ''.join(random.choice(string.digits) for _ in range(10)) # 10 angka acak
    sebagai nomor kartu kredit
)
xpath_element(xpath_list['order_month']).send_keys(random.randint(1,12)) #
Bulan acak antara 1 sampai 12
scroll_to(xpath_element(xpath_list['order_year']))
xpath_element(xpath_list['order_year']).send_keys(random.randint(2025,
2029)) # Tahun acak antara 2025 sampai 2029
```

```
time.sleep(2)
scroll_to(xpath_element(xpath_list['order_btn']))
driver.find_element(By.CSS_SELECTOR, '#orderModal > div > div > div.modal-
footer > button.btn.btn-primary').click()

while True:
    pass
```

Lampiran 2 Source Code Absensi

```

class GroupEvaluation:
    def __init__(self):
        self.group_name = ""
        self.members = []
        self.attendance_records = {} # Dictionary untuk merekam kehadiran
        self.contributions = {} # Dictionary untuk menyimpan kontribusi

    def set_group_name(self, group_name):
        self.group_name = group_name

    def add_member(self, member_name):
        self.members.append(member_name)
        self.contributions[member_name] = 0 # Set kontribusi awal ke 0

    def record_attendance(self, meeting_number, attendance_list):
        if len(attendance_list) == len(self.members):
            self.attendance_records[meeting_number] = attendance_list
        else:
            print("Error: Jumlah anggota tidak sesuai.")

    def generate_attendance_report(self):
        print()
        print("Sistem Absensi Untuk Proyek UTS Testing dan Implementasi",
self.group_name)
        print("=====")
        for meeting_number, attendance_list in
self.attendance_records.items():
            print("Pertemuan", meeting_number)
            # Tabel rekap nama yang hadir
            print("{:<15} {:<10}".format("Nama", "Kehadiran"))
            print("-" * 25)

            for member, attendance in zip(self.members, attendance_list):
                print("{:<15} {:<10}".format(member, "Hadir" if attendance else
"Tidak Hadir"))
                if attendance:
                    self.contributions[member] += 1 # Tambahkan kontribusi
            print()
            # Total kontribusi individual
            total_contributions = sum(self.contributions.values())
            # Jumlah kontributor
            num_contributors = len(self.contributions)
            # Total kontribusi individual terendah
            min_contribution = min(self.contributions.values())
            # Total kontribusi individual tertinggi
            max_contribution = max(self.contributions.values())
            # Quality Score
            quality_score = (total_contributions / num_contributors)

            # Collaboration Score
            collaboration_score = (min_contribution / max_contribution)

        # Menampilkan rekap kontribusi
        print("\nRekap Kontribusi")
        print("=====")
        for member, contribution in self.contributions.items():
            print(f"{member}: {contribution}")
            # Menampilkan Quality Score dan Collaboration Score
            print("\nQuality Score:", quality_score)
            print("Collaboration Score:", collaboration_score)

```

```
# Contoh penggunaan:
group1 = GroupEvaluation()

# Input nama grup
group_name = input("Masukkan nama grup: ")
group1.set_group_name(group_name)

# Input anggota grup
while True:
    member_name = input("Masukkan nama anggota grup (tekan Enter jika sudah tidak ada)")
    if not member_name:
        break
    group1.add_member(member_name)

# Input kehadiran setiap pertemuan
num_meetings = int(input("Masukkan jumlah pertemuan: "))
for i in range(1, num_meetings + 1):
    print("Pertemuan ke-", i)
    attendance_list = []
    for member in group1.members:
        attendance = input(f"Apakah {member} hadir? (Y/N): ")
        attendance_list.append(attendance.upper() == "Y")
    group1.record_attendance(i, attendance_list)

# Menampilkan laporan kehadiran dan rekap kontribusi
group1.generate_attendance_report()
```