

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pengujian perangkat lunak adalah proses krusial dalam pengembangan suatu aplikasi. Pengujian dirancang untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang ditetapkan. Melalui pengujian, kesalahan atau *bug* dalam perangkat lunak dapat diidentifikasi dan diperbaiki sekaligus memverifikasi bahwa perangkat lunak memenuhi semua persyaratan yang diperlukan. Pengujian ini sangat penting untuk menjaga kualitas perangkat lunak serta mengurangi risiko kegagalan pasca implementasi. Shaleh et al. (2021) menekankan pentingnya pengujian perangkat lunak untuk menjamin bahwa seluruh fungsionalitas aplikasi berjalan lancar dan mampu memenuhi kebutuhan pengguna dengan baik.

Dalam siklus pengembangan perangkat lunak, pengujian biasanya dilakukan secara sistematis melalui tahapan *Software Testing Life Cycle* (STLC). STLC adalah kerangka kerja pengujian perangkat lunak yang terdiri dari enam tahap utama, yaitu analisis kebutuhan pengujian, perencanaan pengujian, pengembangan kasus uji, persiapan lingkungan pengujian, eksekusi pengujian, dan penutupan siklus pengujian. Setiap tahap ini dirancang untuk menjamin kualitas perangkat lunak secara menyeluruh dan mengidentifikasi masalah pada tahap awal siklus pengembangan. Menurut Rambe & Prihantoro (2022), penerapan STLC memungkinkan pengujian yang lebih terstruktur, efisien, dan terukur, sehingga membantu tim pengembang dalam mengelola waktu, sumber daya, serta kualitas hasil pengujian.

Pengujian regresi merupakan salah satu jenis pengujian perangkat lunak yang berfungsi untuk memastikan bahwa pembaruan perangkat lunak seperti penambahan fitur atau perbaikan *bug*, tidak mengganggu fungsionalitas yang sudah ada ataupun menimbulkan *bug* baru. Hal ini sangat penting terutama bagi aplikasi seperti Haus! sebagai penyedia layanan pembelian produk minuman dan makanan

secara *online*. Aplikasi Haus! berulang kali melakukan pembaruan agar kinerja aplikasi menjadi lebih baik dari versi-versi sebelumnya. Namun, meskipun penting, pengujian perangkat lunak masih sering kali dilakukan secara manual, salah satunya pada aplikasi Haus!. Tim pengembang aplikasi Haus! menyatakan bahwa ketika sedang melakukan pengembangan versi terbaru dari aplikasi Haus!, pengujian regresi masih hampir seluruhnya dilakukan secara manual. Metode pengujian manual ini menghadapi beberapa tantangan yang signifikan. Menurut penelitian yang dilakukan oleh Kusyadi et al. (2022), pengujian manual sering kali memakan waktu lebih lama, terutama ketika sistem semakin kompleks dan cakupan pengujian bertambah. Selain itu, pengujian manual bergantung pada kemampuan pengujian manusia yang dapat menyebabkan variasi hasil pengujian dan meningkatkan risiko kesalahan. Penelitian lain oleh Desyani et al. (2023) menyatakan bahwa salah satu tantangan terbesar dalam pengujian manual adalah ketidakmampuan untuk menjalankan pengujian berulang kali tanpa adanya ketidakakuratan akibat kelelahan manusia yang dapat menyebabkan penurunan kualitas hasil pengujian.

Dalam dunia yang semakin berkembang dan dituntut untuk memberikan solusi yang lebih cepat dan lebih efisien, munculnya metode pengujian otomatis menjadi solusi yang sangat relevan. Alat pengujian otomatis seperti Katalon Studio telah menjadi sangat populer dalam beberapa tahun terakhir. Katalon Studio merupakan alat pengujian yang mendukung berbagai *platform* dan metode pengujian. Penelitian oleh Kusyadi et al. (2022) menunjukkan bahwa pengujian otomatis dengan Katalon Studio memiliki beberapa keunggulan seperti efisiensi waktu dan pengurangan kesalahan manusia. Pengujian secara otomatis juga akan bermanfaat ketika terdapat pembaruan pada aplikasi Haus!. Tim pengembang Haus! cukup menjalankan kembali skrip pengujian dari setiap fungsi utama yang sudah ada sebelumnya, sehingga akan langsung terdeteksi jika terdapat fungsi utama yang mengalami gangguan dan dapat diperbaiki secepatnya. Penelitian oleh Melia, S., & Putra, F. P. (2023) juga menyatakan bahwa Katalon Studio memiliki keunggulan jika dibandingkan dengan alat pengujian otomatis lainnya yaitu tidak mengharuskan penggunanya untuk menguasai bahasa pemrograman karena Katalon Studio memiliki *user interface* yang simpel dan informatif sehingga mudah digunakan oleh pemula.

Selain Katalon Studio, pendekatan lain yang mulai banyak digunakan dalam pengujian perangkat lunak adalah *Behavior Driven Development* (BDD) yang didukung oleh alat seperti Cucumber. BDD merupakan pendekatan yang mendorong kolaborasi antara tim pengembang dan pemangku kepentingan non-teknis, seperti manajer produk atau pengguna akhir. Dengan menggunakan bahasa yang mudah dipahami oleh semua pihak, yaitu bahasa Gherkin, BDD memungkinkan pengujian dilakukan berdasarkan skenario yang telah dibuat. Menurut penelitian oleh Mughal (2024), penggunaan pendekatan BDD dengan Cucumber tidak hanya mempermudah kolaborasi antar tim, tetapi juga memastikan bahwa setiap fitur yang diuji sesuai dengan kebutuhan dan harapan pengguna.

1.2. Identifikasi Masalah

Identifikasi masalah pada penelitian ini adalah

1. Pengujian regresi pada aplikasi Haus! saat ini masih hampir seluruhnya dilakukan secara manual yang membutuhkan durasi pengujian yang lama serta rentan terhadap kesalahan
2. Belum ada implementasi dari alat pengujian otomatis seperti Cucumber dan Katalon Studio pada aplikasi Haus!

1.3. Pembatasan Penelitian

Penelitian ini memiliki beberapa batasan yaitu,

1. Penelitian berfokus pada proses pengujian regresi aplikasi Haus! pada tingkat *Integration Testing* dan *System Testing*
2. Penelitian berfokus pada pengujian fungsionalitas aplikasi
3. Aplikasi Haus! yang digunakan hanya berbasis Android dan tidak mencakup iOS.
4. Penelitian ini menggunakan aplikasi Haus! versi 2.0.8 dan versi 2.0.9 sebagai objek pengujian.

1.4. Rumusan Masalah

Rumusan masalah dari penelitian ini yaitu,

1. Bagaimana tahapan pengujian untuk aplikasi Haus! versi 2.0.8 dan 2.0.9 dapat dirancang menggunakan Cucumber dan Katalon Studio dalam siklus *Software Testing Life Cycle*?
2. Seberapa efektif skrip pengujian yang dirancang berdasarkan aplikasi Haus! versi 2.0.8 ketika diuji kembali pada aplikasi Haus! versi 2.0.9?
3. Bagaimana perbandingan waktu eksekusi skrip pengujian pada aplikasi Haus! versi 2.0.8 dengan aplikasi Haus! versi 2.0.9?

1.5. Tujuan Penelitian

Tujuan dari dilakukannya penelitian ini adalah

1. Merancang tahapan-tahapan pengujian terhadap aplikasi Haus! versi 2.0.8 dan versi 2.0.9 berdasarkan siklus *Software Testing Life Cycle*
2. Membuat skrip pengujian regresi berdasarkan aplikasi Haus! versi 2.0.8 menggunakan Cucumber dan Katalon Studio
3. Menjalankan skrip pengujian pada aplikasi Haus! versi 2.0.8 dan versi 2.0.9
4. Membandingkan data hasil pengujian dan waktu eksekusi skrip pengujian pada aplikasi Haus! versi 2.0.8 dengan aplikasi Haus! versi 2.0.9.

1.6. Manfaat Penelitian

Manfaat dari penelitian ini adalah

1. Memberikan alternatif metode pengujian regresi yang lebih efisien bagi pengembang aplikasi Haus! dengan menggunakan alat pengujian otomatis seperti Cucumber dan Katalon Studio.
2. Meningkatkan kualitas pengujian pada aplikasi Haus! dengan menerapkan BDD yang mempermudah dalam hal keterbacaan skrip pengujian.