

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Maraknya perkembangan internet saat ini telah menjadikan industri logistik memiliki peran krusial dalam meningkatkan efisiensi distribusi barang dan layanan. Para penyedia layanan logistik sering menghadapi tantangan seperti kerusakan tak terduga, jadwal pemeliharaan yang suboptimal, dan koordinasi kompleks dengan berbagai pemangku kepentingan. Mengelola armada kendaraan menjadi satu dari banyak tantangan krusial yang memerlukan solusi inovatif. Hal ini mendorong TransTrack sebagai perusahaan yang bergerak di bidang industri logistik untuk mengembangkan salah satu produk website mereka bernama *Vehicle Maintenance System* (VMS) sebagai solusi revolusioner berbasis *Software as a Service* (SaaS) yang dirancang khusus untuk mengatasi kompleksitas dalam pengelolaan armada kendaraan.

Vehicle Maintenance System (VMS) dirancang sebagai jawaban cerdas untuk meningkatkan efisiensi operasional dan mengoptimalkan biaya perawatan bagi penyedia layanan logistik. Tantangan yang dihadapi oleh penyedia layanan logistik, seperti kerusakan tak terduga, jadwal pemeliharaan yang tidak optimal, dan kesulitan berkoordinasi, menjadi fokus utama VMS. Metode *maintenance* konvensional seringkali dilakukan secara manual, mencatat jadwal *maintenance*, catatan perbaikan, dan histori servis dalam buku atau lembar kerja. VMS hadir sebagai solusi yang efisien, ter-otomatisasi, dan terintegrasi untuk mengelola perawatan armada kendaraan. Adanya pendekatan ini akan membuat VMS menggantikan metode konvensional yang rentan terhadap kesalahan dan kekurangan informasi. Sistem ini memberikan rekam jejak yang terdokumentasi rapi terkait jadwal *maintenance*, catatan perbaikan, dan histori servis armada. Adanya visibilitas yang lebih baik terhadap kondisi armada, perusahaan logistik dapat meningkatkan efisiensi operasional, mengurangi biaya perawatan, dan memaksimalkan kinerja keseluruhan armada.

Selama masa pengembangan ini pastinya VMS tidak luput dari kesalahan yang berupa *bug* dan *error*. Dalam mencapai tujuan agar *website* ini bisa digunakan oleh para pengguna dengan baik dan lancar, maka perlu diadakannya pengujian secara berkala dengan beberapa metode dalam pengujian *software*. Metode yang digunakan dalam pengujian *software* pada produk VMS adalah pengujian *Black Box* secara manual dan otomatis menggunakan teknologi *framework* katalon. Pengujian *Black box* merupakan suatu metode pengujian perangkat lunak yang tidak memerlukan pengetahuan terkait struktur kode pemrograman terhadap pengujian *website* yang diuji. Pengujian ini hanya berfokus pada fungsionalitas dan persyaratan sistem, tanpa memperhatikan bagaimana sistem bekerja di dalam. Pada produk VMS ini proses pengujian terhadap *software* secara manual dilakukan dengan menguji setiap fungsi dari *website* VMS secara langsung oleh tim *Quality Assurance* dalam mencoba berbagai scenario pengguna untuk memastikan setiap fitur bekerja sesuai spesifikasi yang telah dibuat, sedangkan pengujian secara otomatis menggunakan alat teknologi *framework* bernama katalon. Katalon dirancang untuk menyederhanakan dan mempercepat proses pengujian *software* tanpa harus memiliki pemahaman mendalam terhadap struktur kode pemrograman pada *website* yang diuji. Katalon memiliki antarmuka yang ramah pengguna serta mendukung *scripting* bagi penguji yang ingin mengembangkan tes lebih lanjut, sehingga fitur-fitur yang tersedia akan mendukung pengujian fungsionalitas, regresi, dan bahkan integrasi dengan alat-alat *DevOps* lainnya bagi para pengguna.

Pada pengujian ini tidak menerapkan metode *white box testing* karena memerlukan pengetahuan mendalam tentang struktur internal dan kode sumber aplikasi yang tidak selalu diperlukan untuk memastikan fungsionalitas sistem, sementara fokus utama VMS adalah pada aspek fungsionalitas dan metode *test driven development* juga tidak diterapkan karena mengharuskan penulisan tes sebelum pengembangan kode yang membutuhkan perubahan budaya dan pendekatan tim pengembangan perangkat lunak. *Test driven development* membutuhkan perubahan signifikan dalam proses pengujian yang memengaruhi kecepatan pengujian, kolaborasi tim, dan metode manajemen proyek secara keseluruhan yang dimana tidak sesuai dengan dinamika dan kecepatan produk VMS yang membutuhkan hasil cepat.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas dan fokus permasalahan pada pengujian software, maka isi dari rumusan masalah dan solusi sebagai berikut:

1. Bagaimana cara menguji secara otomatis pada pengujian *website* VMS dengan menggunakan *framework* Katalon?

1.3 Tujuan

Berdasarkan latar belakang di atas dan fokus permasalahan pada pengujian software, maka isi dari tujuan sebagai berikut:

1. Memastikan bahwa penggunaan Katalon sebagai alat pengujian otomatis dapat mempersingkat waktu pengujian terhadap kualitas pengujian *website* VMS.

1.4 Batasan Masalah

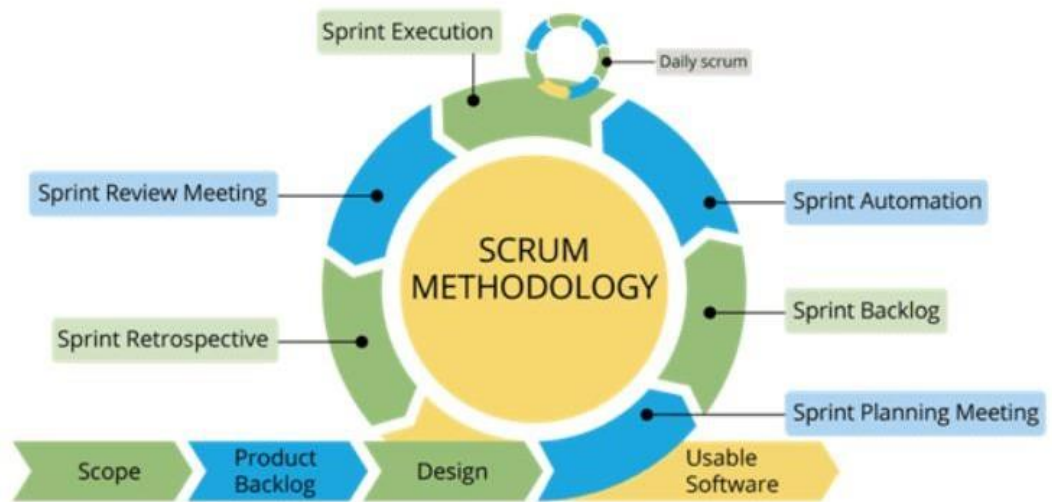
Berdasarkan latar belakang di atas dan fokus permasalahan pada pengujian software, maka isi dari batasan masalah sebagai berikut:

1. Pengujian secara manual menggunakan metode *Black Box Testing*.
2. Pengujian secara otomatis menggunakan *framework* Katalon.
3. Tidak menggunakan *Test Driven Development*.
4. Tidak menggunakan *White Box Testing*.

1.5 Metode Pengerjaan

Vehicle Maintenance System adalah solusi berbasis *Software as a Service* (SaaS) yang dirancang untuk mengatasi kompleksitas pengelolaan armada kendaraan dan dalam memastikan VMS berfungsi secara optimal, maka metode *agile scrum* digunakan dalam pengembangan dan pengujian produk ini. *Agile Scrum* adalah metodologi pengembangan perangkat lunak yang *iteratif* dan *incremental* yang dimana berfokus pada kolaborasi tim, fleksibilitas terhadap perubahan, dan pengiriman produk dalam

siklus pendek yang disebut sprint. Berikut adalah gambar dari sebuah metode *agile scrum*:



Gambar 1. 1 adalah gambaran dari sebuah metode agile scrum yang memiliki tahapan awal dari scope hingga tahapan akhir pada usable software

Peran metode *agile scrum* pada pengujian website VMS adalah sebagai berikut:

1. Proses Pengumpulan Data

- 1) *Sprint Planning*: Melakukan rapat bersama *Project Manager*, *Business Analyst*, *UI/UX Designer*, dan *Tim Developer* untuk memahami persyaratan dan tujuan dari fitur yang akan dikembangkan selanjutnya.
- 2) Studi Dokumen: Mengumpulkan dan menganalisis dokumen pengujian seperti rencana pengujian, kasus uji, laporan *bug*, dan hasil pengujian.
- 3) Eksperimen Pengujian: Melakukan pengujian manual dan otomatis pada fitur-fitur terpilih dengan menggunakan metode *black box* dan *framework* Katalon.

2. Proses Pengujian

- 1) Perencanaan Pengujian: Menyusun rencana pengujian berdasarkan *end-to-end testing* untuk memastikan seluruh alur kerja pada sistem diuji secara komprehensif.

- 2) Desain Kasus Uji: Mengembangkan kasus uji untuk pengujian manual dan otomatis menggunakan metode *black box*. Kasus uji dirancang untuk menguji fungsi tanpa memeriksa struktur internal kode.
 - 3) Pelaksanaan Pengujian Manual: Mengikuti langkah-langkah yang telah ditetapkan dalam kasus uji untuk memverifikasi setiap fitur secara manual.
 - 4) Pelaksanaan Pengujian Otomatis: Mengotomatiskan kasus uji menggunakan *framework* Katalon, menjalankan skrip pengujian otomatis, dan merekam hasil pengujian.
 - 5) Pelaporan *Bug*: Mencatat dan melaporkan *bug* yang ditemukan dalam sistem pelacakan *bug*.
 - 6) *Regresi Testing*: Melakukan pengujian ulang setelah perbaikan *bug* untuk memastikan bahwa perubahan tidak menyebabkan kesalahan baru.
3. Analisis Data dengan menyajikan data mengenai jumlah dan jenis bug yang ditemukan, waktu yang dibutuhkan untuk pengujian manual dan otomatis, serta efisiensi dan efektivitas pengujian menggunakan Katalon.
 4. Alat dan Teknologi Pengujian
 - 1) *End-to-End Testing*: Menguji seluruh proses bisnis dari awal hingga akhir untuk memastikan semua komponen sistem bekerja sesuai harapan dalam skenario penggunaan nyata.
 - 2) Metode *Black Box*: Menerapkan metode pengujian *black box* untuk memfokuskan pada fungsionalitas sistem tanpa memeriksa kode sumber.
 - 3) *Framework* Katalon: Menggunakan Katalon Studio untuk membuat, menjalankan, dan menganalisis skrip pengujian otomatis.
 5. *Sprint Review* dengan mendemonstrasikan dan mengevaluasi hasil kerja, yaitu mempresentasikan fitur yang telah dikembangkan dan diuji, menerima umpan balik dari tim pengembangan dan pengguna, serta mengevaluasi apakah hasil kerja sudah sesuai dengan tujuan *sprint*. *Sprint Review* juga digunakan untuk menyusun *backlog sprint* berikutnya berdasarkan umpan balik yang diterima.

6. Etika Penelitian dengan melakukan semua kegiatan pengujian sesuai kebijakan perusahaan dan menjaga kerahasiaan data pengguna.

1.6 Jadwal Pengerjaan

Magang berlangsung selama 2 semester dengan pelaksanaan aktivitas secara hybrid, baik di kantor maupun di rumah masing-masing. Berikut tabel 1.1 di bawah ini menunjukkan jadwal pelaksanaan kerja selama magang dalam satuan bulan.

Tabel 1. 1 Tabel Pelaksanaan Kerja sebagai Quality Assurance Intern di TransTRACK.ID

No.	Deskripsi Kerja	Agustus				September				Oktober				November				Desember				Januari				Februari				Maret				April				Mei				Juni		
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3				
1.	On Boarding, Mempelajari Automation Testing dan Proses Bisnis produk VMS																																											
2.	Mempelajari Automation Testing dan Test Case produk VMS																																											
3.	Membuat Test Case, Manual Testing, Automation Testing, Membuat dan Menyelesaikan Ticketing (QA Report/Issue) terhadap bug/error, serta Sprint Review dan Sprint Planning																																											