

## BAB II TINJAUAN PUSTAKA

### A. Penelitian Terdahulu

Penelitian terdahulu telah dilakukan Halawa & Saifudin (2023) dengan judul “Pengujian Fungsionalitas Aplikasi Kasir Berbasis Web dengan Metode Blackbox”. Penelitian ini bertujuan untuk memastikan bahwa aplikasi kasir dapat berfungsi dengan baik sesuai kebutuhan pengguna. Metode *black box* digunakan untuk menguji aplikasi dari sudut pandang pengguna tanpa memperhatikan detail implementasi kode. Selenium digunakan sebagai alat otomatisasi untuk menjalankan *test case* pengguna pada *browser*, sehingga memungkinkan pengujian yang lebih efisien dan efektif. Hasil penelitian menunjukkan bahwa Selenium meningkatkan efisiensi dan akurasi pengujian, memastikan sistem berfungsi sesuai kebutuhan pengguna, serta meminimalkan risiko kesalahan. Namun, karena *black box testing* tidak menguji struktur internal perangkat lunak, *white box testing* mungkin diperlukan untuk analisis lebih mendalam guna meningkatkan kualitas aplikasi.

Kusyadi et al. (2023) telah melakukan penelitian dengan judul yang diambil "Pengujian Aplikasi Website Sekolah Menggunakan Selenium WebDriver". Penelitian ini bertujuan untuk memverifikasi bahwa berbagai fitur dalam aplikasi *website* sekolah, seperti *login*, pendaftaran siswa, manajemen pengumuman, dan galeri sekolah, beroperasi sesuai dengan spesifikasi yang ditetapkan. Pengujian dilakukan dengan menerapkan metode *black box testing* dan menggunakan alat bantu otomatisasi Selenium WebDriver. Hasil penelitian mengungkapkan bahwa seluruh fitur yang diuji berfungsi sebagaimana mestinya, sehingga aplikasi *website* sekolah dinilai siap digunakan untuk mendukung aktivitas operasional sekolah.

Nugraha et al. (2020) telah melakukan penelitian berjudul "Pengujian Black Box pada Aplikasi Penghitungan Parkir Swalayan ADA Menggunakan Teknik Equivalence Partitions". Penelitian ini bertujuan untuk mengevaluasi kinerja aplikasi penghitungan parkir guna memastikan fungsionalitasnya

berjalan dengan baik dan bebas dari kesalahan yang dapat merugikan pengelola swalayan. Metode yang digunakan adalah *Black Box Testing* dengan pendekatan *Equivalence Partition*, yang membagi pengujian ke dalam berbagai kategori, seperti validasi fungsi, struktur data, dan interaksi antarmuka. Hasil penelitian mengungkapkan bahwa metode ini efektif dalam mendeteksi kesalahan sistem, seperti ketidaksesuaian dalam pencatatan waktu parkir serta data kendaraan yang tidak terhapus otomatis setelah keluar. Dengan demikian, penelitian ini menegaskan pentingnya penerapan pengujian perangkat lunak untuk meningkatkan keakuratan dan keandalan sistem penghitungan parkir.

Penelitian terdahulu yang dilakukan oleh Kurniawan et al. (2023) dengan judul “Otomatisasi Pengujian Aplikasi Pawoon Menggunakan Selenium IDE”. Masalah yang diangkat dalam penelitian ini adalah adanya kendala seperti bug, ketidaksesuaian dengan kebutuhan pengguna, dan eror pada aplikasi yang telah dibuat. Penelitian ini bertujuan untuk meminimalisir potensi bug melalui pengujian otomatis. Metode yang digunakan adalah *black box testing* dengan alat bantu Selenium IDE, yang dianggap lebih sederhana dan efisien. Hasil penelitian menunjukkan bahwa semua fungsional *input* dan *output* telah berjalan sesuai dengan kebutuhan pengguna, sehingga aplikasi dapat dikatakan layak digunakan.

Penelitian terdahulu telah dilakukan oleh Ayuningtyas et al. (2023) dengan judul "Performance and Functional Testing With The Black Box Testing Method". Penelitian ini bertujuan untuk menguji fungsionalitas dan performa portal akademik Perbanas dengan menggunakan metode *black box*. *Black box testing* digunakan untuk memastikan bahwa fitur-fitur portal, seperti *login* dan pengisian KRS, berjalan sesuai kebutuhan pengguna tanpa memperhatikan struktur internal sistem. Dalam penelitian ini, Katalon Studio digunakan sebagai alat otomatisasi untuk pengujian fungsional, sementara Apache JMeter digunakan untuk pengujian performa. Hasil pengujian menunjukkan bahwa portal Perbanas memiliki tingkat keberhasilan login yang tinggi dengan rata-rata 15 transaksi per detik dan tingkat kegagalan hanya sebesar 0,06% pada pengujian beban 250 pengguna.

Thooriqoh et al. (2021) telah melakukan penelitian berjudul "Selenium Framework for Web Automation Testing: A Systematic Literature Review". Penelitian ini bertujuan untuk menganalisis berbagai studi terkait pemanfaatan Selenium dalam *automation testing*. Dengan menggunakan metode *Systematic Literature Review* (SLR), penelitian ini meninjau 2.418 jurnal dan menyaringnya menjadi 20 penelitian yang paling relevan. Hasil penelitian menunjukkan bahwa Selenium tidak hanya dimanfaatkan untuk pengujian fungsional antarmuka pengguna (*UI testing*), tetapi juga dapat diterapkan dalam pengujian keamanan serta dikombinasikan dengan metode lain seperti data mining, kecerdasan buatan (*AI*), dan *machine learning*. Studi ini menekankan pentingnya peningkatan efektivitas serta kemudahan pemeliharaan (*maintainability*) dalam implementasi Selenium di berbagai pendekatan pengujian perangkat lunak.

Penelitian yang dilakukan oleh Suharyanti & Supendar (2024) dengan judul "Web Program Testing Using Selenium Python: Best Practices and Effective Approaches" berfokus pada evaluasi penggunaan Selenium WebDriver dengan bahasa pemrograman Python dalam pengujian fungsionalitas situs *e-commerce* [www.saucedemo.com](http://www.saucedemo.com). Penelitian ini menerapkan pengujian otomatis terhadap fitur login, transaksi barang, dan pengisian data pelanggan menggunakan teknik identifikasi elemen seperti *id*, *class name*, dan *css selector*. Hasilnya menunjukkan bahwa Selenium Python mampu menjalankan pengujian secara efisien dan konsisten, dengan tingkat keberhasilan mencapai 100% pada semua skenario uji. Selain itu, penelitian ini mengidentifikasi praktik terbaik dalam implementasi Selenium, termasuk penerapan metode *wait* untuk menangani waktu pemuatan halaman serta validasi data.

Ramdhani et al. (2024) telah melakukan penelitian berjudul "Optimalisasi Proses Pembaharuan Locator pada Kode Otomatis Selenium dengan Menggunakan Page Object Model". Penelitian ini berfokus pada peningkatan efisiensi dalam memperbarui *locator* pada kode otomatisasi pengujian berbasis Selenium WebDriver melalui penerapan *Page Object Model* (POM). Dengan metode ini, *locator* disimpan dalam satu repositori

terpusat sehingga perubahan dapat dilakukan tanpa perlu memperbarui setiap skenario secara manual. Hasil penelitian menunjukkan bahwa penerapan POM mampu meningkatkan efisiensi pembaruan *locator* sebesar 14,28% dibandingkan dengan pendekatan manual. Oleh karena itu, penelitian ini memberikan kontribusi dalam mengoptimalkan *automation testing*, khususnya pada sistem yang sering mengalami perubahan elemen antarmuka pengguna.

Penelitian yang dilakukan oleh Inturi (2023) dengan judul “*Implementing Test Automation with Selenium WebDriver*” berfokus pada pengembangan dan penerapan kerangka kerja pengujian otomatis menggunakan Selenium WebDriver dengan bahasa pemrograman Java pada platform NanoHub.org. Temuan dari penelitian ini mengindikasikan bahwa penggunaan Selenium WebDriver secara substansial mampu mengurangi durasi serta beban kerja dalam proses pengujian, terutama pada tahap regresi dan pengujian fungsional, jika dibandingkan dengan metode manual. Framework yang dikembangkan juga mengadopsi pendekatan Page Object Model (POM) guna mempermudah pemeliharaan skrip pengujian

## **B. Landasan Teori**

### **1. Pengujian Perangkat Lunak**

Pengujian perangkat lunak merupakan tahap penting dalam siklus pengembangan perangkat lunak yang memastikan kualitas, kinerja, dan keandalan produk. Tujuan utamanya adalah mendeteksi cacat, kesalahan, atau ketidaksesuaian dengan spesifikasi yang dapat mempengaruhi fungsionalitas atau merugikan pengguna (Hozairi et al., 2024). Proses pengujian perangkat lunak mencakup beberapa tahapan utama, seperti perancangan skenario pengujian, penyusunan kasus uji, serta pelaksanaan pengujian. Dalam proses tersebut melibatkan berbagai aspek, termasuk proses verifikasi dan validasi, pendeteksian serta pemantauan bug, pengujian pada tingkat unit, integrasi, regresi, serta evaluasi kinerja sistem untuk memastikan kualitas dan keandalannya (Fairuz & Sofani, 2024). Pengujian perangkat lunak dapat dilakukan dengan dua cara, yaitu *manual*

*testing* dan *automated testing*. Perbandingan keduanya dijelaskan dalam Tabel 2.1.

Tabel 2. 1 Perbandingan *Manual Testing* dan *Automation Testing*

Aspek	Manual Testing	Automation Testing
Metode Eksekusi	Dilakukan oleh penguji secara langsung pada perangkat lunak	Menggunakan alat otomatis untuk menjalankan pengujian
Kecepatan	Memakan waktu lebih lama	Lebih cepat karena skrip pengujian dapat dijalankan berulang kali
Tingkat Kesalahan	Rentan terhadap kesalahan manusia	Lebih andal karena mengurangi keterlibatan manusia
Adaptasi Perubahan UI	Lebih mudah beradaptasi terhadap perubahan UI	Skrip harus diperbarui ketika UI mengalami perubahan
Kebutuhan Sumber Daya	Mebutuhkan lebih banyak tenaga penguji manual	Mebutuhkan alat pengujian dan <i>engineer</i> otomasi
Efektivitas Biaya	Lebih hemat untuk proyek dengan cakupan pengujian kecil	Lebih hemat untuk proyek dengan cakupan pengujian besar

Sumber: *A Review on the Process of Automated Software Testing* (2022)

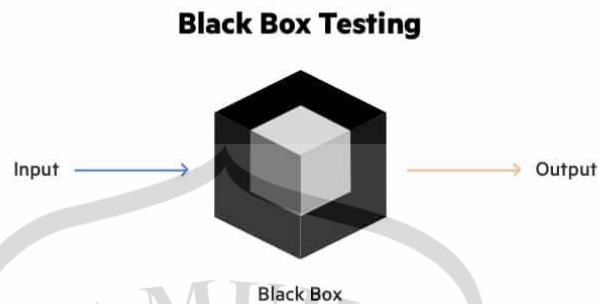
## 2. *Automation Testing*

★ Pengujian otomatis atau *automation testing* adalah metode pengujian perangkat lunak yang dilakukan menggunakan alat uji otomatis khusus untuk menjalankan serangkaian kasus uji. Perangkat lunak pengujian otomatis juga dapat memasukkan data uji ke dalam sistem yang diuji, membandingkan hasil yang diharapkan dengan hasil aktual, serta menghasilkan laporan pengujian yang terperinci (Thooriqoh et al., 2021).

## 3. *Black Box Testing*

Metode *black box testing* merupakan salah satu teknik dalam pengujian perangkat lunak yang menitikberatkan pada evaluasi fungsionalitas sistem tanpa mempertimbangkan struktur internal atau logika pemrogramannya. Dengan kata lain, aspek pengujian hanya mencakup fungsionalitas, antarmuka pengguna, dan proses kerja aplikasi, tanpa melibatkan kode atau skrip yang mendasari sistem tersebut

(Nurkholis & Yunial, 2023). Ilustrasi dari pendekatan ini disajikan dalam Gambar 2.1.



Gambar 2. 1 Konsep *Black Box Testing*  
Sumber: Panduan Komprehensif Pengujian Perangkat Lunak (2024)

#### 4. Selenium WebDriver

Selenium merupakan perangkat lunak *open-source* yang digunakan untuk melakukan proses otomisasi pengujian aplikasi web di berbagai sistem operasi dan platform (Wardhan & Madan, 2021). Dengan menggunakan Selenium WebDriver, pengguna dapat berinteraksi langsung dengan *browser* secara otomatis. Selenium mendukung berbagai peramban populer, termasuk Google Chrome, Mozilla, Firefox, Microsoft Edge, Safari, Opera, serta Internet Explorer (Fairuz & Sofani, 2024). Selain itu, Selenium juga merupakan sekumpulan *library* yang berisi *class* dan *function* untuk merancang dan mendefinisikan suatu pengujian (Ramdhani et al., 2024).

#### 5. *Functional Testing*

Menurut Mustofa & Fajar (2018), *functional testing* atau pengujian fungsional dilakukan untuk mengevaluasi apakah suatu perangkat lunak telah sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Proses ini melibatkan perbandingan antara data yang dimasukkan ke dalam sistem dengan hasil keluaran yang seharusnya diterima.

#### 6. *Page Object Model (POM)*

*Page Object Model* (POM) adalah sebuah pola desain dalam pengujian otomatis yang digunakan untuk memetakan setiap halaman atau bagian dari aplikasi web ke dalam kelas-kelas dalam kode program.

Dengan memisahkan skrip pengujian dari elemen utama, POM menjadikan kode lebih modular dan mudah dikelola. Pendekatan ini juga mendukung pengujian lintas platform dan browser melalui *Base Page Class* serta *Factory Method*, yang memungkinkan penciptaan objek halaman secara dinamis untuk berbagai lingkungan (Rani & Chandrika, 2020). Struktur POM yang terorganisir meningkatkan skalabilitas, keterbacaan kode, dan efektivitas pengujian otomatis.

## 7. Tentang Sobatlazismu

Sobatlazismu adalah sebuah platform hasil kerja sama antara PT Mandatech Mataram Mukti dengan Lazismu Daerah Bantul untuk mendukung operasional Lazismu sebagai lembaga zakat nasional. Platform ini berfungsi sebagai sarana pengelolaan dan pemanfaatan dana zakat, infak, wakaf, serta donasi lainnya dari individu, lembaga, perusahaan, maupun instansi guna meningkatkan pemberdayaan masyarakat secara produktif. Visualisasi antarmuka pengguna dari platform Sobatlazismu ditampilkan pada Gambar 2.2.



Gambar 2. 2 Tampilan Platform Sobatlazismu