BAB III
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan kerja magang yang dilakukan oleh mahasiswa dilaksanakan di perusahaan Ruparupa, yang berlokasi di Kembangan, Jakarta Barat. Sebagai sebuah platform omnichannel, Ruparupa menjual berbagai produk dari Kawan Lama Group, termasuk merek-merek seperti Ace, Informa, dan Selma. Tugas saya sebagai *Quality Assurance* (QA) dalam magang ini adalah memastikan bahwa fitur, modul, dan voucher yang akan diterapkan di website dan aplikasi Ruparupa telah memenuhi standar kualitas yang ditetapkan.

Secara umum, peran seorang QA adalah melakukan pengujian terhadap perangkat lunak atau sistem untuk memastikan bahwa kualitas produk memenuhi standar yang telah ditentukan oleh perusahaan [5]. QA tidak hanya fokus pada kualitas sistem, tetapi juga menilai kehandalan dan fungsionalitas dari sistem tersebut. Setiap sistem harus memiliki tingkat keakuratan yang tinggi dalam pengelolaan data dan alur proses yang ada di dalamnya. Hal ini menjadi tanggung jawab utama QA untuk memastikan bahwa semua elemen ini berjalan dengan baik dan sesuai dengan kebutuhan pengguna. *Quality Assurance* mencakup serangkaian prosedur, proses, dan kebijakan yang dirancang untuk memastikan bahwa produk atau layanan yang dihasilkan oleh perusahaan memenuhi standar kualitas yang telah ditetapkan. Sistem ini berfungsi sebagai kerangka kerja yang terstruktur untuk mengelola dan mengontrol kualitas pada setiap tahap pengembangan produk atau layanan. Tujuan utama dari QA adalah untuk mencegah kesalahan dan cacat produk dengan memastikan bahwa setiap langkah dalam proses produksi atau pengembangan dilakukan sesuai dengan spesifikasi dan standar yang telah ditentukan [5].

Dalam pelaksanaan tugasnya, QA di Ruparupa melibatkan berbagai aktivitas yang bertujuan untuk menjaga dan meningkatkan kualitas produk. Pertama-tama, QA bertanggung jawab untuk merancang dan

mengimplementasikan prosedur operasional standar (SOP) yang mendetail, yang harus diikuti oleh seluruh bagian dalam organisasi. SOP ini mencakup dokumentasi yang lengkap dari setiap langkah dalam proses produksi atau pengembangan produk. Selain itu, QA juga bertanggung jawab untuk melakukan pengujian kualitas pada berbagai tahap dalam proses produksi atau pengembangan. Aktivitas lain yang menjadi tanggung jawab QA adalah pelatihan dan pengembangan karyawan untuk memastikan bahwa mereka memiliki pengetahuan dan keterampilan yang diperlukan untuk mengikuti prosedur kualitas yang telah ditetapkan. Dengan melakukan tugas-tugas ini, QA membantu organisasi mencapai efisiensi yang lebih tinggi, mengurangi biaya yang disebabkan oleh cacat produk, dan meningkatkan kepuasan pelanggan melalui produk atau layanan berkualitas tinggi.

Selama masa magang, koordinasi dan komunikasi di dalam tim *Quality Assurance* dilakukan melalui berbagai saluran komunikasi yang efisien. Selain komunikasi lisan, koordinasi dilakukan melalui aplikasi internal perusahaan yang dirancang untuk memfasilitasi komunikasi dan kolaborasi antar anggota tim. Aplikasi ini memungkinkan pekerja untuk melakukan absensi, mengikuti pelatihan online, dan berkoordinasi dengan tim melalui satu platform yang terpadu. Fitur "*all-in-one*" pada aplikasi ini sangat membantu dalam hal efisiensi waktu, sehingga seluruh proses kerja dapat dilakukan dengan lebih praktis dan terorganisir. Koordinasi yang memerlukan bahasa dan format yang lebih formal dilakukan melalui email resmi perusahaan, yang disediakan khusus untuk keperluan komunikasi antar karyawan. Hal ini memastikan bahwa komunikasi yang berkaitan dengan pekerjaan tidak tercampur dengan email pribadi, sehingga lebih terorganisir dan mudah untuk diakses saat dibutuhkan.

Selama masa magang, saya mendapatkan bimbingan langsung dari atasan saya, yang menjabat sebagai *Section Head Quality Assurance*. Koordinasi dalam tim *Quality Assurance* dilakukan setiap hari melalui aplikasi internal yang telah dijelaskan di atas, dan setiap minggunya diadakan rapat internal tim yang dilaksanakan setiap hari Selasa. Rapat ini digunakan untuk membahas berbagai hal yang berkaitan dengan pekerjaan sehari-hari serta proyek-proyek yang sedang atau

akan ditangani oleh tim. Rapat ini memberikan kesempatan bagi pekerja magang untuk berdiskusi secara langsung mengenai berbagai hal yang berkaitan dengan pekerjaan dan untuk mendapatkan umpan balik dari anggota tim lainnya.

Selain rapat mingguan yang dilakukan bersama tim, setiap bulannya juga diadakan rapat bulanan yang dihadiri oleh seluruh anggota departemen *Quality Assurance*. Rapat bulanan ini bertujuan untuk membahas perkembangan yang sedang berlangsung di setiap seksi dan juga untuk membahas proyek-proyek yang akan datang. Koordinasi dan komunikasi yang baik sangat penting dalam pekerjaan tim, terutama ketika melakukan *System Integration Testing* (SIT). Dalam pekerjaan sehari-hari, komunikasi yang baik memungkinkan setiap anggota tim untuk berkolaborasi dengan lebih efektif, memastikan bahwa semua aspek pengujian berjalan sesuai dengan rencana. Sebagai *Quality Assurance* di Ruparupa, saya terlibat dalam proyek yang berfokus pada fitur-fitur yang terkait dengan keanggotaan pelanggan. Proyek ini bertujuan untuk memberikan nilai tambah kepada pelanggan dengan memberikan diskon dan manfaat lainnya berdasarkan tingkat keanggotaan mereka. Selain itu, saya juga terlibat dalam proyek TAHU, yang berfungsi untuk mendukung tim operasional dalam memasukkan produk, banner, dan konten lainnya ke dalam platform Ruparupa. Setiap fitur yang dikembangkan harus diuji secara menyeluruh untuk memastikan bahwa semuanya telah memenuhi persyaratan pengguna sebelum diimplementasikan ke dalam sistem produksi.

Dalam proses pengembangan fitur hingga *deployment* ke *production*, terdapat banyak tahapan yang harus dilalui untuk memastikan bahwa produk tersebut siap untuk digunakan oleh pengguna. Pertama, *Product Manager* akan mengumpulkan persyaratan dari pengguna dan mengoordinasikan pengembangan dengan tim *developer*. Setelah pengembangan selesai, QA akan melakukan pengujian sesuai dengan persyaratan yang ada. Jika hasil pengujian memuaskan, QA dan *Product Manager* akan melanjutkan dengan melakukan SIT bersama tim internal untuk memastikan bahwa semua aspek sistem bekerja dengan baik. Setelah SIT selesai dan dinyatakan berhasil, tahapan berikutnya adalah *User Acceptance*

Testing (UAT), di mana QA akan mempresentasikan proyek tersebut kepada pengguna. Jika pengguna menyetujui bahwa proyek tersebut sudah memenuhi persyaratan, maka proyek siap untuk dideploy ke production sesuai dengan jadwal yang telah disepakati. Dalam setiap tahapan pengujian, QA harus membuat *test case* yang berisi skenario pengujian yang mungkin terjadi dan memastikan bahwa semua skenario tersebut telah diuji dan memenuhi persyaratan. Jika ditemukan masalah selama pengujian, QA akan melaporkannya kepada developer melalui platform Jira. Setelah masalah diperbaiki dan pengujian berhasil dilakukan, tiket di Jira akan dipindahkan dari "*In Progress Testing*" ke "*Done Testing*", menandakan bahwa proyek siap untuk dilanjutkan ke tahapan berikutnya.

Proses magang ini memberikan saya kesempatan untuk belajar dan berkontribusi dalam berbagai aspek pengembangan produk di Ruparupa, serta meningkatkan keterampilan saya dalam bidang *Quality Assurance*. Pengalaman ini sangat berharga dan memberikan wawasan yang mendalam mengenai bagaimana sebuah perusahaan besar seperti Ruparupa menjaga kualitas produk dan layanan mereka, serta bagaimana komunikasi dan koordinasi yang baik di dalam tim dapat meningkatkan efisiensi dan hasil kerja secara keseluruhan.

3.2 Tugas dan Uraian Kerja Magang

No.	**Tugas**	**Realisasi**	
		Start Date	***End Date***
2.	*Grooming Project*	1 Juni 2024	1 Juni 2024
		1 July 2024	1 July 2024
		1 Agustus 2024	1 Agustus 2024
		1 September 2024	1 September 2024
3.	Mempelajari dokumen project	1 Juni 2024	5 Juni 2024

		1 July 2024	5 July 2024
		1 Agustus 2024	5 Agustus 2024
		1 September 2024	5 September 2024
4.	Membuat *Test Case*	2 Juni 2024	7 Juni 2024
		2 July 2024	7 July 2024
		2 Agustus 2024	7 Agustus 2024
		2 September 2024	7 September 2024
5.	*Testing*	7 Juni 2024	30 Juni 2024
		7 July 2024	30 July 2024
		7 Agustus 2024	30 Agustus 2024
		7 September 2024	30 September
6.	*Update Test Tracking*	1 Juni 2024	14 Juni 2024
		1 July 2024	14 July 2024
		1 Agustus 2024	14 Agustus 2024
		1 September 2024	14 September 2024
7.	SIT& UAT	28 Juni 2024	28 Juni 2024
		18 July 2024	26 July 2024
		7 Agusts 2024	28 Agustus 2024

		10 September 2024	24 September 2024

Tabel 1.1 Realisasi Waktu Pelaksanaan Magang

Selama menjalani kerja magang di Ruparupa, saya ditempatkan di departemen *Quality Assurance* (QA) dengan tanggung jawab yang beragam dan penting dalam memastikan kualitas sistem yang dikembangkan. Tugas pertama saya adalah terlibat dalam *Grooming Project*, yang merupakan tahap awal sebelum proses pengembangan dimulai. Dalam sesi ini, *Product Manager* (PM) memberikan penjelasan mendetail kepada tim pengembang (*developer*) dan QA mengenai proyek yang akan dikerjakan. Tujuan utama dari sesi ini adalah memastikan bahwa semua pihak memahami dengan baik persyaratan, tujuan, serta ruang lingkup proyek. Grooming biasanya berlangsung selama 1-2 jam dalam satu hari, dan diskusi dilakukan untuk menyelaraskan ekspektasi antara tim *developer* dan QA sehingga tidak ada kesalahpahaman selama proses pengembangan.

Setelah sesi grooming selesai, langkah selanjutnya adalah mempelajari dokumen proyek yang dikenal sebagai *Project Requirement Document* (PRD). Dokumen ini memuat detail lengkap mengenai persyaratan, tujuan, dan metrik utama yang harus dicapai oleh proyek tersebut. Sebagai QA, saya bertanggung jawab untuk memahami setiap aspek dari PRD guna memastikan semua kebutuhan user terpenuhi. Proses mempelajari PRD biasanya memakan waktu 1-2 hari, tergantung pada kompleksitas proyek. Pada tahap ini, saya menelaah dokumen dengan cermat untuk mengidentifikasi semua skenario yang mungkin terjadi selama pengujian. Setelah memahami PRD, saya melanjutkan dengan pembuatan test case, yang merupakan dokumen penting sebagai panduan bagi QA dalam melakukan pengujian. Test case yang saya buat harus detail dan mencakup semua skenario, baik positif maupun negatif, yang mungkin terjadi saat pengujian. Pembuatan test case ini biasanya memerlukan waktu sekitar satu minggu, karena saya harus memastikan bahwa semua kemungkinan skenario telah dicakup dengan baik dan

sesuai dengan dokumen PRD. Setiap skenario dalam test case harus disusun dengan jelas dan sistematis agar mudah dipahami dan diikuti selama pengujian.

Setelah developer menyelesaikan pengembangan proyek, tahap berikutnya adalah melakukan testing. Testing dilakukan oleh QA untuk memastikan bahwa semua fitur yang telah dikembangkan sesuai dengan persyaratan yang tercantum dalam PRD. Proses testing ini biasanya berlangsung selama satu sprint, atau sekitar dua minggu. Saya harus melakukan pengujian sesuai dengan skenario yang telah disusun dalam test case, termasuk skenario negatif untuk menguji ketahanan sistem. Pengujian yang saya lakukan mencakup seluruh fungsi dan alur kerja yang ada, memastikan tidak ada bug atau masalah lain yang lolos dari perhatian.

Setelah menyelesaikan proses *testing*, saya memperbarui status di platform manajemen proyek seperti Jira. Saya mengubah status dari "*In Progress Testing*" menjadi "*Done*" untuk menandakan bahwa proyek telah selesai diuji dan siap untuk tahap selanjutnya. Pembaruan ini juga berguna bagi PM dan tim lain untuk memantau perkembangan proyek secara real-time dan memastikan bahwa semua tahapan dilakukan tepat waktu.

Setelah semua testing selesai, tahap berikutnya adalah melakukan *System Integration Testing* (SIT) dengan tim internal Ruparupa. SIT bertujuan untuk memastikan bahwa seluruh sistem yang dikembangkan telah berfungsi dengan baik dan siap untuk digunakan oleh pengguna akhir. Saya terlibat aktif dalam tahap ini, bekerja sama dengan tim internal untuk mengidentifikasi dan memperbaiki masalah yang mungkin muncul. Setelah SIT selesai dan proyek dinyatakan layak, dilanjutkan dengan *User Acceptance Testing* (UAT). Pada tahap UAT, saya berperan dalam mempresentasikan proyek kepada user yang terlibat. Presentasi ini mencakup demonstrasi fitur dan sistem yang telah dikembangkan, serta menjawab pertanyaan atau kekhawatiran yang diajukan oleh user. Jika proyek tersebut disetujui oleh user, langkah terakhir adalah melakukan deployment ke production sesuai jadwal yang telah ditentukan. Proses ini merupakan penutup dari siklus pengembangan, di mana proyek akhirnya siap digunakan oleh pengguna akhir sesuai dengan kebutuhan mereka.

Melalui semua tahapan ini, saya memperoleh pengalaman berharga dalam memastikan bahwa setiap fitur dan sistem yang dikembangkan di Ruparupa memenuhi standar kualitas yang tinggi dan sesuai dengan ekspektasi user.

3.2.1 *Grooming Project*

1	Date Picker
2	Informa App
3	Informa Cohesive
4	No Hp Luar Negeri
5	Tukar Koin ke Produk
6	Revamp Improvement
7	Activate Register
8	Voucher 3rd Party
9	Instant Upgrade Member
10	Suggestion Member
11	Google Sign In
12	Passkey OTP
13	Project Refund Token
14	Project OTP
15	Project Sentry
16	Miss Ace
17	My Order Informa
18	Tracking DC
19	Membership Level
20	Faktur Pajak
21	Shop The Look

Berikut diatas adalah list project yang dikerjakan mahasiswa selama melakukan proses magang di Ruparupa. Project tersebut dilakukan pada rentang

waktu 3-4 minggu per project dan terdapat banyak project yang berjalan bersamaan.

Pada saat memulai masa magang di Ruparupa, saya menghadapi hari *on-boarding* yang dipenuhi dengan berbagai kegiatan pengenalan dan adaptasi terhadap lingkungan kerja yang baru. Mengingat Ruparupa merupakan platform omnichannel dari Kawan Lama Group, yang melayani berbagai merek ternama seperti Ace, Informa, dan Selma, penting bagi saya sebagai pekerja magang untuk memahami tidak hanya peran saya dalam tim, tetapi juga bagaimana seluruh perusahaan beroperasi.

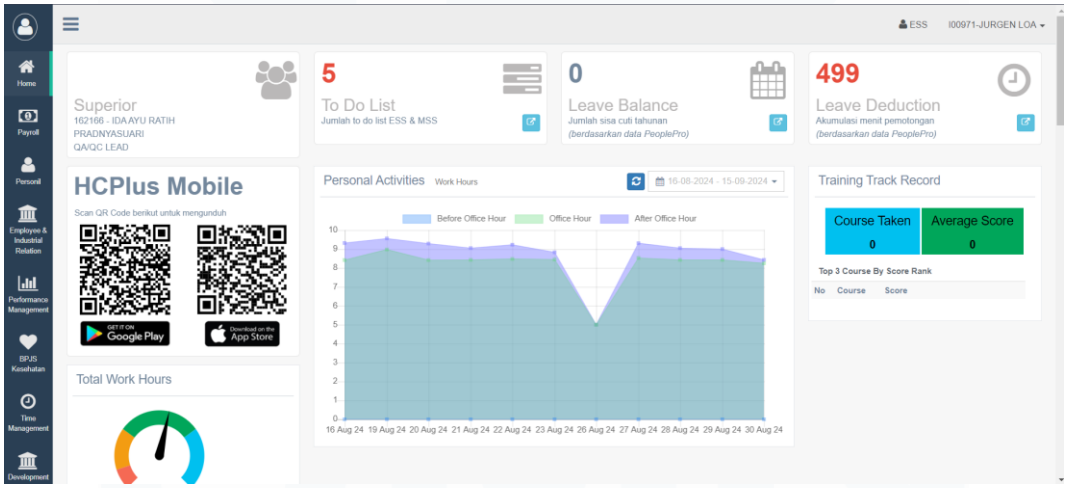
Pada hari pertama magang, saya dan para pekerja magang lainnya dibimbing oleh tim HRD untuk mengenal lebih dalam tentang Ruparupa dan Kawan Lama Group. Kami diajak berkeliling kantor pusat Kawan Lama, yang terdiri dari beberapa lantai, masing-masing ditempati oleh berbagai brand dari grup ini. Setiap lantai memiliki suasana kerja yang khas dan peruntukan yang berbeda, dari ruang kerja yang modern hingga fasilitas pendukung yang menunjang kinerja karyawan. Tur ini sangat bermanfaat bagi saya untuk mendapatkan gambaran umum tentang struktur organisasi dan budaya kerja di perusahaan.

Setelah tur kantor selesai, kami diundang untuk makan siang bersama dengan tim HRD. Kegiatan ini tidak hanya menjadi kesempatan untuk mempererat hubungan dengan tim HRD, tetapi juga untuk mengenal lebih jauh tentang etika dan budaya kerja di Ruparupa. Makan siang bersama ini juga memberikan ruang bagi kami para magang untuk bertanya langsung dan berdiskusi santai tentang pengalaman kerja di perusahaan ini.

Setelah makan siang, saya diantar ke area kantor Ruparupa, di mana saya akan ditempatkan selama masa magang. Di sini, saya diperkenalkan dengan tim *Quality Assurance* (QA), tempat saya akan bekerja. Pertama-tama, saya bertemu dengan *Senior Lead* QA yang memberikan pengenalan mendalam tentang peran dan tanggung jawab saya di tim ini. *Senior Lead* QA menjelaskan

bagaimana tim QA berfungsi, dan bagaimana setiap individu berkontribusi dalam memastikan bahwa semua produk dan fitur yang diluncurkan Ruparupa memenuhi standar kualitas yang telah ditetapkan.

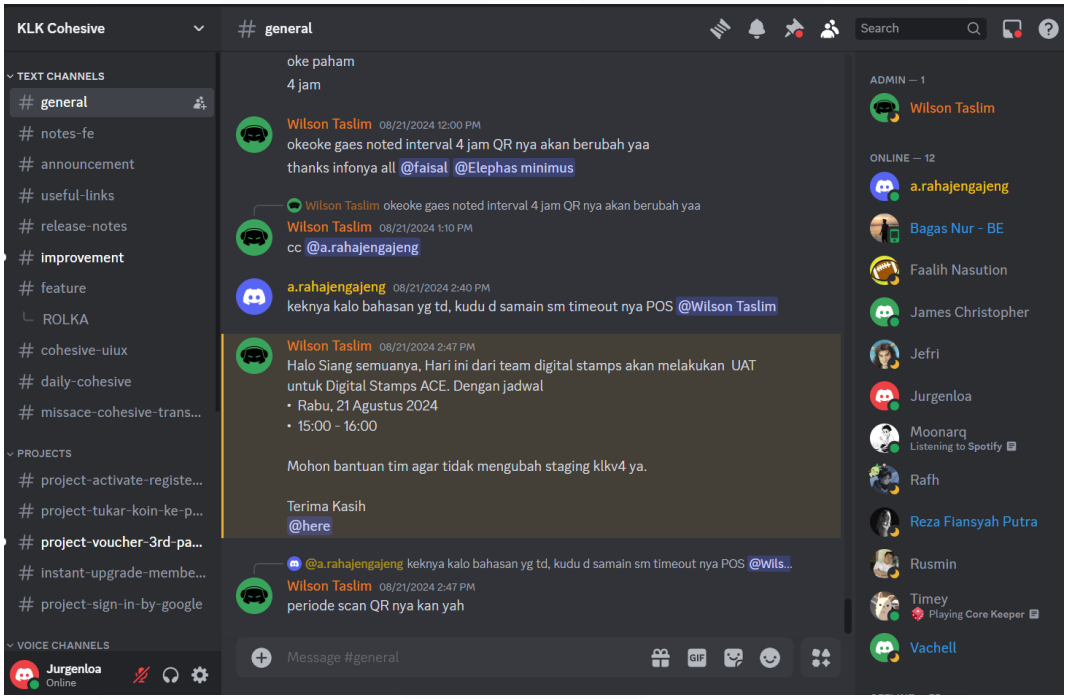
Selanjutnya, saya diperkenalkan kepada Lead QA dan anggota QA lainnya. Mereka menjelaskan secara rinci tugas-tugas harian yang akan saya lakukan, serta bagaimana koordinasi antar anggota tim dilakukan untuk mencapai tujuan bersama. Setelah perkenalan ini, saya diberikan kesempatan untuk menjelajahi situs Ruparupa secara mendalam. Tugas ini bertujuan untuk memastikan bahwa saya memahami semua fitur yang ada di situs, mulai dari fungsi belanja, fitur membership, hingga berbagai layanan yang ditawarkan kepada pelanggan. Pengenalan ini sangat penting agar saya dapat melakukan tugas QA dengan lebih efektif, memahami secara langsung pengalaman pengguna, dan mengidentifikasi area yang memerlukan peningkatan.



Gambar 3.1 Website Internal Perusahaan

Pada gambar 3.1 menampilkan bagian penting lainnya dari proses on-boarding adalah pelatihan mengenai HCPlus, sistem internal *all-in-one* yang digunakan di Ruparupa untuk berbagai keperluan administrasi, termasuk absensi, pengajuan cuti, dan pemesanan makanan. HCPlus hanya bisa diakses melalui jaringan kantor atau VPN khusus, menambah lapisan keamanan yang diperlukan untuk melindungi data karyawan dan perusahaan. Pelatihan ini mengajarkan saya bagaimana cara melakukan absensi harian, revisi absensi, dan

pengajuan cuti dengan efektif. Selain itu, sistem ini memungkinkan saya untuk melakukan pemesanan makanan langsung melalui aplikasi, yang mempermudah proses pemesanan tanpa perlu meninggalkan tempat kerja.



Gambar 3.2 Discord

Pada gambar 3.2 menampilkan tempat berkomunikasi untuk tim *cohesive*. Di Ruparupa, komunikasi dan koordinasi antar tim dilakukan menggunakan platform Discord. Sebagai platform yang sering digunakan dalam lingkungan gaming, Discord di Ruparupa telah diadaptasi untuk kebutuhan profesional. Discord menyediakan berbagai fitur seperti *voice call*, *video call*, *text messaging*, dan *file sharing* yang memudahkan tim QA untuk berkolaborasi, baik dalam diskusi harian maupun dalam pertemuan yang lebih formal. Discord juga memungkinkan kami untuk mengetahui status karyawan lain, seperti "*Available*," "*In a Meeting*," atau "*Do Not Disturb*," yang sangat membantu dalam menjaga komunikasi yang efektif dan efisien.

Proses pengenalan tools ini sangat penting karena membantu saya memahami bagaimana seluruh sistem di Ruparupa bekerja, dan bagaimana saya dapat memanfaatkan tools tersebut untuk mendukung tugas-tugas QA yang

akan saya kerjakan. Selain itu, pelatihan ini juga memperkenalkan saya pada berbagai alat dan teknologi yang akan digunakan dalam proyek-proyek mendatang, termasuk tools untuk manajemen proyek, pelacakan bug, dan pengujian otomatis.

Melalui proses *on-boarding* yang komprehensif ini, saya tidak hanya mendapatkan pemahaman mendalam tentang peran saya di tim QA, tetapi juga tentang bagaimana Ruparupa beroperasi sebagai sebuah perusahaan. Pengalaman ini memberikan fondasi yang kuat bagi saya untuk memulai masa magang dengan penuh percaya diri dan siap memberikan kontribusi maksimal bagi tim dan perusahaan.

Grooming project adalah salah satu tahapan yang sangat krusial dalam siklus pengembangan perangkat lunak, terutama bagi tim *Quality Assurance* (QA) dan tim pengembang (*developer*). Tahap ini dilakukan sebelum proses pengembangan dimulai, bertujuan untuk memastikan bahwa seluruh anggota tim memahami secara menyeluruh tujuan, kebutuhan, dan lingkup proyek yang akan dikerjakan. *Grooming* dilakukan untuk mengidentifikasi dan mengklarifikasi semua elemen yang terkait dengan proyek, termasuk persyaratan (*requirements*), tujuan utama, dan metrik kunci yang akan digunakan untuk mengevaluasi kesuksesan proyek. Pada tahap ini, *Project Manager* (PM) bertanggung jawab untuk memimpin sesi grooming, di mana mereka memberikan penjelasan mendalam kepada tim pengembang dan QA tentang apa yang diharapkan dari proyek tersebut. PM akan merinci spesifikasi yang telah dikumpulkan dari pengguna akhir atau stakeholders, dan memastikan bahwa tidak ada ambiguitas dalam pemahaman tim mengenai persyaratan proyek. *Grooming* biasanya berlangsung selama satu hari, dengan durasi yang dapat berkisar antara satu hingga dua jam, tergantung pada kompleksitas proyek dan kebutuhan diskusi.

Proses *grooming* tidak hanya terbatas pada penyampaian informasi dari PM ke tim, tetapi juga melibatkan diskusi dua arah. Selama grooming, anggota tim

QA dan pengembang diberikan kesempatan untuk bertanya, memberikan masukan, dan mendiskusikan potensi tantangan yang mungkin dihadapi selama pengembangan dan pengujian. Diskusi ini penting untuk mengidentifikasi risiko-risiko yang mungkin tidak terlihat pada tahap awal dan untuk memastikan bahwa semua skenario pengujian telah dipertimbangkan.

QA memegang peran penting selama *grooming* karena mereka perlu memahami setiap detail dari persyaratan proyek untuk dapat merancang *test case* yang efektif. *Test case* yang dibuat oleh QA harus mencakup semua kemungkinan skenario, baik positif maupun negatif, untuk memastikan bahwa produk akhir memenuhi standar kualitas yang diharapkan. Oleh karena itu, QA harus benar-benar mendalami setiap bagian dari dokumen PRD (*Product Requirements Document*) yang dihasilkan setelah sesi *grooming*. Selanjutnya, sesi *grooming* juga mencakup pembahasan mengenai timeline proyek, di mana PM akan menjelaskan batas waktu yang diharapkan untuk setiap tahapan, mulai dari pengembangan hingga pengujian dan akhirnya peluncuran produk. Pemahaman yang jelas mengenai timeline ini penting agar semua anggota tim dapat menyusun rencana kerja yang realistis dan terukur.

Dalam beberapa kasus, grooming juga bisa melibatkan revisi terhadap persyaratan proyek jika selama diskusi terungkap bahwa beberapa aspek dari proyek belum sepenuhnya jelas atau jika ada potensi untuk meningkatkan efisiensi proses pengembangan. Revisi ini bertujuan untuk menghindari potensi masalah di kemudian hari yang bisa mengakibatkan keterlambatan atau biaya tambahan. Secara keseluruhan, grooming project adalah tahap persiapan yang sangat penting untuk memastikan bahwa seluruh tim memiliki pemahaman yang sama mengenai tujuan dan persyaratan proyek. Melalui *grooming* yang efektif, tim dapat meminimalkan risiko kesalahpahaman dan memastikan bahwa pengembangan perangkat lunak berjalan dengan lancar dan sesuai dengan rencana yang telah ditetapkan.

3.2.2 Mempelajari Product Requirment Document

Dalam proses pengembangan proyek di Ruparupa, setelah proses grooming dilakukan, langkah berikutnya adalah mempelajari *Project Requirements Document* (PRD) atau dokumen proyek yang disusun dengan sangat teliti oleh *Project Manager* (PM). Sebelum pembuatan PRD, PM biasanya mengadakan pertemuan terlebih dahulu dengan pengguna atau user untuk memahami kebutuhan mereka secara mendalam. Pertemuan ini bertujuan untuk mengumpulkan semua informasi yang relevan terkait dengan fitur-fitur yang diinginkan, tujuan bisnis yang hendak dicapai, serta kendala-kendala yang mungkin dihadapi. Setelah memahami kebutuhan pengguna, PM kemudian menyusun PRD dengan sangat rinci, memastikan bahwa semua elemen yang diperlukan untuk pengembangan proyek telah terdefinisi dengan jelas. PRD ini disusun agar dapat dimengerti dengan baik oleh semua pihak yang terlibat, termasuk tim pengembang (*developer*) dan tim *Quality Assurance* (QA).

Mempelajari PRD merupakan langkah krusial dalam siklus pengembangan proyek. Sebagai QA, memahami isi PRD secara menyeluruh adalah hal yang wajib. Dokumen PRD berfungsi sebagai panduan utama yang mencakup berbagai aspek penting dari proyek, termasuk *requirement, goals, dan key metrics. Requirement* dalam PRD menjelaskan secara spesifik apa yang harus dibangun atau dikembangkan, mencakup segala fitur dan fungsi yang diinginkan oleh pengguna. Ini adalah dasar dari seluruh pengembangan sistem, karena setiap bagian dari requirement harus dipenuhi untuk memastikan bahwa produk akhir sesuai dengan harapan pengguna.

Selanjutnya, goals dalam PRD menggambarkan tujuan yang ingin dicapai melalui proyek tersebut. Ini bisa meliputi berbagai aspek, seperti peningkatan efisiensi operasional, peningkatan pengalaman pengguna, atau peningkatan penjualan. Goals ini memberikan arah yang jelas bagi seluruh tim proyek dan memastikan bahwa semua orang bekerja menuju tujuan yang sama.

Key metrics adalah indikator yang digunakan untuk mengukur keberhasilan proyek dalam mencapai goals yang telah ditetapkan. *Metrics* ini bisa berupa waktu respon sistem, tingkat kepuasan pengguna, atau tingkat konversi penjualan. *Key metrics* sangat penting karena memberikan tolok ukur yang objektif untuk menilai apakah proyek telah berhasil atau perlu dilakukan perbaikan lebih lanjut.

Proses mempelajari PRD di Ruparupa biasanya memakan waktu 1-2 hari, tergantung pada kompleksitas proyek. Selama periode ini, QA akan mendalami setiap detail dari PRD, memahami seluruh *requirement, goals, dan key metrics* yang telah ditetapkan. QA juga akan mengidentifikasi potensi risiko dan area yang memerlukan perhatian khusus selama proses pengembangan dan pengujian. Dengan pemahaman yang mendalam ini, QA dapat memastikan bahwa seluruh fitur dan fungsi yang dikembangkan sesuai dengan ekspektasi pengguna dan memenuhi standar kualitas yang telah ditetapkan.

3.2.3 Membuat *Test Case*

Pembuatan test case di Ruparupa merupakan salah satu tahapan krusial dalam proses *Quality Assurance* (QA) yang bertujuan untuk menjamin bahwa setiap fitur dan fungsi yang dikembangkan sesuai dengan spesifikasi dan kebutuhan yang telah ditetapkan. Setelah QA mempelajari dokumen PRD (*Product Requirement Document*) dengan seksama dan memahami setiap aspek dari project yang akan diuji, langkah selanjutnya adalah menyusun test case. Test case ini berfungsi sebagai panduan yang memetakan seluruh skenario pengujian yang akan dilakukan untuk memastikan bahwa produk berfungsi dengan baik sebelum diterapkan di lingkungan produksi.

Di Ruparupa, QA menggunakan Google Sheets sebagai alat utama untuk menyusun dan mengelola test case. Penggunaan Google Sheets memungkinkan tim QA untuk secara kolaboratif menyusun, mengedit, dan memantau *test case* secara real-time. Test case ini disusun dengan mengikuti struktur yang jelas dan terperinci untuk memastikan bahwa setiap aspek pengujian tercakup secara

komprehensif. Kolom-kolom yang digunakan dalam Google *Sheets* meliputi beberapa elemen penting.

Pertama, kolom Title merupakan tempat untuk mencantumkan judul atau nama dari test case. Judul ini harus spesifik dan mendeskripsikan secara singkat namun jelas apa yang akan diuji. Misalnya, jika *test case* tersebut berhubungan dengan fitur pencarian produk, maka judulnya mungkin berupa "Pencarian Produk Berdasarkan Nama. Judul yang jelas membantu memudahkan identifikasi test case dan memastikan bahwa setiap anggota tim memahami dengan cepat apa yang sedang diuji.

Selanjutnya, kolom Steps berisi urutan langkah-langkah yang harus dilakukan selama proses pengujian. Langkah-langkah ini disusun secara berurutan dan mencakup semua tindakan yang perlu dilakukan oleh QA. Setiap langkah harus dijelaskan dengan rinci, mulai dari tindakan awal hingga hasil akhir yang diharapkan. Misalnya, langkah pertama mungkin adalah "Buka halaman utama situs Ruparupa," diikuti dengan langkah-langkah berikutnya seperti "Masukkan kata kunci produk di kolom pencarian" dan "Klik tombol Cari" Penguraian langkah-langkah yang terperinci ini memastikan bahwa setiap langkah pengujian dijalankan secara konsisten dan terhindar dari kesalahan interpretasi.

Kolom Pre Condition mencatat kondisi awal yang harus dipenuhi sebelum test case dapat dijalankan. Kondisi awal ini meliputi semua persyaratan atau setup yang harus dilakukan agar test case dapat berjalan dengan benar. Contohnya, jika pengujian berkaitan dengan fitur login, kondisi awalnya mungkin mencakup bahwa pengguna sudah memiliki akun yang valid dan bahwa sistem sudah dalam keadaan login. *Pre Condition* ini penting untuk memastikan bahwa setiap test case dijalankan dalam konteks yang sesuai dan bahwa hasil pengujian yang diperoleh dapat diandalkan.

Kemudian, kolom *Expected Result* menjelaskan hasil yang diharapkan dari test case jika sistem berfungsi sesuai dengan yang diharapkan. *Expected Result* ini merupakan acuan utama untuk menentukan apakah test case tersebut berhasil atau tidak. Misalnya, jika test case tersebut menguji fitur pencarian produk, maka expected result-nya bisa berupa "Sistem menampilkan daftar produk yang sesuai dengan kata kunci pencarian." Dengan menetapkan expected result yang jelas, QA dapat dengan mudah mengevaluasi apakah hasil yang diperoleh sesuai dengan yang diharapkan atau ada penyimpangan yang perlu diperbaiki.

Selanjutnya, kolom *Actual Result* digunakan untuk mencatat hasil nyata dari pengujian setelah test case dijalankan. QA akan membandingkan actual result ini dengan expected result untuk menentukan apakah test case tersebut telah berhasil atau gagal. Jika hasil yang diperoleh sesuai dengan *expected result*, maka test case tersebut dapat ditandai sebagai "*Passed*." Namun, jika terdapat perbedaan antara actual result dan expected result, maka test case akan ditandai sebagai "*Failed*," dan QA harus mencatat detil penyimpangan yang terjadi serta mungkin mengajukan perbaikan kepada tim pengembang.

Terakhir, kolom *Status* merupakan *dropdown* yang digunakan untuk menandai apakah *test case* telah berhasil (*Passed*) atau gagal (*Failed*). QA akan mengisi kolom ini setelah menjalankan pengujian dan mengevaluasi hasilnya. Jika *test case* gagal, QA juga mungkin memberikan catatan tambahan untuk menjelaskan mengapa test case tersebut gagal dan tindakan apa yang diperlukan untuk memperbaikinya. Selain itu, setelah seluruh *test case* disusun dan dijalankan, status setiap test case juga dapat digunakan untuk melacak kemajuan pengujian secara keseluruhan dan memastikan bahwa semua skenario pengujian telah dijalankan sebelum project dipindahkan ke tahap selanjutnya.

Setelah seluruh *test case* disusun dengan lengkap, langkah berikutnya adalah mengajukan *test case* tersebut kepada *Project Manager* (PM) untuk

direview. *review* dari PM ini sangat penting untuk memastikan bahwa semua skenario pengujian yang diperlukan telah dicakup dan bahwa test case tersebut sesuai dengan *requirement* yang tercantum dalam PRD. PM akan meninjau setiap test case dengan teliti, dan mungkin memberikan masukan atau meminta revisi jika ada aspek yang perlu diperbaiki atau ditambahkan. Proses review ini memastikan bahwa tidak ada aspek penting yang terlewatkan dan bahwa pengujian dapat dilakukan dengan tingkat keakuratan yang tinggi.

Setelah *test case* disetujui oleh PM, QA dapat segera memulai proses pengujian. *Testing* ini biasanya dilakukan dalam satu sprint, yang berlangsung selama dua minggu. Dengan mengikuti *test case* yang telah disusun dan disetujui, QA dapat memastikan bahwa setiap fitur dan fungsi yang dikembangkan telah diuji dengan baik dan memenuhi standar kualitas yang diharapkan. Selain itu, dokumentasi test case yang detail juga memungkinkan QA untuk mengulang pengujian dengan konsistensi jika diperlukan, terutama jika ada perubahan atau perbaikan yang harus dilakukan setelah pengujian awal.

Dengan menggunakan struktur yang jelas dan mendetail dalam pembuatan *test case* ini, Ruparupa dapat memastikan bahwa setiap langkah dalam proses QA dilakukan dengan standar yang tinggi, sehingga hasil akhir dari pengembangan project sesuai dengan ekspektasi dan kebutuhan pengguna. Proses ini tidak hanya membantu dalam memastikan kualitas produk, tetapi juga meningkatkan efisiensi *testing* dan meminimalkan risiko kesalahan, sehingga produk yang dihasilkan dapat diluncurkan dengan keyakinan penuh bahwa produk tersebut telah diuji dengan baik dan siap digunakan oleh pengguna.

3.2.4 *Testing*

Pada tahap *Testing* di Ruparupa, QA memegang peran penting dalam memastikan bahwa setiap fitur, modul, dan sistem yang dikembangkan berfungsi dengan baik dan sesuai dengan kebutuhan pengguna sebelum produk tersebut diluncurkan. Proses testing dilakukan setelah developer menyelesaikan tahap pengembangan, dan QA bertanggung jawab untuk menjalankan

serangkaian uji coba yang telah direncanakan dalam *test case*. Testing ini bertujuan untuk mendeteksi potensi bug, memastikan performa sistem stabil, dan mengevaluasi apakah fitur-fitur baru sudah berfungsi sesuai dengan yang diharapkan berdasarkan dokumen PRD.

Salah satu aspek penting dalam proses testing di Ruparupa adalah bahwa QA harus melakukan pengujian di empat platform utama, yaitu mobile web, desktop, aplikasi Android, dan aplikasi IOS. Pengujian di berbagai platform ini sangat penting untuk memastikan bahwa semua fitur dan fungsionalitas dapat berjalan lancar di berbagai jenis perangkat dan sistem operasi yang digunakan oleh customer. Karena Ruparupa adalah platform *omnichannel* yang melayani berbagai macam pengguna, pengalaman pengguna (UX) dan antarmuka pengguna (UI) harus konsisten dan memadai di semua platform.

Selama proses testing, QA harus memastikan bahwa flow dari setiap fitur sudah berjalan dengan lancar. *Flow* ini mencakup seluruh alur interaksi pengguna dengan sistem, mulai dari awal hingga akhir. Misalnya, jika QA sedang menguji fitur *checkout*, mereka harus memastikan bahwa pengguna dapat dengan mudah menambahkan barang ke keranjang belanja, memasukkan informasi pengiriman, memilih metode pembayaran, dan menyelesaikan pembelian tanpa kendala. Setiap langkah dalam *flow* harus bebas dari error dan harus memberikan pengalaman pengguna yang intuitif dan efisien. QA akan mengamati apakah setiap langkah dalam *flow* tersebut sudah sesuai dengan skenario yang telah disusun dalam test case, dan jika terdapat kesalahan atau ketidakcocokan, QA akan mendokumentasikan hal tersebut sebagai bug yang perlu diperbaiki oleh *developer*.

Selain memastikan flow berjalan dengan baik, QA juga harus memperhatikan aspek UI (*User Interface*) dan UX (*User Experience*) selama proses pengujian. Desain antarmuka pengguna harus mudah digunakan, estetis, dan fungsional, sementara pengalaman pengguna harus mulus dan tidak

membingungkan. Misalnya, jika pengujian dilakukan pada aplikasi mobile, QA harus memastikan bahwa tombol-tombol dan elemen UI lainnya tidak hanya tampil dengan baik di layar perangkat mobile, tetapi juga mudah diakses dan digunakan. Ukuran tombol, jarak antar elemen, dan tata letak halaman semuanya harus diperiksa untuk memastikan bahwa pengguna dapat berinteraksi dengan aplikasi tanpa hambatan. Jika ada elemen UI yang tidak responsif atau tidak konsisten, QA akan mencatatnya sebagai masalah yang perlu diperbaiki.

Dari sisi UX, QA juga bertanggung jawab untuk memastikan bahwa setiap fitur yang diuji memberikan kemudahan bagi *customer* dalam menyelesaikan tugas-tugas yang ingin mereka lakukan. Pengalaman pengguna harus dirancang dengan fokus pada kebutuhan dan ekspektasi customer. Misalnya, jika QA sedang menguji fitur pencarian produk, mereka harus memastikan bahwa hasil pencarian ditampilkan dengan cepat dan relevan dengan kata kunci yang dimasukkan oleh pengguna. Jika ada kesulitan dalam menemukan produk atau prosesnya memakan waktu terlalu lama, QA akan menganggap ini sebagai masalah UX yang harus diperbaiki. Pada intinya, QA harus selalu mengevaluasi apakah fitur-fitur yang diuji mampu memberikan nilai tambah bagi pengguna dan mempermudah mereka dalam berinteraksi dengan platform Ruparupa.

Selain UI dan UX, QA juga harus memastikan bahwa setiap fitur yang diuji telah sesuai dengan *requirement* yang telah ditetapkan oleh user. *Requirement* ini biasanya telah dijelaskan secara detail dalam dokumen PRD yang telah dipelajari QA sebelumnya. Setiap skenario pengujian yang dilakukan oleh QA harus memeriksa apakah fitur tersebut sudah memenuhi tujuan dan ekspektasi *user*, baik dari segi fungsionalitas maupun performa. Jika terdapat ketidaksesuaian antara fitur yang diuji dan requirement yang ada, QA akan mengajukan feedback kepada tim pengembang agar perubahan yang diperlukan bisa segera dilakukan.

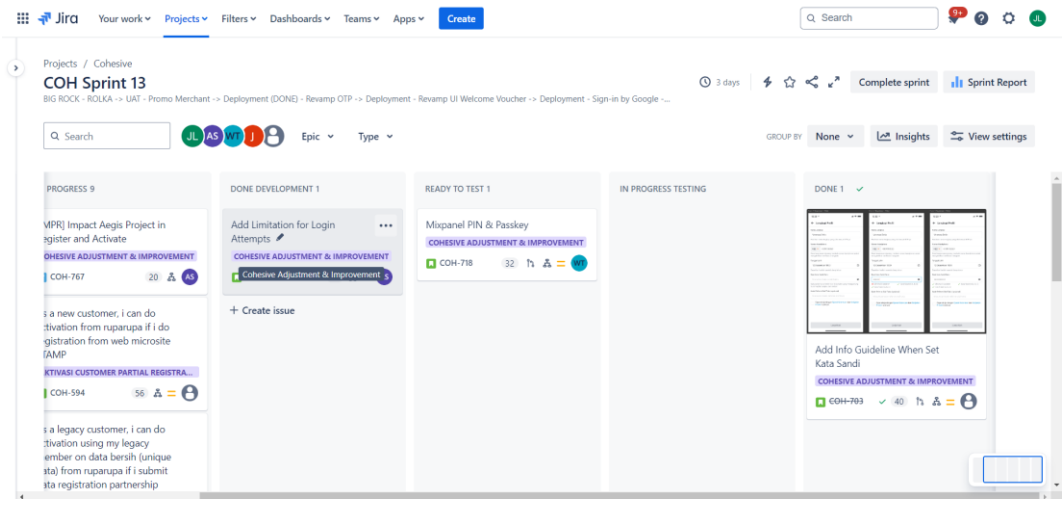
Setelah QA selesai melakukan pengujian, mereka akan memperbarui Test Tracking dan menandai setiap test case dengan status "*Passed*" atau "*Failed*"

berdasarkan hasil pengujian. *Test case* yang telah diselesaikan dan dinyatakan berhasil akan menandakan bahwa fitur tersebut siap untuk dilanjutkan ke tahap SIT (*System Integration Testing*) dan UAT (*User Acceptance Testing*). Namun, jika terdapat bug atau masalah yang ditemukan, QA akan mencatatnya di sistem bug tracking dan berkoordinasi dengan tim untuk memperbaiki masalah tersebut sebelum fitur tersebut bisa diuji ulang.

Secara keseluruhan, proses *testing* di Ruparupa sangat menyeluruh dan detail, dengan fokus pada memastikan bahwa setiap fitur, baik di mobile web, desktop, aplikasi Android, maupun aplikasi IOS, dapat memberikan pengalaman yang mulus dan menyenangkan bagi pengguna. Dengan memperhatikan flow, UI, UX, dan memastikan kesesuaian dengan requirement user, QA berperan penting dalam menjaga kualitas dan kepuasan pengguna terhadap platform Ruparupa.

3.2.5 *Update Test Tracking*

Di Ruparupa, setelah QA menyelesaikan seluruh proses pengujian untuk sebuah proyek, langkah selanjutnya adalah melakukan update status tracking untuk mencerminkan perkembangan terbaru dari pengujian yang telah dilakukan. Proses ini tidak melibatkan pertemuan QA *Review Meeting* seperti di tempat lain, melainkan fokus pada pengkinian status di dalam sistem manajemen proyek yang digunakan, yaitu Jira.



Gambar 3.3 Update Status Jira

Berdasarkan gambar 3.3 bisa dilihat adalah gambar pemindahan status di jira. Setelah semua skenario pengujian telah dijalankan dan hasilnya telah direkam, QA bertanggung jawab untuk memperbarui status Jira dari "*In Progress Testing*" menjadi "*Done*." Perubahan status ini sangat penting karena memberi sinyal kepada *Project Manager* (PM) dan tim lainnya bahwa pengujian telah selesai dan proyek siap untuk melanjutkan ke tahap berikutnya. Dengan status "*Done*," PM dapat menilai bahwa semua kriteria pengujian telah terpenuhi, dan tidak ada masalah yang perlu diselesaikan lebih lanjut sebelum proyek diluncurkan ke tahap berikutnya, seperti *System Integration Testing* (SIT) atau *User Acceptance Testing* (UAT).

Proses update status tracking ini memastikan bahwa komunikasi antar tim berjalan lancar dan setiap anggota tim proyek memiliki informasi terbaru tentang kemajuan proyek. Dengan memanfaatkan Jira sebagai alat pelacak, seluruh proses kerja menjadi lebih transparan dan terstruktur. Setiap perubahan yang dilakukan dapat dilacak kembali, memberikan dokumentasi yang jelas tentang langkah-langkah yang telah diambil selama pengembangan dan pengujian proyek.

Selain itu, dengan adanya update status yang konsisten, QA juga dapat memastikan bahwa waktu proyek dikelola dengan baik dan tidak ada penundaan yang tidak perlu dalam proses transisi dari satu fase ke fase berikutnya. Hal ini membantu menjaga efisiensi dan kualitas dalam keseluruhan siklus hidup pengembangan proyek di Ruparupa.

3.2.6 SIT (*System Integration Testing*) & UAT (*User Acceptance Testing*)

SIT (*System Integration Testing*) dan UAT (*User Acceptance Testing*) merupakan dua tahapan penting dalam pengujian yang dilakukan selama masa magang sebagai *Quality Assurance* (QA) di Ruparupa. SIT adalah salah satu tugas utama yang diemban oleh mahasiswa magang selama masa magang di perusahaan ini. Pada SIT, mahasiswa pekerja magang terlibat secara langsung dalam pengujian sistem secara keseluruhan untuk memastikan bahwa setiap fitur dan modul yang

ada dapat berfungsi dengan baik ketika digabungkan ke dalam satu sistem yang terpadu.

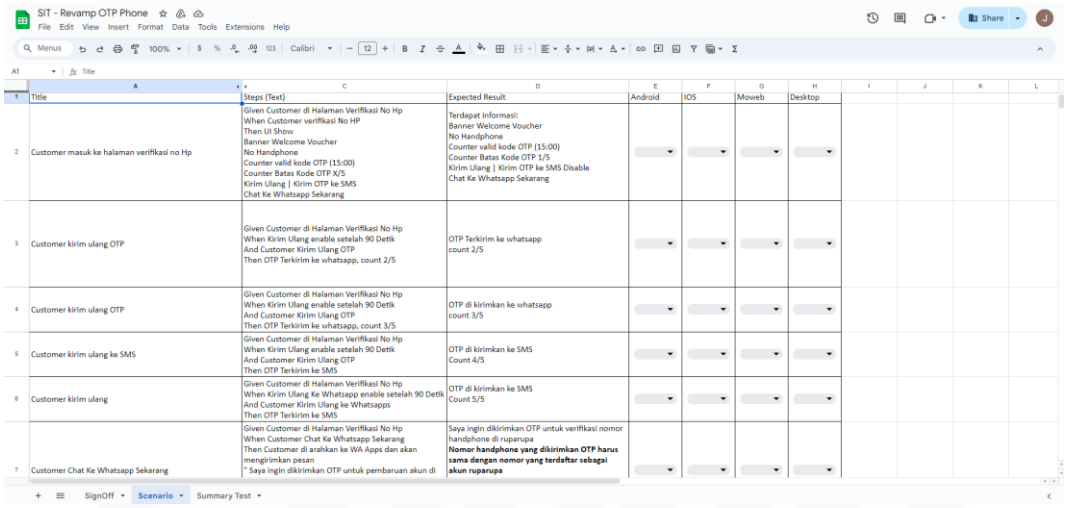
Di awal masa magang, mahasiswa magang bergabung dalam tahap SIT pada berbagai proyek yang sedang berjalan. Mahasiswa tidak hanya bertugas sebagai asisten bagi tester utama, tetapi juga memiliki tanggung jawab langsung dalam beberapa bagian pengujian. Meskipun pada awalnya mahasiswa pekerja magang diberikan tanggung jawab terbatas, keterlibatan ini memungkinkan mahasiswa untuk mempelajari cara kerja pengujian secara keseluruhan, termasuk bagaimana menjalankan skenario pengujian, mengidentifikasi bug, dan mencatat hasil pengujian ke dalam sistem pelacakan seperti Jira.

Selama pengujian SIT, mahasiswa magang di Ruparupa bertanggung jawab untuk memastikan bahwa sistem yang diuji telah memenuhi persyaratan yang ditentukan di dalam PRD (*Product Requirement Document*). Mahasiswa juga terlibat dalam pengujian pada berbagai perangkat, yaitu *mobile web*, *desktop*, aplikasi Android, dan aplikasi iOS. Hal ini dilakukan untuk memastikan bahwa sistem yang diuji berfungsi dengan baik di berbagai platform yang digunakan oleh pelanggan. QA juga harus memastikan bahwa alur kerja (*flow*) sistem sudah berjalan dengan lancar, user interface (UI) dan user experience (UX) telah memenuhi standar kualitas, serta fitur yang diuji dapat memudahkan pengguna sesuai dengan persyaratan yang diajukan oleh user.

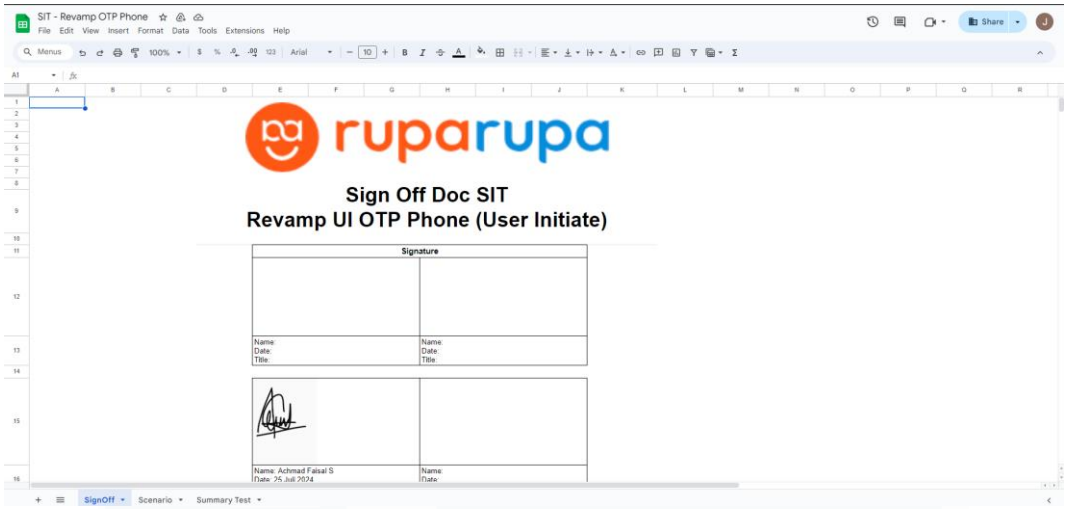
Setelah SIT selesai, tahap berikutnya yang harus dilalui adalah UAT (*User Acceptance Testing*). UAT adalah tahap pengujian di mana pengguna akhir atau user melakukan verifikasi dan validasi terhadap sistem yang telah diuji oleh QA. Dalam tahap ini, user yang terkait dengan proyek akan melakukan pengujian sendiri untuk memastikan bahwa fitur-fitur yang dikembangkan sesuai dengan kebutuhan dan harapan mereka. Peran QA dalam UAT adalah memberikan dukungan kepada user selama proses pengujian, memastikan bahwa setiap bug atau masalah yang dilaporkan user dapat segera ditindaklanjuti, dan mendokumentasikan hasil pengujian UAT tersebut.

SIT dan UAT di Ruparupa tidak hanya memberikan kesempatan kepada mahasiswa magang untuk terlibat langsung dalam pengujian perangkat lunak, tetapi juga memberikan pengalaman berharga dalam memahami bagaimana integrasi sistem diuji secara menyeluruh dan bagaimana pengguna akhir berinteraksi dengan sistem yang sudah selesai dikembangkan. Proses ini memberikan wawasan yang lebih luas tentang siklus pengembangan perangkat lunak dari awal hingga akhir, serta bagaimana kualitas dan kepuasan pengguna menjadi fokus utama dalam setiap pengujian yang dilakukan.

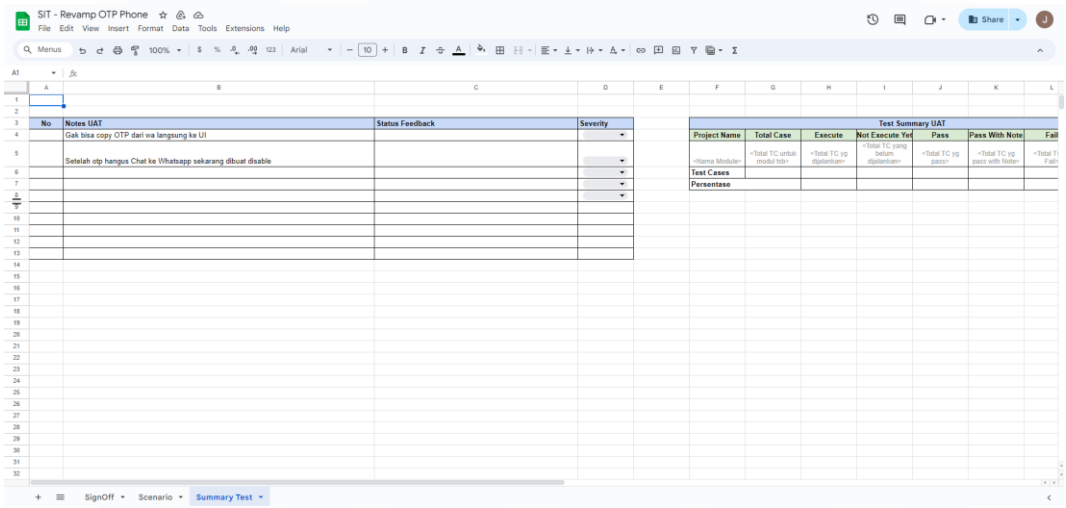
Pada tahap SIT dan UAT di Ruparupa juga memiliki test case yang berbeda. Yang dimana untuk tahap SIT, memiliki test case atau dokumen yang lebih informal sedangkan untuk UAT memiliki testcase atau dokumen UAT yang lebih formal



Gambar 3.4 Skenario SIT

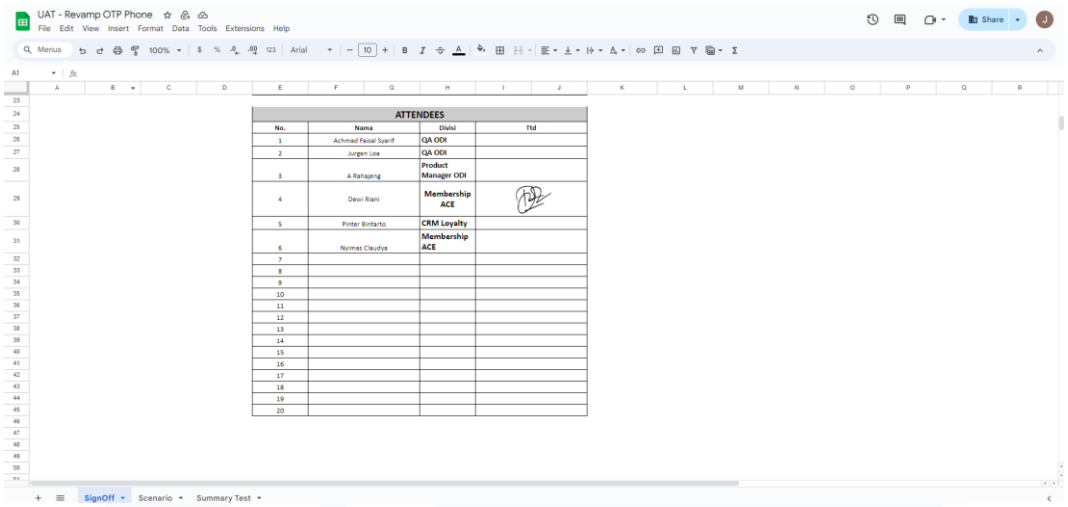


Gambar 3.5 Sign Off SIT



Gambar 3.6 Summary Test SIT

Bisa dilihat dari gambar 3.4 hingga gambar 3.7 bahwa dalam dokumen SIT terdiri dari halaman *Sign Off*, Skenario dan yang terakhir adalah *summary test*. Halaman Sign Off berisi tanda tangan dari tim internal yang menandakan bahwa SIT sudah selesai dan sudah bisa dilanjutkan ke tahap UAT. Selanjutnya dalam halaman Skenario berisi test case yang sudah dibuat oleh QA dan case yang akan dipresentasikan pada tim internal disaat SIT dilakukan. Yang terakhir adalah *Summary Test*, halaman tersebut digunakan untuk mencatat notes yang diberikan oleh tim internal mengenai projeknya agar bisa diperbaiki dan bisa dilanjutkan ke UAT.

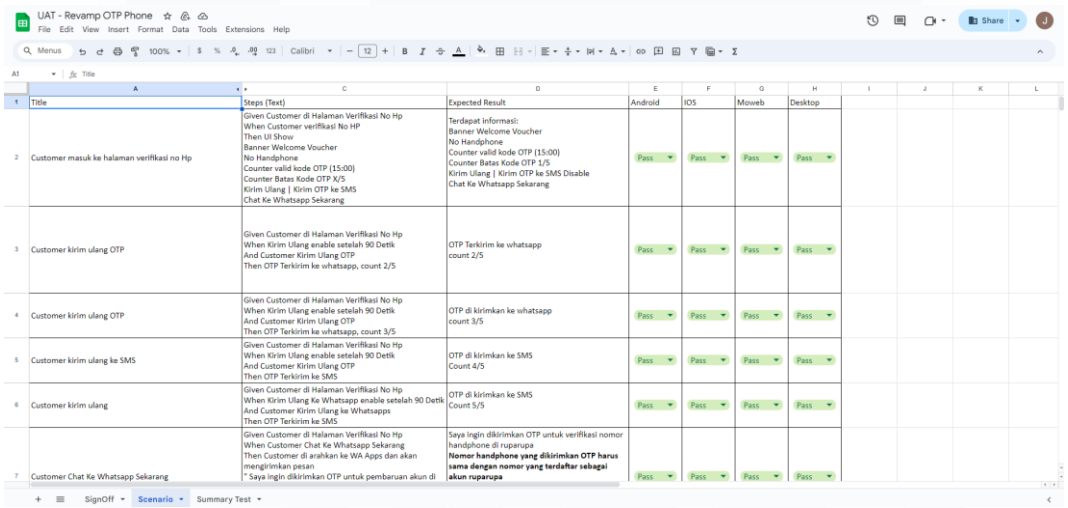


UAT - Revamp OTP Phone

ATTENDEES			
No.	Nama	Divisi	Ttd
1	Achmad Faisal Syarif	QA ODI	
2	Jurgen Loa	QA ODI	
3	A Rahajeng	Product Manager ODI	
4	Dewi Riani	Membership ACE	
5	Pinter Bintarto	CRM Loyalty	
6	Nyimas Claudya	Membership ACE	
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

SignOff | Scenario | Summary Test

Gambar 3.7 Sign Off UAT

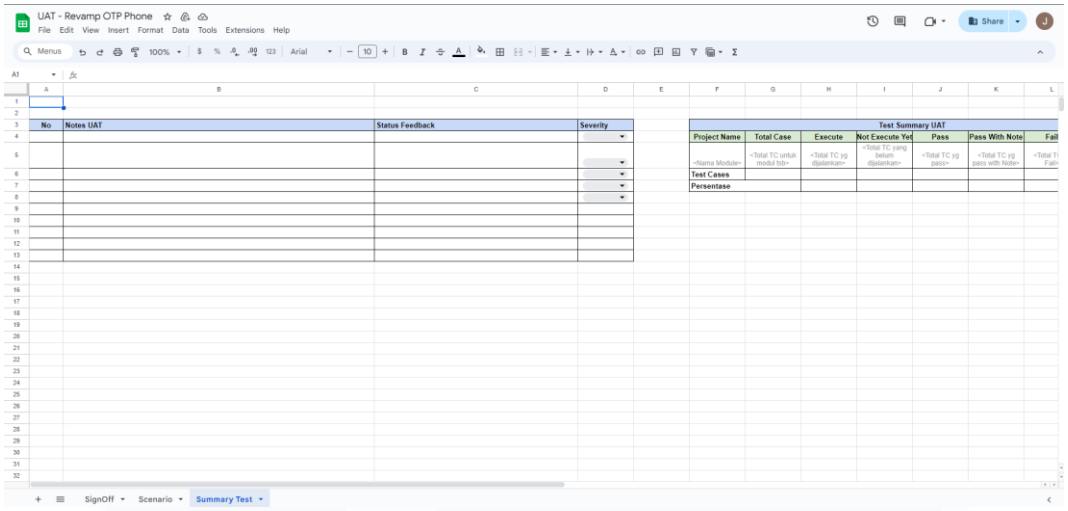


UAT - Revamp OTP Phone

Title	Steps (Text)	Expected Result	Android	IOS	Moweb	Desktop
Customer masuk ke halaman verifikasi no Hp	Given Customer di Halaman Verifikasi No Hp When Customer verifikasi No HP Then UI Show Banner Welcome Voucher No Handphone Counter valid kode OTP (15:00) Counter Batas Kode OTP X/5 Kirim Ulang | Kirim OTP ke SMS Chat Ke Whatsapp Sekarang	Terdapat informasi: Banner Welcome Voucher No Handphone Counter valid kode OTP (15:00) Counter Batas Kode OTP 1/5 Kirim Ulang | Kirim OTP ke SMS Disable Chat Ke Whatsapp Sekarang	Pass	Pass	Pass	Pass
Customer kirim ulang OTP	Given Customer di Halaman Verifikasi No Hp When Kirim Ulang enable setelah 90 Detik And Customer Kirim Ulang OTP Then OTP Terkirim ke whatsapp, count 2/5	OTP Terkirim ke whatsapp count 2/5	Pass	Pass	Pass	Pass
Customer kirim ulang OTP	Given Customer di Halaman Verifikasi No Hp When Kirim Ulang enable setelah 90 Detik And Customer Kirim Ulang OTP Then OTP Terkirim ke whatsapp, count 3/5	OTP di kirimkan ke whatsapp count 3/5	Pass	Pass	Pass	Pass
Customer kirim ulang ke SMS	Given Customer di Halaman Verifikasi No Hp When Kirim Ulang enable setelah 90 Detik And Customer Kirim Ulang OTP Then OTP Terkirim ke SMS	OTP di kirimkan ke SMS Count 4/5	Pass	Pass	Pass	Pass
Customer kirim ulang	Given Customer di Halaman Verifikasi No Hp When Kirim Ulang Ke Whatsapp enable setelah 90 Detik And Customer Kirim Ulang ke Whatsapps Then OTP Terkirim ke SMS	OTP di kirimkan ke SMS Count 5/5	Pass	Pass	Pass	Pass
Customer Chat Ke Whatsapp Sekarang	Given Customer di Halaman Verifikasi No Hp When Customer Chat Ke Whatsapp Sekarang Then Customer di arahkan ke WA Apps dan akan mengirimkan pesan " Saya ingin dikirimkan OTP untuk pembaruan akun di	Saya ingin dikirimkan OTP untuk verifikasi nomor handphone di ruparupa **Nomor handphone yang dikirimkan OTP harus sama dengan nomor yang terdaftar sebagai akun ruparupa**	Pass	Pass	Pass	Pass

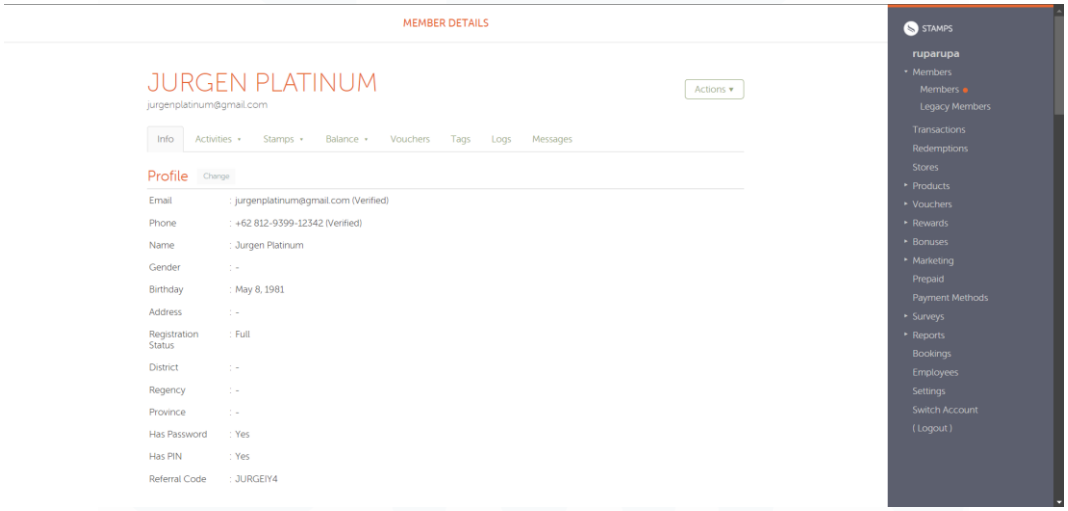
SignOff | Scenario | Summary Test

Gambar 3.8 Scenario UAT



Gambar 3.9 Summary Test UAT

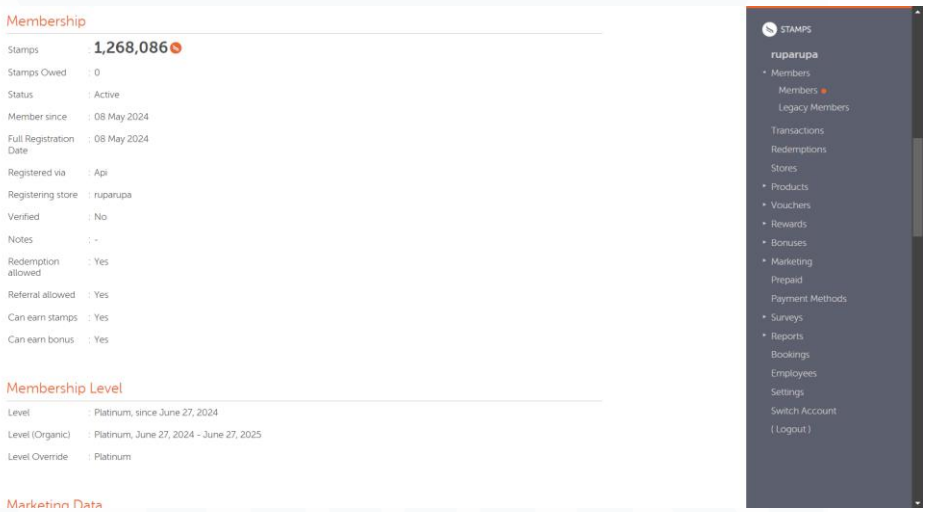
Bisa dilihat dari gambar 3.7 bahwa di *Sign Off* UAT sudah terdapat nama user yang berasal dari Ace. Pada gambar-gambar selanjutnya masih memiliki halaman yang sama dengan dokumen disaat SIT.



Gambar 3.10 Summary Test UAT

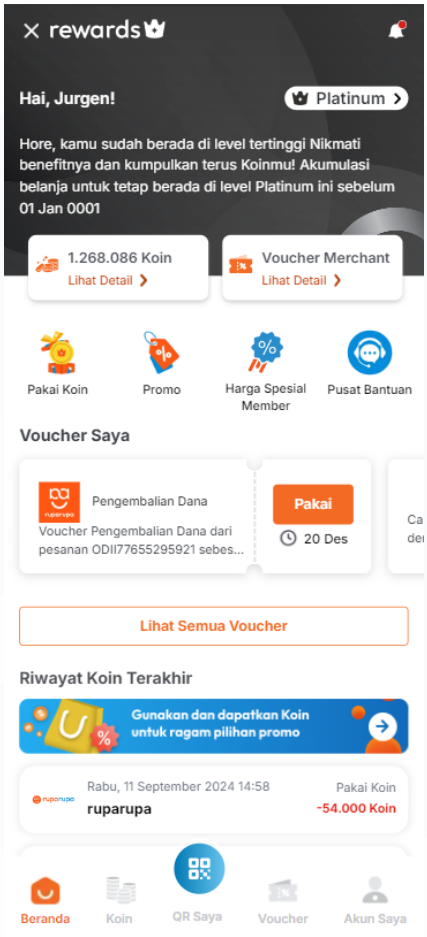
Dalam projek cohesive yang mahasiswa ditempatkan oleh *Lead Tribe* 1. Mahasiswa akan sering membuka dashboard stamps yang dimana tugas utama dari projek *cohesive* adalah untuk melihat dan memastika *flow membership* dan customer berjalan dengan baik. Juga mahasiswa akan melakukan pengecekan terhadap data seperti data *partial register*, data bersih, data kotor dan data *full*

register. Setiap data tersebut akan dilakukan testing oleh QA untuk mengetahui apakah customer dengan data tersebut bisa menjalankan fitur-fitur yang nantinya akan ada di ruparupa. Juga pada projek *Cohesive*, mahasiswa juga harus memastikan setiap level membership akan mendapatkan persenan koin yang berbeda dari setiap pembelian yang dilakukan. Untuk member yang Platinum akan mendapatkan koin 1.5% dari total pembelian, *Gold* akan mendapatkan 1% dan *silver* akan mendapatkan 0.5% dari pembelian. Dalam gambar 3.2.7.6 adalah gambar dashboard stamps yang menunjukkan akun jurgenplatinum@gmail.com yang di *testing* di staging sudah verified dan juga sudah *full register*. Dalam *dashboard* stamps, seorang QA juga bisa melihat *membership* level, memiliki *pin & passkey*, koin yang dimiliki dan juga beberapa data statistik lainnya yang dimiliki oleh customer.

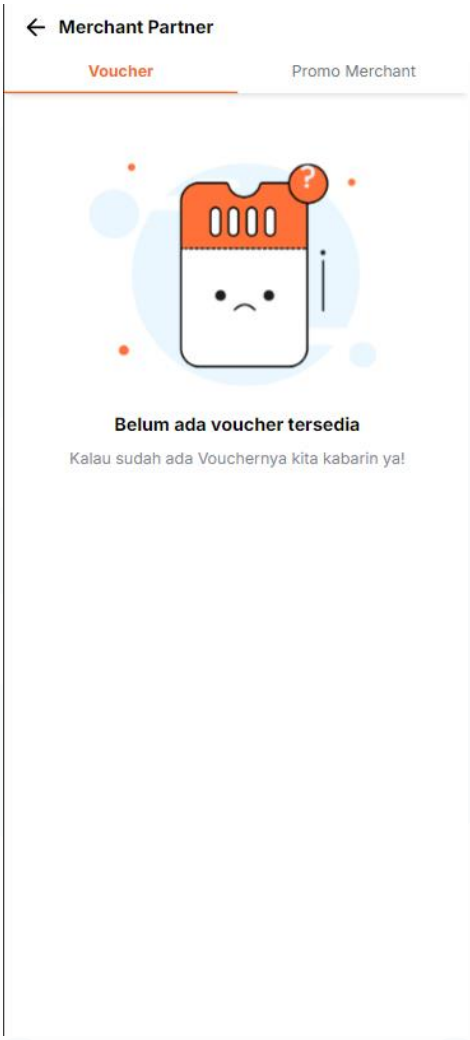


Gambar 3.11 Summary Test UAT

Dari gambar 3.11 menunjukkan bahwa member jurgenplatinum@gmail.com memiliki koin 1,268,086 dan customer memiliki membership level platinum dan juga customer bisa mendapatkan *stamps* dan *bonus*. Juga dalam *dashboard* stamps, QA bisa melihat apakah customer mendapatkan voucher dari registrasi dan apakah customer sudah ada melakukan claim terhadap *voucher*.



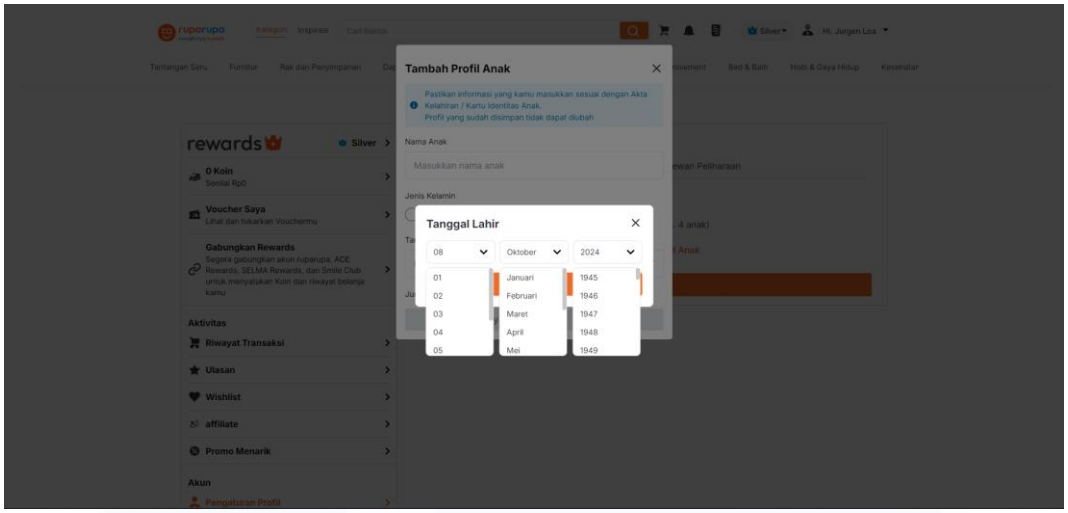
Gambar 3.12 Halaman rr



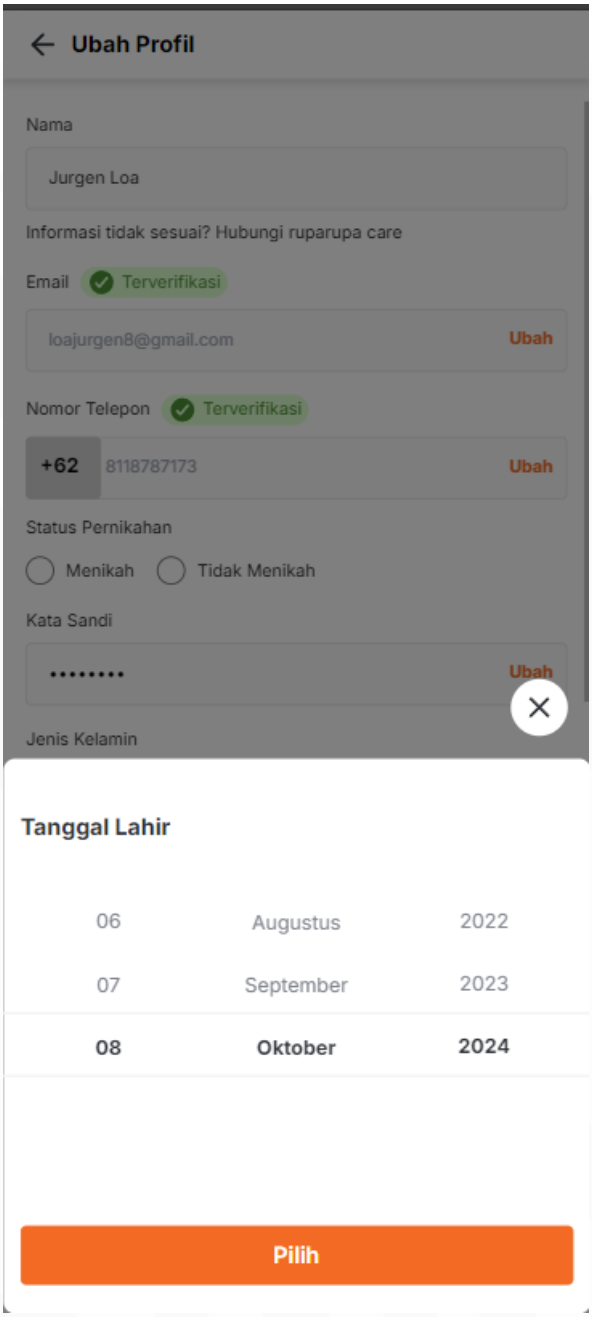
Gambar 3.13. Halaman voucher

Pada halaman Ruparupa di gambar 3.12 menunjukkan *membership level* dari *customer* dan juga pada halaman 3.13 yaitu halaman voucher yang nantinya voucher yang di *claim* oleh *customer* maka voucher tersebut akan masuk ke dalam *dashboard* stamps dan sebagai QA bisa melakukan pengecekan apakah sudah masuk ke *dashboard* stamps atau belum . Setiap *flow testing* yang contohnya seperti gambar diatas akan dilakukan di tahap SIT dan UAT agar bisa memberikan gambar kepada user dan tim internal bagaimana flow *journey* customer hingga tim *internal* ruparupa bisa mendapatkan data *customer*.

Juga dalam projek *cohesive*, mahasiswa sering melakukan *testing* terhadap UI dan UX dari aplikasi Ruparupa. Contohnya adalah seperti :



Gambar 3.14 Date Picker Desktop



Gambar 3.15. Date Picker Mobile

Jika dilihat dari gambar 3.14 menandakan bahwa mahasiswa sedang melakukan testing terhadap fitur *Date Picker* di desktop. Fitur tersebut diubah dari yang sebelumnya berbentuk kalender akan berubah menjadi bentuk dropdown seperti gambar diatas, mahasiswa melakukan testing terhadap fitur tersebut pada

setiap modul yang memiliki *Date Picker*. Juga selain itu mahasiswa melakukan testing di mobile apps dan mobile web yang dimana tampilan mobile memiliki tampilan yang berbeda dengan tampilan desktop. Tampilan mobile akan berbentuk seperti *roullete* hingga membuat *user* lebih mudah untuk menggunakannya. Sebelumya bentuk *Date Picker* di mobile juga berbentuk seperti kalender, namun setelah itu diganti ke berbentuk roullete yang dimana bisa dilihat dari gambar 3.15

Selain manual *testing* yang telah dijelaskan sebelumnya, pengujian integrasi sistem juga dapat dilakukan melalui *automated* testing, yaitu penggunaan perangkat lunak untuk melakukan pengujian secara otomatis, menggantikan *testing* manual yang dilakukan oleh manusia [6]. Proses ini melibatkan penggunaan skrip atau alat yang membandingkan hasil aktual pengujian dengan hasil yang diharapkan (*expected result*) tanpa memerlukan intervensi langsung dari *tester*. Implementasi *automated* testing dapat meningkatkan efisiensi waktu dalam pengujian, memungkinkan tim untuk melakukan lebih banyak pengujian dalam waktu yang lebih singkat, serta mengurangi kemungkinan kesalahan manusia (*human errors*) yang sering terjadi dalam pengujian manual. Dengan demikian, *automated* testing tidak hanya mempercepat proses pengujian, tetapi juga meningkatkan akurasi dan konsistensi hasil pengujian, mendukung kualitas perangkat lunak yang lebih baik.

Automated testing memiliki cakupan pengujian yang jauh lebih luas dibandingkan dengan manual *testing* [7]. Sementara manual *testing* dibatasi oleh sejumlah skenario yang ditentukan oleh logika dan keterbatasan sumber daya manusia, *automated testing* mampu menjangkau berbagai skenario pengujian secara lebih komprehensif. Selain itu, terdapat perbedaan signifikan lainnya antara *automated testing* dan manual *testing*. Manual dan *Automated* memiliki beberapa perbedaan sebagai berikut :

Perbedaan antara *automated testing* dan manual *testing* mencakup berbagai aspek yang dapat memengaruhi efektivitas pengujian sistem. Dari segi akurasi, *automated testing* terbukti lebih akurat untuk pengujian berulang dengan skenario yang sama. Meskipun demikian, akurasi ini bisa berkurang tergantung pada

pengendalian sistem yang dilakukan oleh manusia. Sementara itu, *manual testing* cenderung kurang akurat karena lebih rentan terhadap *human errors*, yang dapat terjadi akibat kelalaian atau kurangnya perhatian dari *tester*. Dalam hal biaya, *automated testing* sering kali membutuhkan investasi yang lebih besar, terutama untuk pengembangan dan pemeliharaan alat uji. Di sisi lain, manual testing lebih hemat biaya karena tidak memerlukan perangkat lunak atau infrastruktur tambahan, sehingga lebih mudah diakses oleh perusahaan dengan anggaran terbatas.

Ketika melihat reliabilitas, automated testing memiliki keunggulan dalam pengujian yang bersifat exploratory, memungkinkan tester untuk menjelajahi dan mengidentifikasi masalah yang mungkin terlewat dalam pengujian yang lebih terstruktur. Sebaliknya, manual *testing* lebih dapat diandalkan untuk pengujian berulang karena akurasinya yang lebih tinggi saat melakukan pengujian yang telah ditentukan sebelumnya. Dari segi *scope testing*, *automated testing* memiliki cakupan yang lebih besar dan kompleks, mampu menangani skenario yang memerlukan pengujian berulang. Namun, metode ini kurang efisien untuk skenario yang membutuhkan campur tangan manusia, di mana interaksi langsung dan penilaian manusia sangat penting. *Manual testing*, meskipun memiliki cakupan yang luas dan mampu menangani skenario yang memerlukan keterlibatan manusia, kurang efisien dalam pengujian yang sangat kompleks.

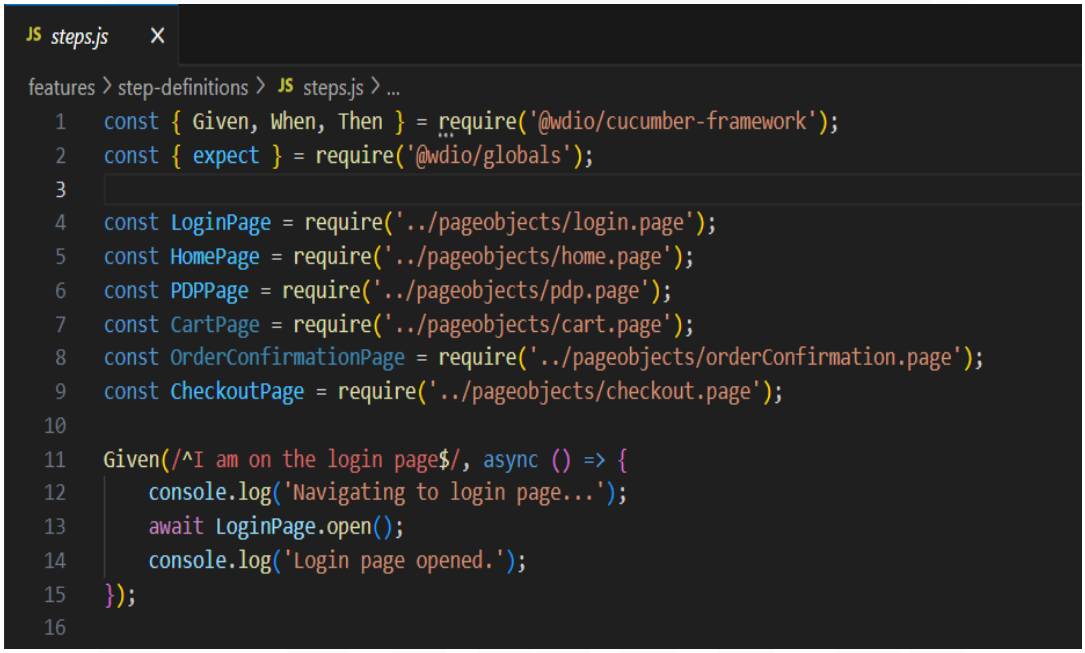
Dalam hal *scalability*, *automated testing* mungkin lebih *time-consuming*, tetapi dapat lebih efektif dalam melakukan pengecekan terhadap antarmuka pengguna (UI), memberikan hasil yang konsisten tanpa dipengaruhi oleh faktor manusia. Di sisi lain, manual testing cenderung lebih cepat dalam melaksanakan pengujian dasar, namun membutuhkan lebih banyak waktu untuk skenario yang kompleks. Akhirnya, terkait dengan keterampilan yang dibutuhkan, *automated testing* tidak memerlukan keahlian pemrograman yang mendalam, memungkinkan lebih banyak orang untuk terlibat dalam proses pengujian. Sebaliknya, *manual testing* sering kali memerlukan keterampilan pemrograman, terutama jika melibatkan skrip atau alat yang memerlukan pemahaman teknis.

Dengan demikian, pemilihan antara *automated testing* dan manual *testing* harus dilakukan dengan cermat, mempertimbangkan berbagai faktor seperti biaya, akurasi, cakupan, dan keterampilan yang dibutuhkan. Keputusan ini akan sangat memengaruhi efektivitas dan efisiensi pengujian sistem yang dijalankan.

Penjelasan di atas menguraikan perbedaan antara *automated testing* dan manual *testing*, serta menyoroti sisi positif dan negatif dari masing-masing metode. Dalam memilih metode yang paling sesuai, perlu dilakukan pertimbangan yang matang terkait sistem yang akan diuji. *Automated testing* menawarkan cakupan pengujian yang lebih luas, serta akurasi dan efektivitas yang lebih tinggi. Namun, metode ini sering kali memerlukan investasi biaya yang lebih besar dan kurang cocok untuk pengujian tampilan atau antarmuka pengguna (UI). Di sisi lain, manual testing tidak memerlukan biaya yang signifikan dan sangat sesuai untuk pengecekan UI serta skenario pengujian yang membutuhkan keterlibatan langsung dari manusia. Namun, metode ini lebih memakan waktu dan memiliki risiko human errors yang lebih besar. Oleh karena itu, pemilihan metode yang tepat harus disesuaikan dengan kebutuhan dan karakteristik proyek pengujian yang sedang dijalankan.

Salah satu alat yang dapat digunakan untuk melaksanakan *automated testing* adalah WebdriverIO. WebdriverIO merupakan sebuah framework pengujian yang berbasis Node.js, yang dirancang untuk memfasilitasi pengujian aplikasi web secara efisien. Keunggulan utama dari WebdriverIO adalah kemampuannya untuk berintegrasi dengan berbagai alat lain, seperti Jenkins untuk *continuous integration* dan berbagai tools pengujian berbasis cloud. Selain itu, WebdriverIO menawarkan fitur debugging yang memungkinkan pengembang untuk mendeteksi dan memperbaiki kesalahan dengan lebih mudah, serta kemampuan untuk mengambil screenshot secara otomatis selama proses pengujian. Proses pengujian dapat dilakukan dengan menulis skrip menggunakan bahasa pemrograman JavaScript, sehingga memudahkan para pengembang yang sudah familiar dengan JavaScript untuk beradaptasi dan memanfaatkan WebdriverIO dalam proyek mereka [9].

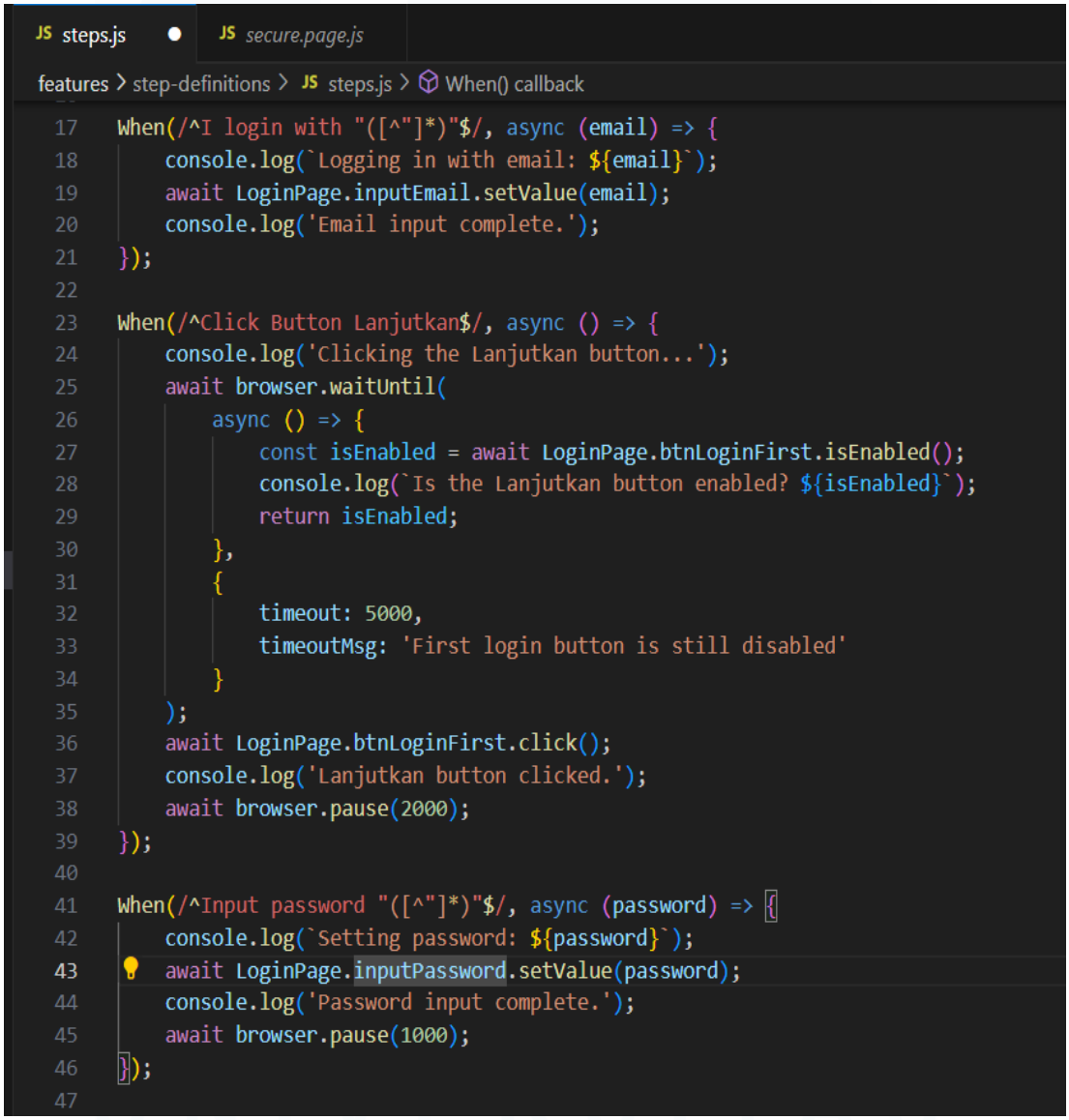
Berikut ini adalah codingan yang digunakan untuk *melakukan automated testing*, mulai dari proses login hingga checkout pada Ruparupa. coding ini ditulis dalam file step.js dengan menggunakan WebdriverIO dan Cucumber sebagai framework untuk menyusun skenario pengujian. Dengan pendekatan ini, pengujian dapat dilakukan secara terstruktur dan terautomasi, sehingga memudahkan pengujian berulang dan meningkatkan efisiensi dalam proses pengujian. Skrip ini mencakup langkah-langkah yang diperlukan untuk mengautentikasi pengguna, menambahkan produk ke keranjang, dan menyelesaikan proses checkout, memastikan bahwa semua fungsi berjalan sesuai harapan dan memenuhi kriteria kualitas yang ditetapkan



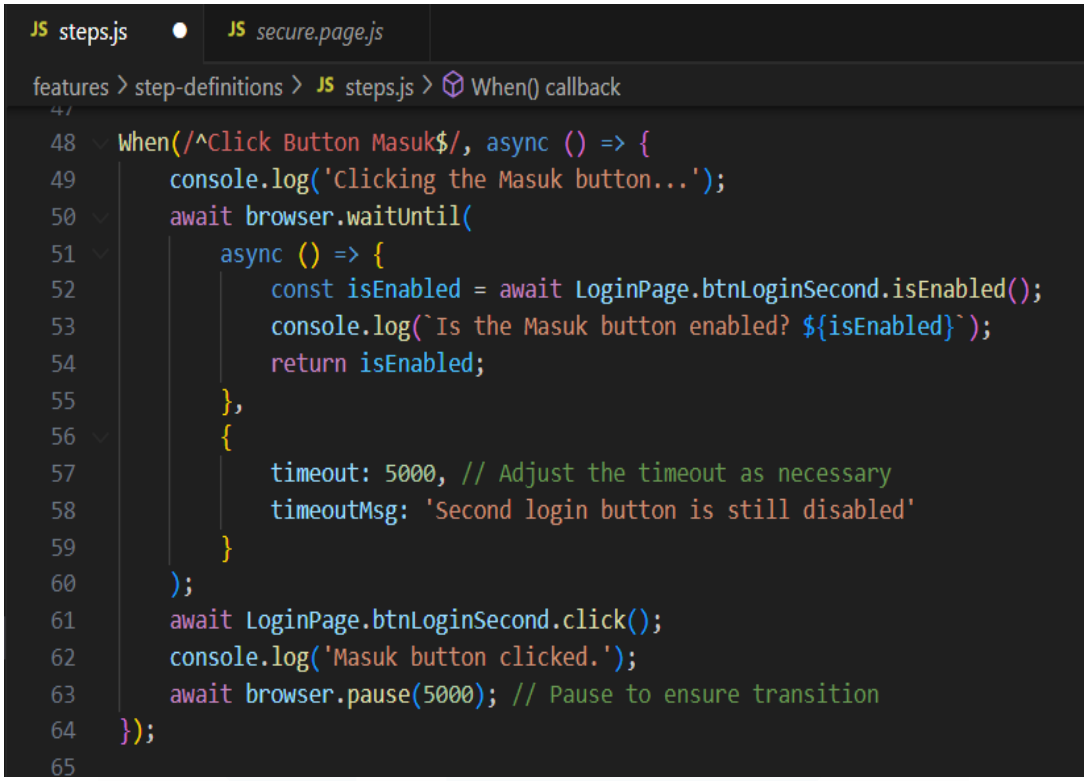
Gambar 3.16 Automation

Given adalah langkah awal dalam skenario pengujian yang menetapkan kondisi atau konteks sebelum tindakan pengujian dilakukan. Dalam codingan di atas, tujuan dari langkah `*Given*` adalah untuk mengarahkan pengguna ke halaman login dari situs web yang sedang diuji. Halaman login tersebut didefinisikan dalam codingan dengan merujuk pada objek `*LoginPage*`, yang diatur dalam file terpisah bernama `*loginpage.js*`. Dengan cara ini, langkah `*Given*` memberikan struktur yang jelas dan memudahkan pemeliharaan, karena semua interaksi dan elemen

yang terkait dengan halaman login terpusat dalam satu file. Pendekatan ini menjadikan proses pengujian lebih efisien dan mudah dipahami, karena setiap langkah dalam skrip terhubung dengan definisi yang lebih rinci di dalam file yang relevan, sehingga memungkinkan pengembang untuk memperbarui elemen halaman login tanpa mempengaruhi keseluruhan skrip pengujian.



Gambar 3.17 Automation

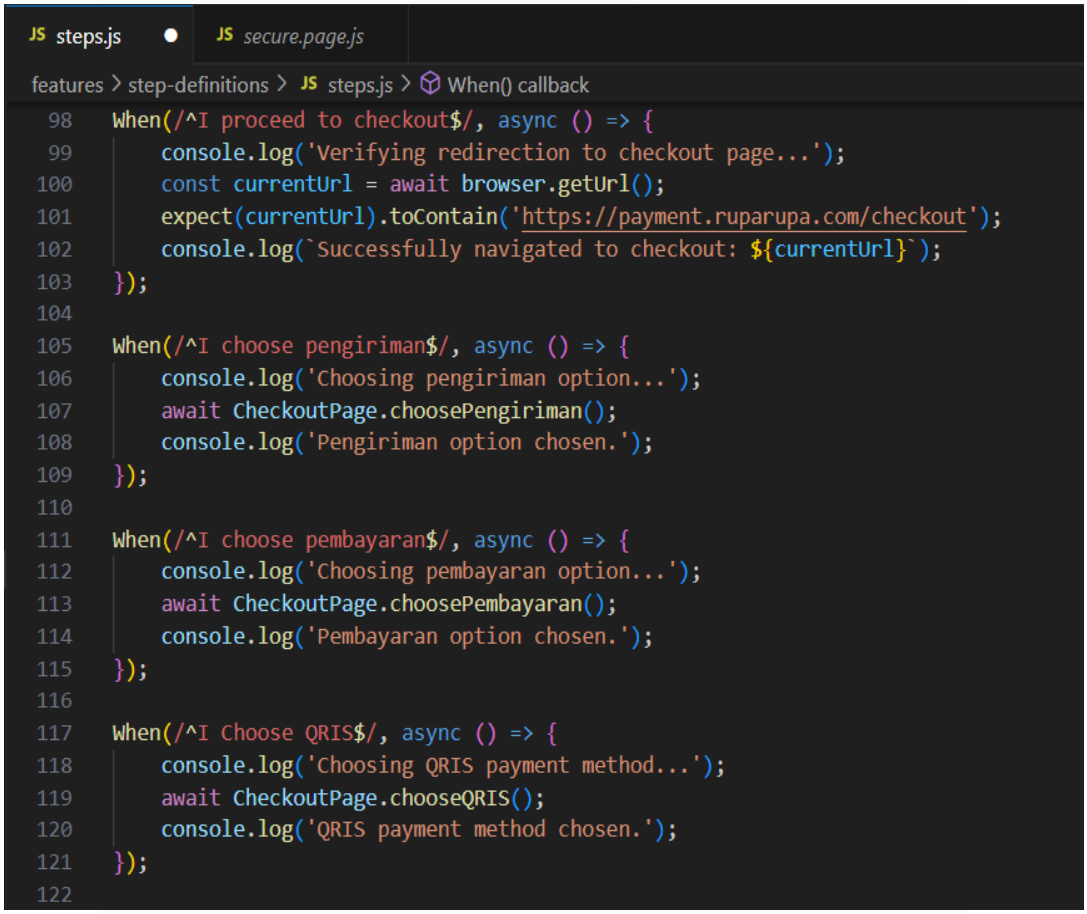


Gambar 3.18 Automation

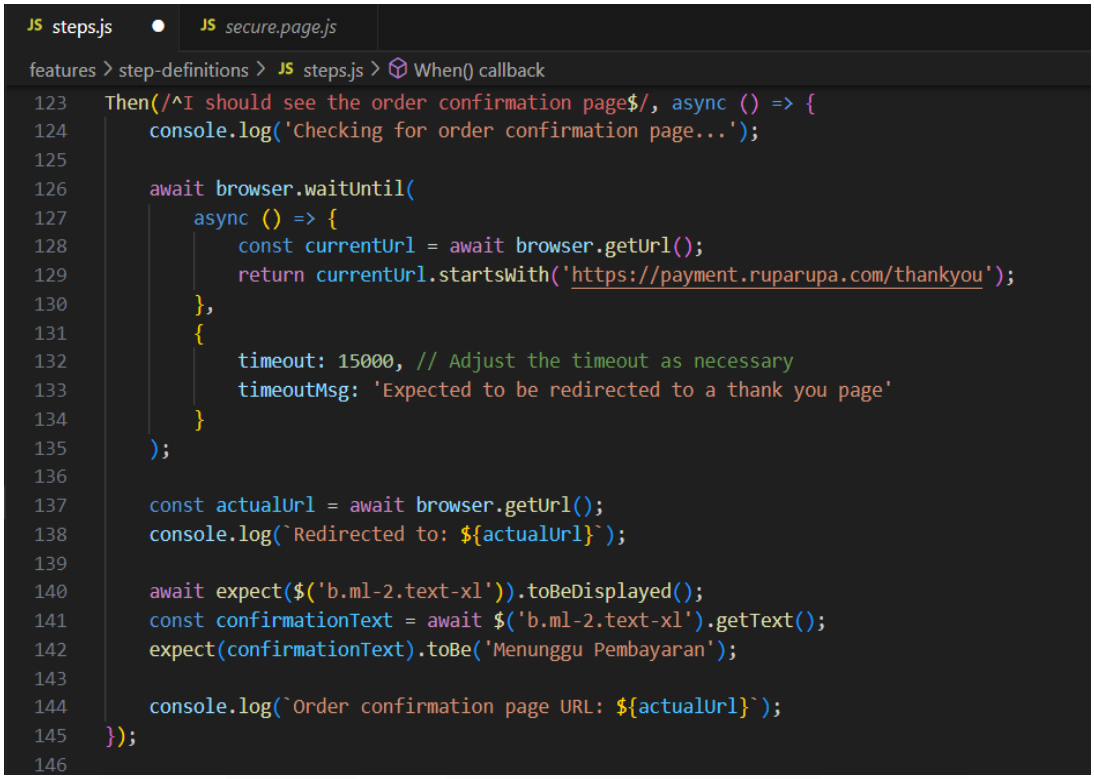
Codingan di gambar 3.18 melanjutkan proses pengujian dengan mendefinisikan bahwa di halaman login terdapat kolom untuk memasukkan alamat email yang sudah terdaftar. Nilai untuk kolom input email tersebut diambil dari file `*order.features*`. Setelah pengguna berhasil memasukkan alamat email, langkah berikutnya adalah mengklik tombol "*Lanjutkan*" pada halaman login. Setelah proses transisi halaman selesai, pengguna kemudian diminta untuk memasukkan kata sandi. Nilai untuk kolom input password juga didefinisikan dalam file `*order.features*`. Selanjutnya, tahap pengujian berlanjut dengan menunggu agar tombol "Masuk" menjadi aktif, sebelum akhirnya mengklik tombol tersebut untuk menyelesaikan proses login. Pendekatan ini memastikan bahwa setiap langkah dalam skenario pengujian terukur dan terdefinisi dengan jelas, sehingga memberikan kejelasan dan konsistensi dalam pengujian fungsional pada halaman login.

Gambar 3.19 Automation

Pengujian berikutnya pada gambar 3.19 dimulai setelah pengguna berhasil login, yang akan mengarahkan halaman ke halaman yang diharapkan, yaitu HomePage. Dalam coding, halaman beranda ini didefinisikan sebagai *HomePage*, yang merujuk pada `*homepage.js*` yang telah ditentukan sebelumnya. Setelah itu, pada halaman beranda, pengguna akan melakukan pencarian produk. Begitu produk ditemukan, pengguna akan mengkliknya, dan di halaman produk tersebut, pengguna akan mengklik tombol "*Beli Sekarang*." Pendekatan ini memastikan bahwa alur pengujian berlangsung sesuai dengan ekspektasi, menciptakan pengalaman pengguna yang mulus dan efisien.

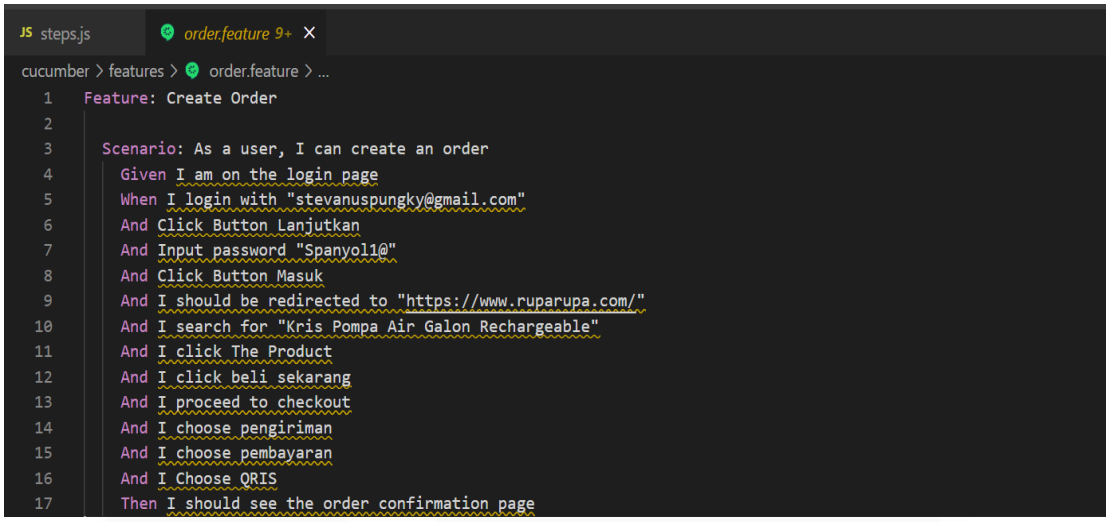


Gambar 3.20 Automation



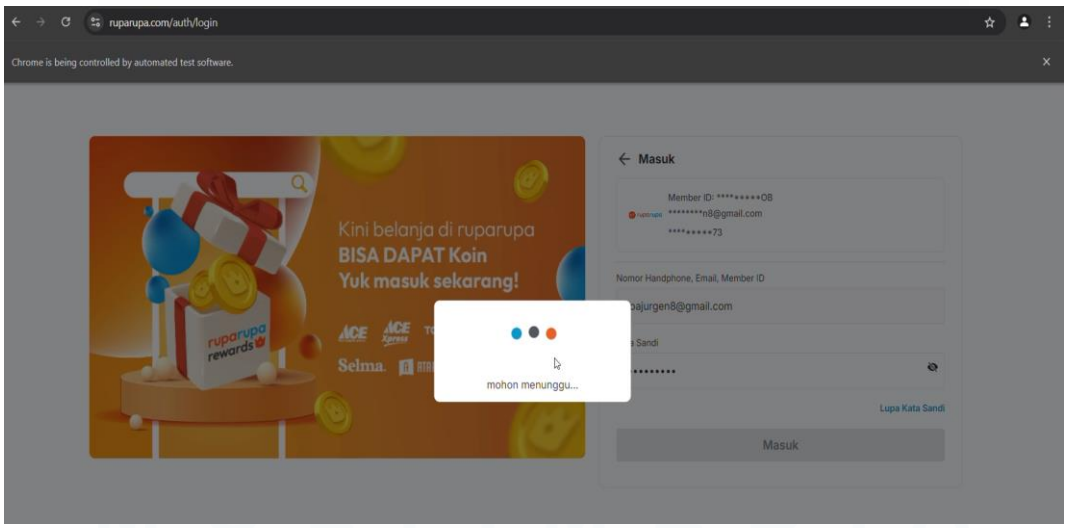
Gambar 3.21 Automation

Coding di pada gambar 3.21 melanjutkan proses pengujian dari langkah sebelumnya. Setelah pengguna mengklik tombol "*Beli Sekarang*," sistem akan mengarahkan mereka ke halaman checkout. Di halaman checkout, pengguna akan memilih metode pengiriman dan metode pembayaran. Metode pembayaran yang dipilih dalam pengujian ini adalah QRIS. Setelah metode pembayaran dipilih, pesanan akan dibuat dan pengguna akan dialihkan ke halaman Konfirmasi Pesanan. Halaman yang diharapkan untuk dituju setelah itu adalah halaman ucapan terima kasih, sesuai dengan URL yang telah ditentukan dalam skrip. Pendekatan ini memastikan bahwa seluruh alur pembelian berjalan dengan lancar dan sesuai dengan ekspektasi pengguna.



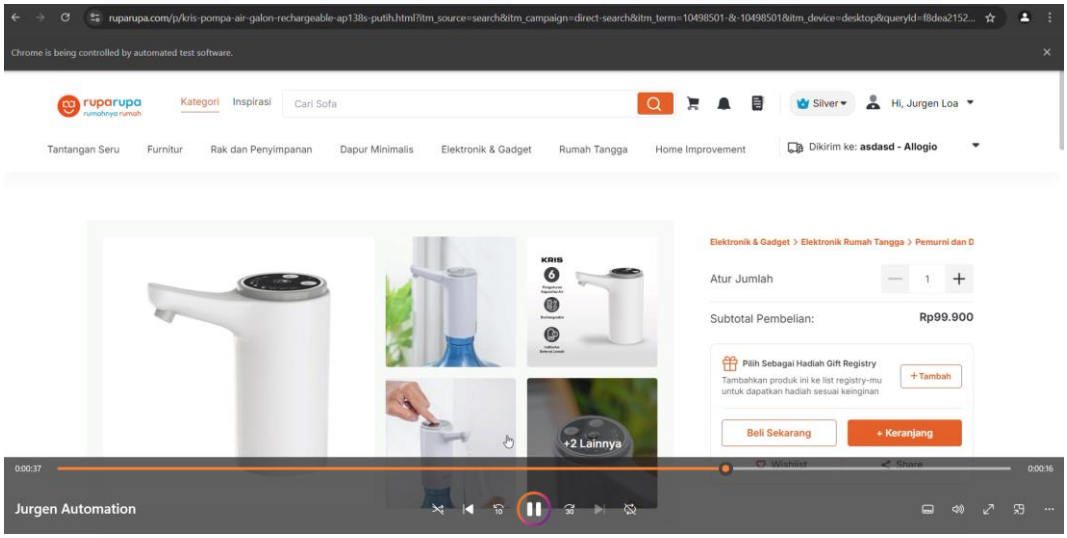
Gambar 3.22 Automation

Gambar 3.22 menunjukkan skrip yang memuat skenario pengujian yang akan dijalankan. Input seperti email, *password*, dan produk yang ingin dicari terdapat dalam skrip skenario pengujian tersebut. Jika diperlukan pengujian berulang pada fitur yang sama, skenario pengujiannya juga dapat didefinisikan dalam file ini. Setelah semua file disimpan dan dipastikan tidak ada kesalahan, skrip dapat dijalankan, dan proses pengujian website akan dilakukan secara otomatis. Juga *test case* tersebut diambil dari *test case* manual yang sudah dibuat.



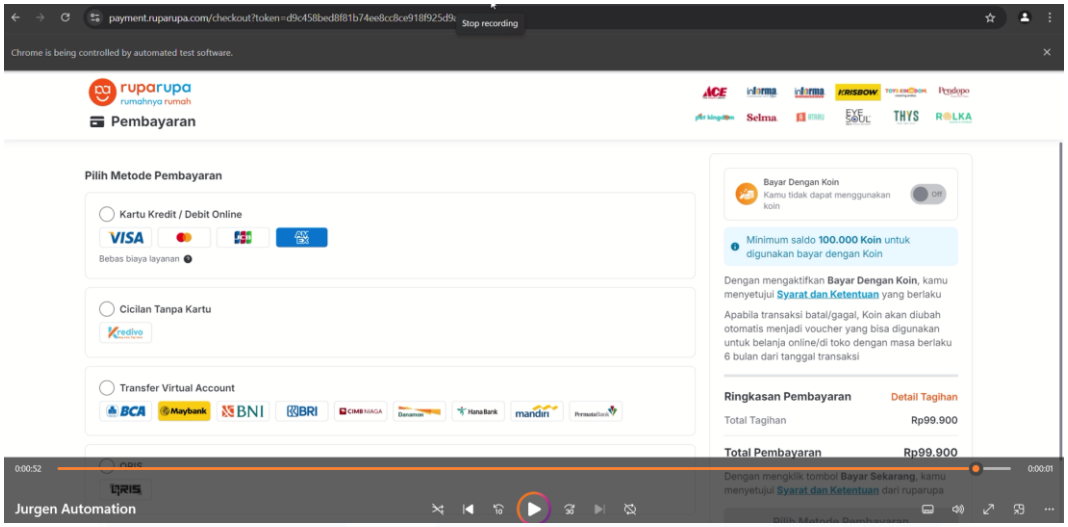
Gambar 3.23 Login Website Ruparupa

Gambar 3.23 menampilkan tampilan website Ruparupa ketika proses automated testing sedang berjalan. Pada tahap ini, setelah pengguna memasukkan password dan menekan tombol "Masuk," halaman yang ditunjukkan pada gambar tersebut muncul. Halaman ini merupakan tampilan saat sistem sedang memverifikasi password yang dimasukkan oleh pengguna. Setelah proses verifikasi selesai, sistem akan secara otomatis mengarahkan pengguna ke halaman utama atau home page. Proses ini menandakan bahwa login berhasil dan user diarahkan menuju tampilan utama untuk melanjutkan aktivitas selanjutnya.



Gambar 3.24 Tampilan PDP

Gambar 3.24 menunjukkan tampilan website Ruparupa saat proses automated testing sedang berlangsung. Gambar ini memperlihatkan halaman produk yang muncul setelah dilakukan pencarian produk di halaman home. Proses pencarian produk menggunakan input value yang sudah didefinisikan sebelumnya dalam file *order.feature.* Pada tahap ini, setelah produk ditemukan, halaman produk akan muncul, dan langkah selanjutnya adalah mengklik tombol "*Beli Sekarang*" untuk melanjutkan proses pembelian. Semua langkah tersebut dilakukan secara otomatis oleh skrip testing yang telah disiapkan sebelumnya.



Gambar 3.25 Tampilan halaman pembayaran

Gambar 3.25 menggambarkan tampilan website Ruparupa saat proses automated testing sedang berlangsung, khususnya pada tahap pemilihan metode pembayaran. Pada halaman ini, pengguna akan memilih metode pembayaran QRIS yang telah disiapkan dalam skenario pengujian. Setelah metode pembayaran dipilih, sistem akan secara otomatis membuat pesanan (*order*). Setelah itu, halaman akan dialihkan ke halaman Order Confirmation, dengan halaman tujuan akhir yang diharapkan adalah halaman Thank You, seperti yang telah ditentukan dalam URL pada skrip di file *steps.js.* Proses ini memastikan bahwa alur pembayaran berjalan dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan.

Dengan rating 4.8 dari 5 di Play Store dan lebih dari 31 ribu ulasan positif, tim QA di Ruparupa telah berhasil memastikan kualitas platform yang memuaskan bagi para pengguna. Upaya teliti dan terstruktur dalam proses *Quality Assurance* terbukti efektif dalam menjaga pengalaman pengguna yang optimal. Hasil ini mencerminkan kinerja tim QA yang telah bekerja keras untuk meminimalisir bug, meningkatkan fungsionalitas, dan menyediakan pengalaman yang sesuai dengan ekspektasi pelanggan, sehingga Ruparupa mendapatkan respons yang baik dan ulasan yang positif dari pengguna.

3.3 Kendala yang Ditemukan

Proses kerja magang yang dilakukan di PT. Omni Digitama Internusa atau ruparupa memiliki beberapa kendala umum, yaitu :

1. **Belum menguasai fitur-fitur, modul dan *flow* ruparupa**

   Saat pertama kali memulai magang di PT. Omni Digitama Internusa atau Ruparupa, salah satu kendala utama yang dihadapi oleh mahasiswa adalah ketidakpahaman mengenai fitur-fitur, modul-modul, serta alur atau *flow* aplikasi dan *website* Ruparupa. Sebelumnya, mahasiswa belum pernah menggunakan ataupun membuka aplikasi dan website Ruparupa, sehingga sama sekali belum familiar dengan bagaimana sistemnya bekerja. Ketika pertama kali diminta oleh *Lead* untuk mengeksplorasi platform Ruparupa, mahasiswa mengalami kebingungan yang cukup signifikan karena tidak mengetahui apa yang harus dilakukan atau fitur mana yang perlu dieksplorasi terlebih dahulu. Hal ini terjadi karena mahasiswa belum pernah menggunakan aplikasi Ruparupa sebelumnya.

   Contoh konkret dari kebingungan yang dialami di awal magang adalah ketika mahasiswa mencoba memahami fitur "*Log Out*" dan halaman "Ruparupa *Rewards*." Pada halaman "Ruparupa *Rewards*," terdapat banyak modul yang belum dipahami fungsinya, dan ini membuat proses eksplorasi menjadi lebih menantang. Dengan minimnya pengetahuan mengenai detail fitur-fitur tersebut, mahasiswa membutuhkan waktu tambahan untuk mempelajari dan memahami bagaimana setiap modul bekerja serta bagaimana masing-masing fitur mendukung pengalaman pengguna.

   Kendala ini juga berkaitan dengan customer journey dalam aplikasi, di mana mahasiswa awalnya tidak mengerti alur perjalanan pengguna, mulai dari masuk ke aplikasi hingga menyelesaikan transaksi atau aktivitas lainnya. Untuk mengatasi kebingungan ini, mahasiswa memerlukan waktu sekitar 1-2 minggu untuk benar-benar memahami lebih dalam mengenai fitur-fitur yang ada, termasuk modul-modul pada halaman rewards dan juga flow customer journey secara keseluruhan. Selama periode ini, mahasiswa melakukan eksplorasi secara mandiri dan berdiskusi dengan tim untuk

memperdalam pemahaman mengenai fitur-fitur yang ada di Ruparupa. Pemahaman ini menjadi sangat penting dalam menjalankan tugas-tugas sebagai QA, terutama dalam memastikan bahwa fitur dan modul yang diuji dapat berfungsi dengan baik sesuai dengan kebutuhan dan ekspektasi pengguna.

2. Private Connection

Selama menjalani magang di PT. Omni Digitama Internusa (Ruparupa), salah satu kendala yang dihadapi oleh mahasiswa adalah masalah terkait *private connection* perusahaan. Sebagai perusahaan yang menangani berbagai data internal penting serta informasi pelanggan yang bersifat sangat sensitif, Ruparupa menerapkan kebijakan ketat mengenai penggunaan jaringan yang aman dan terhubung melalui private connection perusahaan. Pada awal masa magang, mahasiswa seringkali mengalami kesulitan dalam mengakses sistem internal Ruparupa karena adanya kendala teknis saat menyambungkan laptop ke VPN perusahaan.

Proses bekerja di Ruparupa juga melibatkan sistem kerja *hybrid*, di mana karyawan dapat bekerja dari kantor (WFO) maupun dari rumah (WFH). Ketika bekerja dari rumah, mahasiswa diharuskan tetap menggunakan VPN perusahaan untuk memastikan bahwa semua aktivitas yang dilakukan berada dalam lingkungan jaringan yang aman. Namun, pada minggu-minggu pertama masa magang, mahasiswa mengalami kendala dalam menyambungkan koneksi VPN kantor, yang menyebabkan beberapa tugas tertunda dan proses kerja menjadi kurang efisien.

Kendala ini berdampak pada kemampuan mahasiswa untuk mengakses data internal yang diperlukan dalam pekerjaan harian serta memperlambat komunikasi dengan tim secara online. Seperti pada *testing* dan *update test tracking*, akses ke jaringan internal perusahaan sangat krusial. Mahasiswa harus mengatasi kendala ini terlebih dahulu sebelum dapat melanjutkan tugas-tugas yang diberikan. Hambatan teknis ini memakan cukup banyak waktu dan menjadi salah satu faktor yang

memperlambat produktivitas pada awal masa magang, terutama dalam menjalankan tugas-tugas QA yang memerlukan akses ke sistem perusahaan secara penuh dan real-time.

3. Ketidaktahuan Mengenai Tiket dan Pemahaman Alur Proyek

Saat menjalani tugas sebagai QA di PT. Omni Digitama Internusa (Ruparupa), salah satu kendala yang cukup sering terjadi adalah kurangnya pemantauan terhadap pembaruan bug setelah dilakukan perbaikan oleh tim developer. Ketika saya melakukan testing pada fitur-fitur seperti login atau perbaikan tampilan kalender, beberapa *bug* sering kali muncul, dan sesuai prosedur, saya melaporkan bug tersebut kepada tim *developer* melalui sistem Jira. Setelah melaporkan *bug*, tim *developer* akan melakukan investigasi dan perbaikan terhadap masalah tersebut.

Namun, setelah tim developer memperbaiki bug dan mengupdate status tiket di Jira menjadi *'Ready to Test'*, sering kali saya tidak segera memantau pembaruan ini. Kesalahan saya yang cukup berulang adalah tidak melihat jika tiket tersebut telah di-update oleh developer, sehingga bug yang sudah diperbaiki seharusnya dapat segera diujicoba ulang oleh saya sebagai QA. Akibatnya, progres pengujian menjadi semakin lambat karena tiket yang sebenarnya sudah siap diuji kembali masih menunggu di *backlog* tanpa saya sadari. Hal ini tentunya berdampak pada keseluruhan alur proyek yang menjadi tertunda, sebab pengujian yang seharusnya sudah selesai tidak langsung dilanjutkan.

Selain masalah pemantauan tiket, saya juga menghadapi tantangan dalam memahami alur proyek secara menyeluruh. Setiap proyek memiliki *flow testing* dan alur pengguna yang harus diikuti dalam melakukan pengujian, namun terkadang saya merasa kesulitan untuk memahami alur tersebut. Hal ini disebabkan oleh dua faktor utama. Pertama, dokumentasi di PRD (*Product Requirement Document*) yang tersedia kurang lengkap dalam menjelaskan detail alur proyek, sehingga saya sebagai QA tidak

memiliki gambaran yang jelas mengenai bagaimana fitur-fitur tersebut seharusnya berfungsi dalam konteks keseluruhan sistem. Ketika informasi di PRD tidak mencakup aspek-aspek penting atau kurang memberikan penjelasan mendalam tentang *flow testing* yang diharapkan, saya mengalami kebingungan dalam menjalankan tugas pengujian.

Kedua, ada kalanya saya merasa kurang percaya diri untuk bertanya kepada anggota tim lain atau senior mengenai alur proyek yang kurang saya pahami. Meskipun saya sadar bahwa bertanya akan membantu mempercepat pemahaman saya, saya sering merasa malu atau ragu untuk bertanya terlalu banyak, terutama jika saya merasa bahwa pertanyaan saya mungkin sudah jelas bagi orang lain. Akibatnya, saya lebih sering mencoba untuk memahami sendiri, namun ini sering kali memakan waktu lebih lama dan menyebabkan progres pekerjaan menjadi lebih lambat dari yang seharusnya.

Secara keseluruhan, tantangan ini menunjukkan bahwa pemantauan pembaruan bug dan pemahaman yang menyeluruh tentang alur proyek, terutama yang tidak tertulis dengan baik di PRD, adalah area di mana saya perlu meningkatkan keterampilan dan keberanian untuk bertanya. Memperbaiki kendala ini adalah langkah penting dalam meningkatkan efektivitas saya sebagai QA, khususnya dalam memastikan bahwa fitur-fitur diuji dengan tepat waktu dan sesuai dengan ekspektasi pengguna.

4. Automation Testing

Kendala utama saya dalam menjalankan automation testing di Ruparupa dengan WebdriverIO dan Cucumber adalah kurangnya pengalaman sebelumnya di bidang ini, sehingga saya harus memulai dari awal untuk memahami alat dan proses *automation testing*. Tanpa dasar yang kuat, saya menghadapi tantangan dalam menyusun skenario uji yang benar dan efisien, serta memahami prinsip-prinsip dasar seperti BDD (*Behavior-*

Driven Development) yang diterapkan di Cucumber. Hal ini membuat proses belajar terasa lambat, terutama dalam penyesuaian diri terhadap struktur skrip automation yang lebih kompleks.

3.4 Solusi atas Kendala yang Ditemukan

Berdasarkan kendala-kendala yang dijelaskan pada bagian sebelumnya, saya sebagai mahasiswa magang bertanggung jawab untuk mencari solusi dan mengatasi setiap masalah yang muncul selama menjalankan peran sebagai *Quality Assurance* di PT. Omni Digitama Internusa atau Ruparupa. Berikut adalah solusi-solusi yang diambil dalam mengatasi kendala yang dihadapi:

1. **Memperdalam Pemahaman mengenai fitur dan modul Ruparupa**

   Salah satu kendala terbesar yang saya hadapi adalah kurangnya pemahaman terhadap fitur, modul, dan alur kerja aplikasi Ruparupa pada awal masa magang. Sebagai solusi, saya menerapkan strategi eksplorasi mandiri secara intensif. Saya mengambil inisiatif untuk memanfaatkan waktu di luar jam kerja untuk mempelajari setiap elemen yang ada di dalam aplikasi dan website Ruparupa.

   Langkah pertama yang saya lakukan adalah mengidentifikasi fitur-fitur utama seperti halaman "Ruparupa *Rewards*" dan modul *logout*, yang sempat membingungkan di awal. Dengan membedah setiap modul yang ada di halaman rewards, saya mulai membangun pemahaman yang lebih baik mengenai bagaimana modul tersebut berfungsi, tujuan penggunaannya, dan bagaimana fitur-fitur ini berinteraksi satu sama lain dalam konteks customer journey. Saya juga melakukan konsultasi secara rutin dengan senior QA dan tim lainnya untuk mendapatkan panduan mengenai alur *customer journey* secara keseluruhan, mulai dari saat pengguna pertama kali login hingga

menyelesaikan transaksi. Diskusi ini membantu mempercepat proses pembelajaran saya. Selain itu, saya memanfaatkan dokumentasi yang ada dan materi yang diberikan oleh tim untuk mendapatkan informasi mendalam mengenai fitur dan modul di aplikasi. Dalam waktu sekitar 1-2 minggu, saya berhasil menguasai alur dan fitur yang ada, yang memungkinkan saya untuk menjalankan tugas testing dengan lebih efektif.

2. Memperbaiki Koneksi VPN

Masalah teknis terkait *private connection* perusahaan yang sering saya hadapi pada awal masa magang, terutama ketika bekerja dari rumah (WFH), mempengaruhi produktivitas saya dalam melakukan *testing* dan *update tracking*. Untuk mengatasi masalah ini, saya segera berkoordinasi dengan IT *Support* perusahaan untuk menemukan solusi.

Saya melakukan beberapa percobaan untuk menyambungkan laptop saya ke jaringan VPN perusahaan, dan setelah beberapa kali troubleshooting, IT *Support* berhasil membantu saya menemukan konfigurasi yang tepat untuk memastikan koneksi yang aman dan stabil. Dengan tersambungnya koneksi VPN, saya dapat mengakses sistem internal perusahaan, termasuk database dan framework yang sebelumnya tidak bisa diakses. Hal ini meningkatkan efisiensi kerja saya karena saya bisa kembali bekerja dengan akses penuh ke sistem yang diperlukan. Saya juga memastikan bahwa setiap kali ada masalah teknis yang muncul terkait VPN, saya langsung menghubungi IT *Support* agar masalah bisa segera diatasi tanpa menunda pekerjaan lebih lama lagi. Dalam proses ini, saya juga belajar mengenai protokol keamanan jaringan yang diterapkan di Ruparupa dan pentingnya penggunaan VPN dalam menjaga keamanan data perusahaan.

3. Kolaborasi dan Diskusi Aktif Dengan Tim

Sebagai mahasiswa magang, salah satu sumber solusi utama dalam mengatasi kendala adalah berkolaborasi dengan tim. Saya selalu terbuka dalam meminta bantuan dan berdiskusi dengan anggota tim, terutama ketika menghadapi kesulitan yang memerlukan pemahaman lebih dalam mengenai sistem dan framework yang digunakan di Ruparupa.

Diskusi ini tidak hanya membantu dalam menyelesaikan masalah teknis, tetapi juga memberikan saya wawasan baru mengenai proses QA yang lebih efisien dan cara kerja tim di Ruparupa. Dengan keterbukaan dan dukungan dari senior dan kolega lainnya, saya dapat lebih cepat mengatasi berbagai kendala yang dihadapi dan menyelesaikan tugas-tugas yang diberikan dengan baik. Untuk memastikan bahwa semua issue yang ditemukan selama testing dapat terselesaikan dengan baik, saya melakukan monitoring berkala melalui Jira. Setiap kali ada issue yang sudah selesai diperbaiki oleh *developer*, saya melakukan pengecekan ulang untuk memastikan bahwa masalah benar-benar terselesaikan sebelum mengubah status issue menjadi "*checked*" atau "*done*." Jika issue masih terjadi, saya segera melaporkannya kembali dan melampirkan bukti tambahan. Pendekatan ini memastikan bahwa setiap issue dapat diselesaikan secara tuntas dan sistem yang diuji dapat berfungsi sesuai dengan ekspektasi user. Monitoring melalui Jira juga membantu dalam mengelola pekerjaan dengan lebih baik karena setiap issue yang ada dapat dilacak secara terorganisir, dan setiap perbaikan dapat dicatat secara sistematis.

Dengan kombinasi dari eksplorasi mandiri, kolaborasi dengan tim, pemahaman yang lebih baik terhadap sistem, serta monitoring yang ketat, saya berhasil mengatasi kendala yang ditemukan selama masa magang. Ini memungkinkan saya untuk menjalankan peran sebagai QA dengan lebih efektif dan berkontribusi pada keberhasilan proyek-proyek di Ruparupa.

4. Solusi Automation Testing

Sebagai solusi, saya berinisiatif mempelajari konsep automation testing dengan mengikuti panduan resmi dan tutorial untuk WebdriverIO dan Cucumber. Saya mulai dengan skenario sederhana untuk membangun pemahaman, kemudian secara bertahap meningkatkan kompleksitasnya seiring berjalannya waktu. Selain itu, saya sering berdiskusi dengan tim atau komunitas automation untuk memperoleh tips, best practices, dan bantuan dalam mengatasi kendala teknis. Pendekatan ini tidak hanya memperkuat pemahaman saya tetapi juga mempercepat adaptasi terhadap tools yang digunakan, sehingga saya dapat lebih efektif dalam menjalankan tugas-tugas automation di Ruparupa.