

BAB III

PELAKSANAAN KERJA

3.1 Kedudukan dan Koordinasi

Kedudukan dan koordinasi menjelaskan posisi penempatan selama menjalani program magang serta alur koordinasi yang berlangsung dalam pelaksanaan tugas sebagai *Intern Quality Assurance* di PT. Global Loyalty Indonesia. Penjelasan mencakup struktur kedudukan di dalam organisasi, termasuk peran mentor dan keterlibatan dalam proyek, serta bagaimana proses komunikasi dan kolaborasi dilakukan bersama berbagai tim terkait. Selain itu, bagian ini juga menguraikan media komunikasi yang digunakan, pola kerja lintas divisi, hingga bentuk diskusi dan kegiatan internal yang mendukung kelancaran proses pengujian dan pengembangan produk.

3.1.1 Kedudukan

Selama mengikuti kegiatan *Career Acceleration Program Track 2* di PT. Global Loyalty Indonesia, posisi sebagai *Intern Quality Assurance* ditempati di bawah naungan Departemen Product Operations. Departemen ini dipimpin oleh Bapak Benny Chandra selaku penanggung jawab utama yang mengawasi seluruh aktivitas operasional produk. Setiap individu dalam program magang memperoleh mentor pendamping yang berbeda sesuai dengan proyek yang ditangani. Dalam pelaksanaan program ini, bimbingan diberikan oleh Kak Melisa Vania Daniela Samosir sebagai mentor utama. Melalui bimbingan tersebut, arahan langsung terkait proses kerja serta tanggung jawab sebagai *Intern Quality Assurance* diperoleh secara berkelanjutan.

Proyek yang dijalankan melibatkan *Intern Quality Assurance* secara aktif pada setiap tahap pelaksanaannya. Tanggung jawab yang dijalankan meliputi penyusunan skenario pengujian positif dan negatif, melakukan review serta analisis terhadap spesifikasi sistem, serta mengidentifikasi dan melaporkan *bug* maupun *error* yang ditemukan kepada tim developer.

Selain itu, *Intern Quality Assurance* juga berperan dalam menganalisis isu yang muncul selama proses pengujian (*troubleshooting*), menyusun dokumentasi sebagai bagian dari laporan QA, serta mendukung pelaksanaan *automation testing* untuk memastikan kualitas dan stabilitas aplikasi sesuai dengan standar yang telah ditetapkan perusahaan.

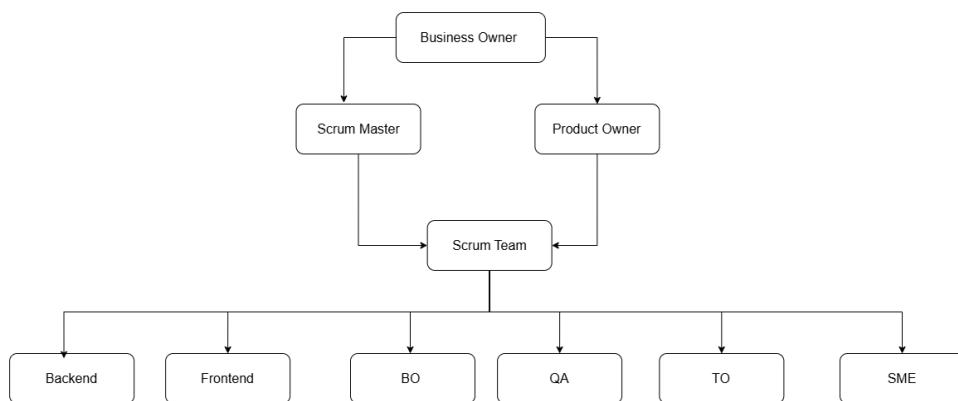
3.1.2 Koordinasi

Bagian ini menjelaskan alur koordinasi pekerjaan yang dilakukan selama menjalani magang di perusahaan. Penjelasan ini mencakup bagaimana proses komunikasi, kolaborasi, serta pembagian tanggung jawab berlangsung dengan berbagai tim terkait dalam satu *Scrum Team* dengan jumlah anggota sekitar 10 orang. Gambar 3.1 Bagan Alur Koordinasi menggambarkan alur koordinasi yang terjadi selama pelaksanaan tugas QA Intern di dalam tim. Bagan alur koordinasi tersebut merepresentasikan struktur kerja proyek yang menerapkan metodologi *Scrum*. Pada tingkat tertinggi, *Business Owner* berperan sebagai pemilik kebutuhan bisnis yang memberikan arahan strategis dan menetapkan prioritas utama. *Business Owner* berkoordinasi dengan *Product Owner* dan *Scrum Master* untuk memastikan kebutuhan bisnis dapat diterjemahkan dengan tepat ke dalam pengembangan produk. *Product Owner* bertanggung jawab mengelola backlog, menentukan prioritas fitur, serta menjelaskan detail *requirement* kepada tim selama *sprint*. Sementara itu, *Scrum Master* memastikan proses *Scrum* berjalan efektif dengan memfasilitasi kegiatan seperti *sprint planning*, daily standup, dan *sprint retro* serta membantu mengatasi hambatan yang dihadapi tim.

Dalam level pelaksana, *Scrum Team* bertindak sebagai inti pengembangan yang berkolaborasi untuk menyelesaikan item *sprint*. Tim ini kemudian berkoordinasi dengan berbagai peran teknis di bawahnya, yaitu *Backend*, *Frontend*, *Back Office*, *Quality Assurance (QA)*, *Technical Operation (TO)*, dan *Subject Matter Expert (SME)*. Setiap role memiliki kontribusi sesuai bidangnya, mulai dari pengembangan logic dan tampilan, pengelolaan operasional sistem, pengujian kualitas, dukungan teknis,

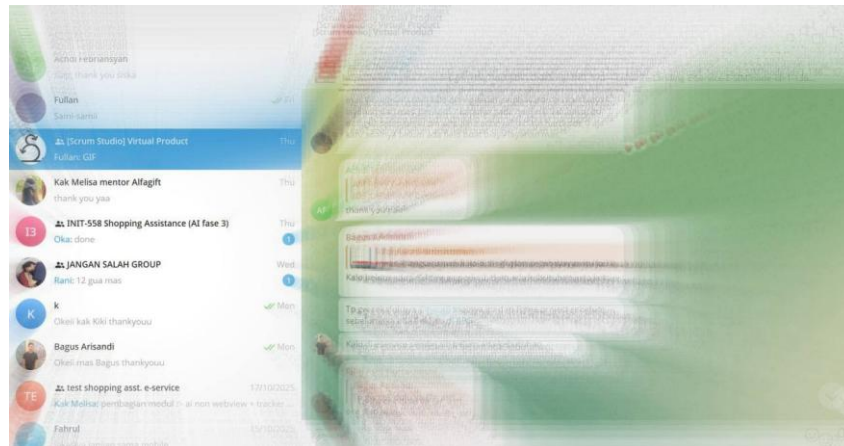
hingga pemberian pengetahuan mendalam terkait domain bisnis. Seluruh aktivitas koordinasi tersebut dilaksanakan dalam siklus sprint yang berlangsung selama dua minggu untuk setiap sprint. Selama satu sprint, koordinasi dilakukan secara intensif melalui *daily scrum* sebagai sarana pemantauan progres pekerjaan, identifikasi kendala, serta penyelarasan aktivitas pengembangan dan pengujian. Dengan pola koordinasi yang terstruktur dalam setiap sprint, proses pengembangan dan pengujian dapat berjalan secara iteratif dan berkelanjutan hingga menghasilkan increment produk yang sesuai dengan kebutuhan bisnis.

Bagan Alur Koordinasi



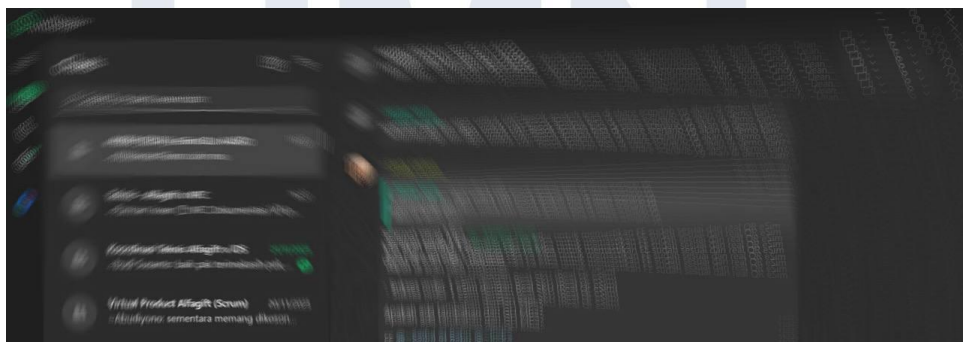
Gambar 3.1. Bagan Alur Koordinasi

Proses komunikasi selama magang juga dilakukan melalui media komunikasi *digital* untuk memastikan informasi dapat tersampaikan dengan cepat dan efektif. Salah satu media yang digunakan adalah Telegram, yang berfungsi sebagai wadah diskusi harian antara QA *Intern* dan anggota tim. Gambar 3.2 Komunikasi Via Telegram menampilkan contoh percakapan yang terjadi di dalam grup, mulai dari proses tanya jawab, penyampaian kebutuhan, klarifikasi fitur, hingga pembahasan detail teknis.



Gambar 3.2 Komunikasi Via Telegram

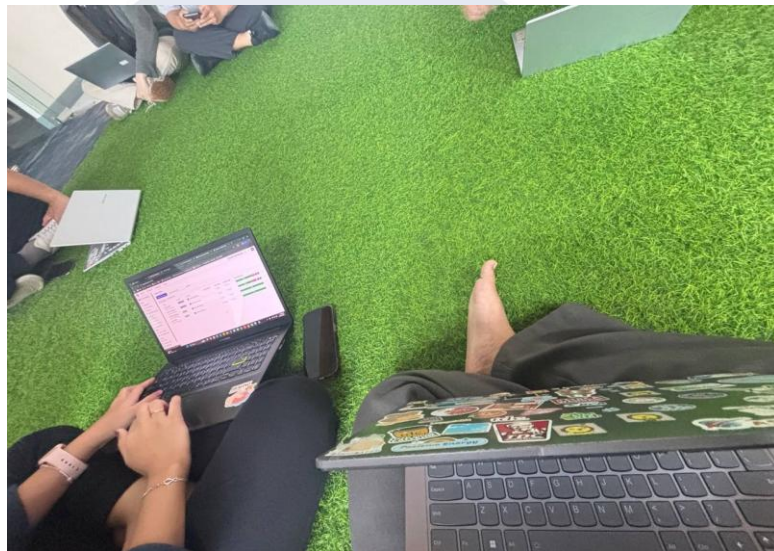
Melalui media ini, anggota tim dapat saling memberikan masukan, melakukan verifikasi kebutuhan data, serta memastikan bahwa setiap informasi yang diperlukan dalam proses pengembangan aplikasi tersampaikan dengan jelas. Pemanfaatan Telegram juga membantu mempercepat koordinasi, terutama pada situasi yang membutuhkan respons cepat tanpa menunggu jadwal meeting formal. Sehingga, komunikasi melalui Telegram berperan penting dalam menjaga kelancaran kolaborasi dan penyelesaian tugas selama masa magang.



Gambar 3.3 Komunikasi Via Whatsapp

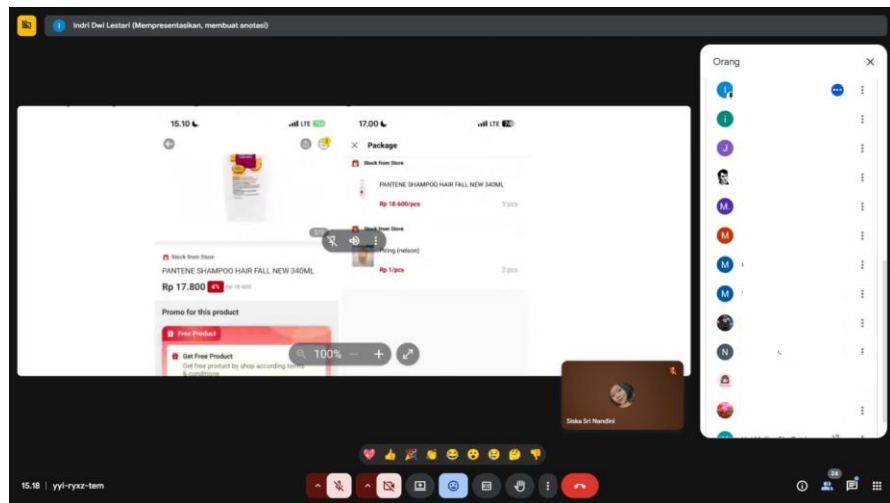
Gambar 3.3 Komunikasi Via Whatsapp menunjukkan bagaimana WhatsApp digunakan sebagai saluran komunikasi utama dalam proses koordinasi dengan pihak eksternal, khususnya vendor. Melalui *platform* ini, proses diskusi menjadi lebih cepat karena setiap pembaruan terkait progres pekerjaan, permintaan data tambahan, hingga penyampaian isu yang perlu

ditindaklanjuti dapat disampaikan secara langsung. Selain itu, penggunaan WhatsApp mempermudah penyimpanan percakapan sehingga setiap keputusan dan arahan yang diberikan dapat terdokumentasi dengan baik. Alur komunikasi yang responsif ini membantu mempercepat penyelesaian kendala serta memastikan seluruh pihak memiliki pemahaman yang selaras selama proses kerja berlangsung.



Gambar 3.4 Offline Team Discussion

Gambar 3.4 Offline Team Discussion memperlihatkan suasana diskusi tim yang dilakukan secara langsung di area kerja beralaskan rumput sintetis. Anggota tim *scrum* duduk melingkar sambil membuka laptop masing-masing untuk membahas progres *sprint* dan pekerjaan yang sedang berjalan. Dengan suasana yang lebih santai dan terbuka seperti ini, komunikasi terasa lebih cair, kolaborasi jadi lebih mudah, dan setiap anggota tim dapat menyampaikan pendapat atau update dengan lebih nyaman.



Gambar 3.5 Kegiatan sharing knowledge

Kegiatan sharing knowledge yang rutin diadakan oleh tim QA sebagai wadah untuk berbagi pemahaman terkait project yang telah mereka kerjakan dan sudah dirilis ke *production*. Gambar 3.5 Kegiatan *Sharing Knowledge* menunjukkan suasana presentasi *online*, di mana setiap QA menjelaskan alur fitur dari project mereka mulai dari tampilan pada *mobile*, proses yang berlangsung di database, hingga langkah kerja pada *Back Office (BO)*. Melalui kegiatan ini, seluruh anggota tim dapat memahami gambaran teknis project secara lebih menyeluruh dan memperoleh insight yang bermanfaat untuk meningkatkan kualitas pekerjaan pada project berikutnya.



Gambar 3.6 Glitz Assemble

Glitz Assemble adalah agenda rutin bulanan yang diselenggarakan oleh Global Loyalty Indonesia sebagai wadah untuk berbagi informasi penting, meningkatkan *awareness*, dan memperkuat pengetahuan seluruh Glitz mengenai berbagai isu di lingkungan kerja. Setiap bulan, Glitz Assemble hadir dengan tema yang beragam, mulai dari teknologi, keamanan data, literasi finansial, hingga pengembangan diri, yang disampaikan langsung oleh C-level perusahaan. Dalam sesi ini, para eksekutif memberikan motivasi, membahas pencapaian target sebelumnya, dan menyampaikan arah serta target perusahaan selanjutnya, sebagaimana ditunjukkan pada Gambar 3.6 Glitz Assemble. Selain itu, Glitz Assemble juga menjadi momen untuk memperkenalkan karyawan baru dan merayakan ulang tahun rekan-rekan Glitz. Seluruh sesi bersifat mandatory karena dirancang untuk memastikan bahwa seluruh Glitz memiliki pemahaman yang sama mengenai topik strategis yang mendukung kebutuhan perusahaan maupun kesejahteraan karyawan.

3.2 Tugas yang Dilakukan

Selama pelaksanaan program, berbagai kegiatan telah dijalankan yang mencakup proses onboarding, tugas harian, keterlibatan dalam beberapa proyek serta aktivitas pengujian dan koordinasi dengan tim. Seluruh kegiatan tersebut dirangkum untuk memberikan gambaran yang jelas mengenai tugas dan tanggung jawab yang telah diselesaikan selama program. Rincian lengkapnya disajikan pada Tabel 3.2 Detail Pekerjaan yang Dilakukan.

Tabel 3.2 Detail Pekerjaan yang Dilakukan

No.	Kegiatan	Mulai	Selesai
Onboarding New Plan for Joiner			
1.	<i>Install & Setup Database Tool</i>	4/8/2025	
2.	<i>Introduction to Qase and Jira</i>	4/8/2025	
3.	<i>Explore & go through Alfigift App</i>	4/8/2025	6/8/2025
Proyek APG Indodana PayLater			
1.	Melakukan <i>testing</i> dan <i>retesting</i> fitur pembayaran Indodana termasuk kombinasi A-Poin	7/8/2025	19/8/2025
2.	Melakukan <i>testing</i> pada service email notifikasi pembayaran		

No.	Kegiatan	Mulai	Selesai
3.	Mengidentifikasi dan melaporkan <i>bug</i>		
Full Cycle Rollout iOS Apps			
1.	Rollout Apps Version 4.42 – Basket, Tebus Murah Product, Address / Pick up, Button ">" / Lihat Promo, Pilih Metode Pemesanan, <i>Order Summary</i> , pilih waktu pengambilan, <i>Delivery Details</i> , APG, Price Change confirmation, Payment, TYP, Orders	12/8/2025	13/8/2025
2.	Rollout Apps Version 4.44 - Address/ <i>Pickup</i> (Basket), Button ">" / Lihat Promo, Pilih Metode Pemesanan, <i>Order Summary</i> , pilih waktu pengambilan, <i>Delivery Details</i> , Account & Membership Alfagift, [FO] Search Official Store, [FO] Search Global > Search Suggestion, Pengaturan Account, tambah alamat	4/11/2025	7/11/2025
Proyek <i>Virtual Product</i> (Microsite Postpaid dan Multifinance)			
1.	<i>Sprint Retro</i>	18/8/2025	
2.	<i>Refinement</i>	18/8/2025	
3.	<i>Sprint Planning</i>	20/8/2025	
4.	Membuat Skenario Multifinance - Validasi Form, <i>Inquiry</i> Tagihan, <i>Place Order</i> , <i>Post Checkout</i> , email	19/8/2025	26/8/2025
5.	<i>Testing</i> Skenario pada <i>mobile</i> untuk skenario postpaid dan multifinance (validasi form, <i>Inquiry</i> Tagihan, <i>place order</i> , <i>post checkout</i> , email)	27/8/2025	19/9/2025
6.	<i>Daily Scrum</i> Update	25/8/2025	19/9/2025
7.	Membuat <i>User Guide</i> Multifinance dan Postpaid	1/10/2025	2/10/2025
8.	<i>Sprint Review</i>	22/9/2025	
Proyek APG Auto Select Last Payment (V2)			
1.	<i>Testing</i> Skenario pada Database dan <i>BO (Back office)</i>	23/9/2025	26/9/2025
Proyek <i>Virtual Product</i> (E-SIM Domestic dan E-SIM Internasional)			
1.	<i>Sprint Retro</i>	22/9/2025	
2.	<i>Refinement</i>	23/9/2025	
3.	<i>Sprint Planning</i>	2/10/2025	
4.	Membuat Skenario <i>Landing Page</i> E-SIM Domestic dan Internasional, Validasi Form, <i>Inquiry</i> Tagihan, <i>Place Order</i> , <i>Oder summary</i> e-sim internasional, <i>Post Checkout</i> , email	3/10/2025	13/10/2025

No.	Kegiatan	Mulai	Selesai
5.	Testing Skenario pada <i>mobile</i> untuk skenario E-SIM Landing, <i>webview</i> untuk provider javamifi, Indosat, im3, dan telkomsel, <i>order summary</i> javamifi, <i>Place order</i> , <i>post checkout</i> , serta email	14/10/2025	
6.	Daily Scrum Update	6/10/2025	
7.	Sprint Review	5/11/2025	
8.	UAT E-SIM Internasional	14/11/2025	
Proyek Shopping Assistant e-service			
1.	Membuat Skenario untuk chat AI <i>Negative case</i> (Keyword Nomor Random, Keyword kalimat random, Irrelevant word > 3x, Respon di waktu service tidak aktif, Session expired)	14/10/2025	
2.	Membuat skenario chat AI groceries (Search Product Groceries, Error receive, Error sent)		
3.	Testing Skenario untuk <i>webview</i> , APG dan <i>order history</i>	11/11/2025	17/11/2025
Proyek Virtual Product (Omni CMV dan BPJS Denda)			
1.	Refinement	4/11/2025	
2.	Sprint Retro	6/11/2025	
3.	Sprint Planning	10/11/2025	
4.	Membuat skenario untuk Indosat CVM (input number, <i>webview</i> Indosat, <i>place order</i> , <i>post checkout</i> , email)	14/11/2025	20/11/2025
5.	Membuat skenario untuk BPJS denda bagian email	18/11/2025	
6.	Daily Scrum Update		
7.	Testing Skenario Indosat CVM		

3.3 Uraian Pelaksanaan Kerja

Secara umum, rangkaian pekerjaan yang dilakukan selama masa magang sebagai *Intern Quality Assurance* di PT Global Loyalty Indonesia (GLI) meliputi aktivitas utama seperti menyusun skenario pengujian positif dan negatif, melakukan review serta analisis spesifikasi sistem, melaporkan *bug* dan *error* kepada tim developer, menganalisis isu yang muncul selama proses troubleshoot, serta membuat dokumentasi pendukung untuk kebutuhan laporan QA. Selain pengujian manual, pekerjaan juga melibatkan penerapan *automation testing* pada fitur tertentu untuk memastikan proses validasi berjalan lebih efisien dan konsisten. Seluruh

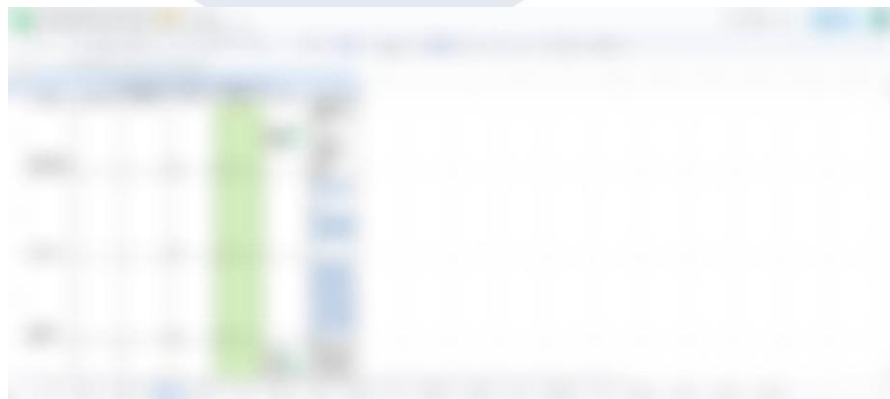
tugas tersebut dipaparkan secara runtut pada subbab berikutnya agar gambaran pelaksanaan pekerjaan selama masa magang dapat dipahami dengan jelas.

3.3.1 Proses Pelaksanaan

Seluruh aktivitas pengujian dilaksanakan berdasarkan proyek yang diberikan, meliputi *Onboarding New Plan for Joiner*, APG Indodana PayLater, Full Cycle Rollout iOS Apps, Microsite Postpaid dan Multifinance, *APG Auto Select Last Payment (V2)*, E-SIM Domestic dan Internasional, *Shopping Assistant e-Service*, serta Omni CMV dan BPJS Denda.

3.3.1.1 *Onboarding New Plan for Joiner*

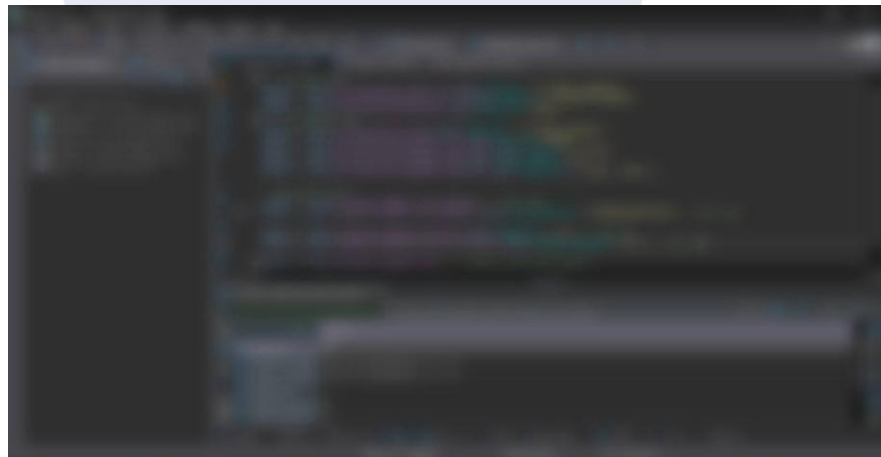
Pada hari pertama masa kerja magang di PT. GLI, pelatihan awal diberikan kepada anak magang sebagai pengantar sebelum menjalani peran sebagai *Intern Quality Assurance*. Materi yang harus dipelajari telah dirangkum dalam Google Sheets berjudul *Onboarding Plan for New Joiner*, sebagaimana ditampilkan pada Gambar 3.7



Gambar 3.7 Sheets *Onboarding Plan for New Joiner*

Dalam pelatihan *Onboarding Plan for New Joiner*, berbagai pengetahuan dan keterampilan dasar diberikan kepada anak magang untuk menunjang kelancaran pekerjaan selama masa magang. Salah satu fokus utama pelatihan ini adalah pengenalan sekaligus praktik instalasi serta konfigurasi *tools* yang akan digunakan sehari-hari dalam proses pengujian dan pengembangan aplikasi. Beberapa *tools* yang diperkenalkan antara lain DBeaver, MongoDB, Swagger, dan

Postman. Masing-masing *tools* memiliki kegunaan yang berbeda namun saling melengkapi. DBeaver berfungsi sebagai antarmuka grafis untuk mengakses dan mengelola berbagai jenis database, MongoDB sebagai database NoSQL memberikan fleksibilitas dalam pengelolaan data berbasis dokumen. Swagger digunakan untuk memahami, menguji, dan mendokumentasikan *endpoint* API. Postman dimanfaatkan sebagai alat utama untuk pengujian API secara manual, termasuk pengiriman request dan penerimaan response dari server. Melalui pelatihan ini, diharapkan tidak hanya memahami fungsi setiap *tools*, tetapi juga mampu menggunakannya secara terintegrasi dalam pekerjaan harian seperti debugging, pengembangan fitur, serta validasi alur data antara aplikasi dan server.



Gambar 3.8 Tampilan Software DBeaver

Gambar 3.8 menampilkan tampilan perangkat lunak DBeaver, sebuah universal database tool yang digunakan untuk mengakses serta mengelola beragam jenis database secara efisien. DBeaver menyediakan dukungan koneksi ke berbagai sistem manajemen basis data (DBMS) populer, seperti MySQL, PostgreSQL, SQLite, Oracle, SQL Server, dan lainnya, sehingga menjadikannya alat yang fleksibel dan sangat berguna dalam pengelolaan data pada berbagai *platform*. DBeaver umumnya dimanfaatkan oleh tim QA sebagai alat pendukung dalam melakukan operasi *CRUD* (*Create, Read, Update,*

Delete). Operasi tersebut dilakukan untuk menunjang aktivitas pengujian, seperti menambahkan test data, membaca informasi, atau memverifikasi data pesanan pada aplikasi Alfagift. Melalui DBeaver, tim QA dapat berinteraksi langsung dengan database secara visual dan terorganisir, tanpa perlu menggunakan command line, sehingga proses kerja menjadi lebih cepat, efisien, dan mudah dipantau.

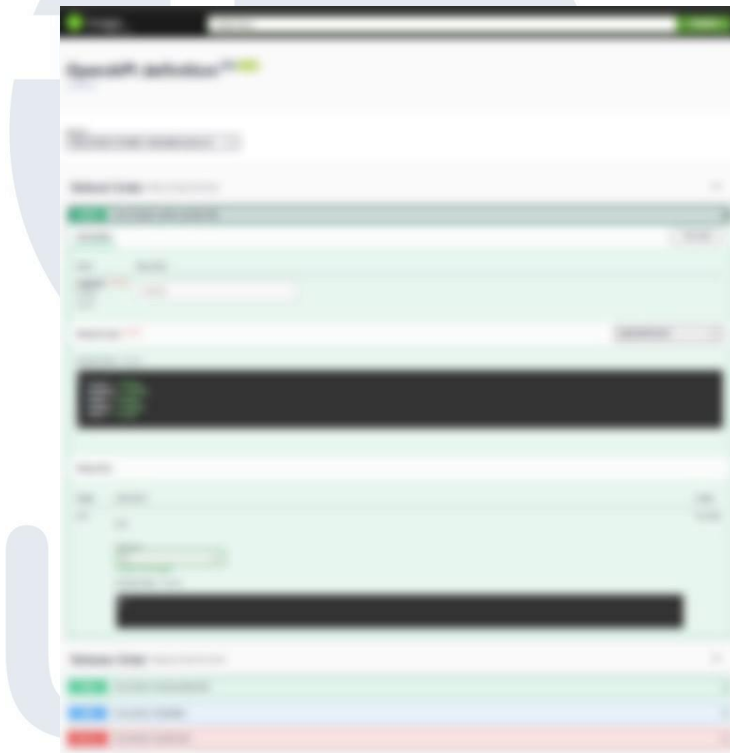


Gambar 3.9 Tampilan software MongoDB

Gambar 3.9 menampilkan antarmuka dari software MongoDB, yang berfungsi untuk mempermudah akses serta pengelolaan database NoSQL pada MongoDB. MongoDB merupakan sistem manajemen basis data berbasis dokumen yang menyimpan data dalam format JSON-like (BSON), sehingga lebih fleksibel dalam menangani data yang tidak terstruktur maupun semi-terstruktur. Melalui tampilan grafis ini, tim QA dapat melakukan eksplorasi data, menjalankan *query*, serta mengelola koleksi dan dokumen dengan cara yang lebih mudah dan intuitif dibandingkan jika menggunakan *Command Line Interface (CLI)*.

Pada proses pengujian aplikasi, khususnya pada aplikasi berbasis layanan *digital* seperti Alfagift, MongoDB sering digunakan oleh tim QA untuk melakukan verifikasi dan validasi terhadap data yang masuk ke dalam sistem. Melalui MongoDB, tim QA dapat

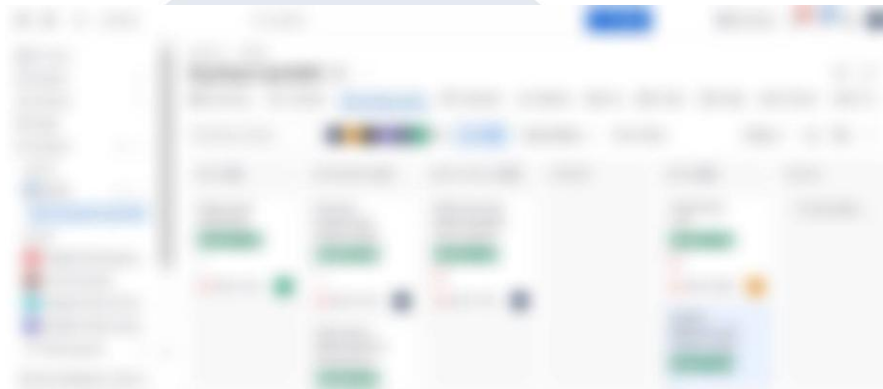
memastikan bahwa data yang diproses oleh *backend* sudah sesuai dengan ekspektasi, baik dari sisi struktur, isi, maupun integritas data. Selain itu, MongoDB juga memudahkan QA dalam melakukan pencarian data spesifik, filter berdasarkan parameter tertentu, hingga pengecekan logis terhadap alur bisnis yang sedang diuji. Penggunaan MongoDB melalui antarmuka GUI seperti ini dapat mempercepat proses pengujian karena pengguna tidak perlu menuliskan *query* yang kompleks secara manual.



Gambar 3.10 Tampilan software Swagger UI

Pada proses pengujian, tim QA juga menggunakan Swagger UI sebagai salah satu *tools* pendukung, terutama untuk pengujian API. Swagger UI pada gambar Gambar 3.10 merupakan antarmuka interaktif berbasis web yang memudahkan pengguna untuk melihat, mencoba, dan berinteraksi langsung dengan *endpoint* API sesuai dengan dokumentasi yang tersedia, sehingga tim QA dapat menguji fungsi *backend* secara lebih mendalam tanpa harus melalui tampilan antarmuka pengguna. Melalui Swagger UI, QA dapat mengirim

berbagai jenis request seperti GET, POST, PUT, dan DELETE ke *endpoint* tertentu dengan memasukkan parameter yang dibutuhkan, yang kemudian membantu tim memverifikasi apakah data yang dikirim maupun diterima sudah sesuai dengan ekspektasi sistem, serta mendeteksi apabila terdapat *error* atau ketidaksesuaian pada respons API.



Gambar 3.11 Tampilan software jira

Gambar 3.11 Tampilan Software Jira menunjukkan tampilan antarmuka Jira yang digunakan sebagai alat manajemen tugas dalam proses pengembangan aplikasi. Melalui *platform* ini, setiap tugas terkait pengujian seperti pembuatan ticket *bug*, pencatatan progres, serta pembaruan status pekerjaan dapat dilakukan secara terstruktur. Jira membantu tim dalam memantau alur kerja, mengatur prioritas, serta memastikan setiap issue ditindaklanjuti sesuai dengan tahapan yang telah ditentukan pada *sprint* yang sedang berjalan. Penggunaan Jira juga mempermudah koordinasi antar anggota tim karena seluruh informasi terkait tugas terdokumentasi dan dapat diakses secara *real-time*.

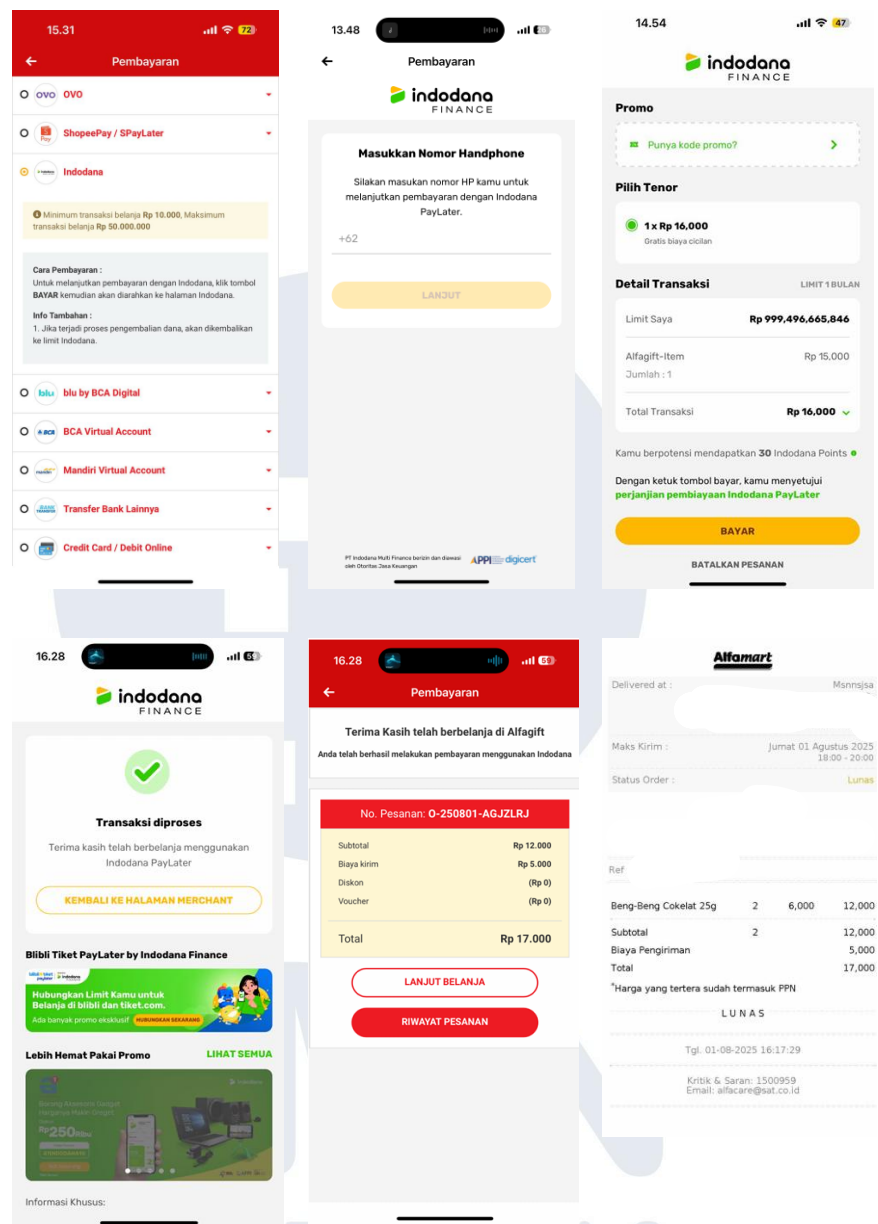


Gambar 3.12 Tampilan software qase

Gambar 3.12 Tampilan Software Qase menampilkan antarmuka Qase, yaitu *platform* yang digunakan untuk mengelola proses pengujian secara terstruktur. Melalui Qase, *test case* dapat dibuat, diorganisasi, dan dieksekusi sesuai kebutuhan pengujian setiap fitur. *Platform* ini mempermudah pencatatan hasil pengujian, seperti status passed, failed, atau blocked, sehingga tim QA dapat memantau kualitas aplikasi secara lebih sistematis. Selain itu, Qase juga memfasilitasi kolaborasi tim dengan menyediakan dokumentasi pengujian yang mudah diakses dan diperbarui, sehingga seluruh anggota tim dapat mengikuti perkembangan hasil pengujian secara aktual.

3.3.1.2 Proyek APG Indodana PayLater

Proyek APG Indodana PayLater merupakan pengembangan yang berfokus pada integrasi metode pembayaran PayLater dari Indodana ke dalam sistem aplikasi melalui *Application Payment Gateway (APG)*. Integrasi ini memungkinkan APG mengelola seluruh proses pembayaran, mulai dari pengiriman *request* transaksi ke Indodana, menerima respons hasil pembayaran, hingga meneruskan status tersebut ke aplikasi. Status transaksi kemudian ditampilkan pada halaman seperti *Thank You Page (TYP)*, *Order History*, *Order Detail*, dan *e-receipt*, sehingga pengguna dapat mengetahui apakah pembayaran berhasil, gagal, dibatalkan, atau masih menunggu verifikasi.



Gambar 3.13 ampilan *webview*

Gambar 3.13 Tampilan *Webview* Metode Pembayaran

Indodana PayLater memperlihatkan alur lengkap proses pembayaran melalui Indodana, dimulai dari pemilihan metode pembayaran di aplikasi hingga pengguna kembali ke halaman merchant setelah transaksi diproses. Pada gambar pertama ditampilkan tampilan pemilihan metode pembayaran Indodana pada aplikasi. Selanjutnya, pengguna diarahkan ke *webview* Indodana untuk memasukkan nomor

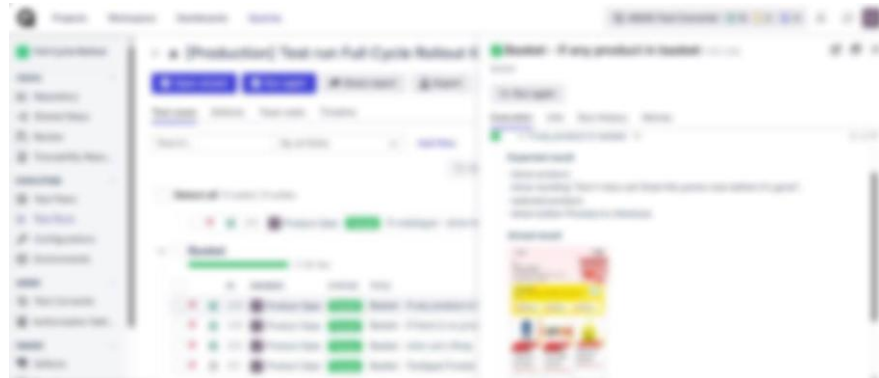
handphone, memilih tenor dan melihat detail transaksi, hingga halaman konfirmasi bahwa transaksi sedang diproses. Setelah itu, aplikasi menampilkan halaman *TYP* yang menunjukkan bahwa pembayaran telah berhasil. Proses ditutup dengan tampilan *e-receipt* yang memuat ringkasan pesanan dan status transaksi.

Dari sisi pengujian, proyek ini menuntut QA untuk melakukan verifikasi menyeluruh terhadap seluruh tahapan transaksi, mulai dari pemilihan metode pembayaran, proses redirect ke *webview* Indodana, input PIN oleh pengguna, hingga kembali ke aplikasi setelah transaksi selesai. QA juga menguji berbagai skenario untuk memastikan setiap status pembayaran muncul dengan tepat pada antarmuka pengguna, serta memastikan alur tampilannya konsisten di seluruh halaman yang berhubungan dengan transaksi. Selain itu, QA melakukan pengecekan bahwa setiap perubahan status pembayaran telah tercatat sesuai pada database

3.3.1.3 Proyek Full Cycle Rollout iOS Apps

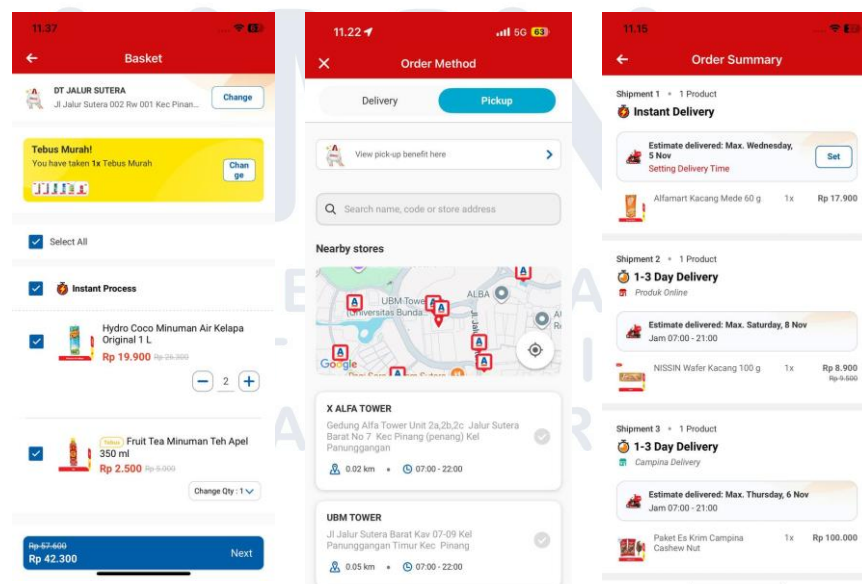
Proyek Full Cycle Rollout iOS Apps merupakan rangkaian aktivitas pengujian menyeluruh yang dilakukan untuk memastikan kualitas aplikasi iOS sebelum dirilis kepada pengguna. Dalam proyek ini, proses pengujian dilakukan secara *end-to-end* meliputi pengecekan fungsi utama aplikasi, validasi perilaku sistem pada berbagai kondisi, serta memastikan stabilitas dan konsistensi pengalaman pengguna. Beberapa fitur yang diuji dalam tahap ini antara lain pengaturan notifikasi untuk memastikan aplikasi dapat meminta izin notifikasi, menyalakan notifikasi, dan mendeteksi status perizinan dengan benar. Selain itu, proses logout juga diuji untuk memastikan pengguna dapat keluar dari aplikasi dengan lancar dan sistem membersihkan sesi pengguna sesuai prosedur. Pengujian turut mencakup fungsi Riwayat Pesanan, termasuk verifikasi tampilan

order dengan status tertentu, navigasi ke detail pesanan, dan kelanjutan proses pembayaran melalui tombol “Pilih Pembayaran”.



Gambar 3.14 Tampilan Testrun

Gambar 3.14 Tampilan Testrun menunjukkan kumpulan skenario pengujian yang dieksekusi oleh QA melalui *platform* testrun. Setiap *test case* menampilkan status eksekusi seperti passed atau failed, beserta langkah-langkah pengujian yang harus dipenuhi. Tampilan ini menjadi acuan utama bagi QA untuk memonitor progres pengujian, mendokumentasikan hasil, serta mengidentifikasi jika terdapat ketidaksesuaian pada fitur-fitur aplikasi.



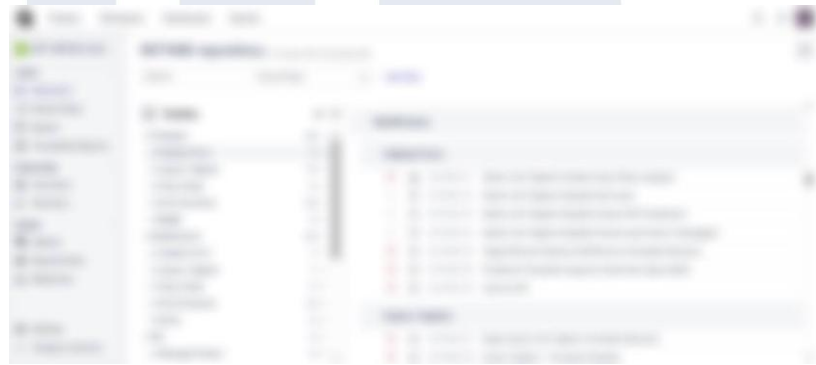
Gambar 3.15 Tampilan *Mobile* pada proses pengujian

Gambar 3.15 Tampilan *Mobile* pada Proses Pengujian menunjukkan beberapa antarmuka aplikasi iOS yang diuji selama pelaksanaan Full Cycle Rollout. Tampilan pertama menampilkan halaman Basket, tempat QA melakukan verifikasi terkait pemilihan produk, jumlah barang, validasi harga, serta pengecekan fungsi *select all* dan tombol *Next*. Tampilan berikutnya memperlihatkan halaman *Order Method* dalam mode *Pickup*, di mana QA memastikan pemilihan lokasi toko berjalan dengan baik, fitur pencarian toko berfungsi sesuai kebutuhan, marker pada peta tampil dengan benar, serta informasi jam operasional toko ditampilkan secara akurat. Tampilan terakhir menunjukkan halaman *Order Summary* yang berisi beberapa *shipment*. Pada tahap ini, QA memvalidasi estimasi waktu pengiriman, konsistensi harga, daftar produk dalam tiap *shipment*, serta memastikan tombol “Set” berfungsi untuk mengatur waktu pengantaran. Seluruh tampilan ini menjadi bagian penting dari proses pengujian untuk memastikan alur transaksi berjalan sesuai standar dan memberikan pengalaman pengguna yang optimal.

Seluruh skenario tersebut dieksekusi sebagai bagian dari siklus pengujian penuh (full cycle), di mana QA memastikan bahwa setiap fitur berjalan secara stabil, tidak menimbulkan *error*, dan sesuai dengan alur bisnis yang telah ditentukan. Pengujian ini juga membantu mengidentifikasi potensi *bug* sebelum aplikasi didistribusikan lebih luas pada tahap rollout. Dengan demikian, proyek Full Cycle Rollout iOS Apps memiliki peran penting dalam menjamin kualitas akhir aplikasi, sehingga dapat memberikan pengalaman pengguna yang optimal dan bebas kendala pada saat dirilis di *platform* iOS.

3.3.1.4 Proyek *Virtual Product* (Microsite Postpaid dan Multifinance)

Proyek Microsite Multifinance dan Postpaid merupakan pengembangan fitur pada aplikasi Alfagift, yang bertujuan untuk mempermudah pengguna dalam melakukan pembayaran tagihan secara *digital*. Melalui fitur ini, pengguna dapat dengan mudah mengakses layanan pembayaran cicilan (multifinance) maupun tagihan pascabayar (postpaid) langsung dari aplikasi, tanpa perlu berpindah ke *platform* lain. Fitur ini termasuk dalam bagian *virtual product*, yaitu layanan *digital* Alfagift yang dirancang untuk mendukung kebutuhan transaksi sehari-hari pengguna secara praktis dan terintegrasi.



Gambar 3.16 *Repository Skenario*

Pengujian fitur Multifinance dilakukan dengan menyusun rangkaian skenario yang mencakup seluruh proses transaksi, dimulai dari tahap validasi data hingga verifikasi notifikasi email setelah pembayaran. Struktur skenario tersebut dapat dilihat pada Gambar 3.16 *Repository Skenario*. Pada tahap Validasi Form, seluruh input yang dimasukkan oleh pengguna diuji untuk memastikan bahwa sistem mampu melakukan pengecekan kelengkapan dan kebenaran data. Pengujian dilakukan pada kondisi data yang lengkap, data kosong, format yang tidak sesuai, serta situasi ketika jaringan tidak stabil. Melalui proses ini, dipastikan bahwa tombol untuk melanjutkan proses hanya akan aktif apabila seluruh persyaratan input terpenuhi.

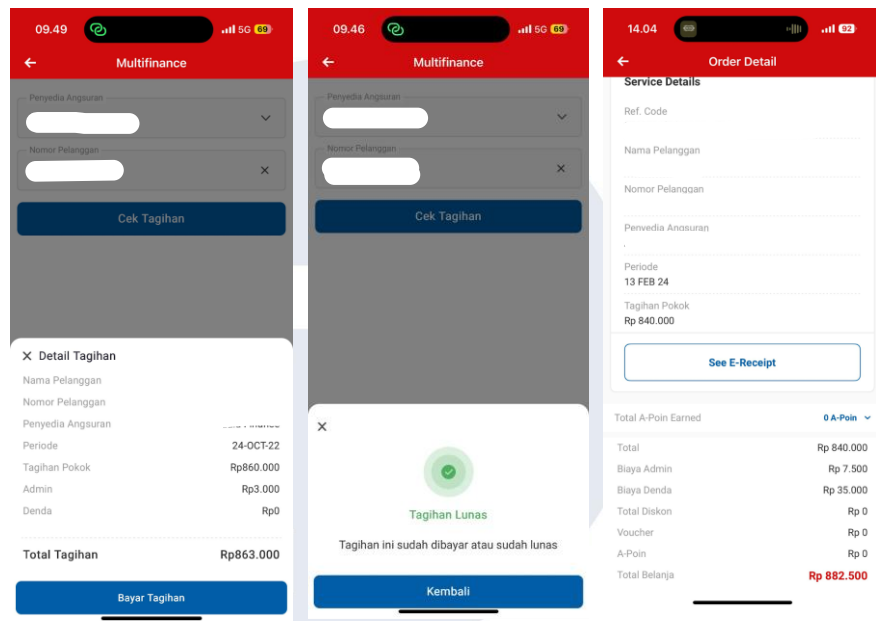
Setelah validasi selesai, pengujian dilanjutkan pada tahap *Inquiry* Tagihan untuk memastikan bahwa sistem dapat menampilkan informasi tagihan secara akurat. Berbagai kondisi diuji, termasuk *inquiry* yang berhasil, data pelanggan yang tidak ditemukan, permintaan yang dilakukan lebih dari satu kali, hingga kegagalan akibat gangguan jaringan. Pengujian ini memastikan bahwa respons sistem konsisten dan mampu memberikan umpan balik yang tepat kepada pengguna.

Tahap berikutnya yaitu *Place Order* dilakukan untuk memverifikasi bahwa proses pembuatan pesanan dapat berjalan sesuai prosedur. Pada bagian ini diuji apakah pengguna dapat memilih metode pembayaran, melakukan konfirmasi, dan menyelesaikan transaksi. Kemampuan sistem dalam mencegah terjadinya pemesanan ganda ketika terjadi klik berulang serta respons terhadap metode pembayaran yang tidak tersedia juga turut divalidasi.

Pada tahap *Post Checkout*, pengujian difokuskan pada perubahan status transaksi berdasarkan notifikasi dari sistem pembayaran. Status pada riwayat pesanan diperiksa untuk memastikan bahwa transaksi tercatat dengan benar ketika pembayaran berhasil, gagal, atau dibatalkan. Selain itu, alur notifikasi email multifinance juga diuji untuk memastikan bahwa setiap status transaksi mengirimkan email yang sesuai. Pengujian dilakukan untuk email Menunggu Pembayaran, Pembayaran Diterima, Pembayaran Gagal, serta Email Tagihan Berhasil Dibayar. Setiap email diverifikasi isi, waktu pengiriman, dan kesesuaiannya dengan status transaksi di sistem.

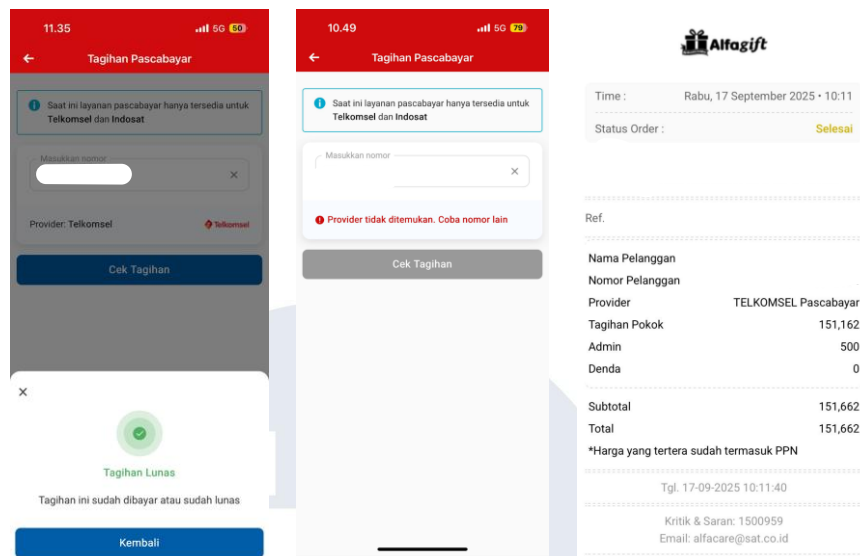
Pada bagian Microsite Multifinance, pengguna dapat memilih menu “Multifinance” untuk melakukan pengecekan cicilan berdasarkan penyedia angsuran yang bekerja sama dengan Alfagift. Setelah diarahkan ke halaman *webview*, pengguna mengisi penyedia

angsuran serta nomor pelanggan sesuai kontrak kredit yang dimiliki. Gambar 3.17 *Webview* Multifinance memperlihatkan tampilan saat sistem menampilkan detail tagihan setelah tombol Cek Tagihan ditekan.



Gambar 3.17 *Webview* Multifinance

Ketika tagihan masih aktif, sistem menampilkan informasi detail seperti nama pelanggan, periode tagihan, tagihan pokok, biaya admin, serta total pembayaran yang harus diselesaikan. Tampilan ini, sebagaimana terlihat pada gambar pertama, menyediakan tombol Bayar Tagihan bagi pengguna untuk melanjutkan proses pembayaran. Jika tagihan sudah dilunasi atau tidak ditemukan, sistem menampilkan pesan sesuai kondisinya, seperti pada gambar kedua yang menunjukkan status Tagihan Lunas. Setelah proses pembayaran berhasil diselesaikan melalui APG, pengguna diarahkan ke halaman *Order Detail* untuk melihat ringkasan transaksi. Gambar ketiga memperlihatkan halaman tersebut, yang berisi nomor referensi, informasi pelanggan, penyedia angsuran, rincian tagihan, total pembayaran, serta akses ke *e-receipt* sebagai bukti resmi.

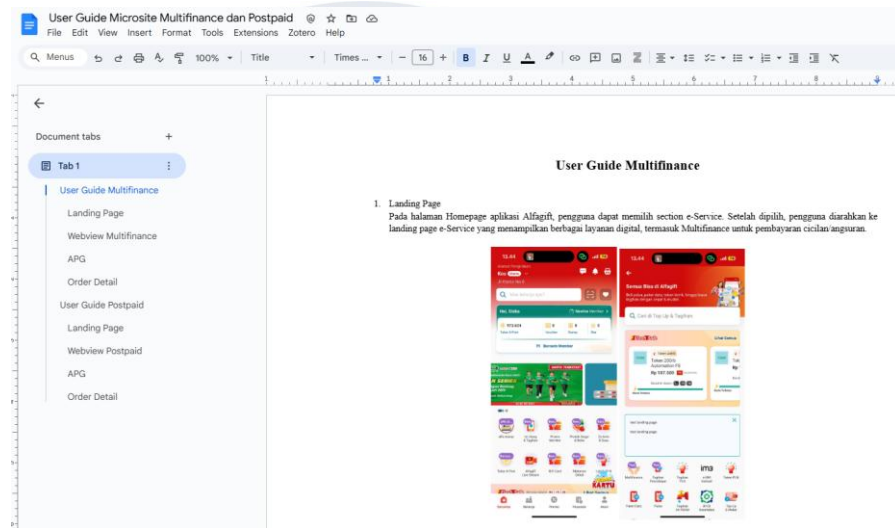


Gambar 3.18 *Webview* Postpaid

Fitur Microsite Postpaid memiliki alur yang mirip dengan Multifinance, namun secara khusus digunakan untuk pembayaran tagihan pascabayar seperti layanan telekomunikasi. Pengguna dapat memilih menu “Postpaid”, lalu diarahkan ke *webview* yang ditampilkan pada Gambar 3.18 *Webview* Postpaid, di mana terdapat satu kolom untuk memasukkan nomor pelanggan. Sistem secara otomatis akan mendeteksi provider layanan berdasarkan empat digit awal nomor pelanggan, dan menampilkan nama serta logo penyedia, seperti Telkomsel atau Indosat. Setelah provider terdeteksi, tombol “Cek Tagihan” akan aktif dan dapat digunakan untuk menampilkan status tagihan.

Hasil pengecekan dapat menampilkan beberapa kondisi yang ditunjukkan pada Gambar 3.18 *Webview* Postpaid. Jika tagihan belum dibayar, pengguna akan melihat detail jumlah tagihan, biaya admin, denda (jika ada), dan total pembayaran, kemudian dapat melanjutkan ke proses pembayaran. Jika tagihan sudah lunas, nomor pelanggan tidak terdaftar, atau provider tidak sesuai, sistem akan memberikan pesan yang sesuai dengan kondisi tersebut, seperti “Tagihan Lunas” atau “Provider tidak ditemukan.” Selain itu, jika terjadi gangguan

jaringan, sistem akan menampilkan notifikasi agar pengguna memeriksa kembali koneksi internet sebelum melanjutkan transaksi. Setelah pembayaran berhasil, pengguna diarahkan ke halaman *Order Detail* yang menampilkan ringkasan pembelian, detail layanan, serta bukti pembayaran dalam bentuk *e-receipt*.



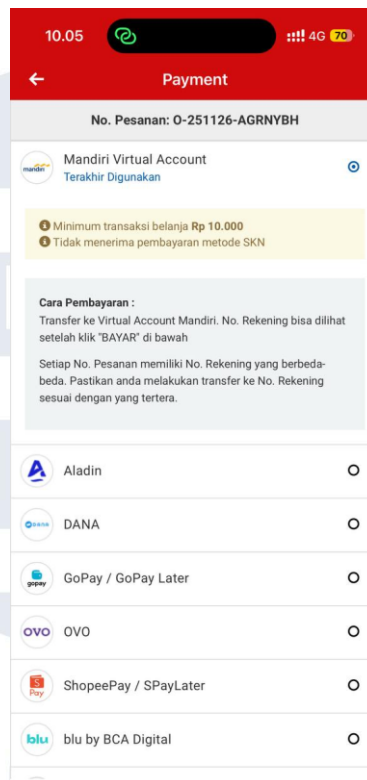
Gambar 3.19 *User Guide* Multifinance dan Postpaid

Gambar 3.19 *User Guide* Microsite Multifinance dan Postpaid memperlihatkan dokumentasi *User Guide* yang digunakan sebagai acuan dalam memahami alur dan tampilan fitur Microsite Multifinance serta Postpaid pada aplikasi Alfagift. Di dalam *User Guide* ini, setiap bagian alur pengguna disusun secara terstruktur mulai dari *Landing Page*, *Webview* Multifinance dan Postpaid, alur APG, hingga tampilan *Order Detail*. Dokumentasi ini menampilkan tangkapan layar dari aplikasi yang membantu menjelaskan bagaimana pengguna berinteraksi dengan masing-masing fitur.

3.3.1.5 Proyek *APG Auto Select Last Payment (V2)*

Proyek *APG Auto Select Last Payment* merupakan salah satu pengembangan fitur pada sistem pembayaran *Alfagift Payment Gateway (APG)*. Tujuan utama dari proyek ini adalah untuk meningkatkan pengalaman pengguna dalam proses transaksi dengan

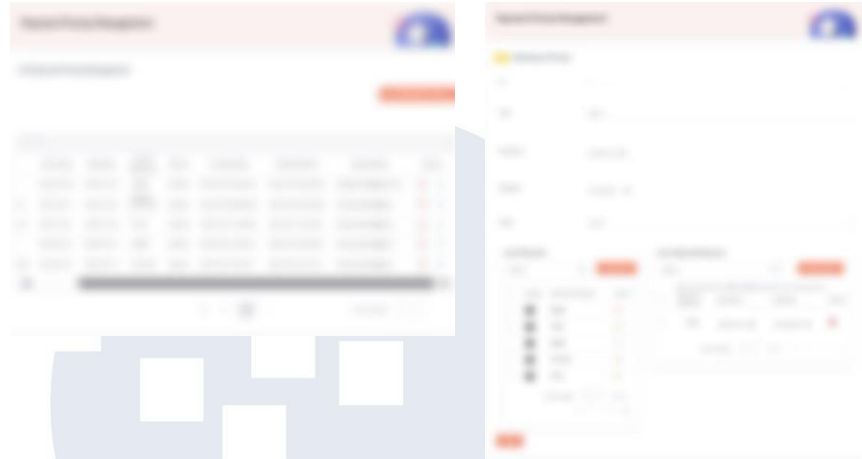
menampilkan metode pembayaran terakhir yang digunakan secara otomatis. Dengan adanya fitur ini, sistem dapat membantu pengguna melakukan pembayaran dengan lebih cepat dan efisien tanpa perlu memilih ulang metode pembayaran setiap kali bertransaksi.



Gambar 3.20 *APG Auto Select Last Payment*

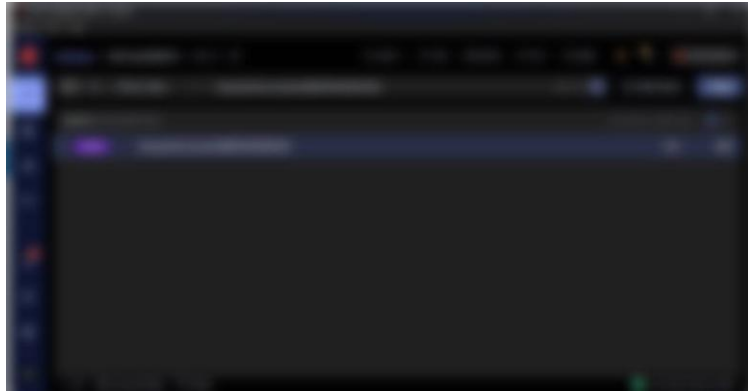
Secara garis besar, sistem ini dirancang untuk dapat mengenali dan menampilkan urutan metode pembayaran berdasarkan prioritas tertentu. Pada Gambar 3.20 *APG Auto Select Last Payment* terlihat bahwa sistem menempatkan Mandiri Virtual Account di posisi pertama karena merupakan metode pembayaran terakhir yang digunakan pengguna. Ketika pengguna membuka halaman pembayaran, sistem akan melakukan pengecekan apakah terdapat riwayat pembayaran sebelumnya. Jika ada, maka metode pembayaran terakhir tersebut akan langsung ditampilkan di posisi pertama dan otomatis terpilih. Sementara itu, jika tidak ditemukan riwayat pembayaran, sistem akan memilih metode pembayaran default, yaitu

Aladin. Dengan logika tersebut, pengguna akan selalu mendapatkan urutan tampilan pembayaran yang relevan, dimulai dari yang paling sering digunakan hingga pilihan default yang disediakan.



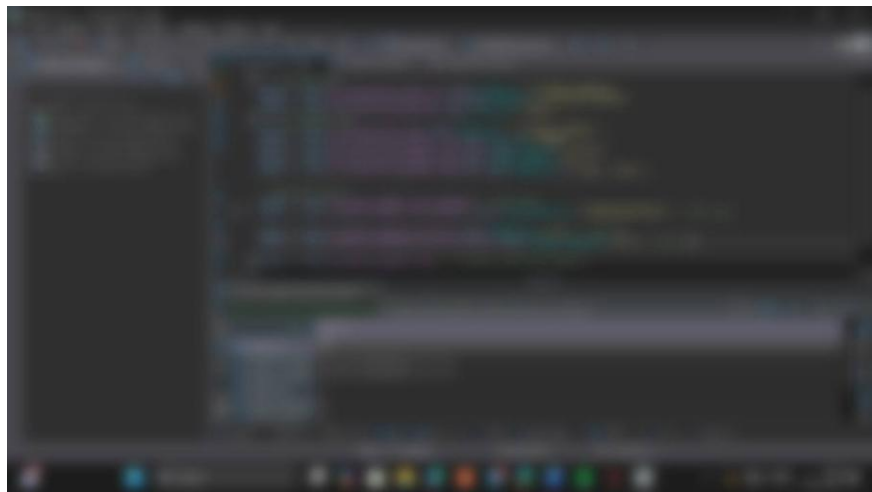
Gambar 3.21 Payment Priority Management

Gambar 3.21 Payment Priority Management memperlihatkan tampilan modul admin yang digunakan untuk mengelola pengaturan prioritas metode pembayaran. Pada modul ini, tim internal dapat menambahkan, mengubah, atau menghapus periode prioritas pembayaran sesuai kebutuhan bisnis. Setiap periode prioritas dapat berisi satu atau beberapa metode pembayaran yang diatur urutannya agar dapat ditampilkan sesuai kebijakan yang berlaku. Fitur ini memberikan fleksibilitas kepada tim bisnis untuk menyesuaikan prioritas pembayaran pada waktu tertentu, misalnya saat adanya kampanye atau kerja sama dengan penyedia layanan pembayaran. Melalui pengaturan ini, sistem dapat menampilkan metode pembayaran yang sesuai dengan strategi bisnis tanpa memengaruhi logika utama pemilihan otomatis berdasarkan riwayat pengguna.



Gambar 3.22 *Data Last Payment* pada Redis

Gambar 3.22 *Data Last Payment* pada Redis menunjukkan proses penyimpanan *data last payment* pada Redis, di mana sistem mencatat metode pembayaran terakhir yang berhasil digunakan oleh pengguna, sehingga dapat diakses kembali secara cepat pada transaksi berikutnya. Apabila data dengan *key* tersebut tersedia, metode pembayaran yang tersimpan akan langsung ditampilkan sebagai pilihan utama. Pendekatan ini memastikan pengalaman pengguna menjadi lebih efisien dan responsif.



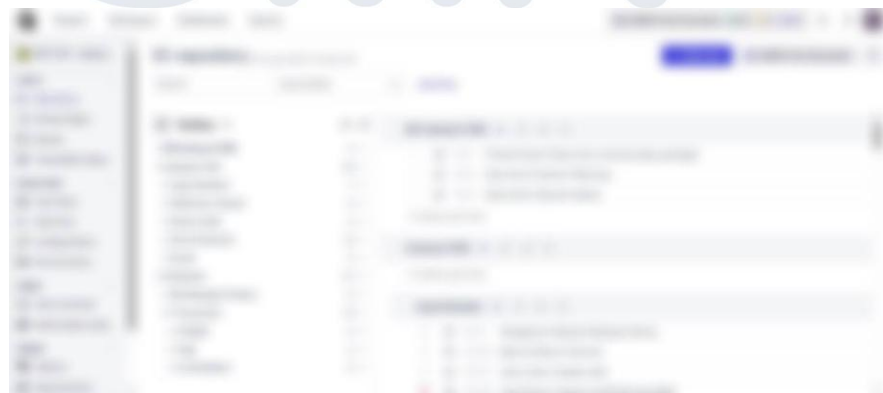
Gambar 3.23 Data Prioritas Pembayaran pada Postgre

Gambar 3.23 Pemeriksaan Data Prioritas Pembayaran pada Postgre menunjukkan proses pengecekan data terkait prioritas metode pembayaran melalui tabel-tabel di database. Data ini digunakan ketika pengguna tidak memiliki riwayat pembayaran pada Redis. Setiap entri

prioritas memiliki informasi seperti periode berlaku, daftar payment, serta nilai sort yang menentukan urutan tampilnya metode pembayaran. Pemeriksaan melalui SQL ini memastikan bahwa sistem menampilkan metode pembayaran sesuai aturan yang aktif pada periode tertentu. Dengan memanfaatkan Postgre sebagai sumber data fallback, sistem tetap dapat menampilkan daftar pembayaran dengan benar meskipun *data last payment* tidak ditemukan.

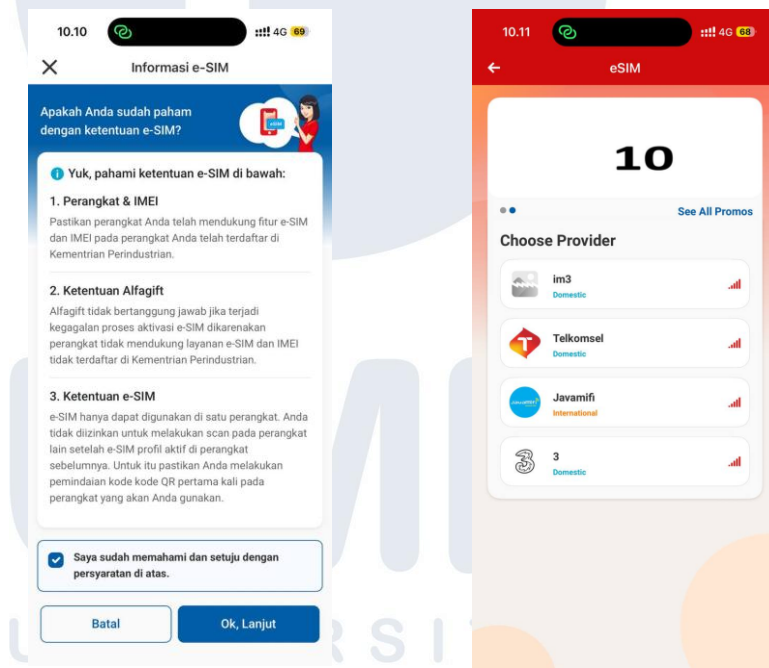
3.3.1.6 Proyek *Virtual Product* (E-SIM Domestic dan Internasional)

Proyek E-SIM Domestic dan Internasional merupakan pengembangan fitur yang memungkinkan pengguna melakukan pembelian e-SIM secara *digital* untuk berbagai provider, baik dalam negeri maupun luar negeri. Gambar 2.24 *Repository* Skenario E-SIM Domestic dan Internasional menunjukkan struktur *repository* skenario pengujian untuk fitur e-SIM, yang dikelompokkan ke dalam beberapa suite yang memisahkan alur e-SIM Domestic dan e-SIM International, sehingga proses validasi dapat dilakukan secara lebih terarah sesuai karakteristik masing-masing layanan. Pada e-SIM Domestic, skenario pengujian mencakup validasi form, *inquiry* tagihan, *place order*, perubahan status transaksi setelah pembayaran, serta pengiriman email notifikasi.



Gambar 3.24 *Repository* Skenario E-SIM Domestic dan Internasional

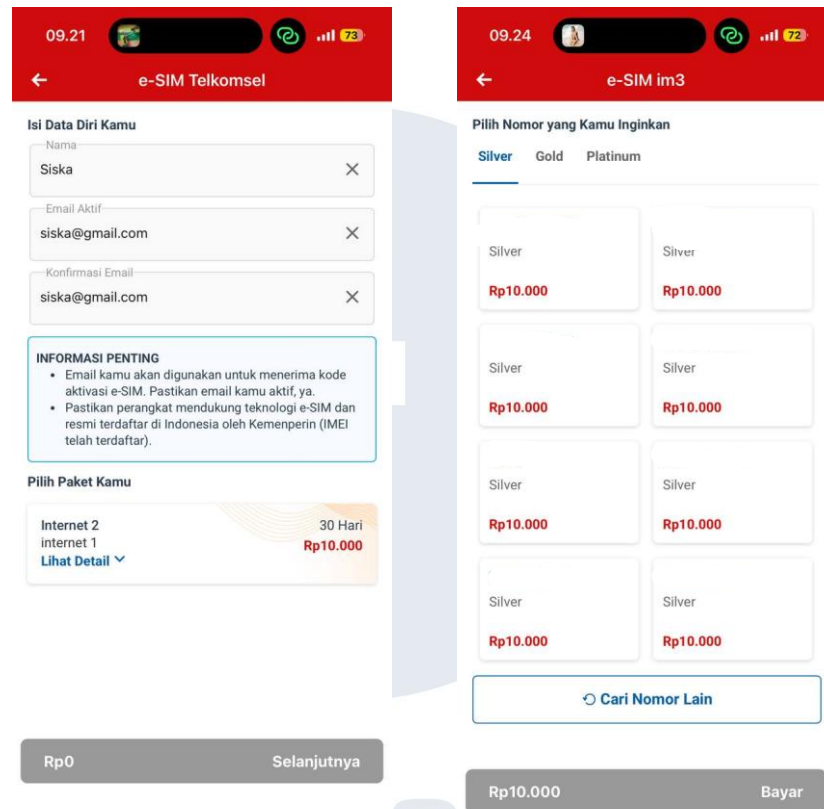
Sementara itu, bagian e-SIM International memiliki cakupan serupa tetapi disesuaikan dengan kebutuhan layanan internasional seperti JavaMifi. Pengujian meliputi validasi data pengguna, tampilan *order summary*, proses pembuatan pesanan, hingga pemeriksaan transaksi untuk paket dengan satu maupun beberapa kode aktivasi. Selain itu, dilakukan verifikasi pada halaman *Order Detail* dan *e-receipt* untuk memastikan seluruh informasi mulai dari harga, kode aktivasi, hingga masa berlaku tampil dengan lengkap dan konsisten. *Repository* ini juga mencakup skenario email yang berfungsi memastikan setiap perubahan status transaksi menghasilkan notifikasi yang sesuai.



Gambar 3.25 Webview E-SIM

Pada awal proses, pengguna akan diarahkan ke halaman informasi e-SIM, seperti terlihat pada Gambar 3.25 Webview E-SIM, yang berisi ketentuan mengenai perangkat, IMEI, dan kebijakan penggunaan e-SIM. Pengguna wajib membaca dan menyetujui persyaratan tersebut sebelum melanjutkan ke tahap berikutnya. Setelah itu, pengguna akan masuk ke halaman pemilihan provider,

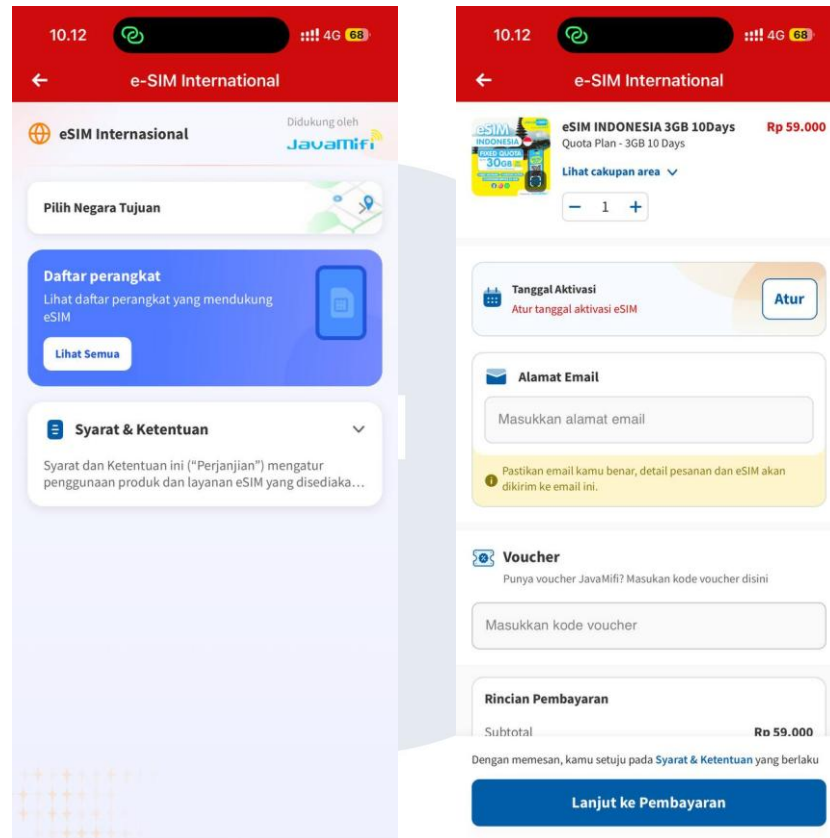
seperti ditunjukkan pada gambar, yang menampilkan pilihan e-SIM dari IM3, Telkomsel, Tri (3), dan Javamifi untuk layanan internasional.



Gambar 3.26 Webview E-SIM Domestic

Gambar 3.26 Webview E-SIM Domestic menampilkan tahapan awal proses pembelian e-SIM untuk provider domestik pada aplikasi Alfagift. Pada tahap ini, pengguna diminta mengisi data diri seperti nama, email aktif, dan konfirmasi email, yang akan digunakan untuk pengiriman kode aktivasi e-SIM. Sistem juga menampilkan informasi penting terkait persyaratan perangkat dan IMEI untuk memastikan bahwa perangkat yang digunakan mendukung teknologi e-SIM. Setelah mengisi data diri, pengguna dapat memilih paket layanan yang tersedia untuk provider tersebut. Pada langkah berikutnya, seperti terlihat pada gambar kedua, pengguna diarahkan

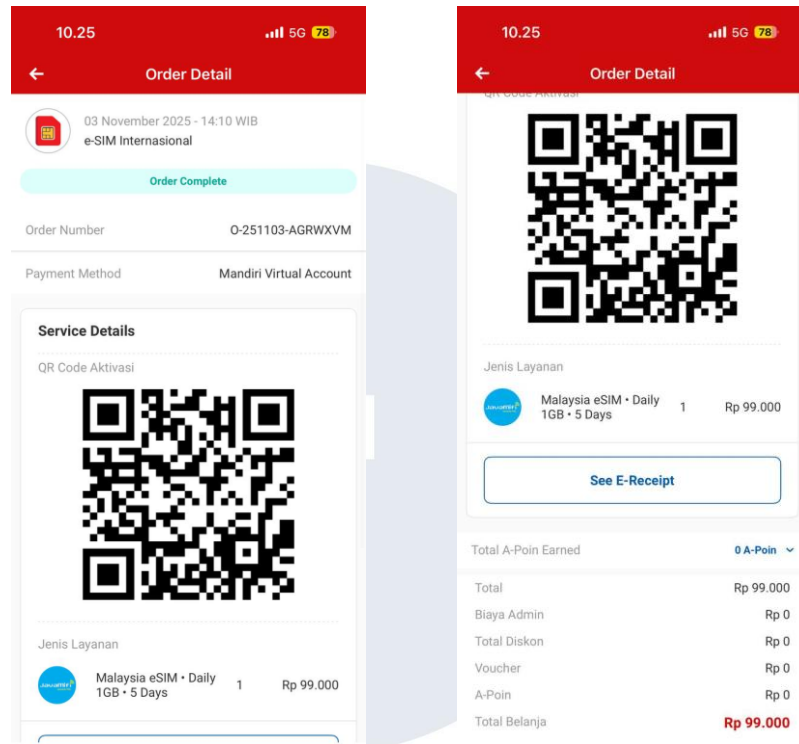
untuk memilih nomor e-SIM berdasarkan kategori yang disediakan, seperti Silver, Gold, atau Platinum.



Gambar 3.27 Webview E-SIM Internasional

Gambar 3.27 Webview E-SIM Internasional menunjukkan tampilan pembelian e-SIM untuk layanan internasional yang didukung oleh Javamifi. Pada tahap awal, pengguna dapat memilih negara tujuan yang ingin dikunjungi serta melihat daftar perangkat yang mendukung teknologi e-SIM agar proses aktivasi dapat berjalan dengan lancar. Sistem juga menyediakan akses ke syarat dan ketentuan penggunaan layanan, sehingga pengguna dapat memahami aturan sebelum melanjutkan transaksi. Setelah memilih negara tujuan dan paket yang sesuai, pengguna diarahkan ke halaman *order summary* seperti terlihat pada gambar kedua. Pada halaman ini, pengguna dapat mengatur tanggal aktivasi e-SIM, memasukkan alamat email untuk pengiriman detail pesanan, serta menambahkan

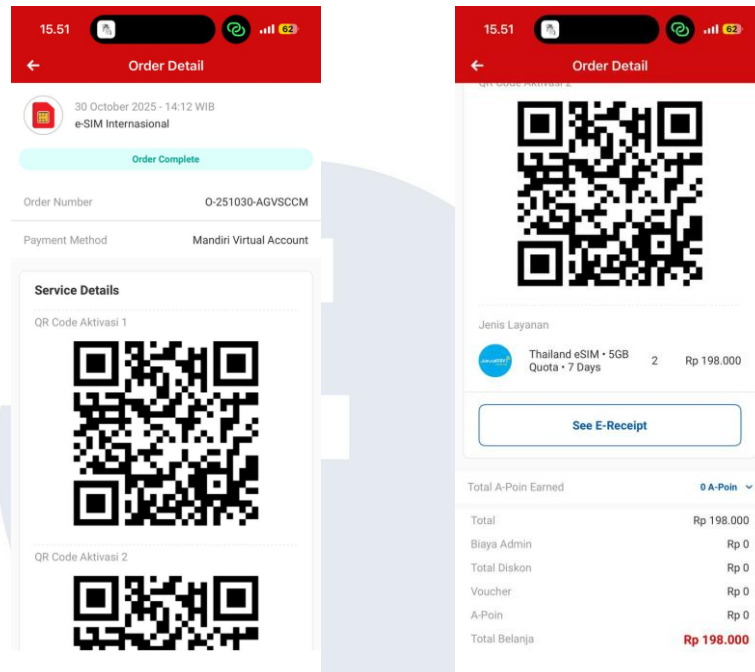
kode voucher apabila tersedia, sebelum melanjutkan ke proses pembayaran.



Gambar 3.28 Order History e-sim internasional 1 code

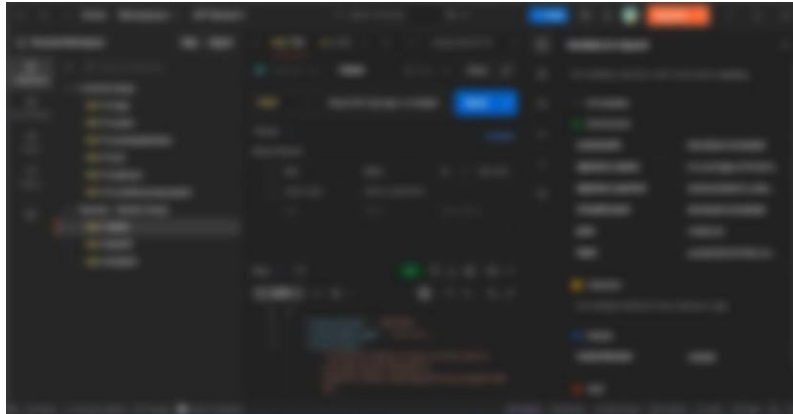
Pada bagian *Order History*, seperti terlihat pada Gambar 3.28 *Order History* e-SIM Internasional 1 Code, pengguna dapat melihat status pembelian e-SIM internasional mereka secara langsung. Informasi yang ditampilkan mencakup detail paket, biaya layanan, serial number, serta QR code aktivasi yang akan digunakan untuk mengaktifkan e-SIM. Jika pesanan dengan jumlah qty satu, sistem hanya menampilkan satu QR code sesuai dengan jumlah e-SIM yang dibeli, sehingga informasi yang diterima pengguna tetap jelas dan tidak membingungkan. Sistem juga menampilkan status transaksi berdasarkan kondisi aktual, seperti menunggu pembayaran, sukses, diproses, selesai, *expired*, atau batal. Selain itu, proses integrasi *backend* turut divalidasi untuk memastikan bahwa data pesanan,

perubahan status, serta proses *top-up* e-SIM tercatat dan tersinkronisasi dengan benar dalam sistem.



Gambar 3.29 Order History e-sim internasional 2 code

Gambar 3.29 Order History e-SIM Internasional 2 Code menampilkan tampilan riwayat pesanan ketika pengguna membeli e-SIM internasional dengan jumlah lebih dari satu unit. Pada kondisi ini, sistem menampilkan dua QR code aktivasi, masing-masing mewakili satu e-SIM yang dipesan. Setiap QR code dilengkapi dengan informasi detail seperti serial number dan paket layanan yang terkait, sehingga pengguna dapat mengaktivasi masing-masing e-SIM secara terpisah sesuai kebutuhan.



Gambar 3.30 Tampilan Postman

Gambar 3.30 tampilan Postman yang digunakan untuk melakukan pengujian API terkait proses pembayaran pada sistem Alfagift. Pada gambar tersebut, terlihat bahwa pengujian dilakukan pada *endpoint* TOKEN untuk mendapatkan access token yang diperlukan pada tahap selanjutnya. Postman menampilkan struktur request beserta parameter yang digunakan dan variabelnya. Bagian output menampilkan respons API dengan status 200 yang menunjukkan bahwa permintaan berhasil diproses dan token berhasil dihasilkan. Tampilan ini digunakan sebagai bagian dari proses validasi integrasi *backend* untuk memastikan bahwa seluruh alur pembayaran berjalan sesuai dengan kebutuhan sistem.

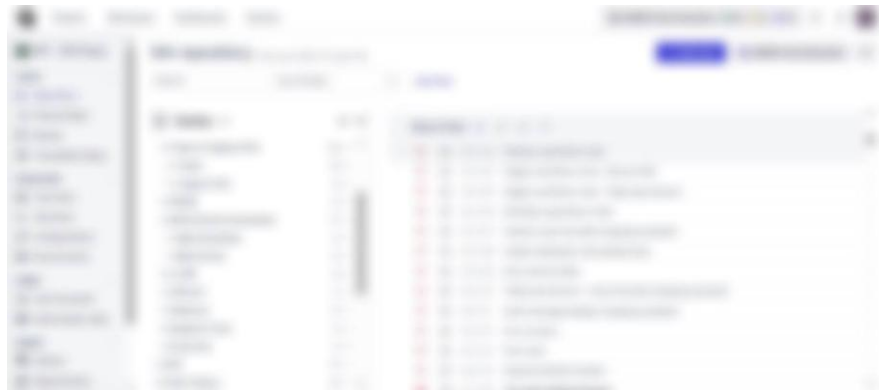


Gambar 3.31 Email Invitation untuk Pelaksanaan UAT

Gambar 3.31 Email Invitation untuk Pelaksanaan UAT menampilkan contoh penyampaian informasi resmi terkait jadwal User Acceptance Test (UAT) yang dikirimkan oleh pihak eksternal kepada tim alfagift. Email tersebut berisi detail pelaksanaan UAT, seperti tanggal, waktu, jenis produk yang diuji, serta lokasi pengujian. Setelah menerima email tersebut, informasi kemudian diteruskan kembali melalui percakapan di grup chat agar setiap pihak dapat mengetahui jadwal UAT dan melakukan persiapan.

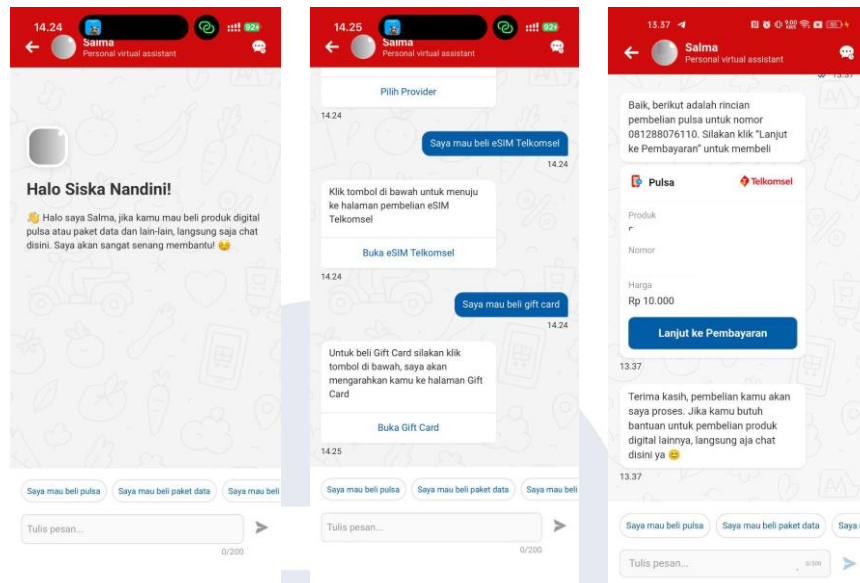
3.3.1.7 Proyek *Shopping Assistant e-service*

Proyek *Shopping Assistant* merupakan pengembangan fitur berbasis kecerdasan buatan yang bertujuan untuk menghadirkan asisten belanja virtual di dalam aplikasi Alfagift. Fitur ini memungkinkan pengguna melakukan transaksi berbagai produk *digital* seperti pulsa, paket data, token listrik, BPJS, e-SIM, *gift card*, dan layanan lainnya hanya melalui percakapan di chatbot. Pada gambar *Repository Skenario Shopping Assistant* ditampilkan berbagai suite skenario terkait layanan *e-service* di aplikasi. Dalam *repository* ini, dilakukan penambahan skenario pengujian untuk fitur *Shopping Assistant*, khususnya terkait validasi alur percakapan dan integrasi dengan data transaksi pengguna. Salah satu pengembangan yang dilakukan adalah penambahan skenario *Order History* untuk status transaksi *create order* dan *paid* berdasarkan daftar produk *e-service* yang tersedia. Penambahan skenario tersebut bertujuan untuk memastikan bahwa *Shopping Assistant* dapat menampilkan riwayat pesanan sesuai status transaksi dan jenis produk yang digunakan oleh pengguna.



Gambar 3.32 Repository Skenario *Shopping Assistant*

Selain itu, dilakukan pula penyusunan skenario *Negative Case* untuk fitur chat AI pada *Shopping Assistant*. Skenario ini mencakup kondisi seperti penggunaan *keyword* nomor acak, kalimat acak, *irrelevant word* lebih dari tiga kali, respons ketika layanan tidak aktif, hingga kondisi *session expired*. Penyusunan skenario ini memastikan bahwa chatbot mampu memberikan respons yang tepat, menampilkan pesan kesalahan yang sesuai, serta tetap stabil meskipun menerima input yang tidak relevan. Selanjutnya, skenario pengujian untuk kategori Groceries juga dikembangkan, yang meliputi fitur pencarian produk melalui chat (Search Product Groceries), kendala pada penerimaan pesan (*error receive*), serta kendala pada pengiriman pesan (*error sent*). Seluruh pengembangan skenario ini bertujuan untuk memastikan bahwa *Shopping Assistant* dapat menangani berbagai kondisi penggunaan secara komprehensif, baik pada alur normal maupun saat terjadi gangguan layanan.



Gambar 3.33 Tampilan Chat *Shopping Assistant*

Tampilan percakapan awal, seperti terlihat pada Gambar 3.33 Tampilan Chat *Shopping Assistant*, menunjukkan salam pembuka dari asisten virtual yang memandu pengguna untuk memulai transaksi. Dengan pendekatan ini, pengguna tidak perlu menavigasi halaman produk secara manual karena seluruh proses dapat dilakukan melalui antarmuka chat yang lebih cepat dan intuitif.

Sistem *Shopping Assistant* mampu mengenali maksud pengguna berdasarkan teks yang dikirimkan. Ketika pengguna mengetik perintah seperti “Saya mau beli eSIM Telkomsel” atau “Saya mau beli *gift card*,” sistem menampilkan respons yang sesuai seperti pada gambar. Asisten akan memberikan tombol aksi yang mengarahkan pengguna ke halaman produk atau *webview* terkait. Kemampuan ini didukung oleh model bahasa yang memproses konteks percakapan dan mengirimkan permintaan ke *backend*.

Pada transaksi pulsa dan produk serupa, *Shopping Assistant* juga menampilkan ringkasan pesanan sebelum diarahkan ke halaman pembayaran. Gambar menunjukkan bagaimana asisten menampilkan detail produk, nomor tujuan, harga, serta tombol “Lanjut ke

Pembayaran.” Penyajian ringkasan ini membantu pengguna memeriksa informasi sebelum melanjutkan transaksi. Setelah itu, pengguna diarahkan ke Payment Gateway untuk memilih metode pembayaran. Pendekatan ini membantu meminimalkan kesalahan input dan menjaga alur transaksi tetap jelas.

Shopping Assistant juga mendukung redireksi otomatis untuk layanan seperti e-SIM, *gift card*, dan berbagai produk *e-service* lainnya. Ketika pengguna mengetik permintaan untuk layanan tertentu, sistem akan menampilkan tombol yang mengarahkan langsung ke halaman layanan terkait. Contohnya dapat dilihat pada gambar ketika pengguna meminta pembelian e-SIM dan *gift card*. Fitur ini membuat akses ke layanan lebih cepat tanpa memerlukan pencarian manual. Hal tersebut meningkatkan efisiensi dalam penggunaan aplikasi.

Sistem juga dilengkapi mekanisme fallback limit. Mekanisme ini aktif ketika pengguna berulang kali mengirimkan pesan yang tidak dapat dipahami oleh sistem. Jika batas dicapai, asisten akan memberi pemberitahuan bahwa percakapan dihentikan sementara. Hal ini membantu menjaga performa dan mencegah pemrosesan permintaan berulang yang tidak valid. Integrasi pembayaran dan *e-receipt* tetap berjalan melalui *Alfagift Payment Gateway*.

3.3.1.8 Proyek *Virtual Product* (Omni CMV dan BPJS Denda)

Proyek Omni CMV merupakan pengembangan fitur kerja yang memungkinkan pengguna membeli paket data Indosat langsung melalui aplikasi Alfagift. Integrasi ini bertujuan menghadirkan proses pembelian paket data yang lebih cepat, seamless, dan terhubung langsung dengan sistem Indosat melalui mekanisme API dan *webview*. Gambar 3.34 *Repository* Skenario Omni CMV menampilkan struktur *repository* skenario pengujian untuk fitur Indosat CVM, yang terdiri dari beberapa suite seperti Input Number,

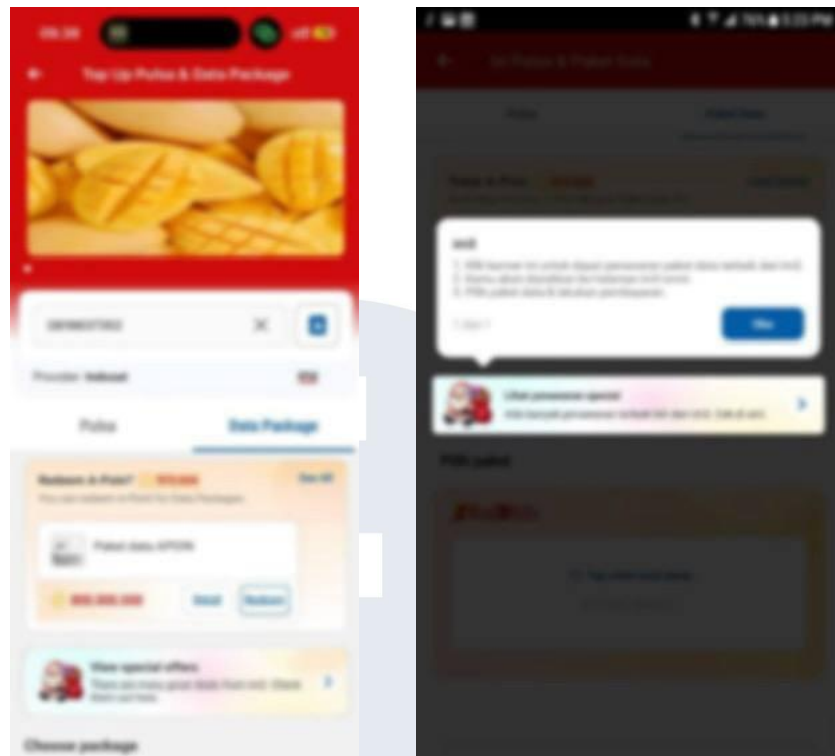
Webview Indosat, *Place Order*, *Post Checkout*, dan Email. Pada bagian ini, serangkaian skenario telah dikembangkan untuk memastikan bahwa seluruh alur layanan Indosat CVM berjalan sesuai ketentuan, mulai dari tahap awal ketika pengguna memasukkan nomor hingga tahap akhir berupa notifikasi dan pembaruan status transaksi.



Gambar 3.34 *Repository* Skenario Omni CMV

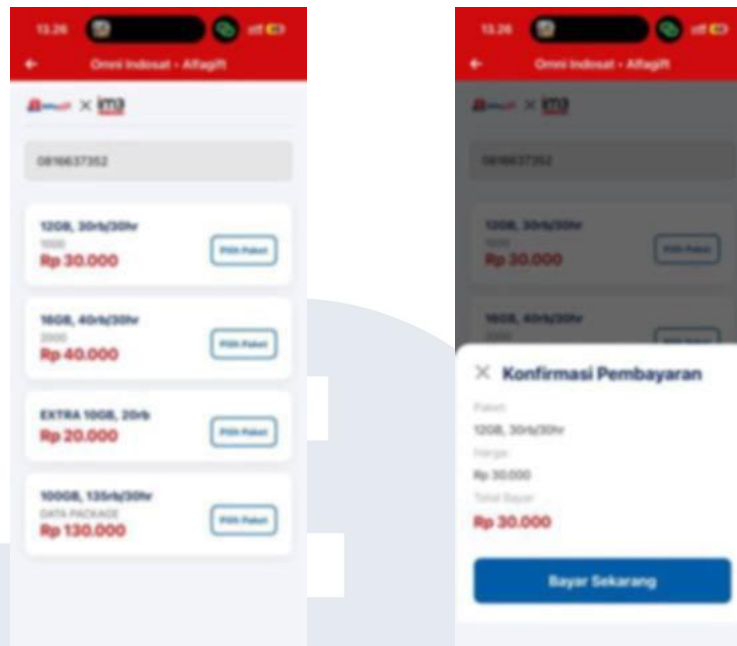
Pada suite Input Number, pengujian difokuskan pada perilaku halaman Masukan Nomor. Skenario yang dibuat meliputi navigasi menuju halaman input, kembali ke menu *E-Service*, pengujian input nomor Indosat valid, serta beberapa kondisi invalid seperti nomor kurang digit dan nomor acak. Selain itu, pengujian juga mencakup skenario pemanggilan daftar kontak serta navigasi kembali ke halaman “Pulsa dan Paket Data.”

Pada suite *Webview* Indosat, pengujian diarahkan untuk memastikan bahwa halaman *webview* yang digunakan untuk menampilkan pilihan paket dan detail layanan termuat dengan benar, termasuk validasi tampilan, pemanggilan data, hingga respons ketika terjadi kondisi tertentu. Sementara itu, pada bagian Email, pengujian dilakukan untuk memastikan bahwa sistem mengirimkan notifikasi email sesuai status transaksi, baik untuk transaksi berhasil, gagal, menunggu pembayaran, maupun kondisi lainnya.



Gambar 3.35 Tampilan Top up pulsa & Paket Data

Alur pembelian dimulai saat pengguna memasuki halaman *Top Up Pulsa & Paket Data*, sebagaimana terlihat pada Gambar 3.35 Tampilan *Top Up Pulsa & Paket Data*. Pada halaman tersebut, pengguna memasukkan nomor handphone yang akan diisi, lalu sistem otomatis mendeteksi provider sebagai Indosat melalui indikator IM3. Setelah itu, pengguna berpindah ke tab *Data Package* untuk melihat pilihan paket atau penawaran khusus yang tersedia. Ketika pengguna menekan banner promo Indosat, sistem menampilkan pop-up panduan, seperti terlihat pada gambar kedua, yang menjelaskan langkah-langkah untuk melanjutkan pembelian paket melalui halaman Omni. Pop-up ini membantu memastikan bahwa pengguna memahami alur sebelum diarahkan ke halaman pembelian paket data Indosat.



Gambar 3.36 Tampilan *Webview* Omni CVM Indosat

Gambar 3.36 Tampilan *Webview* Omni CVM Indosat menunjukkan halaman khusus yang ditampilkan setelah pengguna diarahkan dari menu paket data ke *webview* Omni Indosat. Pada tampilan ini, sistem menampilkan nomor handphone yang telah diinput sebelumnya, lengkap dengan identifikasi provider IM3 sebagai layanan yang digunakan. Kemudian dibawahnya, pengguna dapat melihat daftar paket data yang tersedia, salah satunya adalah paket Freedom Internet 100GB / 30 hari, yang ditampilkan dengan harga dan detail kuota secara jelas. Tampilan ini memudahkan pengguna dalam memilih paket yang paling sesuai sebelum melanjutkan transaksi.

Ketika pengguna menekan tombol Pilih Paket, sistem menampilkan pop-up Konfirmasi Pembayaran seperti terlihat pada gambar kedua. Pop-up ini menyajikan ringkasan paket yang dipilih, termasuk nama paket, harga, dan total pembayaran yang harus diselesaikan. Dengan menyediakan ringkasan ini, sistem memastikan bahwa pengguna dapat memeriksa kembali pilihan mereka sebelum

menekan tombol Bayar Sekarang untuk melanjutkan ke proses pembayaran.



Gambar 3.37 Tampilan *Automation* Pulsa/Paket Data

Gambar 3.37 Tampilan *Automation* Pulsa/Paket Data menggambarkan struktur kode otomatisasi yang digunakan untuk memvalidasi alur transaksi pembelian pulsa dan paket data pada aplikasi Alfagift. Kode otomatisasi ini dikembangkan menggunakan Java karena bahasa tersebut mendukung stabilitas, modularitas, dan praktik pemrograman berorientasi objek yang diperlukan dalam pengujian berskala besar. Appium digunakan sebagai framework utama untuk mengotomatisasi interaksi pengguna pada perangkat *mobile*, dan framework ini dibangun di atas arsitektur Selenium WebDriver sehingga mekanisme identifikasi elemen dan eksekusi perintah mengikuti konsep yang serupa dengan Selenium. Pendekatan Behavior-Driven Development (BDD) diterapkan melalui penyusunan skenario yang menggambarkan perilaku sistem dari sudut pandang pengguna bisnis. Integrasi antara Java, Appium, dan fondasi Selenium WebDriver memungkinkan alur pengujian tereksekusi dengan lebih terstruktur dan presisi sesuai standar validasi.

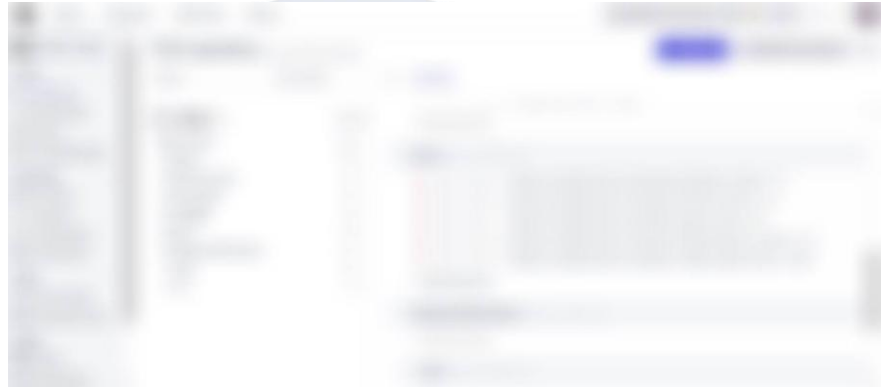
Pada pengujian fitur pulsa dan paket data, proses otomatisasi dimulai dengan menavigasi aplikasi menuju modul *e-service* sebelum memasuki langkah interaksi berikutnya. Skrip otomatis kemudian

melakukan input nomor telepon, memilih nominal pulsa atau paket data, serta memverifikasi kesesuaian harga yang ditampilkan pada ringkasan transaksi. Setiap aksi otomatis diatur melalui step definitions yang bertugas menghubungkan skenario BDD dengan fungsi kode Java yang akan dijalankan. Validasi UI dilakukan untuk memastikan bahwa elemen tampilan seperti tombol, teks, maupun respons layar berfungsi sesuai desain dan *requirement* bisnis. Dengan pendekatan ini, tim QA dapat memastikan akurasi alur bisnis tanpa harus melakukan pengujian manual secara berulang.

Otomatisasi juga mencakup tahap lanjutan seperti simulasi proses pembayaran serta verifikasi status transaksi pada halaman *Order Detail*. Setelah transaksi dijalankan secara otomatis, skrip memastikan bahwa aplikasi menampilkan status yang sesuai dengan logika bisnis, seperti “Berhasil”, “Menunggu Pembayaran”, atau “Gagal”. Proses ini membantu mengidentifikasi inkonsistensi antara tampilan aplikasi dan data hasil pemrosesan *backend* yang mungkin tidak terdeteksi dalam pengujian manual. *Automation* juga dapat memeriksa histori transaksi untuk memastikan bahwa setiap aktivitas tercatat secara akurat dan tanpa anomali. Dengan seluruh langkah yang dijalankan secara konsisten, tim QA memperoleh hasil pengujian yang objektif dan bebas dari variasi yang dapat muncul akibat intervensi pengguna.

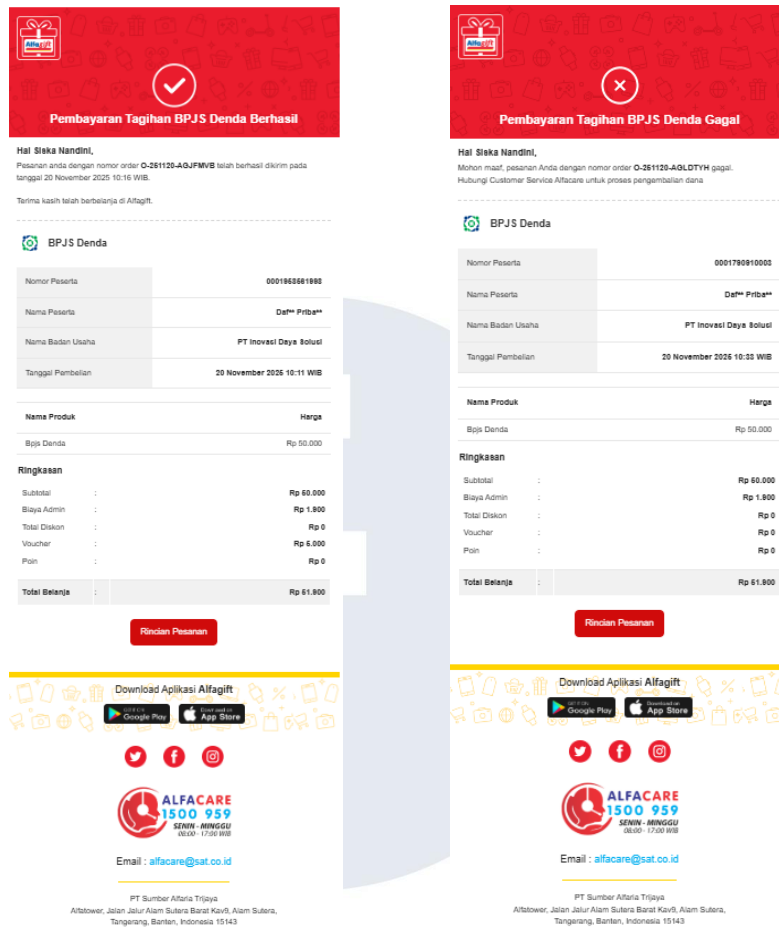
Penerapan *automation* pada fitur pulsa dan paket data berperan signifikan dalam mendukung efektivitas *regression testing* pada setiap siklus pengembangan aplikasi. Skrip otomatis dapat dijalankan ulang dengan cepat setiap kali terjadi pembaruan fitur tanpa perlu menyiapkan ulang skenario pengujian manual. Efisiensi ini membantu tim QA mempertahankan konsistensi kualitas aplikasi dan mengurangi risiko kesalahan yang muncul ketika dilakukan pengujian berulang secara manual. Fondasi Selenium WebDriver yang

digunakan Appium juga membuat skrip lebih mudah dipelihara karena mengikuti standar industri dalam automated testing. Dengan demikian, otomatisasi ini mendukung kestabilan fitur *e-service* pulsa dan paket data serta memastikan pengalaman pengguna tetap optimal pada setiap perilisan aplikasi.



Gambar 3.38 *Repository* BPJS Denda

Gambar 3.38 *Repository* BPJS Denda menampilkan daftar skenario pengujian yang berfokus pada validasi pengiriman email untuk setiap status transaksi pembayaran BPJS Denda. Pada bagian ini, beberapa skenario dibuat untuk memastikan bahwa sistem mengirimkan notifikasi email yang sesuai dengan kondisi transaksi pengguna. Pengujian mencakup verifikasi email “Menunggu Pembayaran,” “Pembayaran Diterima,” “Pembayaran Gagal,” “Pembayaran Tagihan Berhasil,” serta “Pembayaran Tagihan Gagal.” Setiap skenario digunakan untuk memastikan bahwa email terkirim dengan benar, memuat informasi transaksi yang akurat, dan sesuai dengan status yang muncul pada sistem maupun halaman *Order Detail*.



Gambar 3.39 Tampilan Email BPJS Denda

Gambar 3.39 Tampilan Email BPJS Denda menampilkan contoh notifikasi email yang dikirimkan sistem berdasarkan hasil proses *top-up* BPJS Denda. Pada tampilan pertama, ditunjukkan bahwa proses *top-up* berhasil dilakukan, ditandai dengan ikon centang dan judul “Pembayaran Tagihan BPJS Denda Berhasil.” Email ini menampilkan informasi lengkap seperti nomor peserta, nama peserta, nama badan usaha, tanggal pembelian, rincian produk, serta total biaya yang dibayarkan.

3.3.2 Kendala yang Ditemukan

Berikut ini adalah beberapa kendala utama selama pelaksanaan kegiatan magang sebagai *Intern Quality Assurance* di PT Global Loyalty Indonesia yaitu sebagai berikut:

a. Keterbatasan kualitas jaringan Wi-Fi

Selama proses magang, koneksi jaringan Wi-Fi di lingkungan kerja tidak selalu stabil. Kondisi ini menyebabkan keterlambatan dalam mengakses file, melakukan sinkronisasi data, serta membuka *tools* yang memerlukan koneksi internet. Ketidakstabilan jaringan tersebut berdampak pada efisiensi waktu pengerjaan, terutama pada tugas yang membutuhkan akses ke database, Admin *Tools*, maupun Swagger UI.

b. Belum familiar dengan *tools* baru

Selama proses pengerjaan proyek, muncul tantangan terkait penggunaan berbagai *tools* yang sebelumnya belum pernah digunakan dalam perkuliahan maupun proyek kampus. *Tools* tersebut meliputi MongoDB untuk mengakses database NoSQL, Swagger UI dan Postman untuk dokumentasi serta pengujian API, serta DBeaver sebagai alat manajemen database.

c. Koordinasi Internal dan Eksternal belum optimal.

Pada tahap pengujian, khususnya ketika sistem telah memasuki fase integrasi, proses koordinasi melibatkan berbagai pihak seperti Developer, QA, serta unit atau pihak eksternal terkait. Dalam kondisi ini, proses identifikasi sumber *bug* membutuhkan waktu yang lebih panjang karena pengujian dilakukan pada lingkungan yang telah terintegrasi, sehingga permasalahan yang muncul dapat berasal dari berbagai komponen sistem, seperti service, aplikasi *mobile*, *back office*, APG, pihak eksternal, maupun provider. Oleh karena itu, setiap pihak perlu melakukan pengecekan terlebih dahulu pada lingkup tanggung jawabnya masing-masing sebelum dapat ditentukan sumber utama permasalahan. Kondisi ini merupakan hal yang wajar dalam proses pengujian integrasi dan berdampak pada durasi penyelesaian isu.

Selain itu, waktu penyelesaian *bug* juga dapat dipengaruhi oleh perbedaan beban kerja pada masing-masing tim, sehingga proses penanganan isu memerlukan penyesuaian prioritas dan koordinasi lanjutan. Tantangan lain muncul dalam koordinasi dengan pihak eksternal atau vendor, terutama ketika waktu respons yang dibutuhkan relatif lebih lama. Dalam beberapa kasus, adanya perubahan atau penyesuaian dari pihak eksternal juga memerlukan pengujian ulang untuk memastikan konsistensi fungsi dan tampilan sistem sebelum fitur dapat dirilis ke lingkungan *production*. Kondisi tersebut berdampak pada penyesuaian *timeline sprint*, namun tetap menjadi bagian dari upaya menjaga kualitas dan stabilitas aplikasi secara menyeluruh.

3.3.3 Solusi atas Kendala yang Ditemukan

Sebagai upaya mengatasi kendala yang ditemui selama pelaksanaan magang, dilakukan beberapa langkah penyelesaian sebagai berikut:

a. Menyiapkan alternatif koneksi

Sebagai upaya menjaga proses pengujian tetap berjalan lancar meskipun koneksi Wi-Fi mengalami kendala, disiapkan alternatif koneksi seperti menggunakan hotspot pribadi atau meminta dukungan tim *IT Support*.

b. Adaptasi dengan *tools* yang baru

Dalam penggunaan *tools* yang belum pernah digunakan sebelumnya, pemahaman diperoleh melalui dokumentasi proyek sebelumnya serta eksplorasi langsung untuk mengenali fungsi dan alur kerja setiap *tools*. Jika muncul kendala teknis, penjelasan tambahan diperoleh melalui diskusi dengan mentor maupun anggota tim. Pendekatan ini membantu mempercepat proses adaptasi terhadap *tools* seperti MongoDB, Swagger UI, Postman, dan DBeaver sehingga dapat digunakan secara optimal dalam proses pengujian.

c. Meningkatkan Koordinasi Internal dan Eksternal

Sebagai upaya mengelola tantangan pada tahap pengujian integrasi yang melibatkan banyak pihak, koordinasi internal dan eksternal dioptimalkan melalui pembaruan progres secara rutin, baik melalui *daily update* maupun media koordinasi bersama. Penyampaian informasi terkait perubahan, kebutuhan tambahan, serta temuan *bug* dilakukan secara terstruktur agar setiap pihak memiliki pemahaman yang selaras dan dapat menyesuaikan prioritas pekerjaan sesuai dengan kapasitas dan beban kerja masing-masing. Selain itu, koordinasi dengan pihak eksternal atau vendor diperkuat dengan memastikan seluruh penyesuaian maupun proses *deployment* mengikuti alur dan kesepakatan yang telah ditetapkan, khususnya terkait pemisahan *environment* seperti *staging* dan *production*. Pendekatan ini bertujuan untuk menjaga konsistensi proses pengujian, meminimalkan kebutuhan pengujian ulang, serta membantu pengelolaan *timeline sprint* tanpa mengurangi kualitas aplikasi yang dikembangkan.

