

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Terdapat penelitian terdahulu yang dijadikan sebagai sumber referensi dalam penyusunan penelitian. Berikut merupakan Tabel 2.1 yang menjelaskan ringkasan dari penelitian terdahulu.

Tabel 2. 1 Penelitian Terdahulu

| | | |
|-----------------|---|---|
| 1 | Judul | <i>A Study of Automated Software Testing: Automation Tools and Frameworks</i> |
| | Jurnal | <i>International Journal of Computer Science Engineering (IJCSE) Vol 8, No 6</i> |
| | Tahun | 2019 |
| | Penulis | Mubarak Albarka Umar, Chen Zhanfang |
| | Indikator | <i>Automation</i> |
| | Definisi | <i>Automation</i> merupakan sebuah kegiatan yang berjalan atau berproses secara otomatis, tanpa memerlukan banyak tenaga manusia. |
| | Hasil dan temuan kesimpulan | Hasil dari penelian ini adalah sebuah perbandingan dari berbagai <i>tools automation</i> yang sering digunakan dalam melakukan pengujian sebuah perangkat lunak [7]. |
| | Future Research | Penerapan <i>automation</i> terhadap suatu kegiatan khususnya yang menggunakan aplikasi berbasis web. |
| 2 | Judul | <i>Automated driver management for Selenium Webdriver</i> |
| | Jurnal | <i>Empirical Software Engineering, Vol. 26, No. 101</i> |
| | Tahun | 2021 |
| | Penulis | Boni García, Mario Munoz-Organero, Carlos Alario-Hoyos, Carlos Delgado Kloos |
| | Indikator | <i>Selenium Webdriver</i> |
| | Definisi | <i>Selenium Webdirver</i> merupakan salah satu framework untuk melakukan sebuah pengujian otomatis terhadap sebuah website. |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini menunjukkan penilaian yang baik terhadap kemampuan otomatisasi dan kegunaan <i>selenium webdriver</i> oleh pengguna Java, tetapi adopsi yang kurang untuk bahasa pemrograman lainnya [8]. |
| Future Research | Penerapan <i>selenium webdriver</i> pada kegiatan <i>automation</i> dengan menggunakan bahasa pemrograman yang berbeda. | |
| 3 | Judul | Rancangan Program Otomatisasi Otentikasi Pengguna Untuk Otorisasi Pada Website Dengan <i>Python</i> Dan <i>Selenium Web Driver</i> |
| | Jurnal | Jurnal Manajemen Informatika Jayakarta, Vol. 2, No. 4 |
| | Tahun | 2022 |
| | Penulis | Sulthan Muhammad Justice Islam Al Makassar, Thomas Budiman, Akmal Budi Yulianto |

| | | |
|-----------------|---|--|
| | Indikator | <i>Python</i> |
| | Definisi | <i>Python</i> merupakan sebuah bahasa pemrograman yang biasa digunakan untuk pemrosesan data, selain itu <i>python</i> juga bisa digunakan untuk membuat proses <i>automation</i> . |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah berupa sebuah program <i>automation</i> otentikasi pada <i>website</i> . Program tersebut sangat bermanfaat dikarenakan dapat meningkatkan dan mempermudah penjaminan mutu, pengujian kompatibilitas dengan baik [4]. |
| | Future Research | Penerapan <i>selenium webdriver</i> proses <i>automation</i> pengambilan dan pemrosesan data pada aplikasi berbasis web. |
| 4 | Judul | Analisis Pemanfaatan <i>Playwright</i> untuk Automasi Pengujian Aplikasi Berbasis Web (Studi Kasus : Sistem Manajemen Jaringan) |
| | Jurnal | AUTOMATA, Vol. 3, No. 1 |
| | Tahun | 2022 |
| | Indikator | <i>Playwright</i> . |
| | Definisi | <i>Playwright</i> merupakan sebuah <i>tool open source</i> yang dikembangkan oleh Microsoft dan digunakan untuk melakukan sebuah proses otomatisasi <i>website</i> . |
| | Penulis | Anisa Amalia, Andhik Budi Cahyono |
| | Hasil dan temuan kesimpulan | Pada penelitian ini menghasilkan sebuah <i>web automation testing</i> yang dilakukan dengan menggunakan <i>playwright</i> . <i>Playwright</i> sendiri sangat membantu dalam pengujian testing dan pesentase terhadap pengujian yang dilakukan lulus 100%. <i>Playwright</i> cocok diterapkan oleh <i>automation</i> pemula, dikarenakan kemudahan dalam penggunaannya [9]. |
| Future Research | Penerapan otomatisasi web dengan menggunakan <i>playwright</i> pada pengambilan dan pengolahan data. | |
| 5 | Judul | <i>IMPLEMENTATION OF CROWDFUNDING WEB APPLICATION USING AWS AMPLIFY ARCHITECTURE WITH END-TO-END TESTING USING PLAYWRIGHT</i> |
| | Jurnal | Indonesian Journal of Multidisciplinary Science, Vol. 2, No. 12 |
| | Tahun | 2023 |
| | Penulis | Ary Utomo, Ganang Wijaya, Yanto Setiawan |
| | Indikator | AWS Amplify |
| | Definisi | AWS Amplify merupakan sebuah <i>platform</i> dari Amazon Web Services (AWS) yang berfungsi untuk membantu dalam <i>develop</i> dan <i>manage</i> aplikasi berbasis <i>mobile</i> atau web. |
| | Hasil dan temuan kesimpulan | Salah satu hasil dari penelitian ini adalah pengujian otomatis yang dilakukan dengan <i>playwright</i> pada 11 skenario yang telah ditetapkan dengan menunjukkan hasil performa pengujian yang baik dan cepat hanya 1,9 menit. [10]. |
| Future Research | Penerapan <i>automation playwright</i> dengan menggunakan bahasa pemrograman selain <i>JavaScript</i> , contohnya pada penelitian ini menerapkan bahasa pemrograman <i>python</i> . | |
| 6 | Judul | <i>Implementasi Web Scraping untuk Pengambilan Data Pada Website E-Commerce</i> |
| | Jurnal | <i>Jurnal Teknologi Dan Sistem Informasi Bisnis</i> |
| | Tahun | 2023 |

| | | |
|---|-----------------------------|---|
| | Penulis | Apriza Zicka Rizquina, Chanifah Indah Ratnasari |
| | Indikator | <i>Web Scraping</i> |
| | Definisi | <i>Web Scraping</i> merupakan sebuah metode otomatisasi dalam pengambilan data dari sebuah <i>website</i> atau sumber lainnya. |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah pembahasan mengenai implementasi <i>web scraping</i> untuk pengambilan data pada <i>website e-commerce</i> . <i>Website</i> yang digunakan adalah Shopee dan Tokopedia. Pengujian dilakukan sebanyak 10 kali, dan data yang didapatkan setiap kali pengujian adalah sekitar 160-166 data, dengan rata-rata durasi total perpengujiannya adalah 1 menit 0.5 detik [16]. |
| | Future Research | Penerapan pengambilan data dari sebuah aplikasi berbasis web dikombinasikan secara langsung dengan pengolahan dari data yang sudah dilakukan pengambilan sebelumnya. |
| 7 | Judul | Analisis Perbandingan Performansi <i>Framework Selenium</i> dan <i>Cypress</i> dalam Pengujian Otomatisasi Perangkat Lunak pada Aplikasi Berbasis <i>Website</i> |
| | Jurnal | Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Vol 8, No. 3 |
| | Tahun | 2024 |
| | Penulis | Andira Mahendra Syahputra, Muhammad Aminul Akbar, Aryo Pinandito |
| | Indikator | <i>Cypress</i> |
| | Definisi | <i>Cypress</i> merupakan <i>tools</i> pengujian front-end berbasis <i>JavaScript</i> yang dirancang untuk memudahkan pengembangan dan pengujian aplikasi web modern |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah perbandingan performa pengujian terhadap aplikasi portal berita online dengan menggunakan <i>selenium</i> dan <i>cypress</i> . Dari hasil perbandingan tersebut <i>cypress</i> memiliki durasi pengujian lebih baik jika dibandingkan dengan <i>selenium</i> [13]. |
| | Future Research | Melakukan perbandingan <i>tools automation</i> berbeda dan pada objek atau kegiatan berbeda. |
| 8 | Judul | Otomatisasi Pengujian Aplikasi Web Toko Sembako Menggunakan <i>Selenium IDE</i> |
| | Jurnal | Jurnal Ilmu Komputer dan Pendidikan, Vol 1, No. 2 |
| | Tahun | 2023 |
| | Penulis | Alfrianto Pratama Ginting, Zaenal Abidin, Aas Asari, Aries Saifudin |
| | Indikator | <i>Selenium IDE</i> |
| | Definisi | <i>Selenium IDE</i> merupakan alat yang digunakan untuk mengembangkan dan menjalankan tes otomatis untuk aplikasi web. |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah kegiatan pengujian secara otomatis terhadap fungsional dari aplikasi web dengan menggunakan <i>Selenium IDE</i> . Dari hasil pengujian tersebut didapati bahwa <i>selenium IDE</i> secara konsisten dan berhasil dalam melakukan pengujian [14]. |
| | Future Research | Penerapan penggunaan versi <i>selenium</i> lainnya seperti <i>selenium webdriver</i> atau <i>selenium grid</i> dalam melakukan <i>automation</i> terhadap aplikasi web. |

| | | |
|----|-----------------------------|---|
| 9 | Judul | Pemanfaatan <i>Selenium</i> Ide Dalam Otomatisasi Pengujian Black Box Pada Website Umkm Kota Madiun Dengan Teknik Equivalence Partitioning |
| | Jurnal | Jurnal Mahasiswa Teknik Informatika, Vol 8, No. 3 |
| | Tahun | 2024 |
| | Penulis | Eragradiansyah Hardianto, Firza Prima Aditiawan, Andreas Nugroho Sihananto |
| | Indikator | <i>Black box</i> |
| | Definisi | <i>Black box</i> merupakan metode pendekatan dalam pengujian fungsionalitas dari sebuah sistem atau aplikasi. |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah pengujian fungsionalitas dari sebuah aplikasi berbasis web dengan menggunakan <i>selenium IDE</i> dan pendekatan metode <i>black box</i> . Dari hasil pengujian yang telah dilakukan dimana 92% <i>test case</i> sesuai [15]. |
| | Future Research | Penerapan penggunaan versi <i>selenium</i> berbeda atau <i>tools</i> berbeda dalam melakukan <i>automation</i> terhadap aplikasi web. |
| 10 | Judul | Optimasi Metode Scraping Data Produk Dari Platform Tokopedia.Com |
| | Jurnal | Seminar Nasional Mahasiswa Fakultas Teknologi Informasi, Vol 2, No. 2 |
| | Tahun | 2023 |
| | Penulis | Muhammad Daffa Al Farizi, Rahmad Hidayat, Musta'inul Abdi |
| | Indikator | <i>Robotic Process Automation (RPA)</i> |
| | Definisi | <i>Robotic Process Automation (RPA)</i> merupakan teknologi yang memungkinkan penggunaan perangkat lunak untuk mengotomatisasi tugas-tugas yang biasanya dilakukan oleh manusia di dalam aplikasi. |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah sebuah perbandingan HTTPX dengan <i>playwright</i> dalam melakukan proses otomatisasi <i>data scraping</i> . Dari hasil perbandingan tersebut <i>playwright</i> mendapatkan hasil yang lebih efisien dengan catatan waktu 21 menit 7 detik [12]. |
| | Future Research | Penerapan perbandingan antara <i>playwright</i> dengan <i>tools automation</i> lainnya. |
| 11 | Judul | Sistem Informasi Kepegawaian Menggunakan Metode Agile Development di CV. Angkasa Raya |
| | Jurnal | Jurnal Teknologi dan Informasi (JATI), Vol 12, No. 1 |
| | Tahun | 2022 |
| | Penulis | Suhari, A Faqih, F M Basysyar |
| | Indikator | <i>Agile</i> |
| | Definisi | <i>Agile</i> merupakan metodologi manajemen proyek dan pengembangan perangkat lunak yang menekankan fleksibilitas, kolaborasi tim, dan iterasi cepat. |
| | Hasil dan temuan kesimpulan | Hasil dari penelitian ini adalah sebuah aplikasi sistem informasi yang digunakan untuk melakukan proses pengolahan data pegawai agar menjadi lebih efisien. Pengembangan aplikasi tersebut menggunakan metode <i>agile</i> dan pendekatan pengujian <i>black box</i> [17]. |
| | Future Research | Penerapan metode pengembangan <i>agile</i> pada proses pengembangan lainnya, contoh pada penelitian ini menerapkannya pada pengembangan program <i>automation</i> . |

Pada Tabel 2.1 menjelaskan penelitian-penelitian terdahulu yang digunakan sebagai referensi dalam penyusunan penelitian ini. Berdasarkan hasil pembahasan dari penelitian terdahulu, dimana menjelaskan bahwa proses *automation* sendiri dapat menggantikan proses kegiatan yang dilakukan secara manual menjadi lebih efisien terhadap waktu [4], [10], [12], [13]. Terdapat juga pembahasan dari penelitian terdahulu lainnya, dimana adapun berbagai macam *tools* yang dapat melakukan proses *automation*, contohnya seperti *selenium*, dan *playwright* [7], [8], [9], [14], [15]. Namun, hasil pembahasan dari penelitian terdahulu belum dilakukan proses *automation* dalam melakukan proses pengambilan dan pengolahan data pada aplikasi berbasis web, selain itu juga *tools automation* yang digunakan belum terdapat perbandingan dalam memproses kegiatan sejenis. Dari temuan *gap* atau keterbatasan penelitian-penelitian terdahulu, maka pada penelitian ini akan melakukan proses pengembangan *automation* dalam melakukan pengambilan dan pengolahan data tunjangan pada aplikasi berbasis web dengan menggunakan *selenium* dan *playwright*. Setelah kedua *automation* tersebut berhasil dikembangkan, maka akan dilanjutkan ke tahap perbandingan efisiensi antara *selenium* dan *playwright*, dengan tujuan untuk melihat *automation* mana yang memiliki tingkat efisiensi terbaik. *Automation* dengan tingkat efisiensi terbaik akan direkomendasikan kepada PT XYZ untuk diterapkan dalam melakukan proses *automation* pengambilan dan pengolahan data tunjangan.

2.2 Teori tentang Topik Skripsi

2.2.1 Automation

Automation memiliki pengertian sebagai sebuah peralatan yang dioperasikan secara otomatis dengan tujuan untuk menghemat waktu dan tenaga manusia [18]. Terdapat pengertian lainnya, dimana *automation* adalah suatu rencana yang melakukan penggabungan dan penerapan teknologi dalam memperbaiki proses pekerjaan dengan fokus untuk meningkatkan produktivitas dan efektifitas dalam pekerjaan [18]. Dapat disimpulkan bahwa *automation* merupakan sebuah kegiatan atau proses yang dilakukan secara

otomatis yang bertujuan untuk efektifitas dan efisiensi terhadap waktu dengan menghasilkan sebuah *output* cepat dan akurat [2]. *Automation* sendiri memiliki beberapa jenis diantaranya [18]:

1) *Fixed Automation*

Fixed Automation merupakan jenis otomatisasi yang digunakan untuk membuat produk yang sama secara konsisten. Sistem otomatisasi ini dirancang untuk melakukan hal-hal yang sama berulang kali tanpa perlu melakukannya lagi. Keuntungan sistem otomatisasi tetap adalah konsistensi produk yang tinggi dan biaya produksi yang rendah. Kelemahan sistem otomatisasi tetap adalah sulit untuk dimodifikasi dan memerlukan biaya yang tinggi untuk diterapkan.

2) *Flexible Automation*

Flexible Automation merupakan Jenis otomatisasi yang disebut otomatisasi fleksibel menggunakan peralatan robotik dan perangkat lunak untuk mengotomatisasi berbagai tugas. Sistem otomatisasi ini dapat diprogram ulang untuk mengotomatisasi berbagai tugas, dan keuntungan dari otomatisasi fleksibel adalah kemampuan untuk mengotomatisasi berbagai tugas sekaligus fleksibilitas dalam mengubah tugas yang diotomatisasi. Kelemahan sistem otomatisasi ini, bagaimanapun, adalah biaya tinggi untuk membeli dan memelihara peralatan robotik.

3) *Robot Process Automation (RPA)*

Robot Process Automation (RPA) merupakan jenis otomatisasi yang menggunakan perangkat lunak untuk mengotomatisasi pekerjaan yang biasanya dilakukan oleh manusia. RPA dapat digunakan untuk mengotomatisasi pekerjaan rutin seperti pemrosesan pesanan, pengiriman email, dan pengolahan data, di mana ia dapat meningkatkan efisiensi dan produktivitas.

4) *Programmable Automation*

Programmable Automation adalah jenis otomatisasi yang menggunakan peralatan robotik dan perangkat lunak untuk mengotomatisasi berbagai tugas. Keuntungan *programmable automation* adalah fleksibilitas dan kemampuan untuk mengotomatisasi berbagai tugas. Kelemahan *programmable automation* adalah biaya tinggi untuk membeli dan memegangnya.

2.1.2 *Web Automation*

Web automation merupakan salah satu teknologi yang menggunakan perangkat lunak untuk mengotomatisasi tugas berulang di sebuah *browser*, seperti mengisi formulir, mengambil data dari situs web, menguji keandalan aplikasi web, dan menavigasi antar halaman secara otomatis [19]. Dengan kemampuannya mensimulasikan interaksi manusia dengan browser, teknologi ini dapat membuat pengembangan dan pengujian aplikasi web menjadi lebih efisien dan efektif [19]. Misalnya, dalam pengujian aplikasi, otomatisasi web memungkinkan pengguna menjalankan kasus pengujian secara berulang tanpa campur tangan manusia, mengidentifikasi masalah keandalan dan keamanan yang mungkin terlewatkan oleh pengujian manual [19]. Hal ini tidak hanya mempercepat siklus pengembangan, namun juga meningkatkan kualitas produk akhir dengan memastikan aplikasi *web* bekerja sesuai harapan di berbagai lingkungan dan konfigurasi *browser*.

Pengembangan *web automation* dapat dilakukan dengan menggunakan berbagai *tools* atau *framework* yang dirancang untuk memfasilitasi *web automation*. *Tools* atau *framework* seperti *Selenium webdriver*, *Cypress*, *Nightwatch*, dan *Playwright* menawarkan API yang kaya untuk kontrol browser terprogram, memungkinkan pengembang menulis *code* yang secara otomatis melakukan tugas seperti pengujian, penambahan data, dan manajemen konten [20]. Dengan menggunakan bahasa pemrograman populer seperti *Python*, *Java*, *JavaScript*, dan sebagainya, yang dapat memungkinkan pengembang dapat mengembangkan *code* otomatis kompleks yang dapat

menyelesaikan kasus penggunaan tertentu. Manfaat dari pendekatan ini mencakup kemampuan untuk mengintegrasikan otomatisasi web ke dalam pipeline CI/CD (Continuous Integration/Continuous Delivery), mempercepat umpan balik pengembangan, dan memungkinkan pengiriman perangkat lunak yang lebih cepat dan andal [20].

2.1.3 *Flowchart*

Flowchart merupakan sebuah Gambar atau grafik yang didalamnya terdapat alur penjelasan dari suatu fungsi program [21]. Dapat disimpulkan bahwa *flowchart* atau diagram alir merupakan suatu jenis Gambar atau diagram yang menjelaskan logika dan langkah-langkah dari proses awal hingga akhir sebuah program atau sistem [22]. Berdasarkan pengertian *flowchart* itu sendiri dapat disimpulkan fungsi utama dari *flowchart* adalah sebagai pengatur urutan langkah demi langkah dari proses yang akan dilakukan oleh program atau sistem yang akan dibuat [23]. Selain itu adapun fungsi umn dari *flowchart* itu sendiri, antara lain: [22]

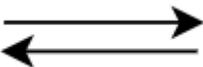
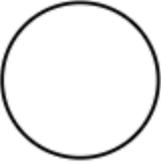
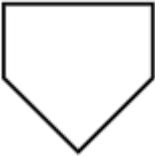
- 1) Membantu dalam perancangan sebuah proyek.
- 2) Dapat mempresentasikan pengolahan alur kerja dari program atau sistem yang dibuat.
- 3) Dapat mempresesntasikan pemodelan alur kerja suatu program atau sistem.
- 4) Membantu dalam mendokumentasikan setiap proses yang dilakukan program atau sistem.
- 5) Mempresentasikan algoritma yang digunakan dalam program atau sistem.
- 6) Membantu proses audit program atau sistem.

Dalam pembuat *flowchart* sendiri tentunya akan menggunakan berbagai simbol-simbol, sehingga dalam *flowchart* sendiri terbagi kedalam tiga kategori jenis symbol yaitu: [22]

- 1) *Flow Direction Symbols*

Sesuai dengan namanya yaitu *flow direction symbols* dimana simbol-simbol yang berada pada kategori ini berfungsi penghubung antara satu proses simbol dengan simbol lainnya. Pada Tabel 2.2 dibawah ini mempresentasikan simbol beserta fungsinya dari masing-masing simbol yang ada pada kategori *flow direction symbols*.

Tabel 2. 2 *Flow Direction Symbols*

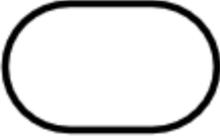
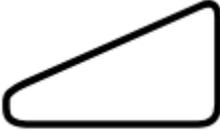
| Simbol | Nama | Fungsi |
|---|--------------------------|--|
|  | <i>Conecting Line</i> | Berfungsi untuk menghubungkan antara simbol-simbol satu sama lain. |
|  | <i>Connector</i> | Berfungsi sebagai penghubung satu proses dengan proses yang lainnya dalam halaman yang sama |
|  | <i>Offline Connector</i> | Berfungsi sebagai penghubung satu proses dengan proses yang lainnya dalam halaman yang berbeda |

2) *Processing Symbols*

Sesuai dengan namanya yaitu *processing symbols* dimana simbol - simbol yang berada pada kategori ini berfungsi sebagai yang menyatakan proses dari sistem. Pada Tabel 2.3 dibawah ini mempresentasikan simbol beserta fungsinya dari masing-masing simbol yang ada pada kategori *processing symbols*.

Tabel 2. 3 *Precessing Symbols*

| Simbol | Nama | Fungsi |
|---|-------------------|---|
|  | <i>Processing</i> | Berfungsi dalam menunjukan proses yang dilakukan oleh sistem. |

| Simbol | Nama | Fungsi |
|---|----------------------------|--|
|  | <i>Manual Operation</i> | Berfungsi dalam menunjukan proses yang dilakukan bukan oleh sistem. |
|  | <i>Decision</i> | Berfungsi sebagai suatu pemilihan kondisi yang akan diambil. |
|  | <i>Predefined Process</i> | Berfungsi sebagai penyimpan proses yang sedang/akan dilakukan oleh sistem. |
|  | <i>Terminal</i> | Berfungsi sebagai pernyataan akhir dari sistem. |
|  | <i>Offline Storage</i> | Berfungsi sebagai petunjuk penyimpanan kedalam media lain. |
|  | <i>Manual Input Symbol</i> | Berfungsi dalam menunjukan penginputan secara manual. |

3) *Input-Output Symbols*

Sesuai dengan namanya yaitu *input/output symbols* dimana simbol - simbol yang berada pada kategori ini berfungsi sebagai yang menyatakan masukan dan hasil dari sistem. Pada Tabel 2.4 dibawah ini mepresentasikan simbol beserta fungsingnya dari masing-masing simbol yang ada pada kategori *input/output symbols*.

Tabel 2. 4 *Input-Output Symbols*

| Simbol | Nama | Fungsi |
|---|----------------------------|---|
|  | <i>Input/Output</i> | Berfungsi dalam menyatakan input atau output. |
|  | <i>Punched Card</i> | Berfungsi dalam menyatakan input atau output berasal dari card. |
|  | <i>Disk Storage</i> | Berfungsi dalam menyatakan input atau output berasal dari disk. |
|  | <i>Magnetic Tape</i> | Berfungsi dalam menyatakan input atau output berasal dari magnetis. |
|  | <i>Document</i> | Berfungsi dalam menyatakan input atau output berasal dari dokumen. |
|  | <i>Manual Input Symbol</i> | Berfungsi untuk menampilkan output kedalam layar. |

2.1.4 Python

Python adalah salah satu bahasa pemrograman yang populer saat ini, hampir semua *programmer* pasti mengetahui Bahasa pemrograman ini. *Python* merupakan salah satu bahasa pemrograman berbasis objek dengan interaksi interaktif [24]. Python pertama kali ditemukan di Amsterdam oleh Guido van Rossum pada tahun 1991 [25]. *Python* dianggap sebagai bahasa pemrograman yang menggabungkan kemampuan dan kejelasan sintaks kode. Dibuat untuk membuat programmer lebih mudah membuat program dengan waktu yang lebih sedikit, pengembangan yang mudah, dan kompatibilitas dengan system [24]. *Python* sendiri merupakan sebuah bahasa pemrograman yang dapat digunakan dalam berbagai pengembangan seperti aplikasi *web*, aplikasi *desktop*, *Internet of Things*, *automation*, dan sebagainya [26]. Tidak hanya itu, *python* juga sering kali digunakan oleh peneliti secara luar, hal tersebut

dikarenakan *python* memiliki kemampuan dalam mengatasi data dengan jumlah data yang besar dan perhitungan matematika yang kompleks [26]. Bahasa pemrograman *python* sendiri memiliki keunggulan-keunggulan, antara lain [21]:

- 1) *Python* merupakan bahasa pemrograman yang dapat dioperasikan diberbagai system operasi seperti windows, macOS, raspberry, linux, dan sebagainya.
- 2) *Python* memiliki struktur penulisan *syntax* atau *code* yang lebih ringkas dibandingkan bahasa pemrograman yang lainnya.
- 3) Bahasa pemrograman *python* memiliki kemiripan dengan bahasa Inggris.
- 4) *Python* mendukung paradigma pemrograman seperti pemrograman berorientasi objek, fungsional, dan pemrograman prosedural. Fleksibilitas ini memungkinkan pengembang memilih pendekatan yang sesuai dengan kebutuhan proyek.
- 5) *Python* dapat dilakukan eksekusi secara cepat, hal tersebut dikarenakan menggunakan interpreter.

2.3 Teori tentang *Framework* / Algoritma yang digunakan

2.2.1 *Selenium*

Selenium merupakan sebuah framework otomatisasi pengujian open-source yang secara khusus dikembangkan untuk mengotomatiskan interaksi pengguna dengan aplikasi web [27]. *Selenium* sendiri dapat digunakan oleh beberapa bahasa pemrograman seperti *Python*, Java, C#, Ruby, dan sebagainya. Mengutip dari halaman resmi *selenium*, dimana *selenium* sendiri pertama kali dikembangkan pada tahun 2004 oleh Jason Huggins, dengan tujuan awalnya hanya untuk pengujian internal terhadap aplikasi berbasis web yang sedang dikembangkan [28]. *Framework selenium* sendiri terbagi kedalam beberapa jenis, antara lain [28],[29],[30]:

1) *Selenium WebDriver*

Selenium WebDriver adalah komponen inti yang memberikan antarmuka langsung ke browser. Digunakan untuk mengotomatiskan interaksi dengan halaman web, *WebDriver* mendukung berbagai bahasa

pemrograman dan memungkinkan pengembang membuat skrip pengujian yang fleksibel dan dapat diandalkan [29].

2) *Selenium* IDE

Selenium IDE adalah alat otomatisasi pengujian yang berfungsi sebagai plugin untuk browser. IDE memungkinkan pengguna merekam, memutar kembali, dan mengedit skenario pengujian tanpa penulisan kode secara manual, membuatnya cocok untuk pemula dan pengujian skenario sederhana [30].

3) *Selenium* Grid

Selenium Grid memfasilitasi pengujian distribusi dengan memungkinkan eksekusi tes secara paralel pada beberapa mesin dan browser. Ini mempercepat waktu pengujian dan mengelola sumber daya secara efisien, memastikan pengujian dapat dilakukan di berbagai konfigurasi [31].

2.2.2 *Playwright*

Playwright merupakan sebuah *framework* bersifat *open source* yang dapat digunakan untuk melakukan proses *automation web browser* [9]. *Playwright* sendiri *framework* yang dikembangkan oleh Microsoft untuk melakukan otomatisasi pengujian aplikasi *web*. *Playwright* sendiri merupakan *framework automation* yang tergolong baru, dikarenakan Microsoft merilis versi pertamanya pada tahun 2020. Implementasi *playwright* dapat dilakukan berbagai cara dengan bahasa pemrograman seperti JavaScript, *Python*, dan Java, .NET, dan TypeScript [20]. Selain dapat dioperasikan oleh berbagai bahasa pemrograman, *playwright* juga mendukung berbagai jenis browser seperti Chromium, Firefox, Safari, dan Edge, Internet Explorer, sehingga memungkinkan pengembang membuat rangkaian pengujian yang dapat berjalan di browser dan sistem operasi berbeda [20].

Kemampuan dalam melakukan *headless testing* yang membuat *automation* akan berjalan secara otomatis tanpa memperlihatkan jendela

browser atau dalam kata lain berjalan di latar belakang sistem. Perangkat *playwright* menawarkan infrastruktur pengujian komprehensif yang mencakup dukungan untuk berbagai kerangka pengujian seperti Jest dan Mocha [20]. Kerangka tersebut menunjukkan integrasi tanpa batas dengan alat Continuous Integration (CI) yang sering digunakan seperti Jenkins dan Travis CI. *Framework playwright* menawarkan fungsionalitas perekaman video terintegrasi yang dapat menangkap secara komprehensif seluruh proses pelaksanaan pengujian, mencakup *screenshots, network traffic, and console logs*. Proses yang disederhanakan memfasilitasi peninjauan yang cepat hasil pengujian oleh pengembang, sehingga memungkinkan identifikasi potensi masalah dengan cepat [20].

2.2.3 Black Box

Pengujian aplikasi sangat penting dalam siklus pengembangan perangkat lunak karena didasarkan pada prinsip dasar keandalan dan kualitas [32]. Proses pengujian memungkinkan pengembang memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasinya, memberikan pengalaman pengguna yang lancar, dan mengurangi kerusakan [32]. Dalam pengujian sebuah aplikasi atau *software* yang telah dikembangkan ada beberapa pendekatan yang dapat dilakukan, salah satunya adalah dengan menggunakan metode *black box*. Black box testing adalah metode pengujian perangkat lunak di mana pengujian dilakukan tanpa memperhatikan struktur internal atau logika kode sumber aplikasi yang diuji [33]. Tujuan dari black box testing adalah untuk mengevaluasi apakah aplikasi berperilaku sesuai dengan ekspektasi pengguna dan apakah fungsionalitasnya sesuai dengan spesifikasi yang telah ditetapkan, tanpa memeriksa bagaimana program mencapai hasil tersebut [33]. *Black Box* sendiri dapat menemukan beberapa kategori kesalahan, seperti: [34]

- 1) Menemukan fungsionalitas yang memiliki kesalahan
- 2) Menemukan kesalahan dalam performa.
- 3) Menemukan kesalahan dalam inisialisasi dan terminasi.

Selain itu terdapat juga terdapat juga keuntungan dalam pengujian dengan menggunakan pendekatan model *black box*, antara lain: [11]

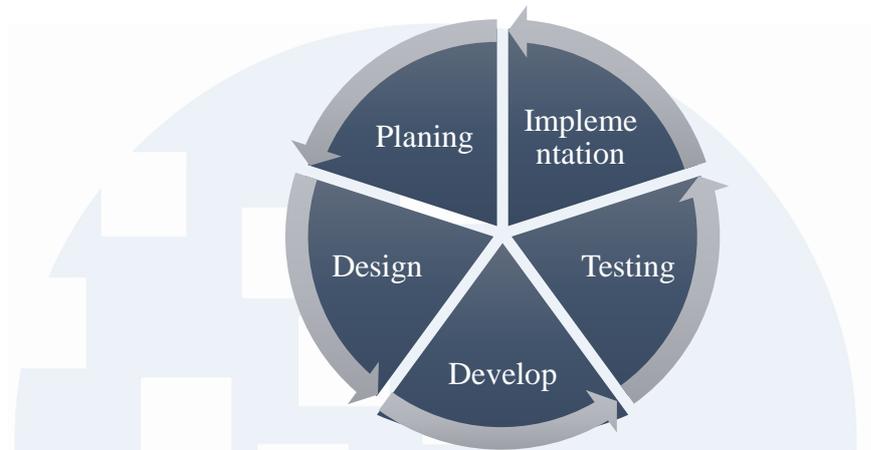
- 1) Dalam pengujian tidak diperlukan pengetahuan mengenai bahasa pemrograman.
- 2) Dalam pengujian sendiri dilakukan dari sudut pandang pengguna, sehingga hal tersebut dapat menemukan ketidakjelasan dalam persyaratan spesifikasi.
- 3) Dalam pengujian tentunya antara *programmer* dan *tester* saling bergantung satu sama lain.

2.2.4 User Acceptance Testing (UAT)

Aplikasi yang telah dikembangkan tentunya harus dilakukan pengujian terlebih dahulu dengan pengguna atau *user* dari sistem atau aplikasi tersebut, pengujian tersebut dinamai dengan *User Acceptance Testing (UAT)* [35]. *User Acceptance Testing (UAT)* adalah sebuah serangkaian pengujian yang dilakukan oleh pengguna terakhir dari aplikasi atau system yang dibuat, dengan bertujuan agar melakukan pengecekan apakah fungsinya sudah berjalan dan sesuai dengan kebutuhan yang inginkan [36]. Sementara itu terdapat pengertian lainnya, bahwa *User Acceptance Testing (UAT)* merupakan sebuah pengujian yang dilakukan terhadap suatu system yang dilakukan oleh penggunanya, dimana dokumen dibuat untuk menunjukkan bahwa pengguna atau pengguna menerima pengembangan aplikasi dan menganggap hasil uji telah memenuhi kebutuhan pengguna [37]. Jadi dapat disimpulkan bahwa *User Acceptance Testing (UAT)* merupakan sebuah rangkaian pengujian yang dilakukan oleh user terhadap system dengan tujuan untuk melakukan pengecekan untuk memastikan bahwa fungsinya sudah sesuai dengan kebutuhan [36].

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

2.2.5 Agile Software Development



Gambar 2. 1 Agile Software Development

Sumber: [38], [39]

Agile software development adalah sebuah metode yang digunakan dalam pengembangan perangkat lunak yang cepat, dan dapat memungkinkan proses pengembangan untuk berjalan secara fleksibel tanpa mengganggu proses pengembangan lainnya [38]. Metode agile dapat membuat keputusan tentang perubahan *software* sesuai dengan kondisi yang diinginkan, sehingga *software* yang dihasilkan dengan metode ini akan lebih efektif dan efisien. Seperti yang terlihat pada Gambar 2.2 dimana menjelaskan alur dari metode pengembangan *agile*, dimana terdapat lima tahapan alur dimulai dari *planning* hingga *implementation*, dan berikut merupakan penjelasan setiap tahapannya. [38], [39]

1) *Planning*

Tahap perencanaan dalam metodologi Agile adalah langkah awal dalam menggambarkan visi dan tujuan pengembangan. Tim bersama-sama mengidentifikasi pekerjaan yang perlu dilakukan, mendefinisikan user stories atau fitur yang akan diimplementasikan, dan menetapkan prioritas. Perencanaan iteratif, seperti sprint planning, membantu tim untuk memfokuskan upaya pada tujuan tertentu selama setiap iterasi. Ini memberikan dasar yang jelas untuk memahami ruang lingkup proyek dan menetapkan ekspektasi yang realistis

2) *Design*

Tahap perancangan memusatkan perhatian pada merancang solusi yang memenuhi kebutuhan yang telah diidentifikasi. Tim bekerja untuk membuat desain arsitektur yang kokoh, merinci spesifikasi teknis, dan merancang solusi secara menyeluruh. Dengan fokus pada aspek perancangan, tim dapat memastikan bahwa solusi yang dikembangkan akan memenuhi persyaratan dan memiliki fondasi teknis yang solid.

3) *Develop*

Pada tahap ini, tim pengembangan menerjemahkan desain yang telah dibuat ke dalam kode nyata. Melibatkan proses menulis dan menguji kode untuk memastikan bahwa setiap komponen berfungsi dengan benar. Pengembangan berdasarkan prinsip-prinsip *clean code* dan pengujian terus-menerus membantu memastikan kualitas dan konsistensi dalam kode yang dihasilkan.

4) *Testing*

Tahap pengujian berfokus pada memastikan bahwa solusi yang dikembangkan berfungsi dengan benar dan memenuhi persyaratan pengguna. Tim melakukan pengujian unit untuk setiap komponen, pengujian integrasi untuk memastikan keterhubungan yang benar antar komponen, dan pengujian fungsional untuk memverifikasi bahwa sistem berperilaku sesuai dengan spesifikasi. Pengujian berkelanjutan memastikan bahwa setiap perubahan diuji secara menyeluruh.

5) *Implementation*

Implementasi adalah tahap di mana solusi yang telah diuji dan disetujui siap untuk digunakan oleh pengguna akhir. Tim meluncurkan solusi ke lingkungan produksi dan melibatkan pemantauan serta pemeliharaan berkelanjutan. Pelatihan pengguna mungkin diperlukan untuk memastikan pemahaman yang baik tentang cara menggunakan solusi. Tahap ini merupakan langkah akhir dalam siklus hidup pengembangan sebelum memasuki iterasi berikutnya.

2.4 Teori tentang Tools / Software yang digunakan

2.3.1 Microsoft Visual Studio Code

Visual Studio Code (VS Code) adalah editor kode sumber terbuka dan lintas platform yang dikembangkan oleh Microsoft. Dirancang untuk memenuhi kebutuhan pengembang perangkat lunak modern, VS Code menonjol dengan antarmuka yang bersih, ringan, dan sangat dapat disesuaikan [40]. Dilengkapi dengan dukungan penuh untuk sejumlah besar bahasa pemrograman, ekstensi, dan integrasi dengan Git, VS Code memungkinkan pengembang untuk menulis kode dengan produktivitas tinggi. Kelebihan utamanya mencakup fitur IntelliSense yang cerdas, debugging langsung dari editor, serta ekosistem ekstensi yang kaya, menjadikannya pilihan populer di kalangan pengembang untuk proyek-proyek yang beragam, mulai dari pengembangan web hingga aplikasi cloud dan pengembangan lintas platform [41].

2.3.2 Microsoft Excel

Microsoft Excel adalah program *spreadsheet* yang dikembangkan oleh Microsoft, yang menjadi salah satu alat produktivitas paling populer di dunia bisnis dan personal [42]. Excel memungkinkan pengguna untuk membuat, mengedit, dan mengelola lembar kerja yang terdiri dari sel-sel terorganisir dalam baris dan kolom. Fungsionalitasnya meliputi pengolahan data, perhitungan matematis, pembuatan grafik, dan analisis statistik. Dengan rumus, fungsi, dan alat analisis data yang kuat, Excel memberikan kemampuan untuk membuat laporan keuangan, grafik, dan tabel pivot untuk mendukung pengambilan Keputusan [42]. Selain itu, integrasi Excel dengan aplikasi Microsoft Office lainnya, seperti Word dan PowerPoint, memudahkan pertukaran dan penggunaan data di seluruh platform. Excel telah menjadi alat yang tak tergantikan untuk mengelola informasi, membuat perkiraan, dan menyajikan data dengan cara yang intuitif dan efisien [42].