

# I. PENDAHULUAN

## 1. Latar Belakang

Pengujian adalah proses yang dilakukan untuk mengidentifikasi kesalahan atau masalah dalam sebuah produk atau sistem dengan tujuan untuk memastikan bahwa produk atau sistem tersebut berfungsi sesuai dengan spesifikasi yang telah ditentukan dan memenuhi kebutuhan. Aktivitas pengujian ini dilakukan selama pengembangan perangkat lunak atau setelah sistem telah selesai dikembangkan. Tujuan utama pengujian adalah memastikan bahwa sistem memenuhi standar kualitas yang telah ditetapkan dan memastikan bahwa produk tersebut berfungsi dan memenuhi kebutuhan. Dengan mengidentifikasi masalah sejak dini dalam siklus pengembangan, pengujian membantu mengurangi resiko masalah pada sistem yang akan muncul saat sistem telah digunakan.

Pengujian website adalah langkah penting dalam pengembangan dan pemeliharaan situs *website* untuk memastikan bahwa situs tersebut berfungsi dengan baik, aman dan dapat memberikan pengalaman memuaskan kepada pengguna. Ada berbagai jenis pengujian yang dapat dilakukan pada sebuah website seperti pengujian fungsional, pengujian *responsive*, pengujian kinerja, pengujian *navigasi* dan *usability*, pengujian perbandingan, pengujian *integrasi*, pengujian manajemen konten, pengujian keamanan. Pengujian aplikasi adalah proses penting dalam pengembangan perangkat lunak yang bertujuan untuk memastikan bahwa aplikasi komputer, perangkat lunak, atau aplikasi mobile berfungsi dengan baik dan aman. Pengujian aplikasi melibatkan sejumlah aktivitas yang dilakukan untuk mengidentifikasi masalah, memvalidasi fungsi dan memastikan kualitas keseluruhan dari suatu aplikasi (Pressman, 2010).

Pengujian aplikasi salah satu komponen penting untuk memastikan kualitas aplikasi dari spesifikasi desain program aplikasi. Perkembangan teknologi informasi saat ini mengembangkan berbagai aplikasi – aplikasi penunjang dibidang pendidikan. Salah satu manfaat teknologi informasi dalam bidang pendidikan adalah sebagai sarana pembelajaran.

Pengujian aplikasi salah satu kegiatan yang memakan banyak usaha dan biaya, Populasi dari pengujian merupakan campuran antara pengujian manual dan pengujian otomatis pada semua bagian. Yang paling utama untuk melakukan analisis survey adalah untuk kepuasan pelanggan. Dalam penemuan *bug* pada aplikasi hal ini berdampak pada kepuasan pelanggan.

Pengujian aplikasi dapat dilakukan secara manual atau otomatis, dan merupakan proses mengevaluasi semua komponen sistem dan memverifikasi bahwa sistem memenuhi persyaratan yang telah ditentukan atau untuk mengklasifikasikan perbedaan hasil yang diharapkan (Bhateja, 2015). Pengujian adalah proses mengidentifikasi semua *bug* pada aplikasi. Proses pengujian yang dilakukan dilakukan secara manual oleh manusia, mencoba berbagai kombinasi kegunaan dan input, lalu membandingkan hasilnya dengan perilaku yang diharapkan, dan mencatat secara manual (R.M. Sharma, 2014). Pengujian manual dilakukan dengan membuat kasus uji secara manual dan menjalankannya tanpa bantuan alat apa pun. pengujian otomatis saat ini merupakan metode terbaik untuk menguji area fungsional pengujian yang luas dari program aplikasi dan mengurangi jumlah waktu dan biaya yang diperlukan untuk pengujian aplikasi.

Pengujian otomatis telah banyak digunakan pengujian manual untuk mempercepat proses menemukan dan memperbaiki *bug*, memungkinkan pengujian berulang dan memberikan umpan balik yang lebih cepat, yang sulit dilakukan oleh manusia. Melakukan pengujian secara manual dapat memakan waktu dan rawan kesalahan. Karena hampir tidak mungkin untuk menghasilkan aplikasi yang bebas kesalahan secara terprogram tanpa melakukan pengujian *unit*, pengujian manual sering diabaikan karena tekanan jadwal meningkat, yang mengakibatkan aplikasi cacat setelah dipublikasikan. Dalam pengembangan aplikasi *terintegrasi*, pengujian unit berulang mengharuskan penggunaan pengujian otomatis (Rajasekhar dan Shafi, 2014).

Selenium IDE adalah aplikasi yang mengotomatiskan pengujian *web*. *Selenium* bekerja dengan melacak aktivitas pengguna di *web* yang sedang diuji. Serta dapat menjalankan Kembali aktivitas yang direkam dan menyimpannya. Karena pengembang aplikasi dapat menjalankan rekaman yang direkam. Ketika proses pengujian tidak lagi memerlukan intervensi manual, *test case* berdasarkan unit *test case* akan berjalan secara otomatis. Sebuah aplikasi khusus dapat digunakan untuk menulis dan menjalankan *test case* pengujian otomatis untuk membandingkan hasil aktual dengan hasil prediksi. Pengujian aplikasi dapat mencapai hal-hal yang tidak dapat dilakukan oleh pengujian manual, dan pengujian otomatis adalah cara yang bagus untuk meningkatkan efektivitas, efisiensi, dan jangkauannya. Dengan cara ini, pengujian otomatis paling cocok untuk lingkungan di mana banyak uji regresi dan kasus uji kompleks harus dijalankan berulang kali dan persyaratan sering berubah. Selain itu, pengujian otomatis berpotensi menurunkan biaya perawatan di masa mendatang dan meningkatkan kualitas pengujian struktural (Sharma, 2012).

Pengembangan aplikasi atau *website* akan lebih efektif dan efisien dari segi waktu, biaya dan tenaga jika digunakan alat pengujian yang dapat berjalan secara otomatis dan terintegrasi

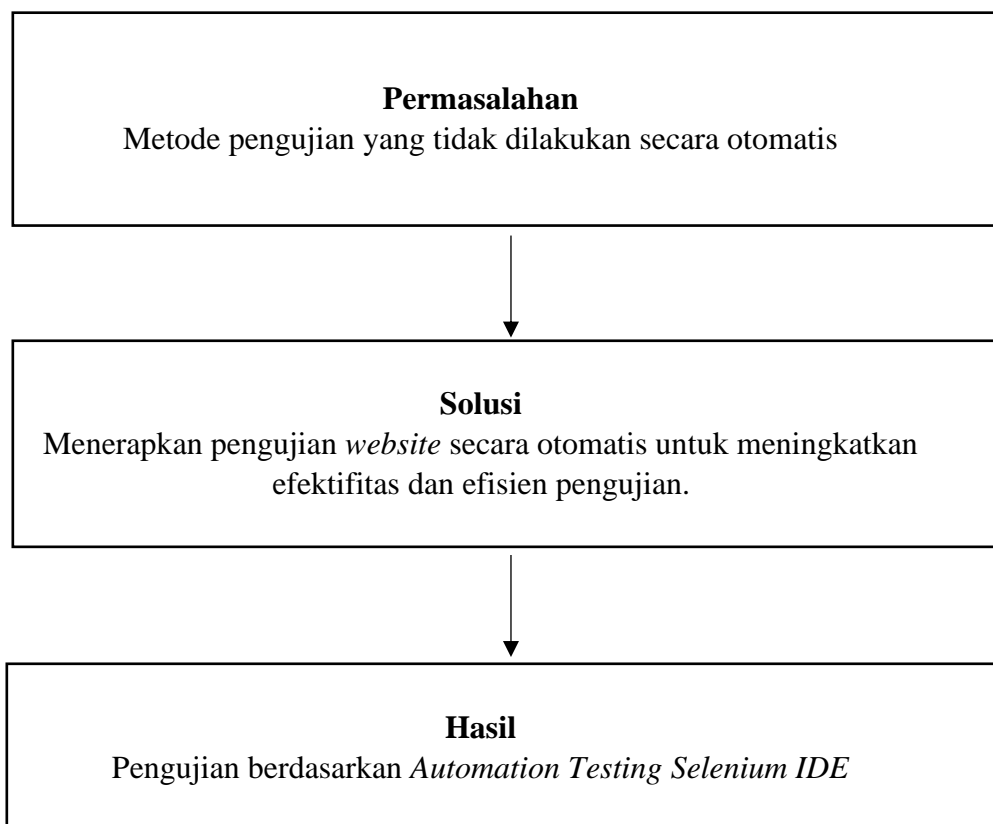
untuk mengatasi permasalahan yang ada saat ini. Selain itu, salah satu ide dalam pengembangan aplikasi adalah pengujian otomatis, yang memiliki banyak manfaat. tujuan dari akhir pengujian otomatis adalah untuk mengurangi jumlah waktu yang diperlukan untuk menghasilkan program siap pakai dari konsep awal. Pembatasan masalah digunakan untuk menghindari adanya penyimpangan manapun serta pelebaran dari pokok masalah. Batasan masalah dalam tugas akhir adalah pengujian otomatis yang akan dilakukan hanya pada bagian *unit testing* seperti *form validation, login test validation read, create, update, delete*. Ruang lingkup hanya meliputi pengujian aplikasi *website* secara otomatis. *Website* yang akan dilakukan pengujian secara otomatis adalah *website* portal unit kegiatan mahasiswa (UKM) Politeknik Negeri Lampung.

### 1.2. Tujuan

Berdasarkan latar belakang permasalahan diatas maka penelitian ini memiliki tujuan untuk menghasilkan alat uji untuk menguji *website* unit kegiatan mahasiswa Politeknik Negeri Lampung dengan *selenium* untuk mendapatkan data rekomendasi.

### 1.3 Kerangka Pikiran

Berdasarkan latar belakang tersebut, dapat disusun suatu kerangka pikiran yang disajikan dalam Gambar 1.



Gambar 1. Kerangka Pemikiran

#### **1.4.Kontribusi**

Dengan dibuatnya pengujian aplikasi *website* secara otomatisasi dengan *selenium ide* diharapkan dapat memberikan kontribusi sebagai berikut:

1. Mempercepat dan mempermudah programmer mengetahui *bug* pada aplikasi
2. Meningkatkan akurasi dan efisiensi dalam pengujian aplikasi, sehingga dapat meningkatkan aplikasi
3. Proses pengembangan aplikasi lebih cepat
4. Memudahkan dalam melihat *unit test case* yang belum selesai dan sudah selesai.

## II. TINJAUAN PUSTAKA

### 2.1 Aplikasi *Testing*

Aplikasi pengujian *testing* adalah perangkat lunak khusus yang digunakan untuk menotomatisasikan dan memfasilitasi proses pengujian perangkat lunak atau sistem komputer. Aplikasi pengujian digunakan pengembang pengujian untuk menguji, memvalidasi, dan memverifikasi kualitas, kinerja, keamanan dan fungsionalitas perangkat lunak. Evaluasi perlengkapan atau kemampuan program dapat menentukan bahwa program tersebut telah memenuhi persyaratan yang telah ditentukan aplikasi *testing* (Hasan Abdurahman dan Asep Ririh Riswaya, 2014) dalam melakukan pengujian pendekatan pada metode pengembangan tergantung aplikasi yang dipakai, tujuan utama dari pendekatan metode pengujian untuk menemukan masalah secepat mungkin untuk mengurangi biaya dan perbaikan di masa yang akan datang. Aplikasi *testing* menentukan apakah suatu produk aplikasi sudah berfungsi dengan benar dan efisien sesuai dengan persyaratan yang telah ditentukan

#### 2.1.1 Performa Pengujian

Pengujian perangkat lunak bisa dilakukan pendekatan pengujian secara manual yang dilakukan oleh manusia dan pengujian perangkat lunak otomatis yang dilakukan oleh komputer.

##### 1. pengujian manual

Pengujian secara manual dilakukan oleh manusia yang mencoba kombinasi penggunaan dan memasukan serta membandingkan menggunakan dengan perilaku yang diharapkan dan mencatat hasil secara manual.

##### 2. Pengujian Otomatis

Pengujian otomatis membutuhkan sebuah pengetahuan berupa skenario pengujian yang dijadikan bahan atau landasan untuk melakukan proses pengujian. Pengujian secara otomatis sendiri dapat dikatakan sebagai proses pengujian yang konsisten karena kita dapat melakukan proses pengujian yang sama dikarenakan berdasarkan pada sebuah skenario pengujian, sehingga apabila terjadi perubahan pada sistem langkah-langkah pengujian masih akan tetap sama. Pada proses pengujian otomatis hal-hal yang dibutuhkan yaitu berupa analisis sistem yang digunakan sebagai dasar untuk membuat skenario pengujian yang nantinya skenario pengujian tersebut digunakan pada alat bantu pengujian otomatis (Pressman, 2010).

### 3. Kelebihan dan kekurangan pengujian manual

Pengujian secara manual dan otomatis memiliki kelebihan dan kekurangan.

#### a. Kelebihan pengujian secara manual

Waktu dalam pengujian secara manual lebih lambat dibandingkan dengan pengaturan ketika melakukan pengujian otomatis.

#### b. Kekurangan pengujian secara manual

Dalam panduan pengujian manual tidak dapat pengujian secara berulang karena bisa menyebabkan hasil yang tidak akurat. Pengujian secara manual juga lebih lambat dibandingkan dengan pengujian secara otomatis dilakukan oleh komputer.

### 4. Kelebihan dan kekurangan pengujian secara otomatis

Pengujian secara otomatis memiliki banyak manfaat yang besar untuk proyek, karena biaya awal untuk otomatisasi dan pemeliharaan pengujian lebih mahal dengan begitu banyak biaya di masa depan (Huska 2014).

a. Pengujian otomatis berjalan lebih cepat daripada pengujian manual karena memiliki eksekusi pengujian yang lebih cepat

b. Pengujian otomatis dapat membantu mengurangi kesalahan yang bisa terjadi oleh *human error* serta masalah waktu yang menjadi kendala dalam pengujian manual yang dilakukan oleh manusia.

c. Pengujian secara otomatis bisa mengurangi biaya anggaran.

Kekurangan pengujian otomatis

a. Beberapa *tools* pengujian otomatis bersifat berbayar

b. Tidak mendapatkan *feedback* dari sisi *user experience*

## 2.1.2 Ruang Lingkup Pengujian

### 1. *Unit Testing*

*Unit Testing* bagian terkecil dari aplikasi yang dapat diuji kemudian dapat diaplikasikan dan uji coba. *Unit testing* melakukan pengujian yang dilakukan sebagai pengujian *white box testing* karena pengujian perlu mengetahui kode dari aplikasi yang akan diuji (Huska 2014).

### 2. *Integration Testing*

*Integration testing* digunakan setelah melakukan pengujian unit dan sebelum fase pengujian aplikasi dilakukan. Tujuan utama dari *integration testing* adalah untuk

menguji kelompok *testing* serta melihat apakah sudah berjalan secara bersama dengan benar dan terintegrasi.

### 3. *Sistem testing*

Gabungan dari beberapa *unit testing* yang menjadi satu komponen, dan beberapa komponen terintegrasi menjadi satu komponen besar yang disebut sistem. Bertujuan untuk menjelaskan masalah yang diekspos dengan melakukan pengujian seluruh sistem yang terintegrasi (Huska 2014).

#### Bentuk Pengujian

##### 1. *Alpha Testing*

*Alpha Testing* dilakukan oleh kelompok pengembang aplikasi yang bertujuan untuk menemukan masalah dan kendala sebelum aplikasi siap untuk dirilis.

##### 2. *Acceptance testing*

Aplikasi yang telah disesuaikan dan dirancang untuk penggunaan tertentu, dalam hal ini pengguna akan melakukan validasi terhadap aplikasi yang sesuai spesifikasi yang dibutuhkan.

##### 3. *Installation testing validation comatibility with hardward platforms*

Portabilitas merupakan kemampuan menjalankan aplikasi pada berbagai *hardware* dan *software* tanpa mengalami perubahan tampilan dan perubahan fungsi yang diperlukan dari pengguna akhir aplikasi.

##### 4. *Stress testing*

Mesnimulasikan perilaku abnormal dalam suatu program aplikasi dengan mempertimbangkan situasi yang menyebabkan akhir program aplikasi yang tidak normal, serta digunakan untuk membaut tolak ukur stabilitas aplikasi.

##### 5. *Smoke testing*

Memberikan konfirmasi bahwa perubahan baru tidak akan mengganggu proses integrasi pada produk aplikasi. *Smoke testing* dilakukan dengan cepat karena sering digunakan untuk mevalidasi integrasi pada produk aplikasi sebelum pengujian yang lebih lama.

## 2.2 Pengertian Website

*Website* atau *World Wide Web* adalah sistem akses informasi dalam internet *web* disusun dari halaman-halaman yang menggunakan teknologi *web* yang saling berkaitan satu sama lain. Sedangkan pengertian lain menyebutkan bahwa *website* sebuah situs yang menyediakan informasi dan pelayanan-pelayanan yang berkaitan dengan informasi, seperti pencarian,

penawaran komersial, aplikasi-aplikasi produktivitas pribadi, fitur membangun komunitas, serta kanal komunikasi bagi pengguna. *Web* dan internet merupakan dua hal yang berbeda (Betha dan Husni, 2010). Internet merupakan perangkat keras dan *web* perangkat lunak. Protokol yang digunakan internet dan *web* juga berbeda, internet menggunakan TCP/IP sedangkan *web* menggunakan HTTP (*HyperText Transfer Protocol*)

### **2.3 Web Portal**

*Web Portal* merupakan sistem informasi berbasis *web* yang menyediakan informasi serta akses ke beberapa fitur lain. Kemampuan *portal* yang lebih spesifik adalah menyediakan informasi yang dapat diakses menggunakan berbagai perangkat, komputer, PDA (*personal Digital Assistant*) atau bahkan handphone (Ganiardi, 2023).

### **2.4 Unit Kegiatan Mahasiswa (UKM)**

Unit Kegiatan Mahasiswa atau UKM adalah organisasi yang mengkhususkan kegiatan pada kegiatan pengembangan minat dan bakat, organisasi ini dibentuk sebagai tempat pengembangan bakat, prestasi, penalaran, minat dan kreativitas mahasiswa (Ulaan, 2020).

### **2.5 Selenium**

*Selenium* dikembangkan oleh Jason Huggins pada tahun 2004 dalam bentuk *library Script*. *Selenium* di relaksasi dengan lisensi FOSS menggunakan klausula lisensi apache sehingga dapat diunduh dan digunakan secara bebas dan gratis. Dari tahun 2006 sampai saat ini *selenium* dikembangkan oleh *google* untuk keperluan sendiri dan publik (Pressman, 2010).

Saat ini, *selenium* direalisasi dalam bentuk beberapa paket perangkat lunak yang memiliki fungsi tersendiri:

1. *Selenium RC (Remote control)* paket utama yang berisi semua fungsi utama pengujian (*selenium Core*). Paket ini dalam bentuk *Java Script* yang dikontrol oleh server berbasis Java
2. *Selenium webDriver selenium RC* dikembangkan oleh Simon Stewart pada tahun 2006 menjadi *WebDriver* yang menyediakan API untuk mengakses browser secara langsung tanpa menggunakan *Java Script*. Pada tahun 2012 Simon Stewart dan untuk menetapkan *WebDriver* sebagai *internet standar*.
3. *selenium Ide* merupakan prototype untuk membuat *test script*. *Selenium* adalah plugin pada browser firofox yang menyediakan antarmuka yang mudah untuk membuat skenario test juga menyediakan recording dari aksi pengguna yang dapat di ekspor ke dalam



berbagai bahasa. Tetapi *selenium ide* tidak memiliki kemampuan untuk menangani pengulangan dan kondisional.

4. *Selenium Grid* untuk menggabungkan beberapa *selenium RC* dalam rangka meingkatkan skalabilitas pengujian yang dijalankan pada beberapa lingkungan pengujian. Dengan demikian dimungkinkan dilakukan pengujian secara paralel dalam skala besar. Akan tetapi dokumentasi belum dibuat.

## 2.6 Selenium IDE

*Selenium* merupakan sebuah alat bantu pengujian sebuah perangkat lunak yang bersifat terbuka untuk sistem *web* pada *browser* yang juga mendukung beberapa bahasa pemrograman (Niranjana Murthy, Kumar, Srinivas, & Manoj, 2014).

*Selenium* terdiri dari beberapa jenis, seperti *Selenium IDE*, *Selenium RC*, *Selenium Web Driver*, dan *Selenium Grid*. Dari setiap jenis, masing-masing *Selenium* memiliki pendekatan tersendiri untuk melakukan proses pengujian secara otomatis. Beberapa jenis *Selenium* digunakan hanya untuk melakukan proses merekam kembali proses pengujian. Seperti *Selenium IDE* adalah sebuah *plugin* yang dapat digunakan pada *web browser Firefox* yang menyediakan sebuah fasilitas untuk merekam, mengedit, dan juga melakukan proses *debugging* selama proses pengujian sehingga pengujian dapat berfokus pada proses pengembangan kasus uji (Hawari, Kusumo, & Firdaus, 2015). Sedangkan *Selenium RC* adalah sebuah alat pengujian otomatis yang bersifat dinamis di karenakan menggunakan bahasa pemrograman. Kemudian *Selenium Web Driver* yang lebih berfokus kepada proses implementasi moderen dalam mengotomasi tindakan pada *web browser*, dengan kata lain proses pengujian dilakukan dengan berkomunikasi langsung dengan *web browser*. *Selenium Grid* merupakan alat yang digunakan secara bersamaan dengan menggunakan *Selenium RC* sebagai alat yang menjalankan tes secara paralel di berbagai mesin dan *browser* secara bersamaan.

Dari berbagai jenis *Selenium* yang ada, pada *Selenium IDE* memiliki beberapa kelebihan seperti yang ditemukan pada penelitian (Yuda, 2019) diantaranya:

- a. *Selenium IDE* merupakan alat bantu pengujian otomatis *open source*.
- b. *Selenium* mendukung bahasa Java, Net, Perl, PHP, Python, dan Ruby.
- c. *Selenium IDE* dapat digunakan pada sistem operasi Windows, MacOS, dan Unix
- d. Untuk melakukan pengujian otomatis dengan *Selenium IDE* dapat dilakukan dengan menambahkan *plugin* pada aplikasi *web*.

*Selenium IDE* juga memiliki kekurangan diantaranya:

- a. *Selenium IDE* hanya mendukung aplikasi berbasis web.

b. *Selenium IDE* tidak dapat merekam aktifitas diluar web browser.

Pada penelitian ini berfokus pada penggunaan *Selenium IDE* sebagai alat bantu pengujian otomatis dengan menerapkan metode regresi. *Selenium IDE* sangat berguna dalam melakukan proses pengujian fungsional sistem dan yang berhubungan dengan *web browser*. Hasil yang diperoleh dari proses pengujian menggunakan *Selenium IDE* juga dapat dikatakan cukup kompleks di karenakan melibatkan proses integrasi dari banyak komponen lainnya. Proses pengujian secara otomatis dengan menerapkan metode regresi dan alat bantu *Selenium IDE* diperlukan untuk membuat pengembang perangkat lunak lebih teratur dalam melakukan proses pengujian.

## **2.7 IDE**

*IDE* adalah program *computer* yang menyediakan fasilitas yang dibutuhkan untuk membangun program aplikasi. *IDE* juga dapat diartikan sebagai program atau alat bantu yang terdiri *compiler*, *debugger*, *editor*, dan *desain* yang terintegrasi dalam aplikasi. Tujuan *IDE* adalah menyediakan semua fasilitas yang dibutuhkan untuk proses pembangunan program aplikasi (Yuda, 2019).

## **2.8 Bug**

*Bug* adalah error pada aplikasi yang tidak sesuai dengan kebutuhan yang telah dibuat dalam sebuah program aplikasi, sebelum aplikasi di sebar luaskan maka harus melalui tahapan pengujian terlebih dahulu supaya ketika aplikasi sudah disebar luaskan tidak terjadi *bug* yang ditemukan pada aplikasi (Azhar 2016).

## **2.9 Test Case**

*Test Case* digunakan penguji untuk memverifikasi aplikasi yang akan diuji berfungsi dengan benar atau tidak. *Test case* membantu untuk menemukan kesalahan dan persyaratan yang dibutuhkan oleh aplikasi.