

## **BAB 3**

### **PELAKSANAAN KERJA MAGANG**

#### **3.1 Kedudukan dan Koordinasi**

Selama pelaksanaan kegiatan magang di PT Asuransi Simas Jiwa, tanggung jawab yang diberikan berfokus pada kegiatan pengecekan serta pengujian manual terhadap fitur, menu, dan aplikasi internal berbasis desktop yang digunakan oleh perusahaan. Kegiatan pengujian dilakukan untuk memastikan bahwa setiap fungsi yang dikembangkan atau diperbarui telah berjalan sesuai kebutuhan operasional dan standar kualitas yang ditetapkan. Selain pengujian manual, dilakukan pula kegiatan eksplorasi awal mengenai penerapan otomasi pengujian pada aplikasi internal berbasis Android untuk menilai potensi implementasinya di lingkungan perusahaan. Penjabaran lebih rinci mengenai bentuk tanggung jawab dan aktivitas koordinasi selama kegiatan magang disampaikan pada bagian berikutnya.

##### **3.1.1 Kedudukan**

Kegiatan magang dilaksanakan di PT Asuransi Simas Jiwa dengan posisi sebagai *Quality Assurance Intern* yang berada di bawah divisi *Quality Control*, sebagaimana ditunjukkan pada Gambar 2.1. Dalam posisi ini, terdapat sejumlah tanggung jawab utama, antara lain:

- Dilakukan pemantauan dan pengingat terhadap tiket atau *timeline* setiap proyek kepada tim QC untuk memastikan bahwa pelaksanaan pengujian berlangsung sesuai rencana serta sinkron dengan hasil rapat internal perusahaan.
- Disusun skrip pengujian serta skenario uji untuk mendukung proses pengujian manual pada aplikasi desktop, termasuk pendokumentasian *evidence* dan *error log* apabila ditemukan ketidaksesuaian selama pelaksanaan pengujian.
- Dilaksanakan pengujian *User Acceptance Test* (UAT) secara manual terhadap aplikasi desktop internal guna memastikan bahwa fungsionalitas sistem telah sesuai dengan kebutuhan operasional perusahaan.

- Dilakukan eksplorasi awal terhadap penerapan otomasi pengujian pada aplikasi internal berbasis Android, termasuk peninjauan alat otomasi yang berpotensi digunakan serta identifikasi peluang dan kendala teknis dalam implementasinya.

### 3.1.2 Koordinasi

Koordinasi dilakukan dengan beberapa pihak, meliputi supervisi, tim pengembang (*Developer*) serta *Business Analyst* (BA), dan rekan satu tim pada divisi QC. Koordinasi dengan supervisi dilakukan secara langsung untuk membahas pembagian tugas serta tanggung jawab yang akan diemban. Sementara itu, koordinasi dengan tim pengembang dan BA dilakukan melalui diskusi, rapat, atau perencanaan terkait proyek baru yang akan dikembangkan. Selain itu, dilakukan pula koordinasi dengan rekan satu tim pada divisi QC untuk saling bertukar pendapat dan memberikan saran terhadap proyek yang sedang maupun akan dikerjakan.

## 3.2 Tugas yang Dilakukan

Pelaksanaan kegiatan magang dilaksanakan secara bertahap dan terstruktur setiap minggu sesuai dengan rencana kerja yang telah ditetapkan oleh perusahaan. Setiap aktivitas yang dilakukan bertujuan untuk mendukung proses pembelajaran serta pencapaian kompetensi di bidang pengujian perangkat lunak. Rincian proyek dan aktivitas mingguan pembelajaran selama pelaksanaan kegiatan magang disajikan secara ringkas dan sistematis pada Tabel 3.1.

Tabel 3.1. Rincian proyek dan aktivitas mingguan pembelajaran

Minggu Ke	Aktivitas yang Dilakukan	Deskripsi Aktivitas
1	Pengenalan dan briefing	Pengenalan terhadap divisi Teknologi Informasi dan anggota tim, penyiapan koneksi jaringan kantor, aktivasi email perusahaan, serta <i>briefing</i> deskripsi pekerjaan.
Lanjut pada halaman berikutnya		

Tabel 3.1 Rincian proyek dan aktivitas mingguan pembelajaran (lanjutan)

Minggu Ke	Aktivitas yang Dilakukan	Deskripsi Aktivitas
2	Pengenalan terhadap dokumen pendukung dan lingkungan kerja proyek	Membaca dan mempelajari dokumen pendukung proyek seperti <i>Business Requirement Document</i> (BRD) dan mempelajari proses pembuatan skenario pengujian dari proyek yang sudah berjalan.
3	Pendalaman terhadap <i>Business Requirement Document</i> (BRD) dan lingkungan kerja divisi TI	Pemahaman istilah dan alur BRD serta pengenalan lingkungan kerja divisi Teknologi Informasi melalui <i>reminder</i> harian pada sistem <i>helpdesk</i> internal perusahaan.
4	<i>Grooming</i> tiket #26854, penyusunan skenario pengujian, dan penyiapan lingkungan pengujian	Pelaksanaan <i>grooming</i> tiket melalui <i>Google Meet</i> , dilanjutkan dengan penyusunan skenario pengujian untuk kebutuhan validasi fungsionalitas sistem.
5	<i>Grooming</i> tiket #30570 dan penyusunan skenario pengujian	Pelaksanaan <i>grooming</i> tiket #30570 melalui <i>Google Meet</i> dan penyusunan skenario pengujian.
6	<i>Grooming</i> tiket #30587, penyusunan skenario pengujian, dan penyiapan lingkungan pengujian	Pelaksanaan <i>grooming</i> tiket #30587 melalui <i>Google Meet</i> , penyusunan skenario pengujian, serta penyiapan lingkungan pengujian.
7	<i>Grooming</i> tiket #30621 dan penyusunan skenario pengujian	Pelaksanaan <i>grooming</i> tiket #30621 melalui <i>Google Meet</i> dan penyusunan skenario pengujian.
8	Eksplorasi dan penerapan awal <i>Katalon Studio</i>	Instalasi dan konfigurasi <i>Katalon Studio</i> sebagai persiapan penerapan <i>User Acceptance Test</i> (UAT).
Lanjut pada halaman berikutnya		

Tabel 3.1 Rincian proyek dan aktivitas mingguan pembelajaran (lanjutan)

Minggu Ke	Aktivitas yang Dilakukan	Deskripsi Aktivitas
9	Eksplorasi dan penerapan awal <i>LambdaTest</i>	Eksplorasi dan percobaan <i>LambdaTest</i> sebagai persiapan penerapan UAT pada aplikasi internal perusahaan berbasis Android.
10	Perbandingan <i>Katalon Studio</i> dan <i>LambdaTest</i> sebagai tools otomasi	Perbandingan <i>Katalon Studio</i> dan <i>LambdaTest</i> sebagai kandidat tools otomasi pengujian perusahaan.
11	Pencobaan penggunaan <i>Katalon</i>	Penerapan awal penggunaan <i>Katalon Studio</i> sebagai platform otomasi pengujian.
12	Penerapan <i>Katalon Studio</i>	Penerapan <i>Katalon Studio</i> dalam pelaksanaan proyek berjalan pada aplikasi internal perusahaan berbasis desktop.
13	Pengerjaan tiket #26854	Pengerjaan tiket berdasarkan hasil <i>grooming</i> dan skenario pengujian yang telah disusun.
14	Pengerjaan tiket #30570	Pengerjaan tiket dengan menerapkan pengujian otomatis berbasis <i>Katalon</i> .
15	Pengerjaan tiket #30621	Pengerjaan tiket dengan menerapkan pengujian otomatis berbasis <i>Katalon</i> .

### 3.2.1 Penjelasan Tabel Tugas Pelaksanaan Kerja Magang

Tabel 3.1 menyajikan ringkasan mengenai berbagai tugas yang dilaksanakan selama kegiatan magang, yang diklasifikasikan berdasarkan minggu pelaksanaan. Setiap aktivitas yang tercantum mencerminkan tahapan persiapan awal hingga pelaksanaan proses pengujian. Uraian lebih mendetail mengenai pelaksanaan setiap tugas akan disampaikan pada subbab ini guna memberikan pemahaman yang lebih

komprehensif terhadap langkah-langkah serta tanggung jawab yang dijalankan selama masa magang.

Selama pelaksanaan kegiatan magang, aktivitas disusun berdasarkan pembagian minggu pelaksanaan. Pada minggu pertama hingga minggu kedua, kegiatan difokuskan pada tahap persiapan perangkat kerja serta pengenalan terhadap lingkungan perusahaan dan lingkungan kerja proyek yang menjadi fokus kegiatan.

Memasuki minggu ketiga, kegiatan diarahkan pada pendalaman serta pemahaman terhadap dokumen pendukung proyek, seperti *Business Requirement Document* (BRD) dan dokumen tambahan lainnya, termasuk data uji dalam bentuk *spreadsheet* serta skenario pengujian yang akan digunakan pada tahap pengujian berikutnya.

Pada minggu keempat hingga minggu ketujuh, kegiatan difokuskan pada pelaksanaan diskusi perencanaan proyek atau yang dikenal dengan istilah *grooming*. Pada periode ini pula mulai dilakukan penyusunan skenario pengujian untuk setiap proyek pengerjaan yang selanjutnya direpresentasikan dalam bentuk *ticket* pengujian.

Selanjutnya, pada minggu kedelapan, dilakukan kegiatan eksplorasi dan penerapan awal terhadap *Katalon Studio* pada aplikasi internal perusahaan berbasis Android. Aktivitas ini bertujuan untuk melakukan percobaan awal pengujian dengan memanfaatkan fitur otomasi yang tersedia pada *Katalon Studio*, serta untuk memahami alur kerja otomasi pengujian yang dapat diterapkan pada lingkungan perusahaan.

Pelaksanaan kegiatan pada minggu kesembilan difokuskan pada proses eksplorasi awal terhadap platform LambdaTest sebagai salah satu kandidat tools otomasi yang berpotensi digunakan dalam pengujian aplikasi internal perusahaan. Aktivitas yang dilakukan mencakup peninjauan fitur-fitur utama, percobaan penggunaan antarmuka, serta pengujian sederhana pada aplikasi Android untuk memahami mekanisme eksekusi pengujian, pengelolaan perangkat virtual, dan kemampuan integrasi platform. Kegiatan eksplorasi ini ditujukan untuk memperoleh gambaran teknis mengenai kelayakan LambdaTest sebagai solusi otomasi, sekaligus sebagai langkah persiapan sebelum penerapannya pada proses *User Acceptance Test* (UAT) di lingkungan perusahaan.

Pada minggu kesepuluh, kegiatan diarahkan pada penyusunan analisis perbandingan antara *Katalon Studio* dan *LambdaTest* sebagai dua platform otomasi yang sedang dipertimbangkan oleh perusahaan. Proses perbandingan dilakukan

dengan meninjau aspek kemampuan teknis, kemudahan penggunaan, dukungan terhadap perangkat Android, potensi integrasi dengan alur kerja tim QC, serta efektivitasnya dalam meningkatkan efisiensi proses pengujian. Melalui analisis tersebut, diperoleh pemahaman awal mengenai kelebihan dan keterbatasan masing-masing tools sebagai dasar pertimbangan dalam pemilihan platform otomasi yang paling sesuai dengan kebutuhan perusahaan.

Mulai minggu ke sebelas dan seterusnya, kegiatan magang difokuskan pada pelaksanaan pengerjaan *tiket* pengujian dengan menerapkan otomasi menggunakan *Katalon Studio*, sebagaimana telah dieksplorasikan pada tahap sebelumnya. Pada tahap ini, skenario pengujian yang telah disusun diimplementasikan ke dalam skrip otomasi untuk mendukung proses pengujian aplikasi internal perusahaan. Penerapan otomasi ini dilakukan sebagai upaya meningkatkan efisiensi dan konsistensi pelaksanaan pengujian, sekaligus sebagai bentuk evaluasi awal terhadap efektivitas penggunaan *Katalon Studio* dalam mendukung proses QC di lingkungan PT Asuransi Simas Jiwa.

### 3.2.2 Proses Pengembangan Perangkat Lunak (SDLC)

Proses pengembangan perangkat lunak pada perusahaan dilaksanakan dengan mengacu pada tahapan *Software Development Life Cycle* (SDLC). SDLC merupakan kerangka kerja yang digunakan untuk mengatur proses pengembangan sistem secara terstruktur agar hasil yang diperoleh sesuai dengan kebutuhan bisnis dan standar kualitas yang ditetapkan. Penerapan SDLC juga membantu meminimalkan risiko kesalahan, meningkatkan efisiensi pengembangan, serta memastikan setiap tahap dilakukan secara terdokumentasi dan terkontrol.

Tahapan pertama yaitu *Planning*, yang dilakukan oleh pihak pengguna (*User*) untuk menentukan kebutuhan sistem yang akan dikembangkan. Pada tahap ini, ditetapkan ruang lingkup, tujuan, serta prioritas pengembangan agar proses berikutnya dapat berjalan secara terarah.

Tahapan selanjutnya yaitu *Requirement Analysis*, yang dilakukan oleh *Business Analyst* (BA). Pada tahap ini, dilakukan proses pengumpulan serta analisis kebutuhan dari pengguna untuk diterjemahkan menjadi spesifikasi fungsional dan nonfungsional yang akan menjadi dasar dalam perancangan sistem.

Berikutnya, pada tahap *System Design*, dilakukan perancangan arsitektur sistem berdasarkan hasil analisis kebutuhan. Rancangan ini mencakup desain basis data, struktur sistem, serta antarmuka pengguna yang akan digunakan.

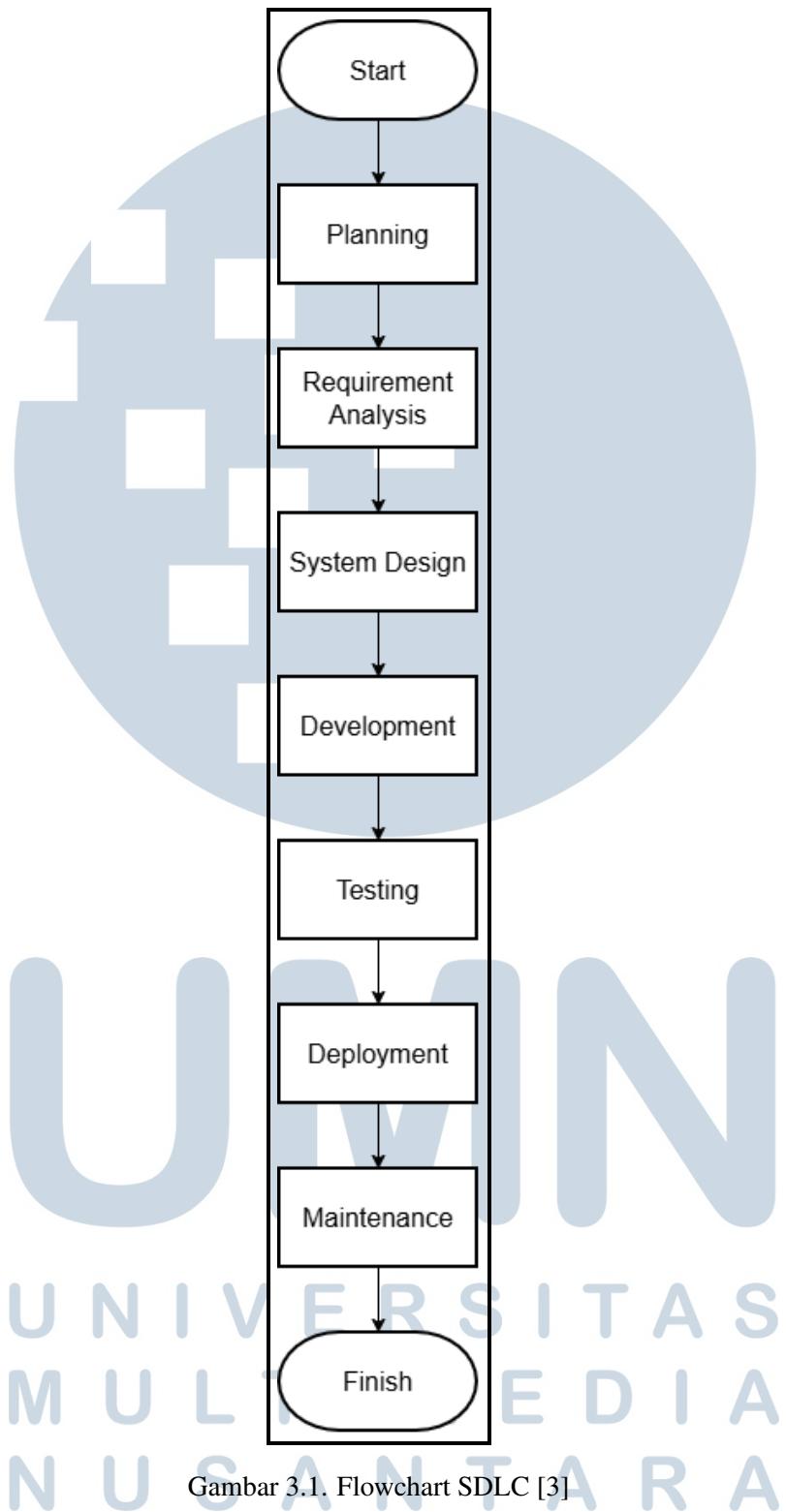
Tahap *Development* dilaksanakan oleh tim pengembang (*Developer*). Proses ini memastikan setiap komponen sistem dikembangkan sesuai spesifikasi dan dapat berfungsi secara optimal.

Setelah tahap pengembangan selesai, dilakukan tahap *Testing* yang menjadi fokus utama kegiatan magang. Pada tahap ini, dilakukan pengujian terhadap fungsi-fungsi sistem untuk memastikan bahwa sistem berjalan sesuai kebutuhan serta bebas dari kesalahan (*bug*). Pengujian dilakukan dengan menggunakan pendekatan manual maupun otomatisasi, salah satunya dengan memanfaatkan Katalon Studio sebagai alat bantu uji.

Selanjutnya, tahap *Deployment* dilaksanakan oleh tim pengembang dengan tujuan untuk mengimplementasikan sistem yang telah lolos pengujian ke lingkungan produksi. Proses ini mencakup instalasi, konfigurasi, serta penyesuaian agar sistem dapat digunakan oleh pengguna.

Tahap terakhir yaitu *Maintenance*, di mana dilakukan pemeliharaan sistem secara berkelanjutan oleh tim terkait untuk menjaga stabilitas dan performa sistem. Kegiatan ini mencakup pembaruan sistem, perbaikan kesalahan yang ditemukan pasca implementasi, serta penyesuaian terhadap kebutuhan baru dari pengguna.





Gambar 3.1. Flowchart SDLC [3]

Sumber: Dokumen internal PT Asuransi Simas Jiwa, 2025

### **3.2.3 Alur Proses Pengujian (UAT)**

Proses *User Acceptance Testing* (UAT) merupakan tahapan akhir dalam siklus pengujian perangkat lunak sebelum sistem dinyatakan siap digunakan oleh pengguna akhir. UAT dilakukan untuk memastikan bahwa sistem telah berfungsi sesuai dengan kebutuhan bisnis dan memenuhi kriteria penerimaan yang telah ditetapkan. Alur pelaksanaan UAT dijelaskan melalui flowchart pada Gambar 3.2, yang terdiri atas beberapa tahapan sebagai berikut.

Tahap pertama yaitu *Preparation*, di mana dilakukan proses persiapan lingkungan pengujian dan seluruh kebutuhan pendukung sebelum pelaksanaan UAT dimulai. Kegiatan pada tahap ini mencakup instalasi aplikasi uji, persiapan data pengujian, serta pengecekan akses akun yang digunakan selama proses pengujian berlangsung.

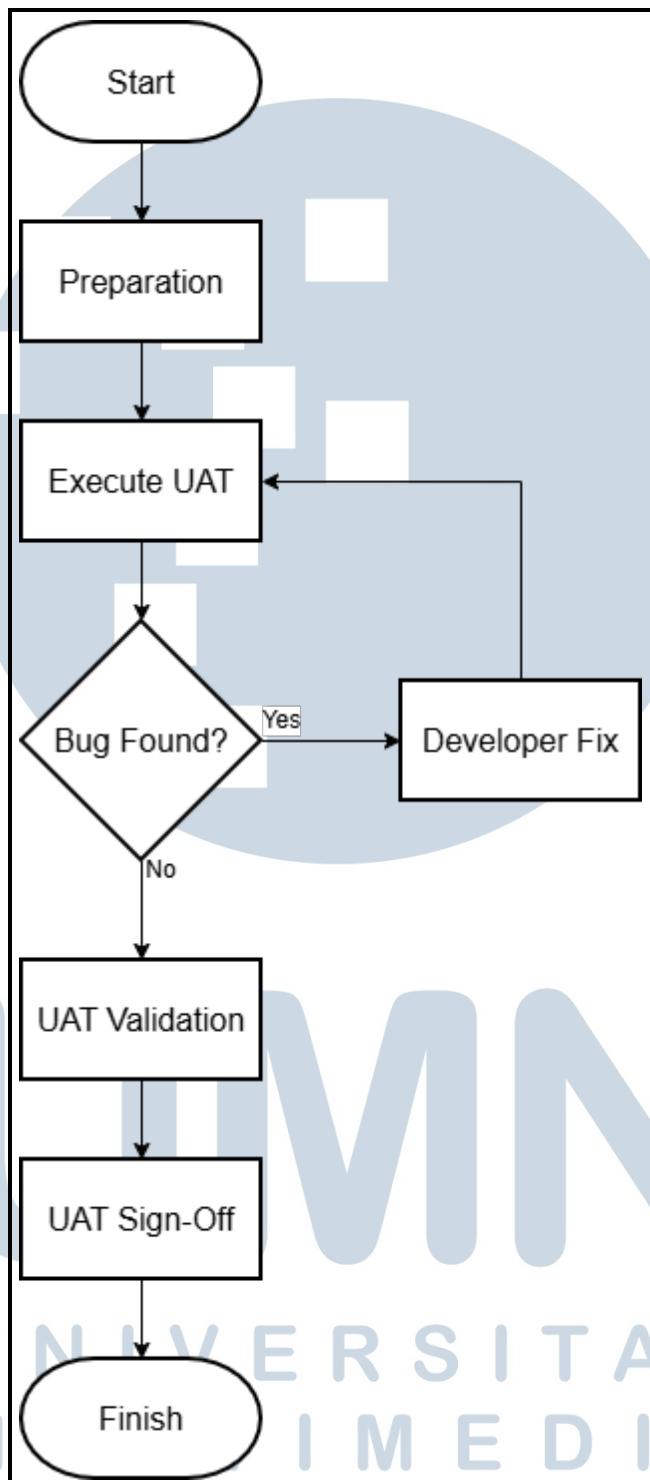
Tahapan berikutnya adalah *Execute UAT*, yaitu pelaksanaan proses pengujian berdasarkan skenario yang telah disusun sebelumnya. Pada tahap ini, setiap fungsi atau fitur diuji untuk memastikan kesesuaianya dengan dokumen kebutuhan bisnis (*Business Requirement Document*). Pengujian dilakukan secara sistematis menggunakan data uji yang telah disiapkan untuk memperoleh hasil yang valid dan terukur.

Selanjutnya, dilakukan proses evaluasi melalui tahap *Bug Found?*. Tahap ini berfungsi untuk menentukan apakah selama pelaksanaan UAT ditemukan kesalahan atau ketidaksesuaian antara hasil aktual sistem dan hasil yang diharapkan. Apabila ditemukan bug, maka proses diarahkan menuju tahapan *Developer Fix*.

Tahapan *Developer Fix* merupakan proses yang dilakukan oleh tim pengembang (*developer team*) untuk memperbaiki kesalahan atau bug yang dilaporkan oleh tim penguji. Setelah perbaikan selesai dilakukan, versi terbaru dari aplikasi dikembalikan kepada tim penguji untuk dilakukan pengujian ulang.

Apabila tidak ditemukan bug atau setelah perbaikan dinyatakan berhasil, maka proses dilanjutkan ke tahap *UAT Validation*. Pada tahap ini dilakukan verifikasi terhadap hasil pengujian untuk memastikan bahwa seluruh fungsi sistem telah berjalan dengan benar sesuai dengan kebutuhan yang telah ditetapkan.

Tahapan berikutnya adalah *UAT Sign-Off*, yaitu proses persetujuan akhir terhadap hasil pengujian. Pada tahap ini, tim pengguna atau perwakilan bisnis memberikan konfirmasi bahwa sistem telah memenuhi seluruh kriteria penerimaan dan siap untuk diimplementasikan pada lingkungan produksi.



Gambar 3.2. Flowchart UAT [3]  
Sumber: Dokumen internal PT Asuransi Simas Jiwa, 2025

### **3.3 Uraian Pelaksanaan Magang**

Section ini menguraikan pelaksanaan kegiatan kerja magang yang dilakukan selama periode magang berlangsung. Uraian disusun untuk memberikan gambaran menyeluruh mengenai aktivitas, proses, serta pengalaman yang diperoleh dalam menjalankan tugas dan tanggung jawab di perusahaan. Pembahasan diawali dengan kegiatan eksplorasi platform otomasi sebagai tahap awal untuk memahami tools, alur kerja, dan teknologi yang digunakan dalam mendukung proses pengujian perangkat lunak.

#### **3.3.1 Eksplorasi Platform Otomasi**

Mengacu pada Tabel 3.1, pada minggu ke-8 pelaksanaan magang telah ditetapkan adanya kegiatan eksplorasi terhadap platform otomasi yang umum serta banyak digunakan dalam lingkungan kerja pengujian perangkat lunak. Kegiatan ini dilaksanakan sebagai bagian dari upaya untuk memperoleh pemahaman yang komprehensif mengenai karakteristik, kemampuan, serta potensi penerapan berbagai platform otomasi dalam mendukung proses *software testing*. Melalui eksplorasi tersebut, diharapkan dapat diperoleh gambaran awal mengenai kesesuaian platform otomasi dengan kebutuhan dan standar pengujian yang diterapkan di perusahaan.

Berdasarkan hasil penelusuran awal melalui berbagai sumber referensi dan mesin pencarian, selanjutnya ditetapkan dua platform otomasi utama yang dinilai paling sering digunakan dan banyak direkomendasikan, yaitu LambdaTest dan Katalon Studio. Penetapan kedua platform tersebut didasarkan pada tingkat popularitas, cakupan fitur, serta relevansi penggunaannya dalam pengujian aplikasi modern. Dengan difokuskannya eksplorasi pada dua platform ini, proses evaluasi diharapkan dapat dilakukan secara lebih terarah dan mendalam.

##### **A Eksplorasi Platform LambdaTest**

Proses eksplorasi terhadap platform LambdaTest dilaksanakan dalam kurun waktu kurang lebih dua minggu dan dilakukan secara paralel dengan eksplorasi pada platform Katalon Studio. Selama periode tersebut, dilakukan pengamatan terhadap antarmuka, alur penggunaan, serta fitur-fitur utama yang disediakan oleh LambdaTest, khususnya yang berkaitan dengan pengujian lintas peramban dan integrasi berbasis *cloud*. Pelaksanaan eksplorasi secara paralel ini bertujuan

untuk mempermudah proses perbandingan antara kedua platform, terutama dari sisi kemudahan penggunaan dan efektivitas penerapannya.

Selain itu, LambdaTest memiliki antarmuka pengguna (*user interface*) yang modern dan intuitif sehingga memudahkan proses adaptasi dalam penggunaan platform. Seluruh aktivitas pengujian, mulai dari konfigurasi lingkungan, pemilihan peramban dan sistem operasi, hingga pemantauan hasil pengujian, disediakan dalam satu lingkungan kerja terintegrasi berbasis *cloud*. Pendekatan ini memungkinkan proses pengujian dilakukan secara lebih efisien tanpa memerlukan pengaturan infrastruktur tambahan secara manual. Kemudahan pengelolaan lingkungan pengujian dalam satu platform terpadu tersebut dinilai mampu mendukung efektivitas pelaksanaan pengujian lintas peramban serta meningkatkan produktivitas selama proses eksplorasi berlangsung.



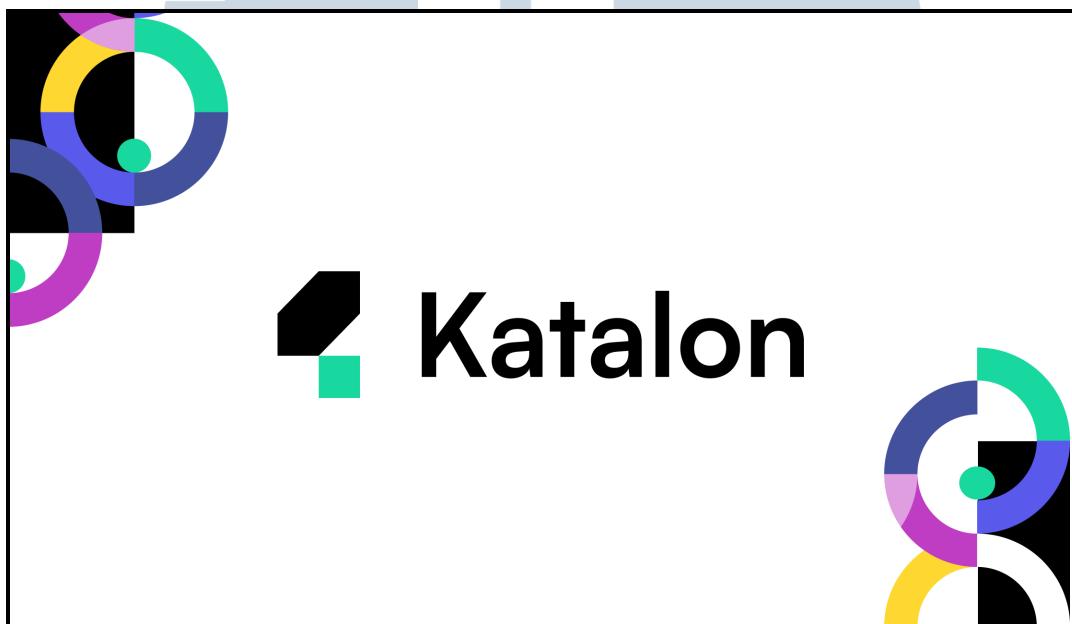
Gambar 3.3. Logo platform LambdaTest [4]

Sumber: Website resmi platform LambdaTest, 2025

## B Eksplorasi Platform Katalon Studio

Sementara itu, proses eksplorasi terhadap platform Katalon Studio dilaksanakan dalam rentang waktu yang relatif lebih panjang, yaitu selama kurang lebih tiga minggu. Pada tahap ini, dilakukan percobaan penerapan berbagai fitur serta keunggulan yang dimiliki oleh Katalon Studio, termasuk dukungan terhadap otomasi pengujian berbasis skrip maupun non-skrip. Selain itu, dilakukan pula pengujian terhadap stabilitas, fleksibilitas, serta kemudahan dalam pengelolaan proyek pengujian.

Setelah melalui rangkaian proses eksplorasi, diskusi internal, serta pelaksanaan sesi demonstrasi dari kedua platform otomasi pengujian, dilakukan evaluasi secara menyeluruh terhadap hasil yang diperoleh. Berdasarkan hasil evaluasi tersebut, disimpulkan bahwa Katalon Studio dinilai lebih sesuai dan sejalan dengan kebutuhan pengujian yang dimiliki oleh perusahaan. Oleh karena itu, diputuskan untuk melanjutkan penggunaan Katalon Studio sebagai platform utama dalam penerapan otomasi pengujian, guna mendukung proses pengujian aplikasi secara lebih efektif, terstruktur, dan berkelanjutan.



Gambar 3.4. Logo platform Katalon Studio [5]

Sumber: Website resmi platform Katalon Studio, 2025

### 3.3.2 Keputusan Penggunaan Platform Otomasi dalam *Software Testing*

Setelah dilaksanakan proses eksplorasi serta ditetapkan kesepakatan untuk menggunakan Katalon Studio, selanjutnya dilakukan percobaan penerapan platform tersebut secara langsung ke dalam proyek yang sedang berjalan maupun proyek yang direncanakan untuk dilaksanakan. Penerapan ini dilakukan untuk menilai tingkat kesesuaian Katalon Studio terhadap alur kerja pengujian yang telah diterapkan di perusahaan, sekaligus untuk mengamati dampak penggunaan otomasi terhadap efektivitas dan efisiensi proses pengujian secara keseluruhan [6].

Berdasarkan hasil penerapan awal tersebut, ditemukan bahwa penggunaan platform otomasi memiliki karakteristik kelebihan dan kekurangan tersendiri

apabila dibandingkan dengan metode pengujian manual. Perbedaan tersebut terlihat dari berbagai aspek, seperti waktu pelaksanaan pengujian, konsistensi hasil, fleksibilitas terhadap perubahan, serta kebutuhan sumber daya. Oleh karena itu, dilakukan identifikasi terhadap kekuatan dan kelemahan otomasi testing sebagai dasar pertimbangan dalam menentukan strategi pengujian yang paling sesuai [7].

Salah satu kekuatan utama dari pengujian berbasis otomasi terletak pada kemampuannya dalam meningkatkan efisiensi waktu, khususnya pada pengujian yang bersifat berulang dan membutuhkan eksekusi dalam jumlah besar. Dengan menggunakan otomasi, proses pengujian dapat dijalankan secara konsisten tanpa memerlukan intervensi manual secara terus-menerus. Selain itu, tingkat konsistensi hasil pengujian dapat lebih terjaga karena skenario pengujian dijalankan dengan langkah yang sama pada setiap eksekusi [8].

Kekuatan lain dari otomasi testing juga terlihat pada kemampuannya untuk mendukung pengujian regresi secara lebih efektif. Setiap perubahan atau pembaruan pada aplikasi dapat diuji kembali dengan cepat untuk memastikan bahwa fungsionalitas yang telah ada sebelumnya tidak mengalami gangguan. Hal ini dinilai sangat membantu pada proyek dengan siklus pembaruan yang cepat [9].

Meskipun demikian, pengujian berbasis otomasi juga memiliki sejumlah kelemahan apabila dibandingkan dengan pengujian manual. Salah satu kelemahan utama terletak pada kebutuhan waktu dan usaha yang lebih besar pada tahap awal pengembangan skrip pengujian. Proses pembuatan, pemeliharaan, serta penyesuaian skrip otomasi memerlukan pemahaman teknis yang memadai, terutama ketika terjadi perubahan pada antarmuka atau alur aplikasi yang diuji [10]. Selain itu, pengujian berbasis otomasi dinilai memiliki keterbatasan dalam beberapa kondisi, khususnya ketika objek dan elemen pada aplikasi tidak dapat terbaca atau tidak terdeteksi secara optimal pada saat pembuatan skrip pengujian. Dalam kondisi tersebut, proses pembuatan dan eksekusi skrip pengujian menjadi kurang optimal, sehingga pengujian manual masih diperlukan sebagai alternatif untuk memastikan fungsionalitas aplikasi dapat diuji secara menyeluruh [11].

Berdasarkan pemaparan tersebut, dapat disimpulkan bahwa pengujian berbasis otomasi dan pengujian manual memiliki peran yang saling melengkapi. Penggunaan otomasi dinilai lebih optimal untuk pengujian yang berulang dan terstruktur, sementara pengujian manual tetap dibutuhkan untuk skenario yang memerlukan fleksibilitas dan evaluasi kualitatif. Oleh karena itu, kombinasi antara kedua metode tersebut dianggap sebagai pendekatan yang paling efektif dalam mendukung kualitas pengujian aplikasi di perusahaan [12].

### **3.3.3 Kebutuhan Pengujian Manual pada Lingkungan Otomasi**

Pada lingkungan pengujian yang telah memanfaatkan otomasi, pengujian manual tetap memiliki peran yang penting dan tidak dapat dihilangkan sepenuhnya. Pengujian manual diperlukan terutama pada skenario yang menuntut penilaian berbasis persepsi manusia, seperti kenyamanan penggunaan, kejelasan tampilan antarmuka, serta kesesuaian alur sistem dengan cara kerja pengguna. Aspek-aspek tersebut tidak hanya berkaitan dengan keberfungsiannya teknis, tetapi juga dengan kualitas pengalaman pengguna, sehingga tidak dapat dievaluasi secara memadai melalui eksekusi skrip otomasi.

Pengujian manual juga dibutuhkan pada tahap awal pengembangan fungsionalitas baru, ketika spesifikasi sistem masih sering mengalami perubahan dan stabilitas aplikasi belum sepenuhnya tercapai. Pada kondisi ini, pendekatan eksploratif lebih sesuai digunakan untuk memahami perilaku sistem secara keseluruhan sebelum skenario pengujian diformalkan ke dalam skrip otomasi. Melalui pengujian manual, potensi kesalahan yang tidak terantisipasi dalam desain skenario uji dapat diidentifikasi lebih dini.

Dari sisi pengendalian kualitas, pengujian manual berperan sebagai mekanisme verifikasi tambahan terhadap hasil pengujian otomasi. Hasil eksekusi otomasi tetap memerlukan konfirmasi secara manual, terutama ketika ditemukan temuan yang meragukan, hasil yang ambigu, atau ketidaksesuaian yang berkaitan dengan aturan bisnis. Dengan demikian, pengujian manual tidak hanya berfungsi sebagai alternatif ketika otomasi tidak dapat diterapkan, tetapi juga sebagai lapisan pengujian tambahan untuk meningkatkan keandalan proses jaminan kualitas perangkat lunak.

Peran pengujian manual juga menonjol pada pelaksanaan *User Acceptance Testing (UAT)*, yaitu tahap pengujian yang melibatkan pengguna akhir untuk memastikan bahwa sistem telah sesuai dengan kebutuhan operasional. Pada tahap ini, penilaian tidak hanya berfokus pada keberhasilan fungsi, tetapi juga pada kemudahan penggunaan dan kesesuaian sistem dengan prosedur kerja yang berlaku. Oleh karena itu, meskipun otomasi pengujian telah diimplementasikan, pengujian manual tetap dibutuhkan sebagai bagian integral dari proses pengujian perangkat lunak secara menyeluruh.

### **3.3.4 Penerapan Platform Otomasi Katalon Pada Aplikasi Internal Perusahaan**

Bagian ini membahas penerapan platform otomasi Katalon dalam proses pengujian aplikasi internal perusahaan. Pembahasan difokuskan pada implementasi otomasi pengujian sebagai upaya untuk meningkatkan efisiensi, konsistensi, dan akurasi dalam pengujian perangkat lunak. Secara khusus, penjelasan diawali dengan penerapan otomasi pengujian menggunakan Katalon pada aplikasi *Simas Salesforce (SSF)* sebagai salah satu sistem internal yang digunakan oleh perusahaan, kemudian dilanjutkan dengan aplikasi *Life Insurance InfOrmation System (Lions)* dan *MoRtgage Insurance (MRI)*.

Dalam proses penerapan otomasi tersebut, rangkaian instruksi pengujian disusun secara terstruktur sebagai pedoman pelaksanaan pengujian pada masing-masing aplikasi. Rangkaian instruksi ini memuat skenario uji yang harus dijalankan, data uji yang digunakan, serta hasil yang diharapkan pada setiap tahapan pengujian. Instruksi tersebut tidak hanya digunakan pada pengujian manual, tetapi juga diimplementasikan dalam bentuk otomasi pada Katalon Studio sehingga langkah pengujian dapat dieksekusi secara konsisten pada setiap pengujian yang dilakukan.

Pada pelaksanaan pengujian manual, rangkaian instruksi tersebut dituangkan ke dalam bentuk *test case* yang memuat deskripsi skenario uji, prasyarat, langkah pelaksanaan, data uji, serta kriteria keberhasilan (*expected result*). Sementara itu, pada pengujian berbasis otomasi, rangkaian instruksi yang sama direalisasikan dalam bentuk skrip pada Katalon Studio yang memungkinkan eksekusi pengujian dilakukan secara otomatis sesuai skenario yang telah dirancang sebelumnya. Dengan penerapan rangkaian instruksi pengujian yang terstruktur tersebut, proses pengujian pada aplikasi internal perusahaan menjadi lebih sistematis, terdokumentasi dengan baik, serta mudah direplikasi pada siklus pengujian berikutnya.

#### **A Simas Salesforce (SSF)**

Aplikasi *Simas Salesforce (SSF)* merupakan sistem yang digunakan untuk mendukung proses pencatatan dan pengelolaan data pemegang polis dan tertanggung secara terstruktur. Aplikasi ini berfungsi sebagai sarana utama dalam melakukan input data pemegang polis dan tertanggung untuk pertama kalinya ke

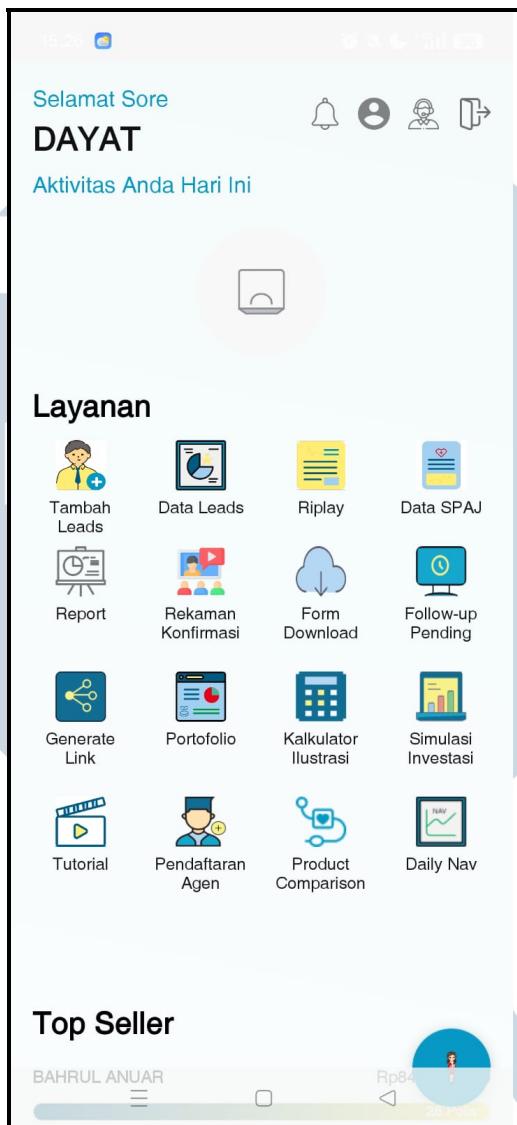
dalam sistem perusahaan, sehingga informasi awal pemegang polis dan tertanggung dapat terdokumentasi secara lengkap, konsisten, dan terintegrasi.

Melalui Simas Salesforce, proses penambahan data dilakukan dengan mengisi berbagai informasi dasar yang dibutuhkan, seperti identitas pribadi, data kontak, serta informasi pendukung lainnya sesuai dengan ketentuan yang berlaku. Data yang dimasukkan akan divalidasi oleh sistem guna memastikan kelengkapan dan kesesuaian informasi sebelum disimpan ke dalam basis data. Dengan adanya mekanisme ini, risiko kesalahan pencatatan data dapat diminimalkan sejak tahap awal.

Selain digunakan untuk pendataan awal, aplikasi Simas Salesforce juga memfasilitasi proses pemilihan produk yang akan dibeli oleh pemegang polis. Setelah data pemegang polis berhasil ditambahkan, pengguna aplikasi dapat melanjutkan ke tahap pemilihan produk sesuai dengan kebutuhan dan preferensi. Proses ini dilakukan secara sistematis, sehingga keterkaitan antara data pemegang polis dan produk yang dipilih dapat tercatat dengan baik dalam sistem.

Dengan demikian, Simas Salesforce berperan penting dalam mendukung alur bisnis perusahaan, mulai dari registrasi awal nasabah hingga penentuan produk yang akan digunakan. Penerapan aplikasi ini diharapkan dapat meningkatkan efisiensi, akurasi data, serta kemudahan dalam pengelolaan informasi nasabah secara menyeluruh.

Pada aplikasi ini, proses otomasi difokuskan pada tahapan awal alur bisnis, yaitu *Tambah Leads*, *Data Leads*, hingga *Ripley*. Tahapan tersebut dipilih karena merupakan proses awal yang krusial dalam pencatatan calon pemegang polis ke dalam sistem. Adapun pengolahan data SPAJ (Surat Permintaan Asuransi Jiwa) belum termasuk dalam ruang lingkup otomasi pada periode ini dan direncanakan untuk dikerjakan pada tahap pengembangan selanjutnya. Dengan pembatasan ruang lingkup tersebut, proses otomasi diharapkan dapat memvalidasi kestabilan fitur utama sebelum dikembangkan ke tahapan lanjutan.

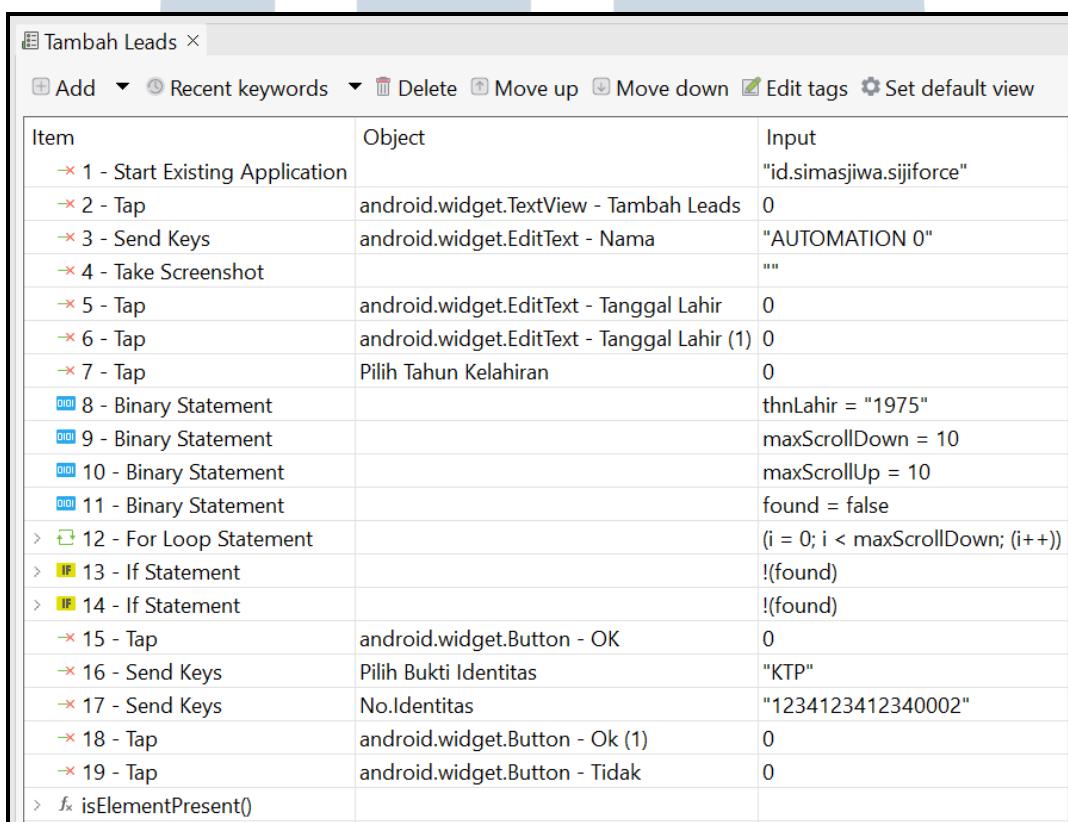


Gambar 3.5. Halaman utama SSF [3]

Sumber: Aplikasi internal PT Asuransi Simas Jiwa, 2025

Proses otomasi pada halaman *Tambah Leads* dilakukan dengan menggunakan skrip otomasi yang dirancang untuk mensimulasikan aktivitas pengguna dalam menambahkan data awal calon pemegang polis. Skrip ini mencakup proses pengisian informasi dasar, pemilihan opsi yang tersedia pada sistem, serta validasi awal terhadap data yang dimasukkan. Setiap langkah dieksekusi secara berurutan untuk memastikan bahwa fungsi penambahan leads dapat berjalan sesuai dengan kebutuhan bisnis dan aturan sistem yang berlaku. Dengan adanya otomasi pada tahap ini, proses pengujian dapat dilakukan secara berulang dan konsisten tanpa ketergantungan pada pengujian manual.

Selanjutnya, proses penambahan *Data Leads* hingga mencapai tahap *Ripley* juga dilakukan melalui skrip otomasi yang terpisah namun saling terintegrasi. Skrip ini dirancang untuk melanjutkan alur dari data leads yang telah berhasil dibuat, kemudian melakukan pengisian data lanjutan serta mengeksekusi proses yang mengarah ke tahapan *Ripley*. Seluruh skenario pengujian disusun untuk memastikan keterkaitan antar data tetap terjaga dan alur proses berjalan sesuai dengan desain aplikasi. Dengan penerapan otomasi pada tahapan ini, efisiensi pengujian meningkat serta potensi kesalahan akibat proses manual dapat diminimalkan.



The screenshot shows a table titled 'Tambah Leads' in Katalon Studio. The table has three columns: 'Item', 'Object', and 'Input'. The 'Item' column lists various test steps, some of which are expanded to show their details. The 'Object' column specifies the type of element being interacted with, such as 'android.widget.TextView' or 'android.widget.Button'. The 'Input' column contains the specific values or conditions for each step, including some Java-like code for loops and conditions.

Item	Object	Input
✓ 1 - Start Existing Application		"id.simasjiwa.sijiforce"
✓ 2 - Tap	android.widget.TextView - Tambah Leads	0
✓ 3 - Send Keys	android.widget.EditText - Nama	"AUTOMATION 0"
✓ 4 - Take Screenshot		...
✓ 5 - Tap	android.widget.EditText - Tanggal Lahir	0
✓ 6 - Tap	android.widget.EditText - Tanggal Lahir (1)	0
✓ 7 - Tap	Pilih Tahun Kelahiran	0
□ 8 - Binary Statement		thnLahir = "1975"
□ 9 - Binary Statement		maxScrollDown = 10
□ 10 - Binary Statement		maxScrollUp = 10
□ 11 - Binary Statement		found = false
> □ 12 - For Loop Statement		(i = 0; i < maxScrollDown; (i++))
> □ 13 - If Statement		!(found)
> □ 14 - If Statement		!(found)
✓ 15 - Tap	android.widget.Button - OK	0
✓ 16 - Send Keys	Pilih Bukti Identitas	"KTP"
✓ 17 - Send Keys	No.Identitas	"1234123412340002"
✓ 18 - Tap	android.widget.Button - Ok (1)	0
✓ 19 - Tap	android.widget.Button - Tidak	0
> fx isElementPresent()		

Gambar 3.6. Tampilan fitur tambah leads pada Katalon Studio

Skrip otomasi tersebut dikembangkan dengan mengombinasikan metode *coding* dan pendekatan *manual testing* atau *Record & Playback*. Penggunaan metode gabungan ini dilakukan untuk menyesuaikan dengan kondisi aplikasi serta kemampuan Katalon dalam mengenali dan mengeksekusi setiap elemen pada halaman yang diuji. Dengan pendekatan tersebut, proses pembuatan skenario pengujian dapat tetap berjalan secara efektif meskipun tidak seluruh elemen aplikasi dapat ditangani menggunakan satu metode saja.

Pemilihan metode antara *coding* dan manual testing atau *Record & Playback* didasarkan pada sejumlah kendala yang ditemukan selama penggunaan Katalon. Beberapa elemen pada aplikasi memiliki karakteristik dan cara kerja yang berbeda dari elemen pada umumnya, khususnya pada proses pencarian (*find*) dan *scroll until* terhadap elemen tertentu. Kondisi ini menyebabkan Katalon tidak selalu dapat membaca atau mengenali objek secara optimal apabila hanya mengandalkan metode otomatis standar.

Oleh karena itu, penggunaan manual testing atau *Record & Playback* hanya dapat diterapkan pada objek atau elemen yang berhasil dikenali dan dibaca oleh Katalon. Apabila elemen tersebut tidak dapat diidentifikasi dengan baik, maka dilakukan penulisan skrip atau kode tambahan sebagai pelengkap dalam penyusunan skenario pengujian. Penulisan kode ini bertujuan untuk mengeksekusi perintah tertentu atau membaca elemen yang tidak dapat ditangani secara langsung oleh Katalon akibat keterbatasan fitur atau mekanisme bawaan yang tersedia. Dengan strategi tersebut, skenario pengujian tetap dapat dijalankan secara menyeluruh dan sesuai dengan kebutuhan pengujian aplikasi.

Pada Gambar 3.7 ditampilkan salah satu contoh penerapan penulisan kode dalam skenario otomasi pada proses pembuatan *leads* baru. Kode tersebut digunakan secara khusus untuk mendukung proses pencarian dan pemilihan tahun yang diinginkan pada saat pengisian data diri, yaitu pada kolom tahun kelahiran pemegang polis. Penggunaan kode ini diperlukan karena mekanisme pemilihan tahun pada komponen antarmuka, sebagaimana ditunjukkan pada Gambar 3.8, tidak dapat sepenuhnya ditangani oleh fitur otomasi standar yang tersedia.

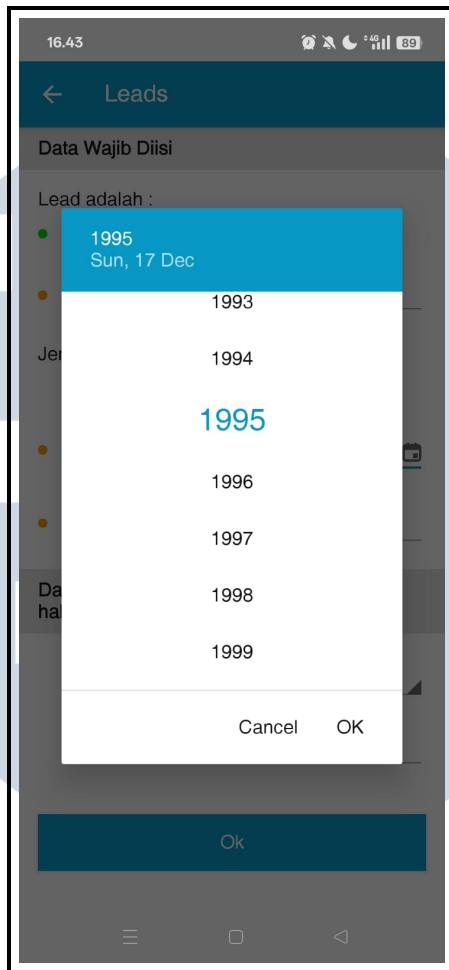
```

37 //Mobile.scrollToText('1985')
38 String thnLahir = '1975'
39
40 int maxScrollDown = 10
41
42 int maxScrollUp = 10
43
44 boolean found = false
45
46 // ↪ Fungsi cek elemen
47 // ↪ Scroll ke bawah terlebih dahulu
48 for (int i = 0; i < maxScrollDown; i++) {
49     if (isElementPresent(thnLahir)) {
50         Mobile.tap(new TestObject().addProperty('text', ConditionType.EQUALS, thnLahir), 5)
51
52         KeywordUtil.LogInfo('↗ Elemen ditemukan saat scroll ke bawah')
53
54         found = true
55
56         break
57     }
58
59     // Swipe dari atas ke bawah (scroll turun)
60     Mobile.swipe(500, 1200, 500, 1900)
61 }
62
63 // ↪ Jika belum ditemukan, scroll ke atas
64 if (!(found)) {
65     for (int i = 0; i < maxScrollUp; i++) {
66         if (isElementPresent(thnLahir)) {
67             Mobile.tap(new TestObject().addProperty('text', ConditionType.EQUALS, thnLahir), 5)
68
69             KeywordUtil.LogInfo('↗ Elemen ditemukan saat scroll ke atas')
70
71             found = true
72
73             break
74         }
75
76         // Swipe dari bawah ke atas (scroll naik)
77         Mobile.swipe(500, 1900, 500, 1200)
78     }
79 }
80
81 // ↪ Jika tetap tidak ditemukan
82 if (!(found)) {
83     KeywordUtil.markFailed(('X Teks \'' + thnLahir) + '\' tidak ditemukan setelah scroll dua arah.')
84 }

```

Gambar 3.7. Potongan kode otomasi pada skenario tambah leads



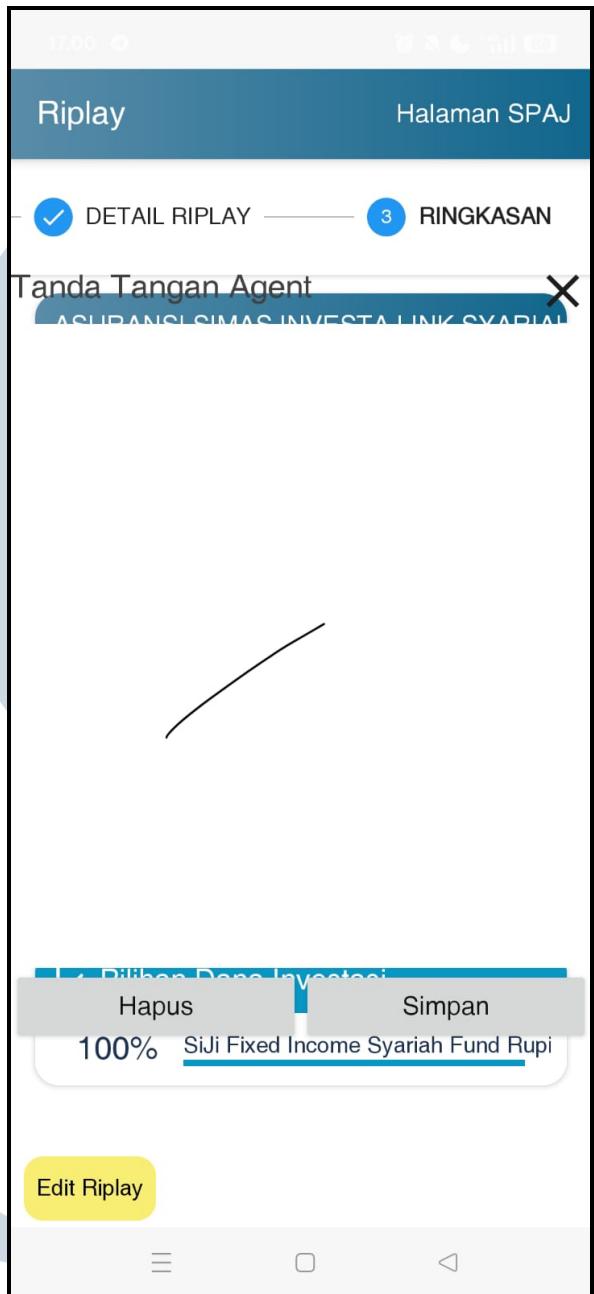


Gambar 3.8. Tampilan pemilihan tahun pemegang polis [3]

Sumber: Aplikasi internal PT Asuransi Simas Jiwa, 2025

Melalui penerapan skrip tersebut, sistem diarahkan untuk melakukan pencarian tahun secara dinamis hingga nilai yang sesuai dengan data uji ditemukan dan dipilih. Pendekatan ini memungkinkan proses pengisian data tahun kelahiran dapat dilakukan secara akurat dan konsisten, meskipun elemen antarmuka yang digunakan memiliki keterbatasan dalam hal keterbacaan oleh Katalon.

Dalam kasus ini, keterbatasan lainnya juga Katalon tidak dapat membaca maupun mengeksekusi tindakan pada komponen tanda tangan dan tombol *save* karena kedua elemen tersebut ditampilkan dalam bentuk *pop-up* pada aplikasi. Keberadaan *pop-up* tersebut menyebabkan terjadinya *layering* antarmuka, sehingga elemen yang berada di dalamnya tidak dapat dikenali secara langsung oleh mekanisme pembacaan objek pada Katalon.



Gambar 3.9. Tampilan *pop-up* tanda tangan pemegang polis [3]

Sumber: Aplikasi internal PT Asuransi Simas Jiwa, 2025

Akibat kondisi tersebut, sistem Katalon mengalami keterbatasan dalam mengidentifikasi dan berinteraksi dengan elemen yang berada pada lapisan terpisah dari halaman utama. Oleh karena itu, diperlukan penulisan kode tambahan untuk menjembatani keterbatasan tersebut, sehingga proses tanda tangan dan penyimpanan data tetap dapat dieksekusi sesuai dengan skenario pengujian yang telah dirancang.

## B #30570 - Life Insurance InfOrmation System (Lions)

Platform otomasi Katalon tidak hanya dapat digunakan untuk pengujian aplikasi berbasis web dan mobile, tetapi juga mendukung pengujian pada aplikasi berbasis desktop. Pada pelaksanaan proyek ini, pengujian dilakukan terhadap aplikasi internal perusahaan, yaitu *Life Insurance Information System (Lions)*, yang mana merupakan aplikasi berbasis desktop dan digunakan sebagai sistem pendukung operasional bisnis perusahaan.

Proyek ini bertujuan untuk melakukan peningkatan dan penyempurnaan fungsi serta fitur aplikasi, dengan fokus utama pada mekanisme notifikasi status pending yang muncul sebagai dampak dari perubahan masa tenggat pembayaran. Pada Gambar 3.10 ditunjukkan tampilan perubahan periode pembayaran yang menjadi salah satu faktor pemicu perubahan status administrasi pada sistem. Perubahan periode pembayaran tersebut berdampak langsung terhadap kelengkapan dan validitas dokumen yang harus dipenuhi.

The screenshot shows a software interface titled 'Worksheet Detail'. At the top, there are buttons for 'Save', 'View Document', and 'Close'. To the right, there is a section for 'Edit Remark' with a text input field. Below these are two main panels. The left panel contains fields for 'Nomor Polis' (W4.185.0000.297), 'Nomor SPAJ' (W4.A190.01278), 'Nama Produk' (SIMAS INVESTA PRIMER), 'Status Polis' (INFORCE), 'Nama Pemegang Polis' (SUI INVESTA PLAT\_2), and 'No. Worksheet' (W4A19001278-1225-03550). It also includes dropdowns for 'Tanggal Pengajuan Formular' (09/12/2025) and 'Tanggal Terima POS' (09/12/2025), and a note field. The right panel displays bank information: 'Bank' (BANK SINARMAS), 'Cabang' (KC JAKARTA - THAMRIN), 'Kota' (JAKARTA), 'Atas Nama' (SUI INVESTA PLAT\_2), and 'No A/C' (5910571056). At the bottom, there is a table titled 'Transaction' with tabs for 'Dokumen' and 'Tanggal Berkas'. The table has columns for 'Jenis Transaksi', 'Beg Date Next MTI/RPI/MPI', 'MTI/RPI/MPI', 'Special Rate', 'ITI/TTI/TPI (Rate)', 'Pembayaran Manfaat', and 'Pilihan JT MTI/RPI/MPI'. It shows two rows: 'Sebelum Perubahan' (Premi Top Up Sekaligus ke-1, 25-Nov-2020, 6 BULAN, 6.750 AKHIR RPI, Tidak Melakukan Penarikan Nilai Polis) and 'Setelah Perubahan' (Premi Top Up Sekaligus ke-1, 25-Nov-2020, 36 BULAN, 5.500 AKHIR RPI, Tidak Melakukan Penarikan Nilai Polis). A dropdown menu next to the second row lists options: 3 BULAN, 6 BULAN, 12 BULAN, 24 BULAN, and 36 BULAN, with '36 BULAN' selected.

Gambar 3.10. Tampilan perubahan periode pembayaran [3]

Sumber: Aplikasi internal PT Asuransi Simas Jiwa, 2025

Selanjutnya, Gambar 3.11 menampilkan halaman tab dokumen yang digunakan untuk memantau status dan kelengkapan dokumen pendukung setelah perubahan periode pembayaran dilakukan. Pengujian terhadap kedua fitur tersebut dilakukan untuk memastikan bahwa sistem mampu menghasilkan notifikasi

status pending yang lebih akurat, tepat waktu, dan konsisten, sehingga potensi keterlambatan tindak lanjut oleh pihak terkait dapat diminimalkan.

The screenshot shows a software interface titled 'Worksheet Detail'. At the top, there are buttons for 'Save', 'View Document', and 'Close'. To the right, there is a 'Edit Remark' field with a text input. The main area contains several input fields and dropdown menus:

Nomor Polis:	19.185.0000.297	Nomor SPAJ:	W4.A190.01278	Nama Produk:	SIMAS INVESTA PRIMER	
Nama Pemegang Polis:	SUJI INVESTA PLAT_2				Status Polis:	INFORCE
No. Worksheet:		W4A19001278-1225-03550	Jenis Worksheet:	PERUBAHAN MTI		
Tanggal Pengajuan Formulir:	09/12/2025	Tanggal Terima POS:	09/12/2025	Note:		

On the right side, there is a panel for banking details:

Bank:	BANK SINARMAS
Cabang:	KC JAKARTA - THAMIRIN
Kota:	JAKARTA
Atas Nama:	SUJI INVESTA PLAT_2
No A/c:	5910571056

Below these sections is a table titled 'Transaction Dokumen Tanggal Berkas' with columns: Mandatory, No., Validasi Dokumen, Status, and Remark. The data in the table is as follows:

Mandatory	No.	Validasi Dokumen	Status	Remark
<input checked="" type="checkbox"/>	1	KTP Aktif	<input checked="" type="radio"/> Ya <input type="radio"/> Tidak	KTP
<input checked="" type="checkbox"/>	2	Formulir	<input checked="" type="radio"/> Ya <input type="radio"/> Tidak	Form
<input checked="" type="checkbox"/>	3	Fotokopi KTP	<input checked="" type="radio"/> Ya <input type="radio"/> Tidak	FKTP
<input checked="" type="checkbox"/>	4	Tanda Tangan Sama	<input checked="" type="radio"/> Ya <input type="radio"/> Tidak	TTD

Gambar 3.11. Tampilan halaman tab dokumen [3]

Sumber: Aplikasi internal PT Asuransi Simas Jiwa, 2025

Pelaksanaan pengujian dilakukan dengan menerapkan sejumlah skenario otomasi yang telah dirancang berdasarkan kebutuhan fungsional aplikasi. Setiap skenario pengujian disusun untuk merepresentasikan alur penggunaan sistem secara aktual, mulai dari proses pemicu notifikasi hingga pengiriman informasi kepada pengguna terkait. Rincian skenario pengujian beserta deskripsi tahapan pengerjaannya disajikan pada Gambar 3.12.

SCENARIO & TESTING RESULTS						
	A	B	C	D	E	
1						
2						
3						
4	Project Name	FU Pending Perubahan RPI				
5	Ticket Number	#30570				
6	Scriptwriter by	Daniel				
7	Start Date Scripting	29 September 2025	End Date Scripting	8 September 2025		
8	UAT Tester by	Daniel				
9	Start Date UAT Testing	10 November 2025	End Date UAT Testing	14 November 2025		
10	% Testing (Percentages of Status Passed)	100%				
11						
12						
13	ID Scenario	Platforms	Functions	Specifications Menu	Scenario Descriptions	
14	PSTR-1.1	Lions System	Input Worksheet Perubahan MTI	>> Entry >> Worksheet >> Perubahan MTI >> Add (Right Click) >> Fill "No. Polis"  >> Entry >> Worksheet >> Perubahan MTI >> Add (Right Click) >> Tambahan >> Klik "Tambahan" >> Pilih Transaksi >> Ubah periode MTI/RPI/MPI	Proses pembahaman Worksheet Perubahan MTI dengan No. polis produk <b>GROUP 1</b> Conditions SPAJ/POLIS Number: - Status (INFORCE)	No. Polis: 211850000236
15	PSTR-1.2	Lions System	Tab "Transaction"	>> Entry >> Worksheet >> Perubahan MTI >> Add (Right Click) >> Tambahan >> Klik "Tambahan" >> Pilih Transaksi >> Ubah periode MTI/RPI/MPI	Melakukan proses input tab transaction dan merubah periode MTI/RPI/MPI menjadi 36 bulan	No. Polis: 211850000236
16	PSTR-1.3	Lions System	Tab "Dokumen"	>> Entry >> Worksheet >> Perubahan MTI >> Add (Right Click) >> Tambahan >> Klik "Tambahan" >> Pilih Transaksi >> Choose tab "Dokumen"	Melakukan proses input tab Dokumen dengan kondisi: - Checklist 2/4 Dokumen Mandatory - 2/4 Dokumen berStatus "Ya" & 2/4 Dokumen berStatus "Tidak" - Remark terisi	No. Polis: 211850000236
17	PSTR-1.4	Lions System	Save & Print	>> Entry >> Worksheet >> Perubahan MTI >> Add (Right Click) >> Tambahan >> Klik "Tambahan" >> Pilih Transaksi >> Fill tab "Transaction" dan "Dokumen"	Melakukan save dan print worksheet setelah semua tab yang diperlukan terisi	No. Polis: 211850000236
18	PSTR-1.5	E-mail	Follow Up Email		Melakukan monitoring worksheet dengan status 'PENDING WORKSHEET' setelah H+7, H+14, H+21, H+28 dari tanggal receipt	Auto Process
19	PSTR-1.6	E-mail	Follow Up Email			Auto Process

Gambar 3.12. Skenario pengujian tiket nomor 30570

Jumlah skenario otomasi yang diterapkan dalam proyek ini relatif terbatas. Hal tersebut disebabkan oleh sifat skenario yang bersifat dinamis dan dapat digunakan secara berulang, di mana satu skenario otomasi dapat dijalankan untuk berbagai kondisi pengujian hanya dengan melakukan penyesuaian terhadap parameter tertentu, seperti jumlah grup penerima atau daftar alamat email yang dituju. Pendekatan ini dinilai efektif dalam meningkatkan efisiensi proses pengujian tanpa mengurangi cakupan validasi fungsi yang diuji. Berikut potongan kode dari projek ini pada Gambar 3.13

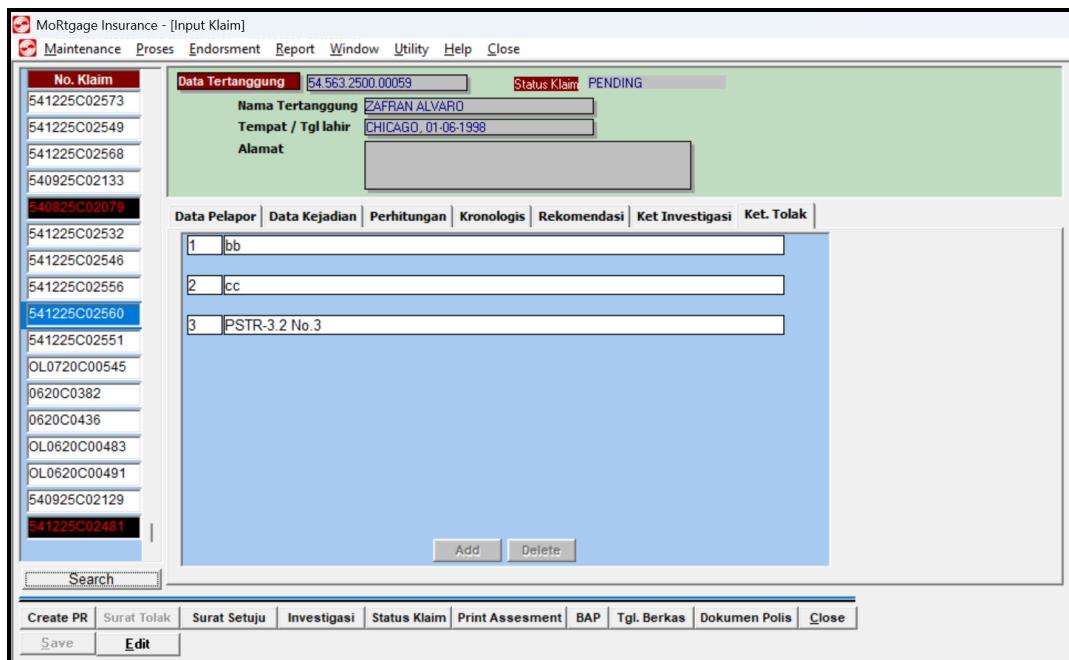
Item	Object	Input	Output	Description
1 - Start Application With Title		"D:\SIMAS JIWA\Lions\ASJ - lions - uat-20250925T071849Z-1-001\ASJ - lions - uat\lions.exe"; "lions"		
2 - Get Driver				
3 - Binary Statement		actions = new Actions(driver)		
4 - Binary Statement		robot = new Robot()		
5 - Delay		2		
6 - Method Call Statement		actions.keyDown(Keys.CONTROL).sendKeys("D").keyUp(Keys.ALT).perform()		
7 - Delay		2		
8 - Method Call Statement		actions.keyDown(Keys.CONTROL).keyUp(Keys.CONTROL).perform()		
9 - Delay		2		
10 - Method Call Statement		actions.keyDown(Keys.ALT).sendKeys("E").keyUp(Keys.ALT).perform()		
11 - Method Call Statement		actions.sendKeys("w").perform()		
12 - Method Call Statement		actions.sendKeys("p").perform()		
13 - Method Call Statement		actions.sendKeys(Keys.ENTER).perform()		
14 - Delay		1		
15 - Method Call Statement		robot.mouseMove(750, 500)		
16 - Method Call Statement		robot.mousePress(InputEvent.BUTTON3_DOWN_MASK)		
17 - Method Call Statement		robot.mouseRelease(InputEvent.BUTTON3_DOWN_MASK)		
18 - Delay		1		
19 - Method Call Statement		robot.mouseMove(760, 510)		
20 - Method Call Statement		robot.mousePress(InputEvent.BUTTON1_DOWN_MASK)		
21 - Method Call Statement		robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK)		
22 - Delay		1		
23 - Switch To Window Title		"Worksheet Detail"		
24 - Set Text	Field_Nopol	"211850000236"		GANTI NO POLISNYA DISINI
25 - Delay		1		
26 - Method Call Statement		actions.sendKeys(Keys.ENTER).perform()		
27 - Delay		5		
28 - Click	Button_Tambah_Transaksi	"List Transaksi Withdraw"		
29 - Switch To Window Title		"Worksheet Detail"		
30 - Click	LTW_OK	"List Transaksi Withdraw"		
31 - Switch To Window Title		2		
32 - Click	Button_Tambah_Transaksi	"C:\Users\himaw\Downloads\Grup 1\2\UPSTR-1.1"		save location
33 - Switch To Window Title		2		
34 - Delay		1		
35 - Take Screenshot				
36 - Delay				
37 - Click	LTW_Close			
38 - Delay				
39 - Click	Close_Message			
40 - Delay				
41 - Method Call Statement				
42 - Method Call Statement				
43 - Method Call Statement				
44 - Delay				
45 - Method Call Statement				Pilih nerahan hulan

Gambar 3.13. Kode otomasi tiket nomor 30570

## C #30621 - MoRtgage Insurance (MRI)

Aplikasi MRI merupakan aplikasi internal perusahaan yang dirancang untuk mendukung dan mengelola proses operasional perusahaan yang berkaitan dengan AJK (Asuransi Jiwa Kredit). Aplikasi ini berperan sebagai sistem pendukung utama dalam pengelolaan data, proses administrasi, serta alur kerja operasional yang berkaitan dengan layanan asuransi jiwa kredit, sehingga pengembangan fitur pada aplikasi MRI menjadi aspek penting dalam meningkatkan efektivitas dan efisiensi operasional perusahaan secara keseluruhan.

Pada proyek ini dilakukan pengembangan dan penyempurnaan fungsionalitas aplikasi internal perusahaan MRI melalui penambahan fitur baru berupa tab Keterangan Tolak. Penambahan tab tersebut bertujuan untuk menyediakan informasi yang lebih terperinci dan sistematis terkait alasan penolakan polis, yang selanjutnya digunakan sebagai dasar penyampaian keterangan penolakan kepada pemegang polis melalui mekanisme pengiriman email penolakan. Dengan adanya fitur ini, proses penyampaian informasi penolakan diharapkan dapat dilakukan secara lebih jelas, konsisten, dan terdokumentasi dengan baik.

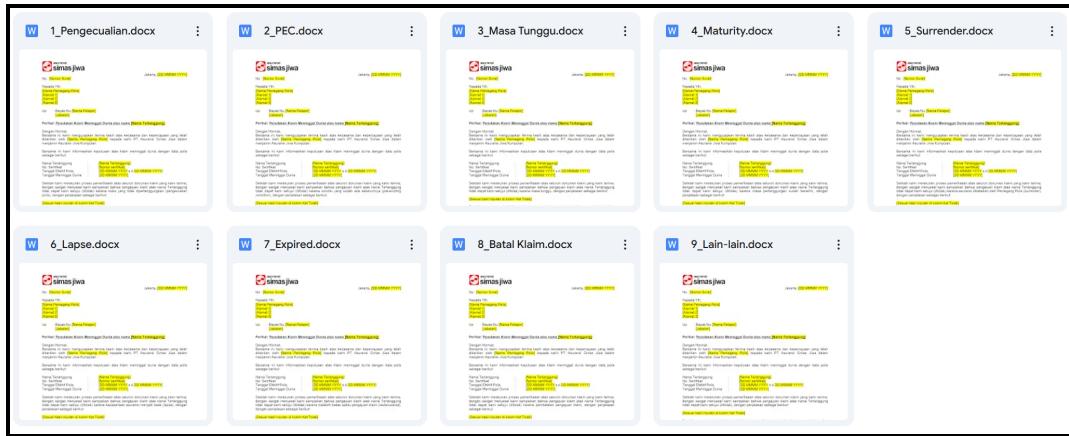


Gambar 3.14. Fitur tab keterangan tolak [3]

Sumber: Aplikasi internal PT Asuransi Simas Jiwa, 2025

Selain penambahan tab Keterangan Tolak, dilakukan pula pengembangan pada aspek klasifikasi penolakan dengan menerapkan penggolongan tipe penolakan ke dalam sembilan kategori penolakan yang telah ditetapkan. Penggolongan tersebut dilengkapi dengan pembagian dua jenis tipe penolakan, yaitu penolakan konvensional dan penolakan syariah, guna menyesuaikan karakteristik produk dan ketentuan operasional yang berlaku di perusahaan. Implementasi penggolongan ini bertujuan untuk meningkatkan akurasi pencatatan data penolakan serta mempermudah proses analisis dan pelaporan pada tahap selanjutnya.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.15. List kategori surat tolak [3]

Sumber: Dokumen internal PT Asuransi Simas Jiwa, 2025

*Test scenario* yang disusun disesuaikan secara menyeluruh dengan kebutuhan pengujian yang telah dijabarkan pada tahap sebelumnya. Lingkup pengujian mencakup proses verifikasi fungsional pada halaman tab Keterangan Tolak, termasuk validasi input dan tampilan data, hingga pengujian terhadap proses pembuatan serta pengiriman informasi penolakan polis kepada pemegang polis melalui media surat elektronik (*email*). Dengan cakupan pengujian tersebut, diharapkan seluruh alur penolakan polis dapat divalidasi secara end-to-end untuk memastikan bahwa sistem telah berjalan sesuai dengan kebutuhan fungsional dan aturan bisnis yang berlaku.

UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

SCENARIO & TESTING RESULTS						
	ID Scenario	Platforms	Functions	Specifications Menu	Scenario Descriptions	Positive Scenario Data Input
				>> Cari (klik)		- Button sc >> Button /
15	PSTR-#35283_2_1	MRI System	Pop Up "Update Status Klaim"	>> Proses >> Akseptasi Klaim >> Search (Klik) >> Input No. Polis >> Cari (klik) >> Status Klaim	Proses mengubah "Status Klaim" menjadi "TOLAK" dan memilih jenis tolak yang akan digunakan	No Polis: 54563250000046  >> Apabila : digunakan >> Apabila :
16	PSTR-#35283_3_1	MRI System	Dropdown "Pemilihan Jenis Penolakan" (SYARIAH)	>> Proses >> Akseptasi Klaim >> Search (Klik) >> Input No. Polis >> Cari (klik) >> Status Klaim >> Update Status Klaim (STATUS KLAIM = TOLAK)	Proses pengisian dropdown "Pemilihan Jenis Penolakan" dengan memilih jenis tolak no.1 dan 2, no.8 dan 9 tanpa adanya keterangan tolak	No Polis: No. 1 = 54563250000047 No. 2 = 54563250000048 No. 3 = 54563250000049 No. 4 = 54563250000050 No. 5 = 54563250000051 No. 6 = 54563250000052 No. 7 = 54563250000053 No. 8 = 54563250000054  After fix No. 9 = 54563250000090  No Polis:
17	PSTR-#35283_3_2	MRI System	Dropdown "Pemilihan Jenis Penolakan" (SYARIAH)	>> Proses >> Akseptasi Klaim >> Search (Klik) >> Input No. Polis >> Cari (klik) >> Status Klaim >> Update Status Klaim (STATUS KLAIM = TOLAK)	Proses pengisian dropdown "Pemilihan Jenis Penolakan" dengan memilih jenis tolak no.3-7 dengan keterangan tolak	No. 3 = 54563250000044 No. 4 = 54563250000045 No. 5 = 54563250000052 No. 6 = 54563250000053 No. 7 = 54563250000054  After fix No. 3 = 54563250000081  No Polis:
18	PSTR-#35283_3_3	MRI System	Dropdown "Pemilihan Jenis Penolakan" (KONVENTIONAL)	>> Proses >> Akseptasi Klaim >> Search (Klik) >> Input No. Polis >> Cari (klik) >> Status Klaim >> Update Status Klaim (STATUS KLAIM = TOLAK) >> Akseptasi Klaim	Proses pengisian dropdown "Pemilihan Jenis Penolakan" dengan memilih jenis tolak no.1 dan 2, no.8 dan 9 tanpa adanya keterangan tolak	No Polis: No. 8 = 54562250000349  Proses dap button 'Sun'

Gambar 3.16. Skenario positif pengujian tiket nomor 30621

Pada proyek ini, turut disusun beberapa *negative test scenario* sebagai bagian dari proses pengujian. Penyusunan skenario negatif tersebut bertujuan untuk mengeksplorasi berbagai kemungkinan terjadinya kesalahan atau kondisi tidak valid yang berpotensi muncul di luar skenario normal, namun masih berada dalam lingkup dan batasan proyek yang dikerjakan. Melalui pengujian ini, diharapkan kemampuan sistem dalam menangani input yang tidak sesuai, proses yang gagal, maupun kondisi pengecualian (*exception handling*) dapat dievaluasi secara lebih komprehensif.

**UNIVERSITAS**  
**MULTIMEDIA**  
**NUSANTARA**

Gambar 3.17. Skenario negatif pengujian tiket nomor 30621

Pada tahap ini, dilakukan penyesuaian terhadap pembuatan skrip otomasi, agar selaras dengan kode *test scenario* yang telah dirancang dan ditetapkan pada tahap sebelumnya. Penyesuaian tersebut bertujuan untuk memastikan bahwa setiap skrip otomasi merepresentasikan satu skenario pengujian secara spesifik, sehingga keterlacakkan (*traceability*) antara dokumen test scenario dan implementasi skrip otomasi dapat terjaga dengan baik. Dengan adanya kesesuaian tersebut, proses eksekusi, evaluasi hasil pengujian, serta pemeliharaan skrip di tahap selanjutnya dapat dilakukan secara lebih sistematis dan terstruktur.

Meskipun demikian, perubahan yang diterapkan pada skrip otomasi relatif terbatas karena perbedaan yang ditemukan pada masing-masing skenario pengujian hanya berkaitan dengan variasi tipe penolakan dan kategori penolakan yang digunakan. Secara umum, alur pengujian, struktur skrip, serta logika eksekusi tetap dipertahankan tanpa mengalami perubahan yang signifikan. Oleh karena itu, penyesuaian yang dilakukan lebih difokuskan pada penggantian parameter data uji yang relevan, sehingga skrip otomasi tetap dapat digunakan kembali (*reusable*) dengan tingkat konsistensi dan efisiensi yang tinggi.

Kode otomasi yang digunakan dalam proses pengujian dirancang agar dapat digunakan kembali pada skenario pengujian lainnya. Namun demikian, penerapan kode otomasi tersebut dibatasi hanya pada skenario positif atau

*Positive Scenario Testing Result (PSTR).* Pengujian terhadap skenario negatif tidak diimplementasikan menggunakan otomasi karena membutuhkan waktu yang lebih lama dalam penyusunan skenario dengan alur pengujian yang berbeda.

```
1* import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint()
2
3 //Windows.startApplicationWithTitle('D:\\SIMAS JIWA\\MRI\\ASJ - mri - uat\\mri.exe', 'MoRtgage Insurance')
4 Windows.startApplication('D:\\SIMAS JIWA\\MRI\\ASJ - mri - uat\\mri.exe')
5
6 def driver = Windows.getDriver()
7
8 Actions actions = new Actions(driver)
9
10 Robot robot = new Robot()
11
12 actions.keyDown(Keys.CONTROL).sendKeys('D').keyUp(Keys.CONTROL).perform()
13
14 Windows.delay(2)
15
16 actions.keyDown(Keys.ALT).sendKeys('P').keyUp(Keys.ALT).perform()
17
18 Windows.delay(1)
19
20 actions.sendKeys('a').perform()
21
22 actions.sendKeys('a').perform()
23
24 Windows.delay(1)
25
26 actions.sendKeys(Keys.ENTER).perform()
27
28 Windows.delay(20)
29
30 Windows.switchToWindowTitle('MoRtgage Insurance')
31
32 Windows.click(findWindowsObject('35283 - Enhance MRI/Button Search No Polis'))
33
34 'EDIT NO POLIS DISINI'
35 Windows.setText(findWindowsObject('35283 - Enhance MRI/Field No Pol'), '5456325000049')
36
37 Windows.delay(1)
38
39 Windows.click(findWindowsObject('35283 - Enhance MRI/Button Cari No Pol'))
40
41 Windows.delay(2)
42
43 // pindah mouse ke koordinat ABSOLUT
44 'KLIK TAB KET. TOLAK'
45 robot.mouseMove(689, 183)
46
47 robot.mousePress(InputEvent.BUTTON1_DOWN_MASK)
48
49 robot.mouseRelease(InputEvent.BUTTON1_DOWN_MASK)
50
51 'GANTI SAVE LOCATION'
52 Windows.takeScreenshot('D:\\SIMAS JIWA\\#30621\\#35283 - Enhance MRI\\PSTR-3\\No8\\PSTR-3.1_No8_Ket_Tolak.png')
53
54 Windows.delay(2)
55
56 Windows.click(findWindowsObject('35283 - Enhance MRI/Button Status Klaim'))
```

Gambar 3.18. Katalon Studio "MRI"

### 3.4 Kendala dan Solusi yang Ditemukan

Bagian ini membahas berbagai kendala yang ditemukan selama pelaksanaan kerja magang, khususnya dalam kegiatan pengujian aplikasi baik secara manual maupun dengan menggunakan otomasi. Identifikasi kendala dilakukan untuk memahami faktor-faktor yang memengaruhi kelancaran dan efektivitas proses pengujian. Selain itu, section ini juga menguraikan solusi yang diterapkan sebagai

upaya untuk mengatasi kendala tersebut, sehingga pelaksanaan pengujian dapat berjalan lebih optimal dan terstruktur.

### 3.4.1 Kendala

Selama pelaksanaan pengujian manual maupun otomasi, ditemukan beberapa kendala yang memengaruhi kelancaran proses pengujian aplikasi. Kendala-kendala tersebut dirangkum sebagai berikut:

1. **Keterbatasan Ketersediaan Waktu Pengujian** Keterbatasan waktu yang dialokasikan pada setiap proyek dapat menjadi kendala, khususnya pada proyek yang memerlukan persiapan skrip otomasi serta pemahaman alur fitur aplikasi, terutama ketika terjadi perubahan fungsionalitas maupun tampilan antarmuka.
2. **Keterbatasan Platform Otomasi dalam Mendeteksi Elemen Aplikasi** Platform otomasi memiliki keterbatasan dalam membaca atau mendeteksi elemen dan *class* tertentu pada aplikasi, sehingga beberapa objek pengujian tidak dapat terbaca dan berdampak pada proses pembuatan maupun eksekusi skrip otomasi.
3. **Kebutuhan Adaptasi Penggunaan Katalon Studio** Penggunaan Katalon Studio memerlukan proses adaptasi terhadap fitur, antarmuka, dan konsep otomasi yang diterapkan, terutama pada tahap awal implementasi, sehingga berpengaruh terhadap kecepatan pelaksanaan pengujian.

Upaya untuk mengatasi kendala-kendala tersebut akan dibahas pada subbab selanjutnya yang memuat solusi dan strategi pendukung pelaksanaan pengujian secara lebih efektif dan terstruktur.

### 3.4.2 Solusi

Untuk mengatasi kendala yang telah diidentifikasi, berikut adalah solusi yang dapat diterapkan:

#### 1. Keterbatasan Ketersediaan Waktu Pengujian

Keterbatasan waktu pengujian diatasi melalui perencanaan yang lebih terstruktur, penentuan prioritas pada fitur kritis, serta penerapan *reusable*

*test case* dan *modular scripting* pada skrip otomasi untuk meminimalkan waktu penulisan ulang skrip.

## 2. Keterbatasan Platform Otomasi dalam Mendeteksi Elemen Aplikasi

Kendala keterbacaan elemen aplikasi diatasi dengan penyesuaian metode identifikasi objek menggunakan *XPath*, *CSS Selector*, dan *attribute-based locator*, serta koordinasi dengan tim pengembang untuk meningkatkan stabilitas dan keterbacaan elemen.

## 3. Kebutuhan Adaptasi Penggunaan Katalon Studio

Adaptasi terhadap Katalon Studio dilakukan melalui pembelajaran bertahap, pemanfaatan dokumentasi resmi, serta pengembangan skrip dari skenario sederhana ke kompleks guna meningkatkan efektivitas dan efisiensi pengujian.

