

## BAB I. PENDAHULUAN

### 1.1 Latar Belakang

Pada era teknologi yang berkembang pesat saat ini, banyak sistem informasi baik berbasis *web* maupun *mobile* yang diluncurkan untuk memberikan kemudahan dan *fleksibilitas* dalam pekerjaan manusia (Danuri, 2019). *Automation testing* menjadi sangat penting untuk menemukan kesalahan yang mungkin terjadi akibat dari pengembangan perangkat lunak (Rizky, 2019), tanpa dilakukannya *automation testing* perangkat lunak dapat memiliki *bug*, kesalahan, atau masalah yang tidak terdeteksi (Herlinda et al., 2019). Hal ini dapat mengakibatkan kualitas yang buruk dan pengalaman pengguna yang tidak memuaskan, rentan terhadap kegagalan atau *crash*, sehingga dapat mengganggu penggunaan normal dan mengakibatkan kerugian finansial.

*Automation testing* berperan penting dalam memastikan kualitas sistem informasi yang sedang dikembangkan (Rizky, 2019). *Automation testing* berbeda dari pengujian manual karena melibatkan penggunaan skrip pengujian yang telah disusun sebelumnya. Keuntungan dari *automation testing* dibandingkan dengan pengujian manual dari kemampuannya untuk dijalankan berulang kali dengan biaya yang lebih murah dan kecepatan yang lebih baik. Saat ini, pengujian otomatis menjadi sangat penting dalam proses pengembangan perangkat lunak (Wang, 2018). Melakukan *automatic testing* pada setiap komponen perangkat lunak, pengembang dapat menemukan dan memperbaiki *bug*, kesalahan, atau masalah lainnya sebelum, sistem tersebut digunakan oleh pengguna (Andry & Reinaldo, 2017). Dengan demikian, *automation testing* membantu mencegah terjadinya kegagalan sistem yang dapat berdampak pada kerugian finansial atau reputasi yang buruk.

Sistem informasi Indo Towing merupakan *platform website* untuk memudahkan pemesanan pelanggan. Berbagai fungsi seperti pemesanan *online* dan pengecekan tarif towing di seluruh Indonesia dapat diakses dengan mudah melalui *platform* ini. Namun pada *platform website* belum dilakukan *automation testing* untuk sistem salah satu proses yang tidak boleh terlewat dalam proses pengembangan sistem, hal tersebut digunakan untuk mengetahui kualitas dari sistem. Jika *automation testing*

tidak dilakukan, pengembang tidak akan dapat mengetahui kualitas sistem, yang dapat menyebabkan terjadinya kesalahan yang merugikan sistem tersebut. Kualitas sistem pengukuran proses sistem informasi yang berfokus pada hasil interaksi antara pengguna dan sistem.

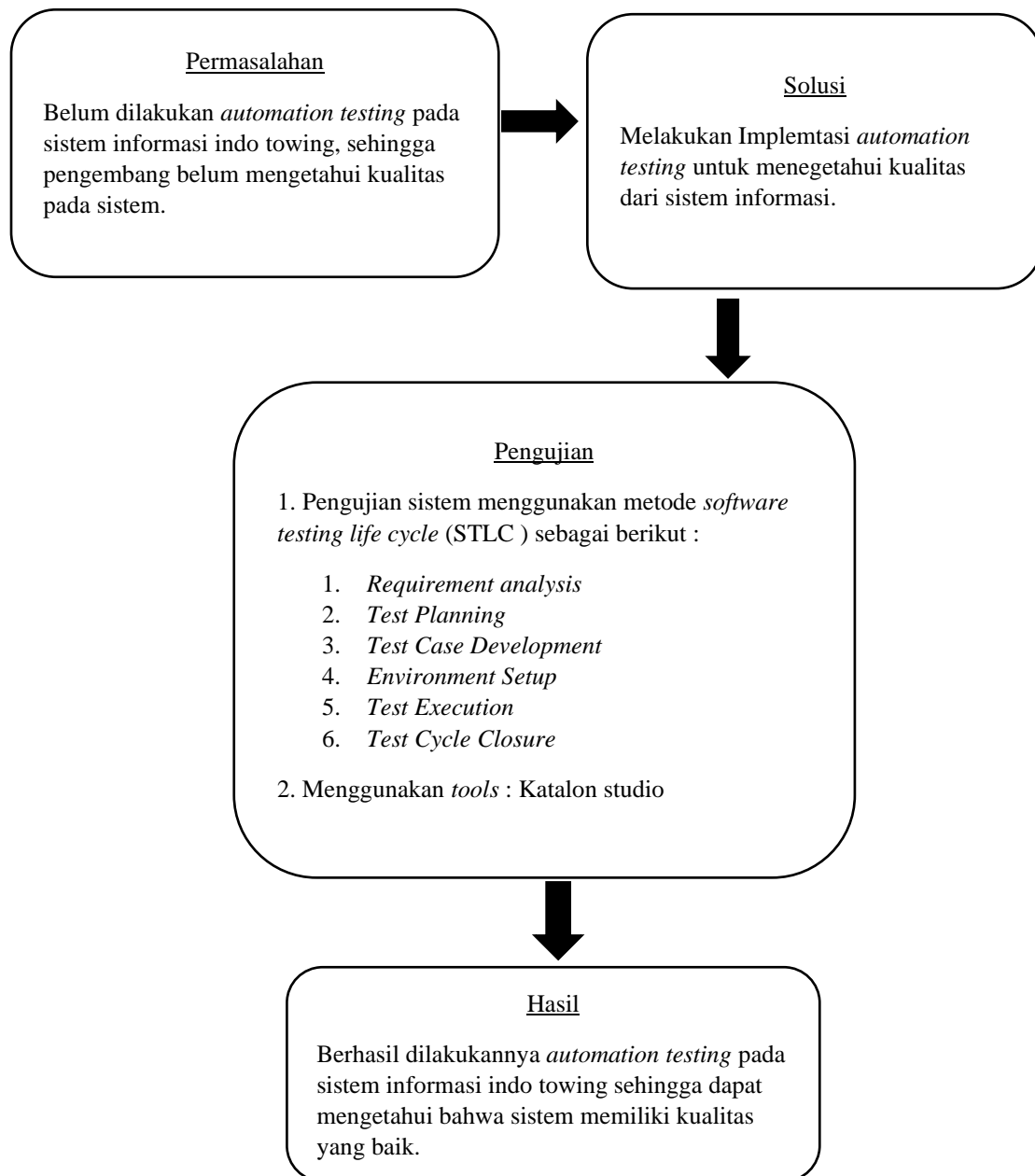
Berdasarkan permasalahan yang ada, maka dibutuhkan *automation testing* sistem untuk mendukung keberhasilan sebuah *web* yang dapat mengetahui kualitas sistem. Solusi dari masalah tersebut maka dilakukan *automation testing* pada sistem informasi indo towing menggunakan metode STLC dengan teknik *equivalence partitioning*. Pengerjaan *automation testing* pada sistem informasi indo towing menggunakan *tools* bernama Katalon Studio merupakan perangkat lunak dalam proses *automation testing* dalam pengembangan perangkat lunak, tersedia secara gratis dengan mendukung pengujian *web*, desktop hingga seluler (Prakoso, B., & Sujarwo, A.,2022).

## **1.2 Tujuan**

Adapun tujuan dari tugas akhir ini yaitu untuk dapat menerapkan *automation testing* pada sistem informasi indo towing dan dapat mengetahui keberhasilan sebuah sistem berjalan dengan tugas program tersebut serta tidak memiliki *bug* pada sistem, tanpa mengetahui kode program yang digunakan.

### 1.3 Kerangka Pemikiran

Berdasarkan latar belakang yang ada, dapat ditarik kesimpulan bahwa terdapat suatu kerangka pemikiran yang diilustrasikan dalam gambar 1.



Gambar 1. Kerangka Pemikiran

#### **1.4 Kontribusi**

Dengan dilakukannya implementasi *automation testing* pada sistem informasi indo towing menggunakan Katalon Studio. Diharapkan bisa memberikan kontribusi sebagai berikut :

1. Dapat membantu memastikan kualitas sistem berjalan dengan baik.
2. Dengan melakukan *automation testing* yang lebih efektif, sistem informasi dapat dikirim dengan sukses serta dapat membantu mengidentifikasi masalah, meminimalkan resiko kegagalan serta memastikan sistem informasi dapat digunakan oleh pengguna.

## **BAB II. TINJAUAN PUSTAKA**

### **2.1 Pengujian Perangkat Lunak**

Pengujian perangkat lunak sebagai langkah untuk menjalankan suatu program dengan maksud untuk menemukan masalah yang mungkin ada. Sebuah tes dianggap berhasil jika dapat menemukan kesalahan yang sebelumnya tidak terdeteksi melalui tes yang telah dilakukan sebelumnya (Ahrizal et al., 2023). Tujuan dari pengujian perangkat lunak adalah menemukan *bug* untuk mendapatkan produk berkualitas dengan kinerja tinggi.

### **2.2 Implementasi**

Implementasi merupakan suatu proses melakukan tindakan yang bertujuan untuk mencapai tujuan yang ditetapkan dalam keputusan. Tujuan dari tindakan ini adalah untuk mengubah keputusan ke dalam bentuk operasional dan mencoba membawa perubahan sesuai dengan yang telah diputuskan sebelumnya, baik dalam skala besar maupun kecil. Implementasi juga mencakup pemahaman tentang apa yang harus terjadi setelah program implementasi dilakukan (Mulyati & Watini, 2022). Implementasi melibatkan langkah lebih spesifik untuk menerapkan keputusan dan mencapai hasil yang diharapkan.

### **2.3 Automation Testing**

*Automation testing* sebagai teknik pengujian perangkat lunak yang secara otomatis tanpa campur tangan manual, hanya dengan merekam dan memutar ulang kasus uji. Ini memungkinkan pengujian/penguji untuk mengulangi pekerjaan mereka pada setiap siklus pengujian (Prabowo et al., 2023).

Dalam metode pengujian tersebut, pengujian melakukan akan memasukkan data uji ke dalam sistem pengujian dan membandingkan hasilnya dengan yang diharapkan. Tujuan dari pengujian otomatisasi untuk meningkatkan efisiensi dan efektivitas pengujian perangkat lunak.

## 2.4 Sistem Informasi

Sistem informasi suatu sistem yang terdiri dari berbagai komponen, seperti perangkat lunak, perangkat keras, dan sumber daya manusia, yang bekerja sama untuk mengolah informasi menjadi hasil yang bermanfaat guna mencapai tujuan tertentu dalam suatu organisasi (Mulyanto et al., 2017). Sistem informasi bertujuan untuk mengelola informasi dengan efisien dan efektif.

## 2.5 Indo Towing

Indo towing merupakan jasa towing. Layanan towing ini memberikan solusi alternatif untuk mengantarkan kendaraan menggunakan kendaraan khusus sehingga proses lebih efisien dalam mencapai tujuan. bekerja sama dengan mitra transportasi yang andal dan terpercaya, percaya bahwa dapat dengan aman mengantarkan kendaraan ke tujuan.

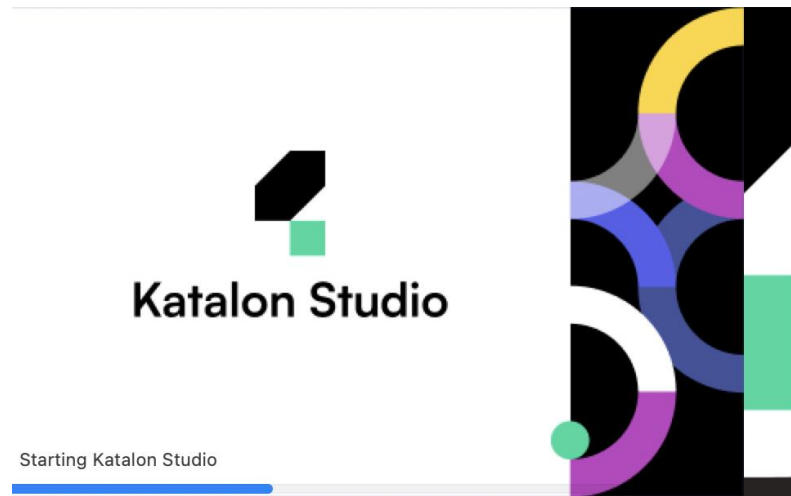


Gambar 2. Indo towing

## 2.6 Katalon Studio

Katalon Studio adalah sebuah solusi otomatis yang terjangkau, karena didasarkan pada sumber terbuka (*open source*) dan dirancang dengan kesederhanaan yang maksimal. Dengan adanya Katalon Studio, diharapkan dapat membantu para profesional maupun pemula dalam bidang QA (*Quality Assurance*) untuk melakukan pengujian dengan mudah. Katalon Studio mencakup berbagai jenis pengujian, seperti *Web UI*, *API*, *Mobile*, dan *Scripting* (M. M. Muhtadi, 2019). Katalon Studio

memungkinkan pembuatan pengujian otomatis dengan cepat, yang sangat membantu para penguji dalam pekerjaan mereka di berbagai *platform*.



Gambar 3. Katalon studio

### 2.6.1 *Test Case*

*Test case* adalah metode pengujian perangkat lunak yang menggunakan serangkaian skenario eksekusi untuk pengujian sistem apakah sistem informasi yang dikembangkan memenuhi standar kualitas yang diinginkan. Pada dasarnya, pengujian tidak dapat membuktikan kebenaran semua kemungkinan yang terjadi selama eksekusi *test case*, pendekatan yang tepat dalam perencanaan dan perancangan *test case* dapat memastikan bahwa kinerja perangkat lunak sampai batas tertentu seperti yang diharapkan (Hasibuan & Dirgahayu, 2021). Dengan demikian, *test case* dapat memberikan jaminan efektivitas perangkat lunak hingga tingkat tertentu sesuai dengan harapan yang diinginkan *test case* biasanya dibuat berdasarkan pada analisis kebutuhan dan desain sistem, dan digunakan oleh tim pengujian untuk menguji fungsionalitas, keandalan, dan kinerja sistem.

### 2.6.2 *Test Suites*

*Test suite* kumpulan tes kasus yang telah disiapkan untuk diuji secara bersamaan dengan data. Digunakan untuk memastikan bahwa semua aspek yang relevan telah diuji, dan memudahkan dalam mengidentifikasi dan memperbaiki masalah yang mungkin muncul (Ardi, 2021). *Test suites* biasanya dibuat berdasarkan pada persyaratan dan desain sistem, dan dapat mencakup beberapa skenario pengujian yang

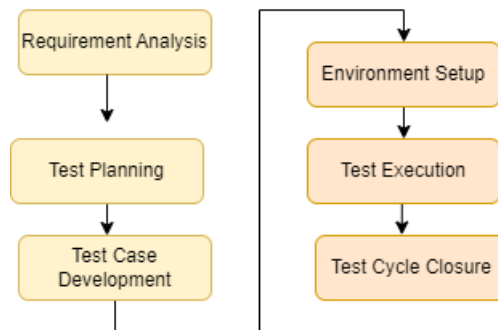
berbeda. Mereka membantu dalam mengelola dan melacak proses pengujian dengan lebih baik, memastikan bahwa semua aspek yang relevan telah diuji, dan memudahkan dalam mengidentifikasi dan memperbaiki masalah yang mungkin muncul.

### 2.7 Metode *Equivalence Partitioning* (EP)

Metode *Equivalence Partitions* adalah metode pengujian yang melibatkan penginputan data ke dalam setiap formulir dalam sistem pemilihan terbaik. Setiap menu *input* akan diuji dan dikategorikan berdasarkan fungsinya, apakah *valid* atau tidak *valid* (Hidayat & Muttaqin, 2018). Asumsi ini digunakan untuk satu kelas mewakili nilai *input* pada seluruh kelas ekuivalensi.

### 2.8 *Software Testing Life Cycle* (STLC)

*Software Testing Life Cycle* (STLC) merupakan rangkaian aktivitas yang dilakukan untuk menguji perangkat lunak (Min et al., 2020). STLC sebagian dari siklus hidup pengembangan perangkat lunak. STLC memiliki proses yang terdiri dari beberapa tahap, setiap tahap memiliki pilihan dan hasil yang telah ditemukan sebelumnya (Bhor et al., 2018). Dalam STLC, setiap kegiatan dilakukan secara terencana dan sistematis. Tahapan-tahapan dalam STLC meliputi seperti gambar berikut ini :



Gambar 4. *Software Testing Life Cycle*

Sumber : (Bhor et al., 2018)



Proses yang terlibat dalam STLC terdiri dari beberapa tahap, setiap tahap memiliki kriteria dan hasil yang telah ditetapkan :

*A. Requirement Analysis*

*Requirement Analysis* merupakan tahap pertama, yang dilakukan dalam pengembangan perangkat lunak untuk memahami kebutuhan dan persyaratan pengguna.

*B. Test Planning*

*Test Planning* merupakan tahapan dalam perencanaan yang dilakukan dalam pengujian perangkat lunak.

*C. Test Case Development*

*Test Case Development* pada tahap ini proses yang dilakukan dalam pengujian perangkat lunak untuk merancang dan mengembangkan kasus uji yang akan digunakan dalam pengujian.

*D. Environment Setup*

*Environment Setup* merupakan tahapan yang dilakukan dalam pengembangan perangkat lunak untuk menyiapkan lingkungan yang diperlukan agar perangkat lunak dapat dijalankan dan diuji dengan baik.

*E. Test Execution*

*Test Execution* adalah tahap dalam siklus pengujian perangkat lunak di mana test case yang telah dikembangkan dijalankan pada sistem yang sedang diuji.

*F. Test Cycle Closure*

*Test Cycle Closure* merupakan tahapan akhir dari siklus pengujian di mana semua aktivitas terkait pengujian dievaluasi dan dilaporkan.

## **2.9 Bug**

*Bug* merupakan kegagalan dalam perangkat lunak yang menyebabkan program tersebut tidak berjalan sesuai yang diharapkan (Noviansyah & Saiyar, 2023). *Bug* pada *software* ataupun *hardware* dapat menyebabkan sistem tidak dapat dijalankan dengan sebagaimana mestinya.

## **2.10 Internet**

Internet adalah suatu sistem global yang menghubungkan komputer-komputer melalui jaringan komunikasi seluruh dunia, tanpa memandang perbedaan sistem

operasi dan perangkat yang digunakan. Internet juga berfungsi sebagai penghubung antara berbagai jaringan di seluruh dunia, dan menggunakan protokol komunikasi yang sama, yaitu TCP/IP (*Transmission Control Protocol/ Internet Protocol*) (Nofyat, 2018).

### **2.11 Web Browser**

*Web Browser* merupakan halaman situs sistem informasi yang dapat diakses secara cepat. Pengelolaan data sesuai dengan layar pengguna dan dikembangkan lagi menjadi lebih responsif (Komang et al., 2022).

### **2.12 Website**

*Website* adalah sekumpulan halaman yang digunakan untuk menampilkan informasi dalam bentuk teks, gambar, animasi diam atau bergerak, suara, dan kombinasi dari elemen-elemen tersebut. Halaman-halaman ini dapat memiliki sifat statis atau dinamis, dan saling terkait serta terhubung melalui jaringan internet (Rivanthio, 2020).

### **2.13 Penelitian Jurnal Terkait**

Jurnal yang terkait dalam penyusunan tugas akhir ini merupakan referensi yang dapat digunakan sebagai contoh dalam menyusun tugas akhir. Terdapat beberapa sumber referensi yang dapat dijadikan contoh, seperti buku, jurnal, artikel, dan lainnya. Berikut ini adalah contoh jurnal terkait yang dapat digunakan sebagai referensi.

Jurnal yang disusun oleh Virginia Tirza Tempomona dan Hendry (2022) dengan judul “Penerapan Metode *Black box* Pada Perangkat Lunak Menggunakan Katalon Studio (Studi Kasus: Aplikasi Absensi Di PT. Astra Sedaya Finance)”. Dalam jurnal ini berhasil menerapkan *automation testing* menggunakan katalon studio pada aplikasi absensi sesuai dengan skenario dan data yang telah dibuat.

Jurnal yang disusun oleh Yulianti, Teti Desyani, Redo Ramadham Chaniago, Hadi Iswanto, Eko Suroso, Taufik Dan Surya Hermanto (2022) dengan judul “Pengujian Aplikasi Sistem Informasi Akademik Berbasis Website Menggunakan Teknik *Equivalence Partitioning* dan Metode *Black Box*”. Dalam jurnal ini bertujuan untuk menemukan kesalahan pada *interface* tanpa mengetahui kode program sistem.

Jurnal yang disusun oleh Ibnu Adha Shaleh, Juma Prayogi, Perdi Pirdaus, Riky Syawal, Aries dan Saifudin (2021) dengan judul “Pengujian *Black Box* pada Sistem Informasi Penjualan Buku Berbasis *Web* dengan Teknik *Equivalent Partitions*”. Dalam jurnal ini tidak ditemukan kesalahan fungsi yang ada pada program dengan demikian program sudah berjalan dengan apa yang diharapkan.

Jurnal yang disusun oleh Eland Vaskita Nugraha, Renna Yanwastika Ariyanna, dan Erna Kumalasari Nurnawati, (2022) dengan judul “Uji Black Box Tes Aplikasi *Software Development System Information* (Sodevi) PT. Dimata Sora Jayate Menggunakan Katalon Studio”. Dalam jurnal ini menggunakan metode sistem berjalan sesuai dengan fungsi dan tidak ditemukan *bug* atau *error* dari sistem.

Tabel 1. Jurnal Terkait

Nama dan Tahun	Judul	Hasil
(1)	(2)	(3)
Virginia Tirza Tempomona dan Hendry (2022)	Penerapan metode black box pada perangkat lunak menggunakan katalon studio(studi kasus:aplikasi absensi di pt.astra sedaya.	Diterapkan metode balckbox pada aplikasi absensi pt.astra sedaya.
Yulianti,Teti Desyani,Redo Ramadhan Chaniago,Hadi Iswanto,Eko Suroso dan Taufik Surya Hermanto (2022)	Pengujian Aplikasi Sistem Informasi Akademik Berbasis Website Menggunakan Teknik <i>Equivalence Partitioning</i> dan Metode <i>Black Box</i>	Berhasil melakukan pengujian pada sistem aplikasi unfirmasi akademik menggunakan teknik <i>equivalence partitioning</i>
Ibnu,Adha Shaleh,Juma Prayogi,Perdi Pirdaus,Riky Syawal dan Aries Saifudin (2021)	Pengujian <i>Black Box</i> pada Sistem Informasi Penjualan Buku Berbasis <i>Web</i> dengan Teknik <i>Equivalent Partitions</i> .	Pengujian <i>Black box</i> berhasil dilakukan pada sistem infromasi penjualan.
Eland vaskita nugraha , rena yanwastika ariyana, dan erna kumalasari nurnawat (2021)	Uji Black Box Tes Aplikasi Software Development System Information (Sodevi) PT. Dimata Sora Jayate Menggunakan Katalon Studio	Berjalan sesuai dengan fungsi dan tidak ditemukan bug atau error dari Software Development System Information (Sodevi) PT. Dimata Sora Jayate.