

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Saat ini perkembangan teknologi merupakan hal yang sangat penting, salah satunya dalam ranah bisnis. Dengan penggunaan teknologi yang tepat sesuai kebutuhan suatu perusahaan dapat meningkatkan produktifitas dan efektifitas, kedua hal tersebut merupakan suatu indikator yang penting bagi industri bisnis. Saat ini penggunaan teknologi dalam dunia bisnis juga sudah mulai diterapkan salah satunya pada industri *shipping company* yang merupakan suatu perusahaan dalam bidang pelayanan jasa pengiriman maupun ekspedisi suatu barang/muatan. Dalam hal ini beberapa industri *shipping company* bekerja sama dengan agregator logistik, dimana agregator logistik juga merupakan suatu perusahaan yang berintegrasi dengan perusahaan jasa pengiriman, sehingga baik *shipping company*, agregator dan jasa pengiriman dapat saling berintegrasi.

Berdasarkan hal tersebut saat ini mulai dikembangkan suatu sistem Logistic Service Integrator (LSI) dimana sistem ini merupakan suatu pengembangan teknologi guna melakukan integrasi antara *shipment company* sebagai perusahaan produksi barang dengan agregator selaku perusahaan yang akan menghubungkan dengan fleet owner sebagai *vendor* yang menyediakan kendaraan sebagai jasa pengiriman barang. Sehingga dengan dikembangkannya sistem tersebut dapat membantu dan mempermudah suatu bisnis dalam hal operasionalnya.

Selama masa pengembangan ini pastinya LSI 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. Salah satu metode yang digunakan dalam pengujian adalah pengujian *Black Box* dengan *framework* Katalon. *Black Box testing* merupakan suatu metode pengujian perangkat lunak yang tidak memerlukan pengetahuan terkait struktur kode pemrograman terhadap pengembangan website yang diuji. Pengujian ini hanya berfokus pada fungsionalitas dan persyaratan sistem, tanpa memperhatikan bagaimana sistem bekerja di dalam. Metode *Black Box* dapat dilakukan pengujian secara manual dan otomatis. Pada produk LSI ini pengujian terhadap software secara otomatis menggunakan alat teknologi *framework* bernama Katalon. Pada pengujian otomatis menggunakan Katalon ini, tim pengujian menggunakan fitur *Record and Playback* untuk mempermudah proses pengujian dalam mempercepat dan meningkatkan kualitas pengujian software pengembangan website LSI.

Pengujian secara manual dalam *Black Box testing* melibatkan pengujian langsung terhadap fungsionalitas sistem oleh tester tanpa menggunakan alat bantu otomatisasi. Proses pengujian manual memungkinkan untuk menangkap *bug* atau kesalahan yang mungkin terlewatkan oleh alat otomatisasi dan sangat efektif untuk menguji aspek-aspek yang memerlukan penilaian subjektif manusia, seperti antarmuka pengguna dan pengalaman pengguna.

Metode *White Box testing* tidak digunakan dalam kasus ini karena fokus utama adalah pada pengujian fungsionalitas eksternal dan persyaratan pengguna, bukan pada struktur internal atau alur logika kode program. *White Box testing* memerlukan pengetahuan mendalam tentang struktur kode dan sering kali lebih tepat digunakan oleh pengembang selama tahap pengembangan awal untuk menguji alur logika dan jalur eksekusi kode. *Test Driven Development* (TDD) tidak diterapkan di sini karena pendekatan ini memerlukan pengembangan tes sebelum kode aplikasi ditulis, yang tidak selalu praktis untuk semua tim pengembangan atau jenis proyek. TDD juga memerlukan disiplin dan keterampilan khusus dalam menulis tes terlebih dahulu, yang mungkin mempengaruhi kecepatan pengembangan jika tim tidak terbiasa dengan metode ini. Pemilihan Katalon sebagai alat uji otomatis didasarkan pada beberapa alasan.

Automatisasi pengujian dalam pengembangan Website LSI menggunakan Katalon adalah langkah yang penting dan tepat dalam meningkatkan efisiensi serta kualitas pengujian. Pertama-tama, dengan mengadopsi automasi pengujian, PT. Indo Trans Teknologi (TransTRACK.ID) dapat mengurangi ketergantungan pada pengujian manual yang cenderung lebih lambat dan rentan terhadap kesalahan manusia. Automatisasi memungkinkan pengulangan pengujian yang konsisten dan dapat diandalkan, yang mengarah pada penemuan *bug* lebih cepat dan efisien dalam siklus pengembangan.

Katalon menawarkan antarmuka yang *user-friendly* dan fitur *Record and Playback* yang memudahkan pembuatan skrip uji tanpa memerlukan keterampilan pemrograman yang mendalam. Hal ini sangat berguna untuk tim yang mungkin tidak memiliki latar belakang teknis yang kuat dalam pengembangan perangkat lunak. Selain itu, Katalon mendukung berbagai jenis pengujian, seperti pengujian web, API, mobile, dan desktop, serta memiliki integrasi dengan berbagai alat lain seperti Jenkins, JIRA, dan Git, yang meningkatkan efisiensi dan kolaborasi dalam proses pengembangan dan pengujian perangkat lunak.

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 melakukan automasi pengujian pada pengembangan Website LSI menggunakan Katalon?

1.3 Tujuan

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

1. Memanfaatkan Katalon dalam mengotomasi pengujian dalam proses pengembangan Website LSI.

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*.
5. Fitur Yang di Automasi:
 - *Login dan Logout*
 - *Create Shipment*
 - *History Shipment*
 - *Detail History Billing*
 - *Invoice Vendor*
 - *Filter dan Grid*

1.5 Metode Pengerjaan

Penulis menggunakan metode agile yang berfokus pada fleksibilitas, kolaborasi, dan responsivitas dalam pengembangan perangkat lunak atau produk lainnya. Proses ini dibagi menjadi iterasi pendek yang disebut *sprint*, di mana tim bekerja untuk menghasilkan hasil yang dapat diuji dan dinilai secara teratur. Keunggulan utama metode agile adalah kemampuannya untuk menyesuaikan perencanaan dengan perubahan yang terjadi selama proyek berlangsung, memungkinkan tim untuk dengan mudah merespons masukan dan permintaan perubahan dari pemangku kepentingan.

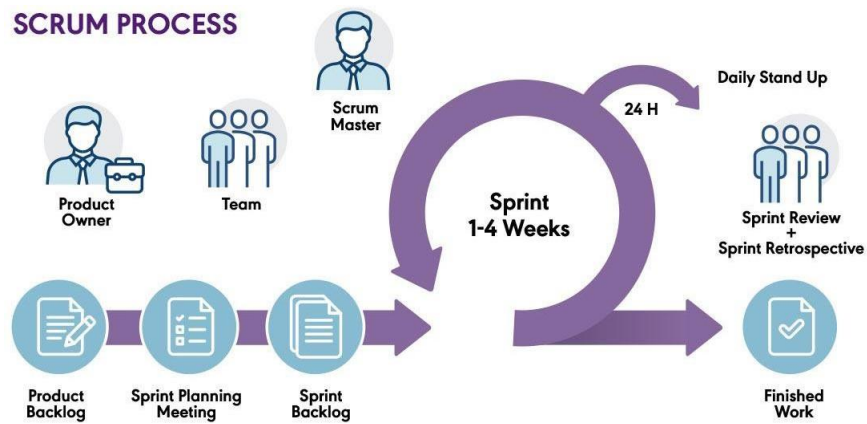
Metode agile mendorong kolaborasi erat antara anggota tim, pemilik produk, dan pemangku kepentingan, dengan komunikasi terbuka dan transparan yang sangat penting untuk menghasilkan produk yang sesuai dengan harapan pengguna dan

kebutuhan pasar yang berubah. Siklus pengembangan yang berulang memungkinkan tim untuk terus memperbaiki dan meningkatkan produk berdasarkan umpan balik pengguna, memastikan produk tetap relevan dan kompetitif. Secara keseluruhan, metode agile dikenal karena kemampuannya menghasilkan produk berkualitas tinggi dengan cepat, mengurangi risiko kegagalan proyek, dan memberikan nilai yang jelas bagi pemangku kepentingan.

Penulis menggunakan model agile Scrum, yang merupakan salah satu *framework* yang paling populer dan sering digunakan dalam metode agile. Scrum bekerja dengan membagi proyek menjadi serangkaian *sprint*, biasanya berlangsung selama 2 hingga 4 minggu. Berikut adalah elemen utama dari model Scrum:

1. *Product Backlog*: Daftar prioritas tinggi dari semua fitur, perbaikan, dan perubahan yang diinginkan dalam produk. Product Owner bertanggung jawab untuk memelihara dan memprioritaskan backlog ini.
2. *Sprint Planning*: Pertemuan yang diadakan pada awal setiap *sprint* di mana tim memilih item dari product backlog yang akan dikerjakan selama *sprint* tersebut.
3. *Sprint Backlog*: Subset dari product backlog yang dipilih untuk dikerjakan selama *sprint*. Tim bertanggung jawab untuk menyelesaikan item ini selama *sprint*.
4. *Sprint Review*: Pertemuan yang diadakan di akhir *sprint* untuk meninjau hasil kerja yang telah diselesaikan dan mendapatkan umpan balik dari pemangku kepentingan.
5. *Sprint Retrospective*: Pertemuan setelah *sprint* review di mana tim mengevaluasi proses kerja mereka selama *sprint* dan mengidentifikasi cara untuk meningkatkan kinerja dan efisiensi di *sprint* berikutnya.

Berikut adalah gambar model Scrum untuk lebih memahami prosesnya:



Gambar 1. 1 Scrum Proses

Dalam penerapan metode Scrum pada pengembangan sistem Logistic Service Integrator (LSI), tim pengembangan dan pengujian bekerja sama dalam *sprint* untuk memastikan setiap iterasi menghasilkan peningkatan yang berarti pada sistem. Proses pengujian, termasuk pengujian *Black Box* dengan *framework* Katalon, dilakukan dalam setiap *sprint* untuk memastikan kualitas dan fungsionalitas sistem secara berkelanjutan. Dengan pendekatan ini, tim dapat merespons umpan balik dengan cepat dan memastikan produk yang dikembangkan sesuai dengan kebutuhan pengguna dan pasar.

1.6 Jadwal Pengerjaan

Berikut adalah timeline pelaksanaan kerja yang dilakukan untuk mencapai solusi yang ditawarkan.

Tabel 1. 1 Tabel Pelaksanaan Kerja Magang

No	Deskripsi Kerja	Agustus				September				Oktober				November			
		2023				2023				2023				2023			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Onboarding pengenalan perusahaan, Pengenalan Logistic Service Provider (LSI).																
2	Membuat flow LSI v.1 & v.2, Membuat 2 Card Issues, Membuat test data untuk skenario forgot pin, Pembuatan Testcase LSI v2 feature secure PIN di																

	role <i>Product Owner</i> , Logistic Provider, dan <i>Vendor</i> .																			
3	Membuat 4 Card Issues, Pembuatan Testcase Transtrack Web feature Login, Pembuatan Testcase Transtrack Go featute Login, Search & Add Device, <i>Filter Map</i> , <i>Zoom</i> , <i>History</i> ,																			
4	Meeting LSI, Pembuatan Testcase Transtrack Go: <i>History</i> - Routes , Graph, Maps, Log, Button, Membuat <i>Test Case</i> Transtrack Web Feature : Login, Dashboard, Navigation Panel, Membuat 1 card Issues Transtrack Go: Button - See <i>History</i> - Sensor, Membuat Testcase untuk App TransTrack Web: List, Group, Detail Unit, Tools, Membuat 1 card Issues Transtrack Go : Login - Butuh Bantuan																			
5	Membuat Dokumentasi Testcase App TransTrack web: Alerts, Type, geofences, Notifications, Rute, Report, POI, Ruler, Maintenance, Tasks, Sharings, Membuat Testcase Untuk App TransTrack Web Fitur : Dashboard, Map Control, My Account, Show Name																			
6	Membuat <i>Test Case</i> Untuk App Transtrack Web: Admin <i>User</i> , Object, Content, Logo, Email, SMS <i>Sprint</i> Review LSI, Membuat Dokumentasi Untuk <i>Test Case</i> App Transtrack Web Fitur: Setup Events, SMS gateway, Show Point, Send Command, Forgot Password.																			

7	<p>Membuat <i>Test Case</i> Untuk App Transtrack Web: Admin <i>User</i>, Object, Content, Logo, Email, SMS, <i>Sprint</i> Review LSI, Membuat Dokumentasi Untuk <i>Test Case</i> App Transtrack Web: Setup Events, SMS gateway, Show Point, Show Address, Send Command, Forgot Password.</p>																				
8	<p><i>Testing</i> LSI Apical, Membuat Test data untuk <i>Product Owner</i>, Membuat <i>Test Case</i> LSI Role SC/PO Module <i>Invoice Vendor</i> <i>Testing</i> card LSI Ready To Test fitur <i>Billing</i> Report Bulanan Team <i>Product</i> Intern, Membuat <i>Test Case</i> LSI Role <i>Vendor: Invoice Shipment, Invoice Order Shipment, Billing Shipment</i>.</p>																				
9	<p>Mempelajari ERD, Flowchart, DFD <i>Order Planning, Testing</i> API <i>Order Planning</i> Corporates, Depots, Goods, Drivers, Orders, Entities, Status, Epods, Schedule, Assignment, Membuat Dokumentasi API <i>Testing</i> <i>Order Planning</i> Drivers, Schedule, Assignment, <i>Testing</i> LSI Apical</p>																				
10	<p>Mempelajari <i>Automation Testing</i> WDIO, <i>Testing</i> LSI Apical role LP, <i>Vendor</i> fitur <i>Create Shipment, Add Driver, Retesting</i> LSI Apical, Mempelajari <i>Automation Testing</i> Tools Katalon.</p>																				