

BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

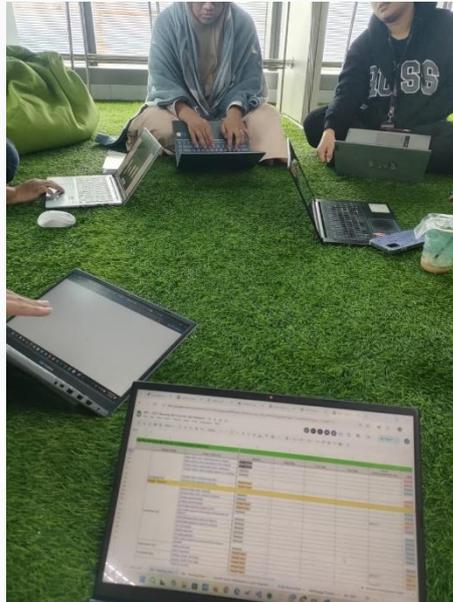
Selama mengikuti kegiatan magang *track 1* di PT. Global Loyalty Indonesia, penulis menempati posisi sebagai *Intern Quality Assurance* di bawah naungan Departemen *Product Operations* yang dipimpin oleh Bapak Benny Chandra dan dibimbing oleh Kak Alpha Adarrani sebagai *mentor* selama program kerja magang. Proyek yang tengah dikerjakan oleh *mentor* akan didampingi oleh *Quality Assurance Intern*, dimana para mahasiswa magang berperan aktif dalam mendukung proses kerja sesuai dengan alur dan tanggung jawab yang harus dipenuhi pada setiap proyeknya.

Selama masa pengerjaan proyek di PT. Global Loyalty Indonesia, untuk berkoordinasi dengan rekan-rekan tim lainnya seluruh karyawan menggunakan aplikasi Telegram dan Google Meet. Aplikasi Telegram menjadi sarana komunikasi antar tim, sedangkan Google Meet digunakan untuk sarana komunikasi secara lisan.

3.1.1 *Daily Meeting*

Daily Meeting biasa dilakukan setiap seminggu sekali di pagi hari menggunakan sarana Google Meet untuk menyampaikan *update* perkembangan tugas masing-masing anggota tim, kendala yang dihadapi, serta rencana kerja selanjutnya. Selain itu *meeting* ini juga menjadi ajang koordinasi untuk memastikan semua tim berada pada jalur yang sama dan dapat saling membantu jika terdapat hal-hal yang menjadi hambatan jalannya proyek, seperti yang tertampil pada Gambar 3.1.

mendiskusikan suatu hal secara langsung. *Internal Meeting* berfokus untuk membahas berbagai aspek teknis yang berkaitan dengan proses pengujian aplikasi, seperti yang terlihat pada Gambar 3.3.



Gambar 3.3 *Internal Meeting*

3.2 Tugas dan Uraian Kerja Magang

Selama melakukan kegiatan praktik magang di PT. Global Loyalty Indonesia, penulis mendapatkan kesempatan untuk terlibat langsung dalam proses pengujian kualitas aplikasi Alfagift melalui beberapa proyek yang sedang berjalan maupun sudah berjalan pada PT. GLI. Selama pengerjaan proyek, penulis didampingi oleh seorang *mentor* yang berperan dalam membimbing selama pengerjaan proyek.

Secara umum, ruang lingkup pekerjaan sebagai *Quality Assurance Intern* melibatkan beberapa tanggung jawab untuk melakukan pengetesan pada beberapa *environment* yakni; *iOS, Android, Database* hingga *Service*. Alur kerja sebagai *Quality Assurance Intern* dimulai dari partisipasi tim teknis termasuk QA dalam kegiatan *Sprint Planning Meeting*, di mana pada tahap ini tim QA akan menerima berbagai dokumen seperti *System Design, Story List* serta tampilan *User Interface* dari berbagai tim teknis lainnya. Dokumen-dokumen tersebut akan menjadi acuan

utama dalam proses selanjutnya, yaitu penyusunan *scenario case* yang menggambarkan langkah-langkah untuk menguji fungsional maupun non-fungsional pada aplikasi.

Setelah *scenario case* telah disusun, tim QA akan mempertimbangkan apakah pengujian tersebut bersifat *repetitive* atau tidak. Jika proyek akan dilakukan pengujian secara berulang, biasanya tim QA akan melakukan pengujian secara otomatis (*automation*) menggunakan jenis *Regression Test*. Pengujian secara otomatis akan menggunakan *software* Appium dengan menggunakan bahasa pemrograman Java. Sebaliknya, jika proyek tidak akan dilakukan pengujian secara berulang, pengujian pada proyek tersebut akan dilaksanakan secara manual. Untuk pengujian secara manual, pengujian hanya perlu mengikuti tahap-tahap *scenario case* yang telah dibuat sebelumnya.

Saat melaksanakan pengujian atau *testing* baik manual maupun otomatis, tim QA akan mencatat semua temuan *bug* dalam bentuk tiket menggunakan layanan berbasis *website* bernama Jira Cloud. Tiket ini nantinya akan langsung tersampaikan kepada tim *developer* untuk dapat dilakukan perbaikan. Setelah proyek selesai, langkah terakhir yang dilakukan oleh QA adalah menyusun dokumentasi proyek sebagai catatan akhir dari proses pengujian.

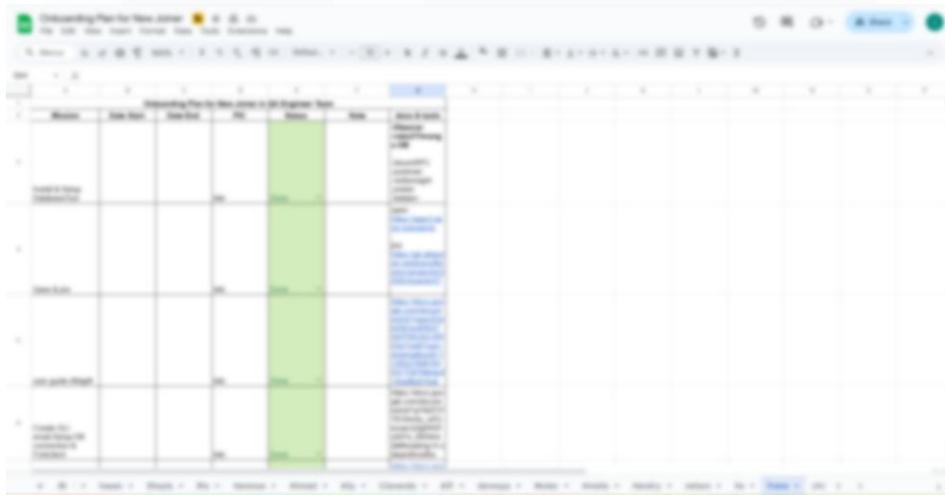
Adapun rincian mengenai periode kegiatan magang selama enam bulan disajikan dalam Tabel 3.1, yang menggambarkan periode pelaksanaan proyek dan tugas yang telah dilaksanakan di PT. Global Loyalty Indonesia.

Tabel 3.1 Waktu Pelaksanaan dan Kegiatan Kerja Magang

Kegiatan	Mulai	Selesai	Minggu
Onboarding New Plan for Joiner			
<i>Install and setup database tools</i>	3 Februari 2025	4 Februari 2025	1
<i>Introduction to Qase and Jira</i>	4 Februari 2025	4 Februari 2025	1
<i>Rollout Cycle test</i>	4 Februari 2025	5 Februari 2025	1
Revamp Voucher and Redeem A-Poin			
<i>Scenario testing Back Office Alfagift-Promotion Campaign and Reward</i>	11 Februari 2025	17 Februari 2025	2, 3
<i>Membuat scenario test Tooltip A-Poin page, Pre-Checkout, Back Office Dashboard Order Redemption</i>	3 Maret 2025	4 Maret 2025	1
<i>Testing scenario Tooltip A-Poin page, Pre-Checkout, Back Office Dashboard Order Redemption</i>	4 Maret 2025	14 Mei 2025	1, 2, 3, 4
Full Cycle Rollout IOS			
<i>Full Cycle Rollout iOS V. 4.37</i>	12 Februari 2025	14 Februari 2025	2
<i>Full Cycle Rollout iOS V. 4.38</i>	15 April 2025	17 April 2025	3
<i>Full Cycle Rollout iOS V. 4.39</i>	22 Mei 2025	23 Mei 2025	3
<i>Full Cycle Rollout iOS V. 4.40</i>	17 Juni 2025	19 Juni 2025	3
Dark Store Sebagai Hub Tag I APO			
<i>Membuat scenario test List History Pesanan, Detail History Pesanan, Filter & Search History, Monitoring SAPA</i>	21 April 2025	6 Mei 2025	4
<i>Testing scenario Detail History Pesanan, Order List & Order Detail Listing Order, POD Pickup</i>	6 Mei 2025	17 Juni 2025	1, 2, 3,4

3.2.1 Onboarding Plan for New Joiner

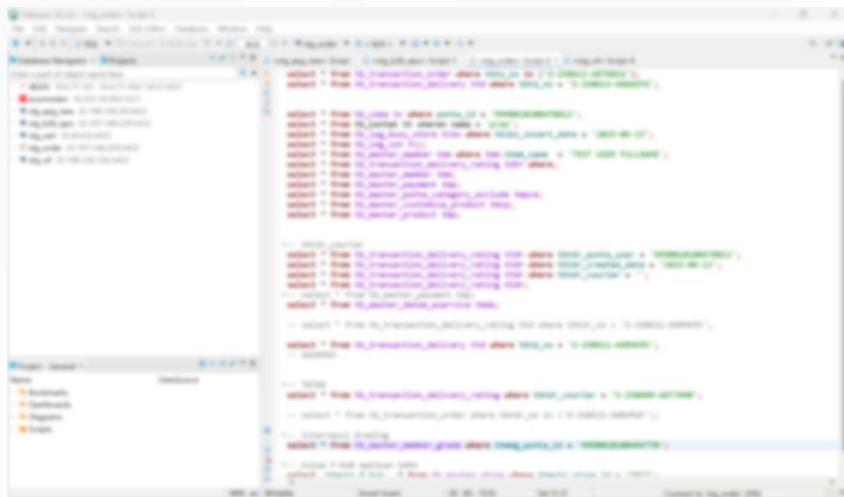
Pada hari pertama masa kerja magang di PT. GLI, terdapat sebuah pelatihan terlebih dahulu yang diberikan kepada para peserta magang sebagai permulaan untuk kedepannya sebagai *Quality Assurance Intern*. Hal-hal yang harus dipelajari telah dirangkap di dalam sebuah *google sheets* bernama *Onboarding Plan for New Joiner* seperti yang ditampilkan pada Gambar 3.4.



Gambar 3.4 Onboarding Plan for New Joiner

Pada pelatihan *Onboarding Plan for New Joiner*, para peserta magang dibekali dengan berbagai pengetahuan dan keterampilan dasar yang penting untuk menunjang kelancaran mereka selama masa kerja. Salah satu bagian utama dari pelatihan ini adalah pengenalan serta praktik langsung dalam melakukan instalasi dan setup berbagai tools yang akan digunakan sehari-hari dalam pengujian dan pengembangan aplikasi. Beberapa *tools* penting yang diperkenalkan antara lain DBeaver, MongoDB, Swagger, dan Postman. Masing-masing tools tersebut memiliki fungsi dan kegunaan yang berbeda namun saling melengkapi. DBeaver digunakan sebagai antarmuka grafis untuk mengakses dan mengelola berbagai jenis database secara efisien. MongoDB, sebagai salah satu database NoSQL populer, memberikan fleksibilitas dalam menyimpan dan mengelola data dalam format dokumen. Swagger digunakan untuk mendokumentasikan, menguji, dan memahami *endpoint-endpoint* API

yang tersedia dalam sebuah sistem. Sementara itu, Postman menjadi alat bantu utama dalam proses pengujian API secara manual, termasuk untuk mengirim permintaan (*request*) dan menerima respons (*response*) dari *server*. Melalui pelatihan ini, para peserta diharapkan tidak hanya memahami fungsi masing-masing *tools*, tetapi juga mampu mengintegrasikan penggunaannya dalam aktivitas kerja harian, seperti debugging, pengembangan fitur, dan validasi alur data antara aplikasi dan server.

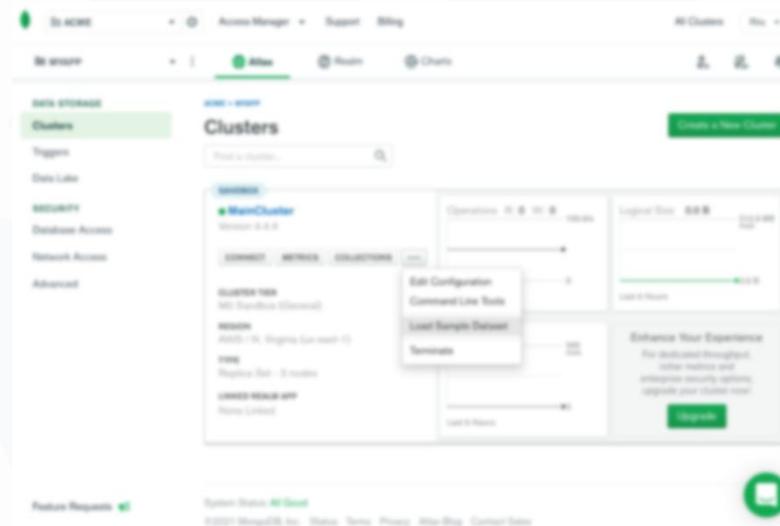


Gambar 3.5 DBeaver

Gambar 3.5 merupakan tampilan dari perangkat lunak DBeaver, yaitu sebuah *universal database tool* yang digunakan untuk mengakses dan mengelola berbagai macam jenis *database* secara efisien. DBeaver mendukung koneksi ke berbagai jenis sistem manajemen basis data (DBMS) populer, seperti MySQL, PostgreSQL, SQLite, Oracle, SQL Server, dan banyak lagi, sehingga menjadikannya alat yang fleksibel dan sangat bermanfaat dalam pengelolaan data lintas platform.

DBeaver biasanya digunakan oleh tim QA sebagai alat bantu untuk melakukan operasi CRUD (*Create, Read, Update, Delete*). Operasi ini dilakukan untuk mendukung proses pengujian, misalnya dengan menambahkan data uji (*test data*), membaca atau memverifikasi data pesanan aplikasi Alfacift. Dengan menggunakan DBeaver, tim QA dapat berinteraksi

langsung dengan database secara visual dan terstruktur, tanpa harus menggunakan *command line*, sehingga pekerjaan menjadi lebih cepat dan mudah dipantau.



Gambar 3.6 MongoDB

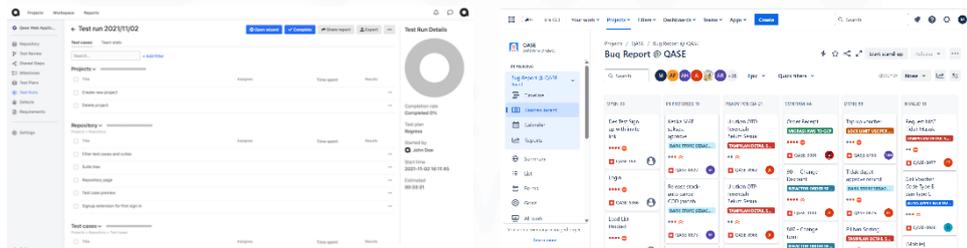
Gambar 3.6 merupakan tampilan dari *software* MongoDB, yang dirancang untuk memudahkan akses dan pengelolaan database berjenis NoSQL milik MongoDB. MongoDB sendiri merupakan sistem manajemen basis data yang berbasis dokumen, yang menyimpan data dalam format JSON-like (BSON), sehingga sangat fleksibel dalam menangani data tidak terstruktur atau semi-terstruktur. Antarmuka grafis ini memungkinkan tim QA untuk melakukan eksplorasi data, membuat query, serta mengelola koleksi dan dokumen dengan cara yang lebih intuitif dibandingkan penggunaan *Command Line Interface* (CLI).

Dalam proses pengujian aplikasi, khususnya aplikasi yang berbasis layanan digital seperti Alfagift, MongoDB sering kali digunakan oleh tim QA untuk melakukan proses verifikasi dan validasi terhadap data yang masuk ke dalam sistem. Tim QA dapat memastikan bahwa data yang dihasilkan oleh sistem backend sesuai dengan ekspektasi, baik dari segi struktur, isi, maupun integritasnya. Selain itu, MongoDB juga memungkinkan tim QA untuk melakukan operasi seperti pencarian data spesifik, filter berdasarkan parameter

tertentu, hingga pemeriksaan logis terhadap proses bisnis yang sedang diuji. Penggunaan MongoDB dengan antarmuka GUI seperti ini sangat membantu mempercepat proses pengujian karena pengguna tidak perlu menulis query kompleks secara manual.

Selain itu, tim QA juga memanfaatkan Swagger UI sebagai salah satu alat bantu dalam proses pengujian, khususnya dalam konteks pengujian *Application Programming Interface* (API). Swagger UI merupakan sebuah antarmuka interaktif berbasis web yang memungkinkan pengguna untuk melihat, mencoba, dan mengintervensi *endpoint-endpoint* API secara langsung berdasarkan dokumentasi yang tersedia. Penggunaan Swagger UI sangat membantu tim QA dalam melakukan pengujian terhadap fungsi *backend* secara lebih mendalam, tanpa harus melalui antarmuka pengguna (UI). Melalui Swagger UI, tim QA dapat melakukan berbagai permintaan (*request*) seperti GET, POST, PUT, dan DELETE ke *endpoint* API tertentu, dengan memasukkan parameter yang sesuai. Hal ini memungkinkan tim untuk memverifikasi apakah data yang dikirim atau diterima sudah sesuai dengan ekspektasi sistem, serta mengidentifikasi jika terdapat *error* atau ketidaksesuaian pada respons yang diterima.

Setelah itu, peserta magang juga melakukan sebuah sesi pengenalan singkat terkait *platform* yang akan digunakan untuk tim QA, yakni Qase sebagai tempat / *repository* penyimpanan *scenario test* dan juga *testing scenario* dan juga Jira Cloud sebagai *platform* untuk melaporkan *bug* kepada developer, seperti yang tertampil pada Gambar 3.7.



Gambar 3.7 Gambar Qase (Kiri) dan Jira Cloud (Kanan)

Di akhir pelatihan *onboarding*, peserta magang juga diberikan pengenalan mengenai proses testing aplikasi Alfagift melalui metode *Full Cycle Rollout Test*. Proses ini merupakan bagian penting dari tahapan QA sebelum versi aplikasi dirilis secara resmi ke pengguna melalui Play Store maupun App Store. Dalam sesi ini, dijelaskan bahwa *Full Cycle Rollout Test* dilakukan secara *manual* untuk memastikan bahwa semua fitur utama yang ada di dalam aplikasi Alfagift dapat berjalan dengan lancar, stabil, dan sesuai dengan kebutuhan pengguna, terutama setelah dilakukan perubahan atau pembaruan pada versi terbaru aplikasi. Pengujian ini mencakup berbagai macam modul penting yang merupakan bagian dari core system aplikasi, seperti *Welcome Screen*, *Login*, *Register*, *Homepage*, *Shopping*, *Promo*, *Order Summary*, *Basket*, hingga halaman *Thank You Page (TYP)*. Setiap modul diuji satu per satu menggunakan skenario positif (*positive case*) untuk memastikan bahwa fungsi berjalan normal sesuai alur yang seharusnya. *Full Cycle Rollout Test* dilakukan pada versi Play Store atau App Store aplikasi Alfagift sebagai bahan percobaan dan *training*. Seperti yang terlampir pada Gambar 3.8.

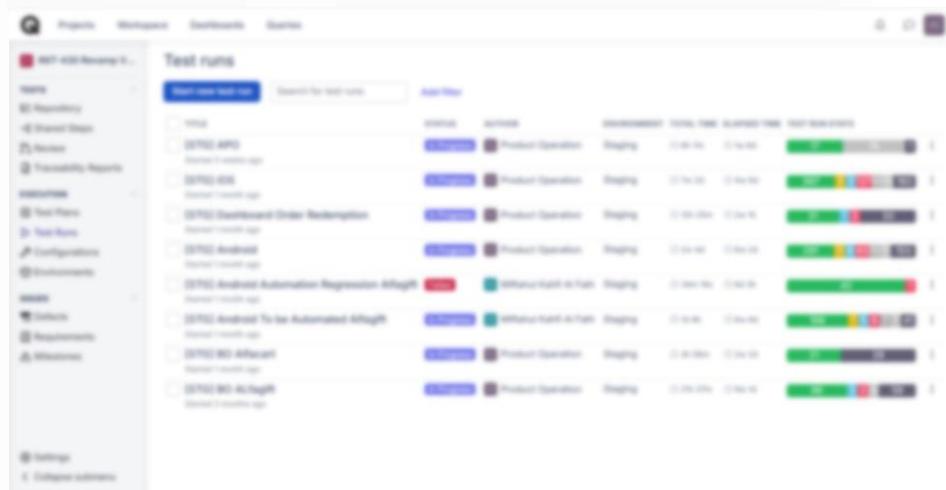
Gambar 3.8 Full Cycle Rollout Test

3.2.2 Revamp Voucher dan Redeem

Proyek pertama yang dikerjakan adalah proyek aplikasi Alfagift *Revamp Voucher dan Redeem*. Proyek ini bertujuan untuk memperbarui tampilan antarmuka (*user interface*) pada halaman *Voucher* dan *A-Poin* pada aplikasi Alfagift. Selain itu, proyek ini juga bertujuan untuk menyediakan lebih banyak

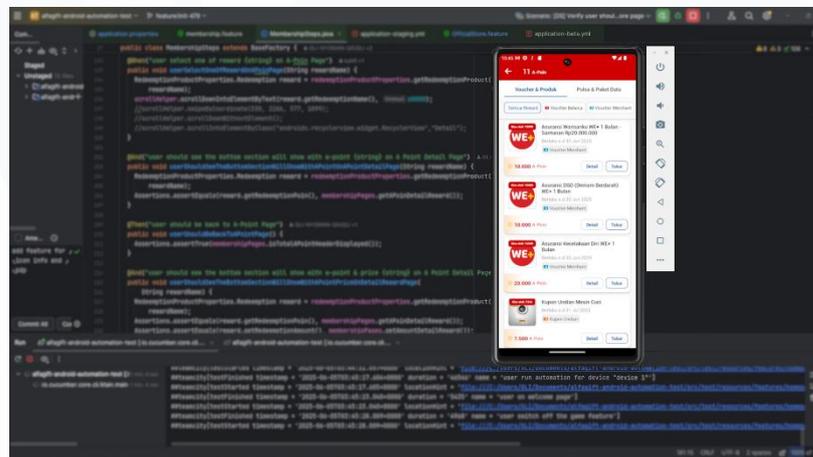
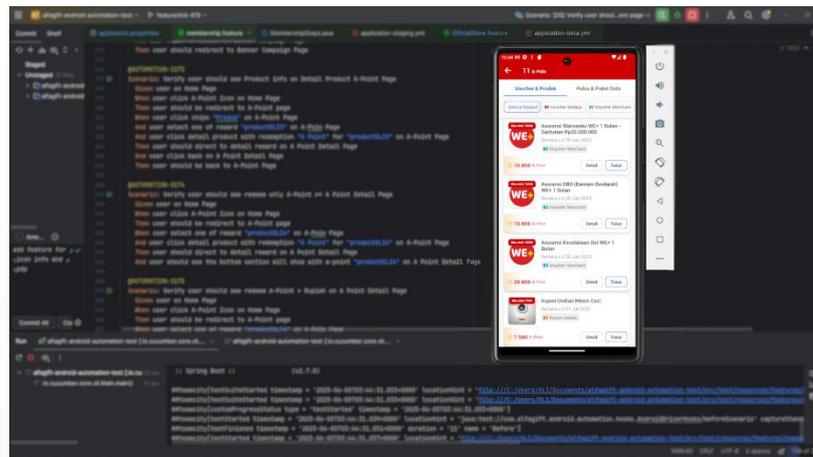
variasi produk dari produk Alfamart, produk GLI hingga produk Digital yang dapat ditukarkan menggunakan A-Poin.

Dalam proyek ini, tim QA berperan untuk memastikan bahwa seluruh sistem baik dari segi fungsionalitas maupun non-fungsionalitas dapat berjalan dengan spesifikasi yang telah ditentukan. Proses pengujian mencakup identifikasi dan pelaporan *bug* secara fungsional, serta pengujian regresi untuk memastikan perubahan tidak merusak atau menyenggol fitur lama. Dengan demikian, tim QA dapat membantu menjaga kualitas aplikasi secara menyeluruh sebelum fitur dirilis ke *user*.



Gambar 3.9 Testrun Revamp Voucher dan Redeem

Gambar 3.9 merupakan tampilan dari *testrun* dari proyek *Revamp Voucher dan Redeem*. *Testrun* tersebut merupakan kumpulan dari *scenario test* yang sudah disusun secara teratur untuk segera dilakukan pengujian. *Testrun* tersebut terdiri dari beberapa *environment* yang nantinya akan dilakukan pengujian yakni *device* iOS dan Android, *database* hingga *Back Office*.



Gambar 3.10 Automation Testing on A-Poin Page

Pada proyek ini, tim QA memutuskan untuk melakukan dua jenis pengujian aplikasi yakni *manual* dan *automation*. *Scenario test* yang dilakukan dalam proses *testing automation* merupakan *scenario repetitive* atau *scenario* pengulangan dari proyek sebelumnya. Tampilan aplikasi Appium dapat dilihat pada Gambar 3.10 dimana disitu dilakukan pengetesan secara *automation* untuk fitur halaman A-Poin khususnya untuk *device* Android. Halaman A-Poin merupakan salah satu fitur yang memungkinkan *user* dapat menukar A-Poin mereka terhadap berbagai macam produk yang ditawarkan mulai dari produk GLI, produk reguler hingga produk digital.



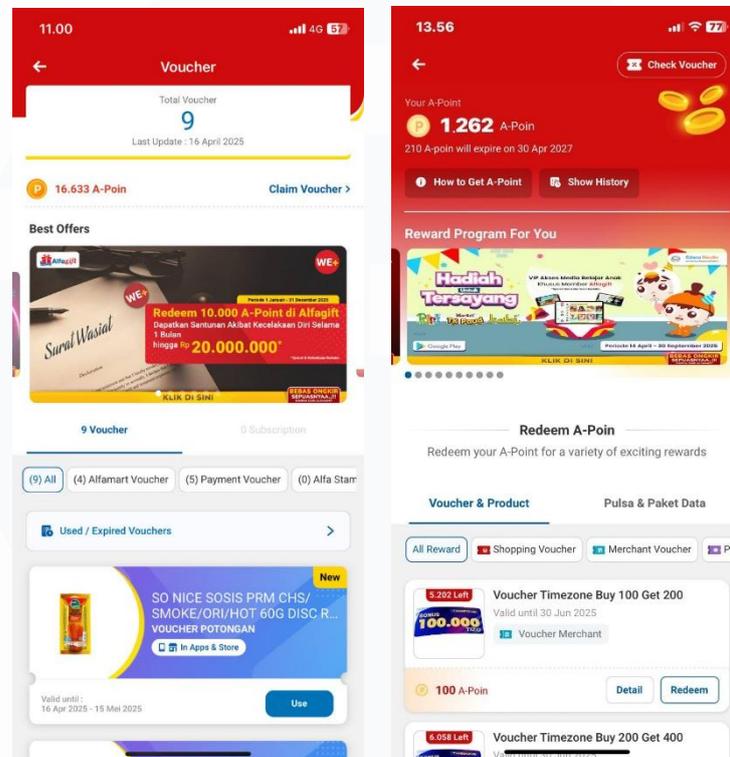
Gambar 3.11 Emulator Automation Testing A-Poin Page

Dapat dilihat pada Gambar 3.11 bahwa modul yang akan di uji menggunakan *automation testing* adalah Halaman A-Poin. Berikut beberapa elemen yang akan dicari:

- Cek *Filter Chips*
- Cek “Detail” *Button*
- Cek “Redeem” *Button*

Dalam skenario pengujian yang ditulis dalam Bahasa Gherkin dengan konsep BDD (*Behavior Driven Development*) yang terlampir pada Gambar 3.10. Tim QA mensimulasikan beberapa elemen yang akan diuji, antara lain *filter chips* sebagai kategori penyaring jenis *reward* (misalnya: *All Reward*, *Shopping Voucher*, *Merchant Voucher*, *Product*, dll), tombol “Detail” untuk memverifikasi informasi produk, termasuk nama produk, masa berlaku, harga A-Poin serta stok. Kemudian tombol “Redeem” yang berfungsi untuk menukarkan A-Poin dengan *reward* tersebut. Setiap proses pengujian *automation* yang selesai, sistem akan menghasilkan *report* yang menunjukkan

success rate yang menandakan beberapa skenario yang berhasil dijalankan dan yang gagal dijalankan.



Gambar 3. 12 Halaman A-Poin Versi Lama (Kiri) dan Versi Revamp (Kanan)

Gambar 3.12 menunjukkan perbedaan antarmuka pengguna (*user interface*) pada aplikasi Alfagift khususnya untuk fitur halaman A-Poin. Perbandingan antara tampilan versi lama dan versi *revamp* menunjukkan beberapa perbedaan yang signifikan baik itu dari segi visual maupun segi fungsi. Dari segi tampilan visual, versi *revamp* menampilkan desain antarmuka yang lebih interaktif dan dinamis. Sedangkan dari segi fungsi, halaman A-Poin menampilkan beberapa pengembangan fitur seperti jenis *reward* yang telah diklasifikasikan ke dalam tiga kategori yaitu:

- Produk GLI

Produk GLI merupakan produk B2B atau produk perusahaan lain yang dijual di aplikasi Alfagift.

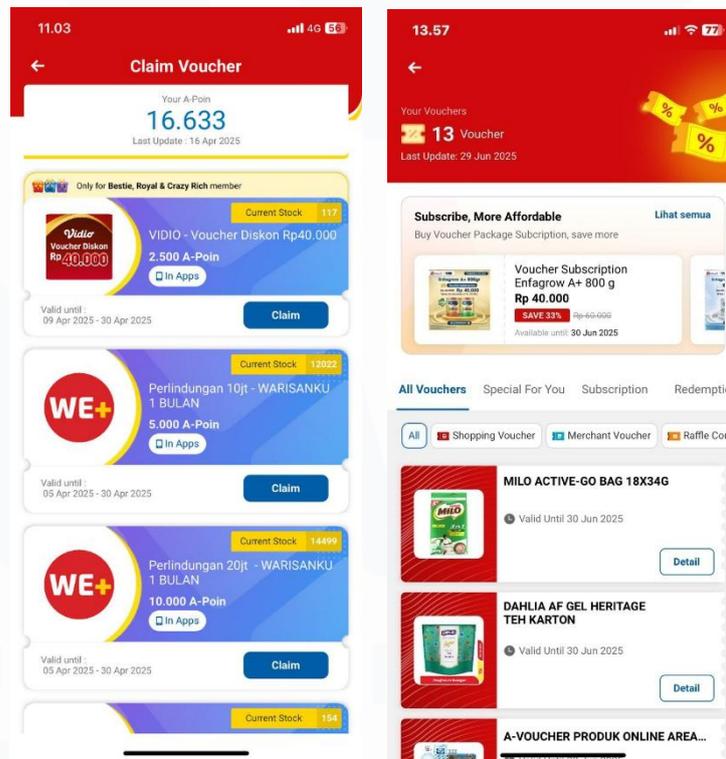
- Produk Reguler

Produk Reguler merupakan produk Alfamart yang dijual di aplikasi Alfagift.

- Produk Digital

Produk Digital merupakan produk yang dijual dalam bentuk digital seperti *voucher* pulsa, *voucher* makanan, dll.

Selain itu, halaman A-Poin juga menampilkan beberapa fitur baru seperti informasi mengenai apa itu A-Poin serta bagaimana cara mendapatkan A-Poin dalam tombol *How to Get A-Point*. Kemudian, pengguna juga dapat melihat riwayat transaksi A-Poin yang dilakukan melalui tombol *Show History*. Dengan adanya pembaruan antarmuka dan fitur ini, memungkinkan para pengguna agar lebih nyaman, efisien dalam menggunakan fitur A-Poin di aplikasi Alfagift.



Gambar 3.13 Halaman Voucher Versi Lama (Kiri) dan Versi Revamp (Kanan)

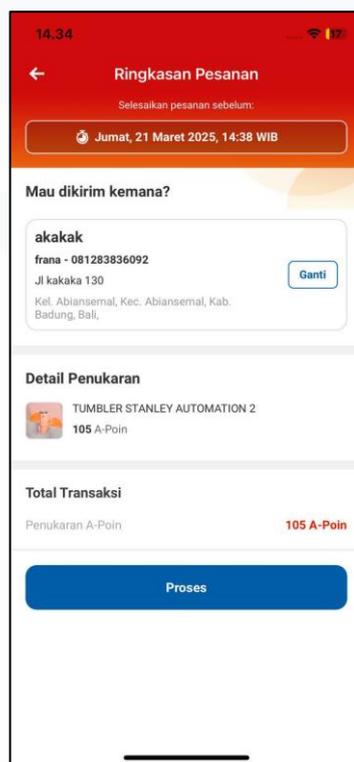
Halaman *check voucher* pada Gambar 3.13 juga mengalami perubahan tampilan visual. Pada versi lama, tampilan halaman berfokus pada penukaran A-Poin dengan menunjukkan jumlah poin yang dimiliki pengguna secara dominan di bagian atas, lengkap dengan informasi pembaruan terakhir. Setiap *voucher* ditampilkan dalam format kotak sederhana berwarna biru, dilengkapi dengan deskripsi jumlah A-Poin yang diperlukan, masa berlaku, serta tombol klaim. Desain ini bersifat informatif, namun relatif monoton dan kurang menarik secara visual. Sementara itu, pada versi *revamp* desain mengalami perubahan yang lebih dinamis dan informatif dengan pemilihan warna dan tata letak yang lebih menarik. Selain itu, *voucher* dikategorikan dengan lebih baik menggunakan beberapa *chips* seperti *All Vouchers*, *Special for You*, *Subscription* dan *Redemption*.



Gambar 3.14 Redeem Produk Reguler

Produk reguler yang tersedia pada aplikasi Alfagift dapat ditukar oleh pengguna dengan dua metode penukaran, yaitu penukaran menggunakan A-Poin secara penuh dan penukaran menggunakan kombinasi antara A-Poin dan

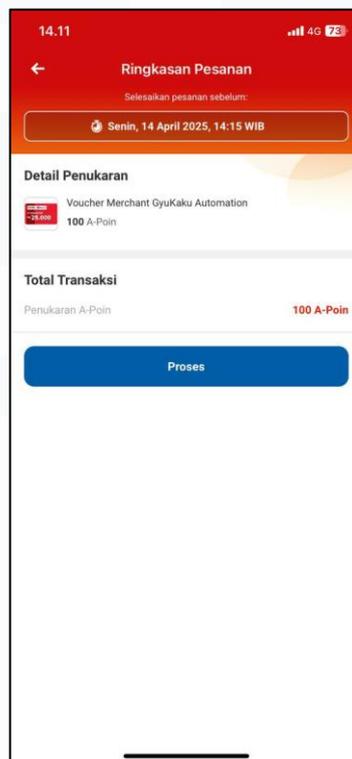
Rupiah. Mekanisme ini memberikan fleksibilitas lebih kepada pengguna, khususnya ketika jumlah A-Poin yang dimiliki tidak mencukupi untuk menukarkan produk secara penuh. Selain itu, produk reguler ini juga dapat dipilih untuk ditukar dengan dua jenis sistem pengiriman, yakni melalui layanan *Delivery* (pengiriman ke alamat pengguna) dan *Pickup* (pengambilan langsung di toko Alfamart yang dipilih). Pilihan sistem ini dapat disesuaikan dengan preferensi serta kebutuhan pengguna, baik dari segi waktu maupun kenyamanan. Fitur ini ditampilkan secara jelas dalam aplikasi, seperti yang diperlihatkan pada Gambar 3.14, dan menjadi salah satu bentuk kemudahan layanan yang ditawarkan oleh Alfagift dalam mendukung program loyalitas pelanggan.



Gambar 3.15 Redeem Produk GLI

Produk GLI yang tersedia pada aplikasi Alfagift juga dapat ditukar oleh pengguna melalui dua metode penukaran, yaitu penukaran menggunakan A-Poin secara penuh serta penukaran dengan kombinasi antara A-Poin dan Rupiah. Berbeda dengan produk reguler, produk GLI hanya dapat ditukar

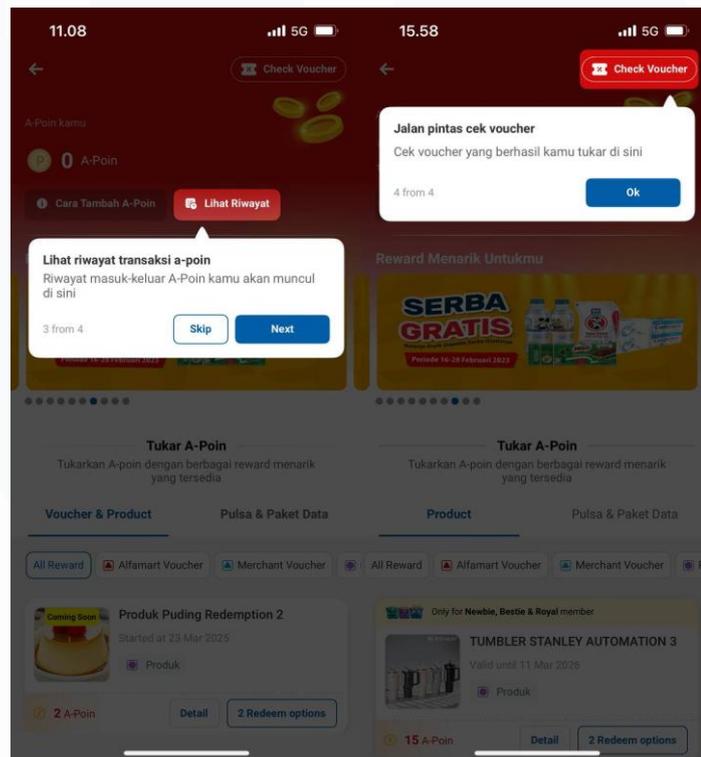
menggunakan satu jenis sistem pengiriman, yaitu melalui metode *Delivery* atau pengiriman langsung ke alamat tujuan pengguna. Dengan demikian, opsi pengambilan langsung di toko (*Pickup*) tidak tersedia untuk jenis produk ini. Informasi mengenai sistem penukaran dan metode pengiriman untuk produk GLI ditampilkan secara jelas dalam aplikasi Alfagift, sebagaimana yang dapat dilihat pada Gambar 3.15. Hal ini bertujuan untuk memastikan transparansi serta memudahkan pengguna dalam melakukan transaksi penukaran poin dengan tepat sesuai kebijakan yang berlaku.



Gambar 3.16 Redeem Produk Digital

Produk Digital yang tersedia pada aplikasi Alfagift yang ditampilkan pada Gambar 3.16 juga dapat ditukar oleh pengguna melalui dua metode penukaran, yaitu menggunakan A-Poin secara penuh maupun dengan kombinasi antara A-Poin dan pembayaran dalam bentuk Rupiah. Berbeda dengan produk reguler maupun produk GLI, produk digital tidak memerlukan proses logistik fisik sehingga tidak menyediakan opsi sistem pengiriman seperti *Delivery* atau *Pickup*. Hal ini dikarenakan produk digital, seperti *voucher* elektronik, pulsa,

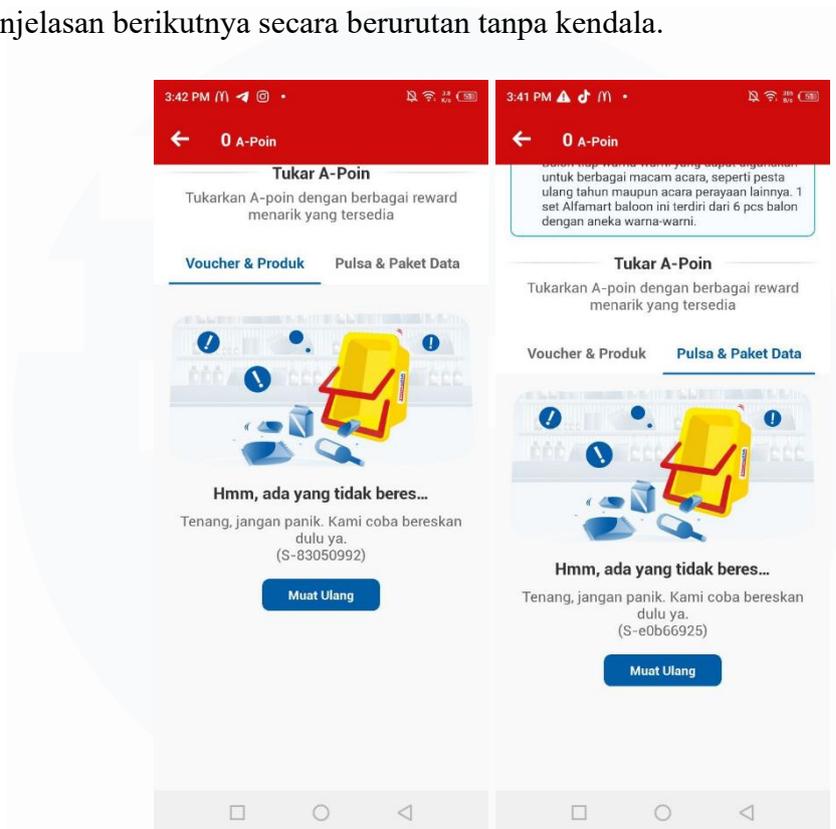
atau layanan streaming, dikirimkan secara otomatis dalam bentuk digital langsung ke akun pengguna atau melalui notifikasi aplikasi setelah transaksi berhasil. Oleh karena itu, pengguna tidak perlu menunggu pengiriman fisik maupun datang langsung ke toko untuk mengambil produk tersebut.



Gambar 3.17 *Tooltip A-Poin Page*

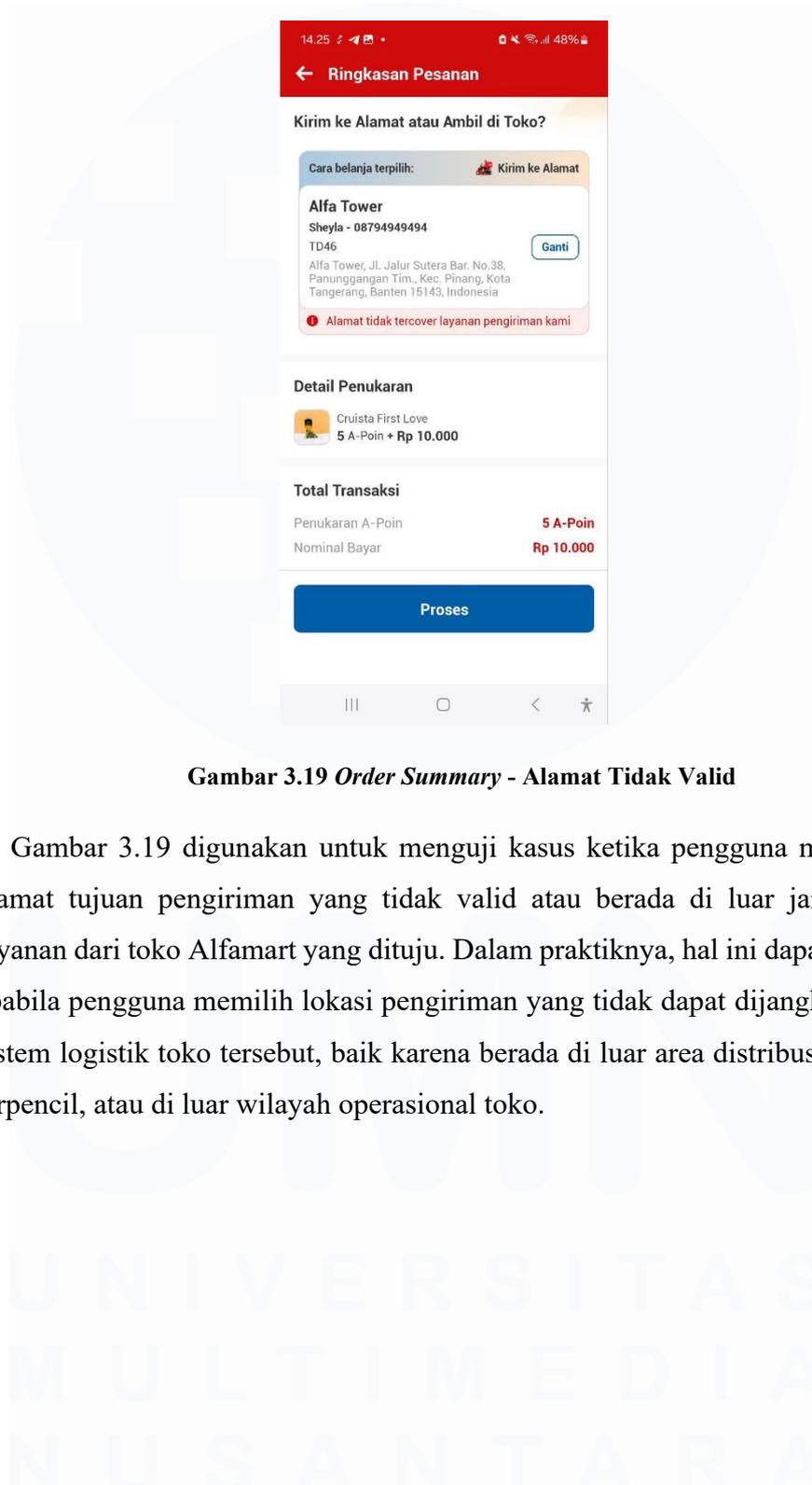
Tim QA juga melakukan pengujian terhadap fitur *tooltip* yang terdapat pada halaman A-Poin guna memastikan bahwa fungsionalitas dan informasi yang disampaikan kepada pengguna berjalan dengan baik dan sesuai harapan. Pengujian ini mencakup tiga aspek utama. Pertama, tim memverifikasi bahwa *wording* atau teks yang ditampilkan dalam tooltip sudah sesuai, jelas, dan informatif, khususnya dalam menjelaskan fungsi dari elemen yang sedang diperkenalkan kepada pengguna seperti pada Gambar 3.17 yang menampilkan penjelasan terkait riwayat transaksi A-Poin. Kedua, QA memastikan bahwa tombol "Skip" berfungsi dengan benar, yaitu dapat digunakan untuk melewati langkah penjelasan yang sedang ditampilkan dan langsung menuju akhir atau keluar dari rangkaian tooltip. Ketiga, tombol "Next" juga diuji untuk

memastikan bahwa saat ditekan, sistem dapat menampilkan langkah atau penjelasan berikutnya secara berurutan tanpa kendala.



Gambar 3.18 *E Services and Groceries*

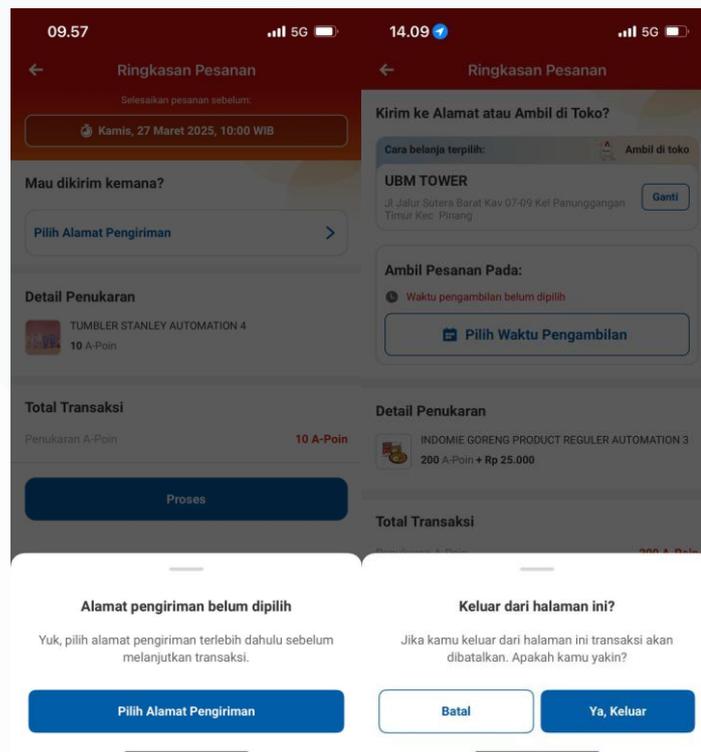
Pengujian pada Gambar 3.18 ini merupakan bagian dari skenario *negative test* yang dilakukan oleh tim QA sebagai upaya untuk memastikan sistem mampu menangani kondisi kegagalan layanan secara tepat. Dalam pelaksanaannya, tim QA melakukan koordinasi secara langsung dengan tim *service* untuk melakukan simulasi kondisi *error*, yakni dengan cara menonaktifkan atau mematikan *endpoint service* pada fitur tertentu secara sementara. Dengan begitu, tim QA dapat menguji bagaimana aplikasi merespons ketika terjadi gangguan pada sisi layanan *service*, seperti kegagalan dalam memuat data dari *server* atau ketidakterediaan respons dari API yang dibutuhkan. Melalui pendekatan ini, QA dapat mengevaluasi apakah tampilan sudah sesuai mampu menunjukkan informasi error dengan tepat, serta memastikan bahwa pesan (*wording*) yang ditampilkan kepada pengguna sudah sesuai.



Gambar 3.19 Order Summary - Alamat Tidak Valid

Gambar 3.19 digunakan untuk menguji kasus ketika pengguna mengatur alamat tujuan pengiriman yang tidak valid atau berada di luar jangkauan layanan dari toko Alfamart yang dituju. Dalam praktiknya, hal ini dapat terjadi apabila pengguna memilih lokasi pengiriman yang tidak dapat dijangkau oleh sistem logistik toko tersebut, baik karena berada di luar area distribusi, lokasi terpencil, atau di luar wilayah operasional toko.

Tim QA melakukan pengujian ini untuk memastikan bahwa sistem mampu menangani kondisi tersebut dengan tepat, tanpa menimbulkan kebingungan atau kesalahan lanjutan pada sisi pengguna. Pengujian difokuskan pada validasi terhadap munculnya pesan kesalahan (*inline error*) secara langsung di tampilan antarmuka pengguna, tepat pada bagian pengisian atau pemilihan alamat pengiriman.



Gambar 3.20 Handling Bottom Sheet

Pada Gambar 3.20 tim QA juga melakukan pengujian terhadap berbagai skenario *handling* lainnya untuk memastikan bahwa aplikasi dapat merespons tindakan pengguna dengan tepat dan memberikan arahan yang jelas dalam alur pemesanan. Salah satu skenario yang diuji adalah ketika pengguna menekan tombol “Proses” pada halaman *Order Summary*, namun belum memilih alamat pengiriman. Dalam kondisi ini, sistem diharapkan menampilkan *bottom sheet* secara otomatis, yang berfungsi untuk mengarahkan pengguna agar segera memilih atau menambahkan alamat pengiriman terlebih dahulu sebelum dapat melanjutkan ke tahap selanjutnya.

Selain itu, QA juga menguji skenario ketika pengguna menekan tombol “*Back*” dari halaman *Order Summary*. Dalam hal ini, sistem dirancang untuk menampilkan *bottom sheet* konfirmasi yang berfungsi sebagai mekanisme perlindungan terhadap tindakan yang tidak disengaja. Pengguna akan diberikan opsi untuk membatalkan dan tetap berada di halaman tersebut, atau melanjutkan keluar dari proses pemesanan. Dengan begitu, risiko kehilangan data atau pembatalan pesanan secara tidak sengaja dapat diminimalkan.

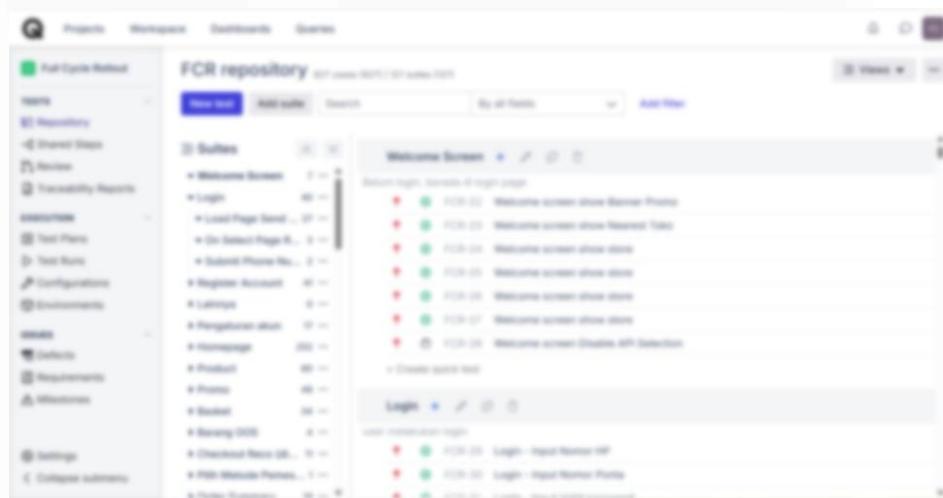
Pengujian ini bertujuan untuk memastikan bahwa sistem tidak hanya mengandalkan validasi di akhir proses, melainkan juga memberikan *guidance* secara *real-time* sehingga pengguna tidak mengalami kebingungan atau kesalahan proses transaksi. Kemunculan *bottom sheet* yang informatif ini menjadi bagian penting dari pengalaman pengguna yang intuitif dan ramah terhadap kesalahan input.

3.2.3 Full Cycle Rollout Alfagift

Full Cycle Rollout Alfagift merupakan pengujian terhadap aplikasi Alfagift yang dilakukan ketika aplikasi akan merilis sebuah versi baru ke dalam Play Store atau App Store. *Full Cycle Rollout* biasanya hanya terdiri dari *positive scenario test* untuk memastikan semua fitur dapat berjalan dengan lancar dan baik di *device* iOS maupun Android. *Full Cycle Rollout* biasanya dilakukan sebulan sekali dengan jenis pengetesan regresi yang dimana hanya dilakukan pengetesan terhadap modul atau fitur yang tidak mengalami perubahan di versi terbaru atau modul yang akan tetap selalu sama fungsinya.

Full Cycle Rollout Alfagift merupakan proses pengujian menyeluruh terhadap aplikasi Alfagift yang dilakukan secara rutin menjelang perilisasi versi baru aplikasi ke dalam *platform* distribusi resmi seperti Google Play Store dan App Store iOS. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa semua fitur yang terdapat dalam aplikasi dapat berjalan dengan baik, lancar, dan tanpa kendala pada kedua jenis perangkat, baik itu berbasis sistem operasi Android maupun iOS. Pengujian ini biasanya difokuskan pada *positive*

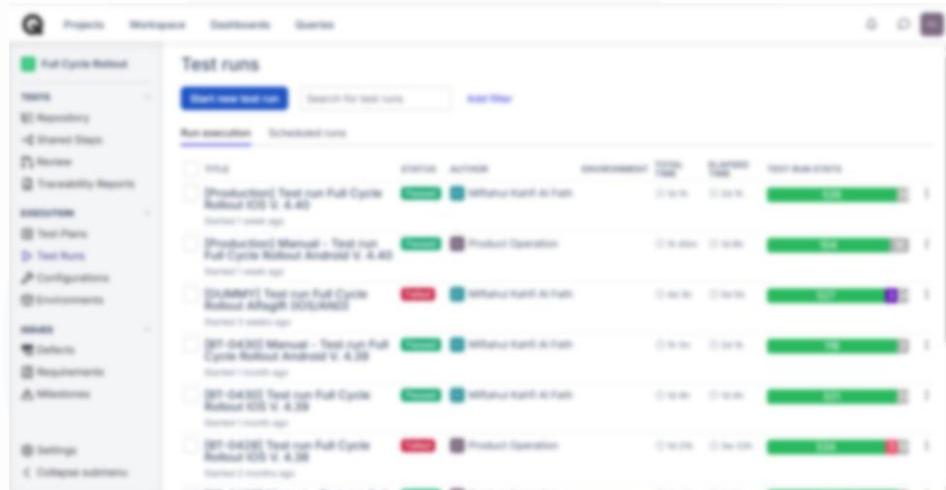
scenario test, yaitu skenario uji di mana pengguna menjalankan fungsi aplikasi sesuai alur penggunaan normal. *Full Cycle Rollout* umumnya dilaksanakan setiap satu bulan sekali, mengikuti jadwal rilis versi terbaru aplikasi. Jenis pengujian yang dilakukan lebih mengarah pada *regression testing*, yakni pengujian ulang terhadap fitur atau modul yang tidak mengalami perubahan signifikan pada versi terbaru. Tujuannya adalah untuk memastikan bahwa fungsionalitas dasar dari modul-modul tersebut tetap stabil dan tidak terpengaruh oleh perubahan kode atau pembaruan lain yang terjadi di versi tersebut.



Gambar 3.21 Full Cycle Rollout Scenario Test

Gambar 3.21 merupakan tampilan dari *repository scenario test* yang digunakan khusus untuk proses *Full Cycle Rollout* aplikasi Alfagift. *Repository* ini terdiri dari sejumlah modul utama yang menjadi fokus pengujian, antara lain: *Welcome Screen*, *Login*, *Register*, *Homepage*, *Product*, *Promo*, *Basket*, *Order Summary*, *Thank You Page (TYP)*, dan beberapa modul penting lainnya. Modul-modul ini dikategorikan sebagai modul inti yang bersifat statis, dalam arti umumnya tidak mengalami perubahan atau pembaruan fungsional secara signifikan ketika Alfagift melakukan rilis versi terbaru ke Play Store maupun App Store. Oleh karena itu, skenario pengujian terhadap modul-modul ini

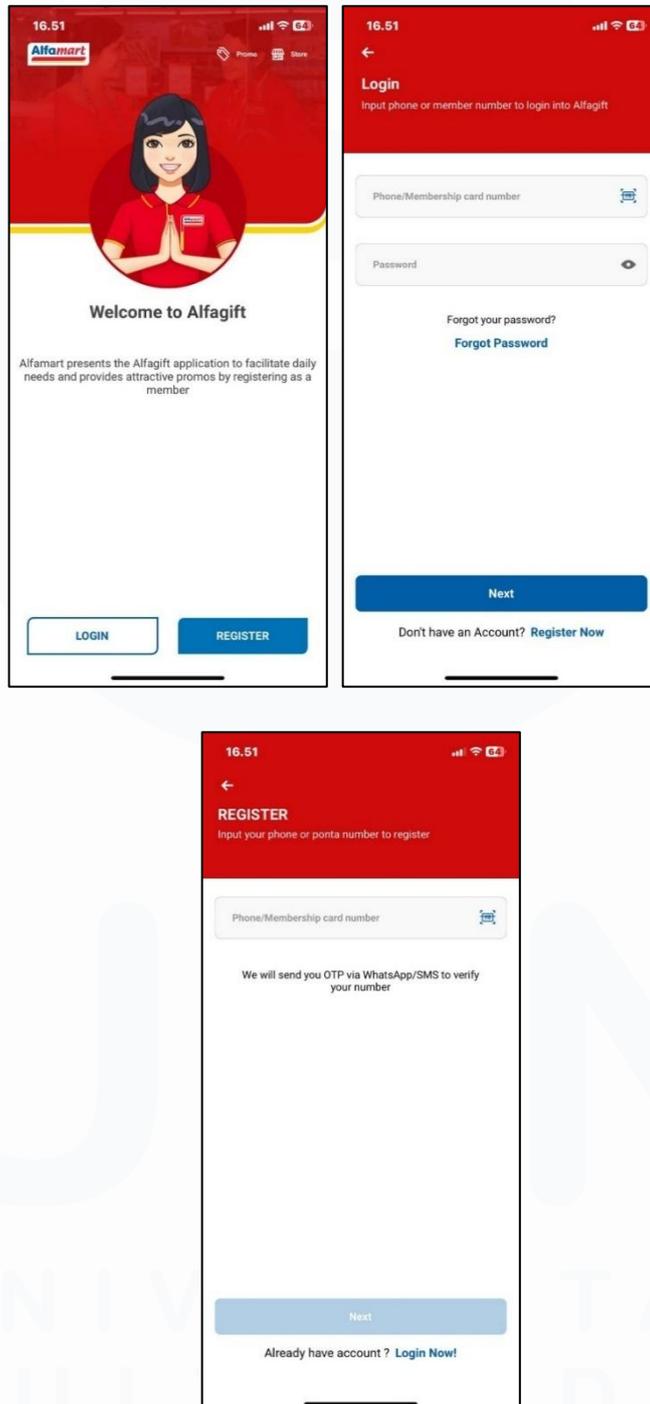
disusun secara permanen dan menjadi bagian dari *regression test* yang wajib dilakukan pada setiap *Full Cycle Rollout*.



Gambar 3.22 Testrun Full Cycle Rollout Alfagift

Ketika melakukan *Rollout*, tim QA biasanya tidak perlu membuat ulang skenario pengujian dari awal. Sebaliknya, mereka cukup melakukan proses *import* atau pengambilan ulang dari kumpulan *scenario test* yang sebelumnya telah disusun dan tersimpan secara sistematis di dalam *repository*. Setelah skenario tersebut berhasil di-import, tim QA akan memasukkannya ke dalam *testrun* yang sedang dijalankan seperti yang tertampil di Gambar 3.22. Pemilihan skenario pengujian disesuaikan dengan kebutuhan pengujian pada versi aplikasi yang akan dirilis, apakah mencakup seluruh modul, atau hanya modul-modul tertentu yang relevan. Proses ini memungkinkan pelaksanaan

pengujian dilakukan dengan lebih cepat, efisien, dan terorganisir karena semua skenario telah terdokumentasi dan terstruktur dengan baik.



Gambar 3.23 Welcome Screen Alfagift

Gambar 3.23 menampilkan contoh pengujian pada modul *Welcome Screen*, *Login* dan *Register*. Pengujian fitur login mencakup berbagai skenario seperti login dengan data valid, login dengan data tidak valid, pengisian *field* kosong, serta respon aplikasi terhadap kredensial yang salah. Tujuannya adalah untuk memastikan bahwa sistem dapat secara akurat mengenali akun pengguna, menjaga keamanan data, dan memberikan *feedback* yang jelas saat terjadi kesalahan. Sementara itu, pada fitur register, QA akan memastikan bahwa proses pendaftaran akun baru dapat dilakukan dengan mulus, mulai dari pengisian data diri, validasi nomor telepon/email, hingga penerimaan OTP (*One Time Password*) dan aktivasi akun.

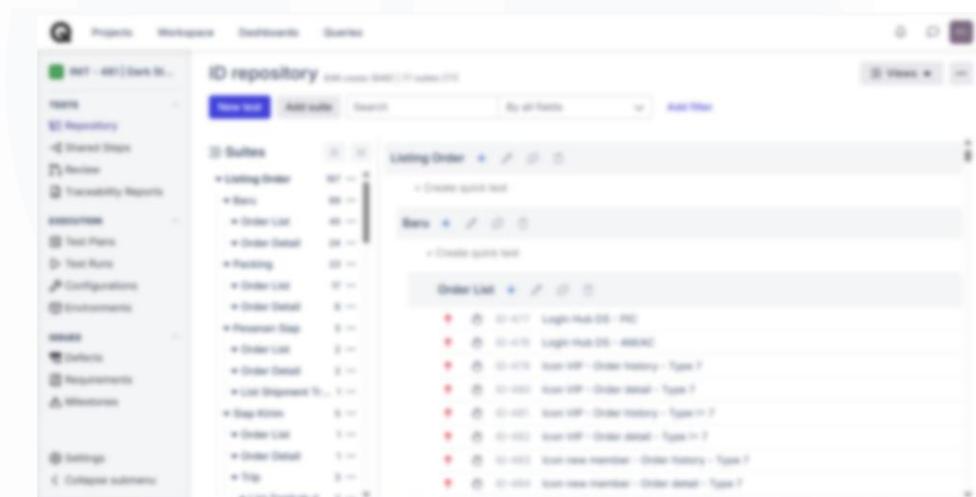
Karena fitur *login* dan *register* bersifat fundamental dan tidak berubah dari satu versi ke versi lain, maka pengujiannya hampir selalu dilakukan dalam setiap siklus Full Cycle Rollout. Dengan memastikan kedua fitur ini berjalan tanpa kendala di berbagai jenis perangkat dan versi sistem operasi (Android maupun iOS), tim QA dapat memberikan jaminan kualitas awal bahwa pengguna akan dapat mengakses aplikasi tanpa hambatan. Ini menjadi dasar yang penting sebelum melanjutkan pengujian terhadap modul-modul lanjutan lainnya dalam Alfagift.

3.2.4 Dark Store Sebagai Hub Tag I APO

Proyek *Dark Store Hub Tag I* sebenarnya merupakan proyek yang berfokus pada pengembangan aplikasi internal milik Alfamart yakni, Aplikasi Pesanan Online (APO). APO merupakan platform internal yang digunakan oleh kurir SAPA Alfamart dalam menjalankan tugasnya mengantarkan pesanan pelanggan.

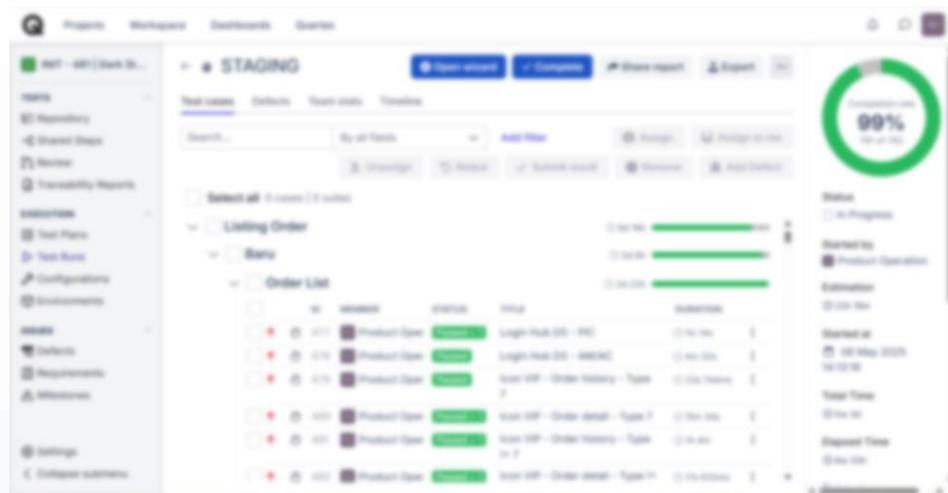
Layanan SAPA atau Siap Antar Pesanan Anda merupakan layanan pesan antar resmi dari Alfamart yang memungkinkan pelanggan memesan produk dari toko Alfamart dan mendapatkan pengiriman langsung ke alamat tujuan. Melalui program SAPA ini, Alfamart berupaya memberikan kemudahan dan kenyamanan dalam berbelanja tanpa harus datang langsung ke toko fisik.

Dalam proyek *Dark Store sebagai Hub Tag I APO* ini, fokus utamanya adalah optimalisasi alur kerja pengantaran melalui sistem APO, termasuk integrasi titik distribusi yang disebut *Dark Store* sebagai hub atau pusat pengiriman yang ditugaskan dalam sistem. Kurir SAPA menggunakan aplikasi APO untuk menerima, melihat detail pesanan, serta memproses dan mencatat status pengantaran, sehingga pengelolaan pengiriman menjadi lebih efisien dan terkontrol lebih cepat.



Gambar 3.24 Scenario Test Dark Store Sebagai Hub Tag I APO

Gambar 3.24 merupakan tampilan dari *repository scenario test* yang digunakan dalam proyek ini, di mana *repository* tersebut berisi total 846 *test case* yang mencakup skenario pengujian *positive case* maupun *negative case*. Seluruh *test case* yang terdapat dalam *repository* ini disusun secara sistematis dan menyeluruh berdasarkan *story list* yang telah disepakati sebelumnya pada tahap *sprint planning*. *Story list* tersebut berisi daftar kebutuhan dan fungsionalitas yang akan dikembangkan dalam siklus sprint tertentu, sehingga menjadi acuan utama dalam menyusun *scenario test*.



Gambar 3.25 Testrun Dark Store Sebagai Hub Tag I APO

Gambar 3.25 merupakan tampilan dari *testrun* yang digunakan dalam proses pengujian aplikasi APO (Aplikasi Pesanan Online). *Testrun* ini merepresentasikan keseluruhan skenario pengujian yang telah disusun dan dijalankan oleh tim QA untuk memastikan bahwa seluruh fitur dan alur kerja dalam aplikasi APO berjalan dengan baik sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Pada proses ini, setiap *test case* diuji satu per satu untuk memverifikasi fungsionalitas, performa, integritas data, serta kompatibilitas aplikasi, khususnya untuk mendukung operasional para kurir SAPA Alfamart dalam melakukan pengantaran barang pesanan pelanggan.

Dalam proyek ini, tim QA baru saja mendapatkan status QA *Pass* pada tanggal 18 Juni 2025. Status ini menandakan bahwa aplikasi telah berhasil melewati tahapan pengujian *staging*. Meski telah mendapat QA *Pass*, aplikasi ini belum memasuki tahap perilisan resmi ke pengguna internal. Hal ini dikarenakan masih adanya proses lanjutan yakni pengujian ke tahap *beta* oleh tim *Test Operation*.



Gambar 3.26 Tampilan Aplikasi Pesanan Online (APO)

Gambar 3.26 merupakan tampilan antarmuka Aplikasi Pesanan Online (APO), di aplikasi tersebut kurir SAPA biasanya menggunakan untuk mengakses informasi terkait pesanan-pesanan yang harus dikirimkan kepada pelanggan. Dalam tampilan *homepage* aplikasi ini, kurir dapat melihat status terkini dari pesanan seperti jumlah pesanan baru, pesanan dalam proses *packing*, hingga selesai. Selain fungsionalitas utama untuk mengelola pengiriman, aplikasi APO juga dilengkapi dengan beberapa fitur penunjang seperti:

- History Pesanan

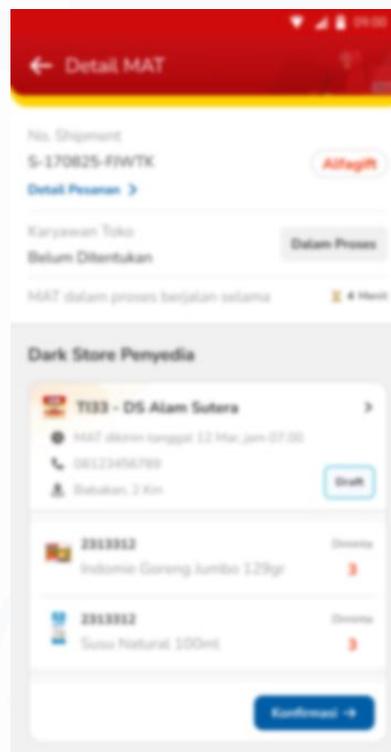
Fitur ini digunakan untuk melihat riwayat pengiriman yang telah dilakukan sebelumnya, sehingga memudahkan pelacakan dan dokumentasi aktivitas pengantaran.

- MAT (Mutasi Antar Toko)

Fitur ini berisi data Barang Masuk dan Barang Keluar yang berhubungan dengan proses perpindahan stok antar toko Alfamart. Fitur ini penting untuk mencatat dan memverifikasi transaksi logistik internal antar gerai.

Dalam mengakses APO, biasanya kurir akan menggunakan NIK dan *password* sehingga orang lain yang bukan kurir tidak akan bisa melakukan proses *login* ke APO walaupun aplikasi tersebut tersedia di layanan PlayStore. Terdapat juga beberapa *level user* yang dapat login di APO serta kekuasaannya dalam mengakses berbagai macam fitur di APO, yakni:

- AC (*Area Control*)
- AM (*Area Manager*)
- PIC (*Person in Charge*)

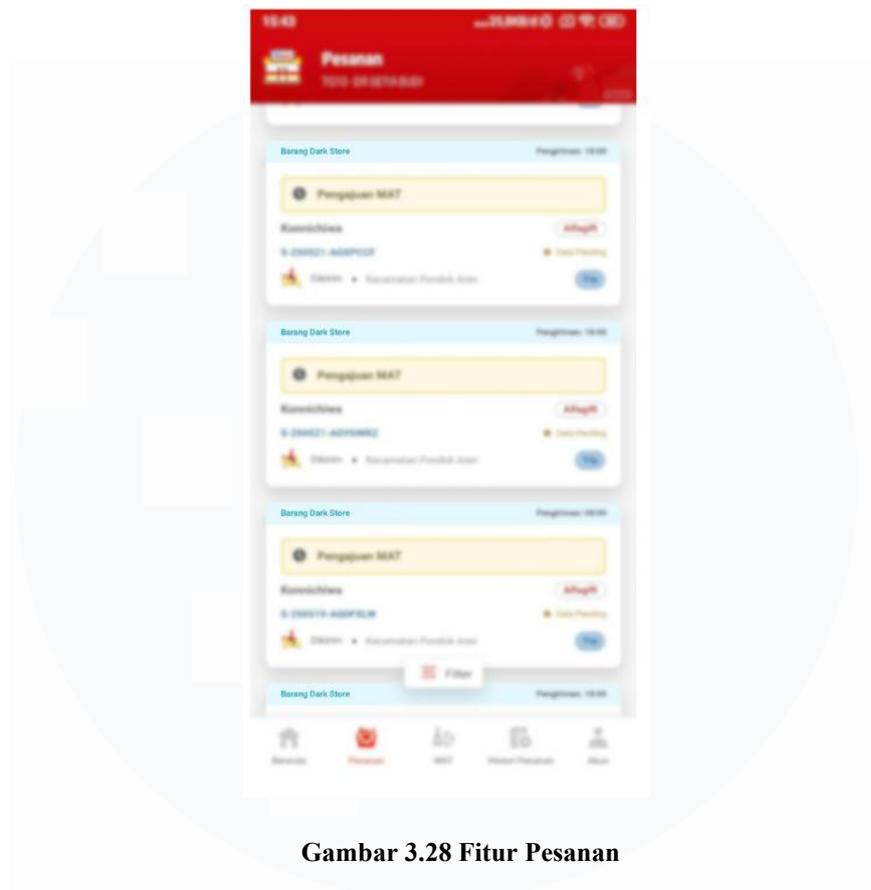


Gambar 3.27 Fitur MAT

Salah satu perbaruan fitur utama dalam proyek ini, seperti yang ditampilkan pada Gambar 3.27, adalah penyederhanaan proses MAT (Mutasi Antar Toko) khusus untuk barang *Dark Store*. Dalam versi sebelumnya, proses MAT mengharuskan staf toko untuk melakukan scan manual terhadap setiap barang yang akan dikirim hal ini memakan waktu, terutama jika jumlah barang cukup banyak atau proses dilakukan di jam sibuk.

Dengan pembaruan di proyek ini, sistem secara otomatis melakukan *generate* data barang MAT tanpa memerlukan proses scan satu per satu oleh staff toko. Begitu pesanan diterima dan diverifikasi dari *Dark Store*, data barang akan langsung berstatus “*Draft*” dan siap untuk dikonfirmasi tanpa perlu aktivitas scan fisik oleh staff. Proses ini secara signifikan mempercepat waktu pengiriman, mengurangi beban kerja staf, dan mengurangi risiko *human error* yang mungkin terjadi dalam proses input manual.

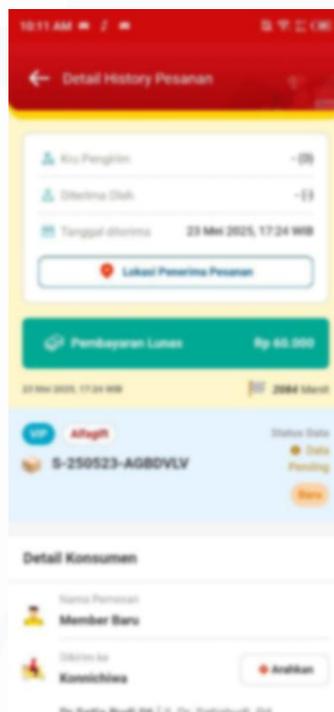
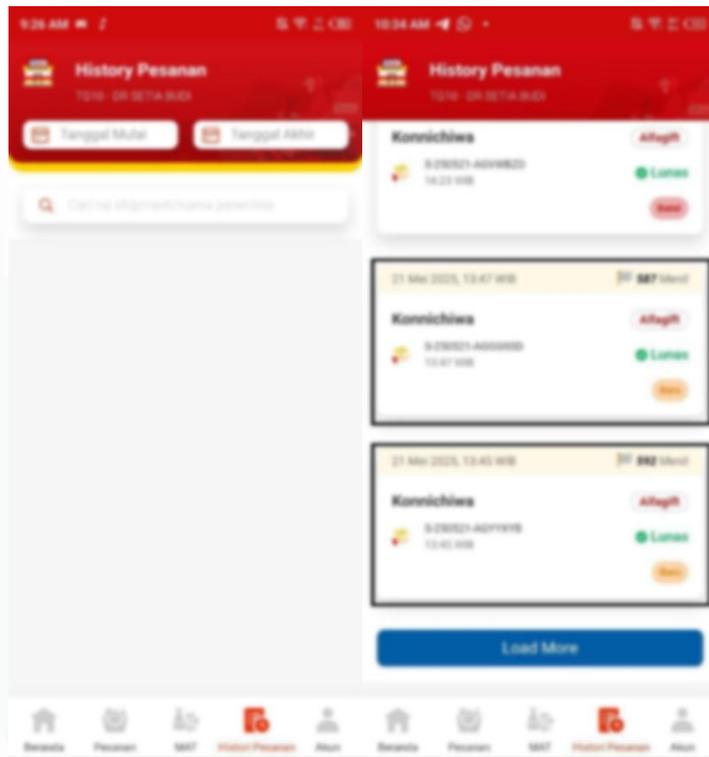
Fitur ini khusus diterapkan untuk barang yang berasal dari stok *Dark Store*, mengingat seluruh data inventori sudah terintegrasi dan dikontrol secara digital. Dengan demikian, sistem mampu menjamin bahwa data barang yang dikirim adalah valid dan akurat sesuai permintaan. Hal ini berbeda dari barang *non-Dark Store* yang masih memerlukan verifikasi fisik dan proses *packing* manual.



Gambar 3.28 Fitur Pesanan

Salah satu fitur baru yang diimplementasikan dalam proyek APO (Aplikasi Pesanan Online) ini adalah penambahan label “Barang *Dark Store*” yang secara eksplisit ditampilkan pada setiap pesanan yang berasal dari *Dark Store*. Fitur ini dapat dilihat pada Gambar 3.28, di mana label tersebut muncul di bagian atas setiap kartu pesanan yang berkaitan.

Fungsi utama dari penambahan label ini adalah untuk mempermudah identifikasi jenis pesanan oleh para pengguna aplikasi APO, khususnya kurir SAPA dan staf toko yang menangani proses pengiriman. *Dark Store* sendiri merupakan gudang khusus (bukan toko fisik yang dikunjungi pelanggan) yang digunakan untuk menyimpan stok barang dan melayani pesanan online secara lebih efisien. Dengan adanya identifikasi yang jelas terhadap pesanan dari *Dark Store*, proses sortir, *packing*, dan pengiriman dapat dilakukan dengan lebih cepat dan akurat.



Gambar 3.29 History Pesanan

Gambar 3.29 merupakan tampilan dari fitur Histori Pesanan dalam aplikasi APO. Meskipun pada proyek ini fitur Histori Pesanan tidak mengalami pembaruan atau perubahan fungsionalitas masih menggunakan struktur dan alur yang sama seperti pada versi sebelumnya namun fitur ini tetap menjadi bagian dari cakupan pengujian regresi oleh tim QA. Selain itu, QA juga memastikan bahwa fungsi *Search* dan *Filter* Histori Pesanan juga berjalan dengan lancar dalam *regression testing*. Hal ini dilakukan sebagai langkah antisipasi terhadap kemungkinan terjadinya penyenggolan yang bisa saja timbul akibat perubahan kode pada fitur-fitur lain dalam aplikasi yang berhubungan secara langsung maupun tidak langsung.



Gambar 3. 30 Fitur Load Monitoring SAPA

Fitur Monitoring SAPA dirancang untuk memantau aktivitas pesanan pada tiap-tiap toko Alfamart, sehingga memiliki peran penting dalam mendukung transparansi, akuntabilitas, serta pengambilan keputusan yang berbasis data seperti yang terlampir pada Gambar 3.30. Oleh karena itu, QA perlu memastikan bahwa setiap elemen dalam fitur ini mulai dari tampilan

antarmuka, fungsi fitur. Regresi testing dilakukan untuk memastikan bahwa fungsi-fungsi yang sudah ada pada fitur tersebut tetap berjalan dengan baik dan tidak terdampak oleh perubahan kode, penambahan fitur lain, ataupun pembaruan sistem secara keseluruhan. Meskipun tidak ada penambahan fungsi baru pada fitur ini, proses pengujian tetap penting untuk menjamin stabilitas dan integritas sistem secara menyeluruh.



Gambar 3. 31 *No Internet Case*

Gambar 3.31 merupakan salah satu pengujian skenario *negative case* pada APO. Dalam skenario ini, tim QA memverifikasi bagaimana sistem merespons ketiadaan koneksi internet. Fokus utama pengujian adalah memastikan bahwa *user interface* dapat menampilkan halaman kesalahan secara tepat, lengkap dengan informasi yang mudah dipahami oleh pengguna. Tampilan yang diuji mencakup adanya ilustrasi atau ikon yang menggambarkan masalah koneksi, serta pesan atau wording yang informatif dan sesuai standar kebahasaan yang digunakan dalam sistem secara keseluruhan.

Selain itu, tim QA juga memastikan bahwa tombol “Refresh Halaman” yang disediakan pada tampilan *error* tersebut berfungsi dengan baik. Tombol ini

menjadi bagian penting dari *user experience* karena memberikan opsi bagi pengguna untuk mencoba kembali memuat halaman setelah koneksi internet tersedia. QA menguji fungsi tombol ini dalam berbagai kondisi untuk memastikan bahwa ketika koneksi telah pulih, sistem dapat secara otomatis menampilkan ulang konten tanpa kendala.

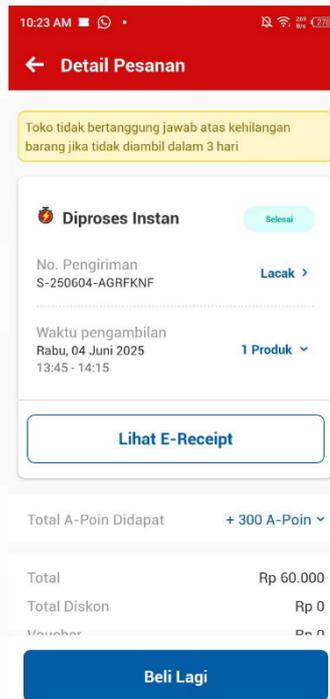


Gambar 3. 32 Order History - POD Manual

Pada proyek ini, selain melakukan proses pengetesan pada aplikasi APO, tim QA juga turut melakukan pengujian pada aplikasi Alfagift, karena dalam proyek ini terdapat sejumlah fitur yang beririsan langsung antara kedua aplikasi. Salah satu fitur yang diuji di Alfagift adalah peningkatan (*enhancement*) pada fitur POD, khususnya untuk skenario pemesanan dengan metode *pickup* oleh pelanggan.

Gambar 3.32 menampilkan salah satu contoh implementasi dari fitur POD *manual* di aplikasi Alfagift. Dalam skenario ini, setelah pelanggan melakukan pemesanan dan memilih metode pengambilan barang secara langsung (*pickup*),

pelanggan diberikan opsi untuk melakukan konfirmasi penerimaan barang secara mandiri melalui halaman *Order History*. Pada halaman tersebut, pelanggan dapat menekan tombol "Pesanan Sudah Diterima" sebagai bentuk konfirmasi bahwa barang telah diterima dengan baik.



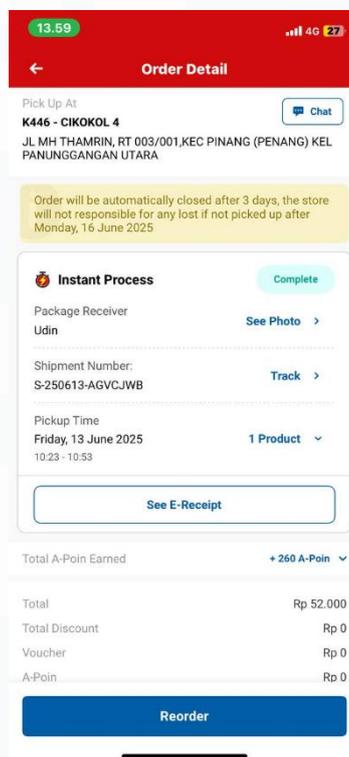
Gambar 3. 33 Order History - POD by System

Selain fitur *POD manual* yang memungkinkan *user* menekan tombol “Pesanan Sudah Diterima” secara langsung pada halaman *Order History*, terdapat juga fitur *POD otomatis* yang berbasis sistem, guna menyederhanakan proses konfirmasi pesanan yang belum diselesaikan oleh pelanggan secara *manual* seperti yang terlampir pada Gambar 3.33.

Pada skenario ini, jika *user* telah melakukan pemesanan dengan metode *pickup* dan status pesanan telah mencapai tahap “Ready to Pickup”, namun *user* tidak melakukan tindakan apapun (tidak menekan tombol konfirmasi penerimaan barang), maka sistem akan secara otomatis mengubah status pesanan menjadi “Selesai” setelah 3 hari terhitung dari waktu pesanan berada dalam status *Ready to Pickup*. Dengan kata lain, *POD* akan diproses secara

otomatis oleh sistem sebagai bentuk asumsi bahwa pesanan telah diambil dan diterima oleh pelanggan dalam kurun waktu tersebut.

Setelah itu ada juga sistem POD secara system dimana ketika *user* sudah memesan barang dengan status “Ready to Pickup” dan user tidak mengclick button pesanan sudah diterima. Pesanan akan otomatis selesai secara sistem dan berubah menjadi status selesai setelah 3 hari. Tim QA memastikan bahwa flow ini berjalan dengan lancar



Gambar 3. 34 Order History - POD by Crew

Fitur terakhir dalam rangkaian proses POD pada proyek ini adalah POD by Crew Toko, yaitu metode konfirmasi pesanan yang dilakukan secara manual oleh staf toko melalui aplikasi APO (Aplikasi Pesanan Online). Dengan demikian, crew toko diberikan otoritas untuk menyelesaikan pesanan secara langsung dari sisi *backend* operasional seperti yang terlampir pada Gambar 3.34.

Dalam alur kerjanya, *crew* toko akan memindahkan pesanan ke tab “Selesai” di dalam aplikasi APO setelah memastikan bahwa barang benar-benar telah diterima oleh pelanggan. Ketika tindakan ini dilakukan, status pesanan yang sebelumnya “Dalam Pengiriman” atau “*Ready to Pickup*” akan langsung berubah menjadi “Selesai” pada halaman *Order History* pengguna di aplikasi Alfagift.

Selain perubahan status menjadi “Selesai”, sistem juga akan menampilkan detail tambahan berupa nama penerima paket (*package receiver*) dan bukti visual berupa foto yang diunggah oleh *crew* toko saat penyerahan barang. Dalam tahap pengujian, tim QA bertanggung jawab untuk memastikan bahwa seluruh alur POD ini berjalan dengan sesuai, baik dari sisi sistem selama 3 hari, sisi manual, maupun integrasi dengan sistem APO.

3.3 Kendala yang Ditemukan

Selama pelaksanaan kegiatan magang sebagai Quality Assurance QA di PT. GLI, terdapat sejumlah kendala baik dari sisi teknis maupun non-teknis yang mempengaruhi kelancaran dalam melaksanakan tugas-tugas yang diberikan sebagai:

1. Penggunaan *Tools* Baru

Selama proses pengerjaan proyek, ditemukan kebutuhan untuk menggunakan berbagai *tools* dan *platform* yang sebelumnya belum pernah digunakan, seperti MongoDB untuk mengakses database NoSQL, Swagger UI dan Postman untuk dokumentasi serta pengujian API, dan DBeaver sebagai alat manajemen *database*.

2. *Uptime Service* yang Sering *Error*

Salah satu kendala yang cukup sering dihadapi selama proses pengujian adalah terkait dengan *uptime service* yang tidak stabil, terutama pada sistem *service* Alfagift. Selama proses pengetesan berlangsung, cukup sering muncul error dengan kode status 500 *Internal Server Error*, yang menandakan adanya gangguan pada sisi server. Kondisi ini terjadi karena *service* utama dari Alfagift kerap kali mengalami *downtime* atau mati secara tiba-tiba, sehingga menyebabkan berbagai fitur yang sedang diuji menjadi tidak dapat diakses.

3. Adaptasi Lingkungan Kerja yang Lambat

Salah satu kendala yang dialami adalah ketidakterbiasaan sebagai anak magang terhadap suasana kerja di lingkungan kantor, terutama dalam mengikuti alur kerja proyek nyata yang melibatkan banyak pihak dan tanggung jawab yang kompleks. Di awal masa magang, sering muncul perasaan canggung dan kurang percaya diri karena belum terbiasa dengan ritme kerja yang cepat, penggunaan *tools* profesional, serta komunikasi tim yang dinamis. Selain itu, adanya tekanan untuk menyesuaikan diri dengan ekspektasi tim dan memahami proses kerja yang sudah terstruktur juga menjadi tantangan tersendiri. Butuh waktu untuk memahami standar kerja profesional, mengikuti alur *sprint*, memahami *story list*, hingga terlibat langsung dalam proses *testing*.

4. Kurangnya Ketekunan dan Ketelitian yang Lebih

Kurangnya ketekunan dan ketelitian dalam melakukan pengetesan juga terkadang menjadi hambatan, khususnya pada awal-awal. Hal ini terlihat dari beberapa kesalahan minor seperti salah input data, kurang teliti saat melakukan verifikasi hasil pengujian, hingga terlewatnya beberapa skenario uji yang seharusnya dicoba. Meskipun kesalahan tersebut tidak berdampak besar terhadap proses secara keseluruhan, namun tetap menunjukkan bahwa dibutuhkan perhatian lebih terhadap detail dan konsistensi dalam bekerja. Kurangnya ketekunan ini juga berpengaruh terhadap efisiensi waktu, karena harus dilakukan revisi atau pengecekan ulang.

5. Keterbatasan Akses

Keterbatasan akses juga menjadi kendala yang cukup signifikan selama menjalani masa magang. Sebagai peserta magang, akses yang diberikan memang cenderung terbatas demi menjaga keamanan dan kerahasiaan data perusahaan. Namun, kondisi ini berdampak langsung terhadap proses pengujian yang dilakukan, terutama ketika harus melakukan verifikasi data secara langsung ke *database* atau mengintervensi API.

3.4 Solusi atas Kendala yang Ditemukan

Dalam mengatasi berbagai kendala tersebut, terdapat beberapa langkah aktif dan strategis yang tidak untuk menyelesaikan permasalahan tersebut, antara lain:

1. Melakukan Eksplorasi Lebih

Untuk mengatasi keterbatasan pengetahuan terhadap tools baru seperti MongoDB, Swagger UI, Postman, dan DBeaver, dilakukan pendekatan pembelajaran mandiri secara bertahap melalui dokumentasi resmi, tutorial video, dan forum komunitas. Seiring waktu dan frekuensi penggunaan, pemahaman terhadap *tools* menjadi lebih baik dan mampu mendukung aktivitas pengujian dengan lebih efektif.

2. Mencatat dan Melaporkan di *Group Project*

Dalam menghadapi kondisi sistem yang tidak stabil, khususnya seringkali muncul error 500 *Internal Server Error* akibat *downtime* pada service Alfagift, tim *Quality Assurance* menerapkan pendekatan strategis guna meminimalisir gangguan terhadap proses pengujian. Salah satu solusi utama yang diterapkan adalah dengan melakukan pencatatan secara rinci terhadap waktu terjadinya error, termasuk mencatat tanggal, jam, durasi, serta endpoint yang terdampak. Data pencatatan ini kemudian secara berkala atau real-time dilaporkan kepada tim pengelola service atau tim backend untuk segera ditindaklanjuti dan dilakukan analisis akar masalah (*root cause analysis*).

Dalam menghadapi seringnya error 500 akibat *downtime service* Alfagift, solusi yang diterapkan adalah dengan melakukan pencatatan waktu error secara rinci dan melaporkannya ke tim *service* untuk ditindaklanjuti. Pendekatan ini membantu menjaga alur pengujian tetap berjalan meskipun terdapat kendala sistem.

3. Lebih Aktif dan Beradaptasi

Langkah yang dilakukan antara lain adalah dengan mulai berani bertanya kepada *mentor* atau rekan tim lainnya jika ada hal yang belum dipahami, baik terkait teknis maupun alur kerja. Selain itu, juga diupayakan untuk selalu hadir dan terlibat dalam setiap forum diskusi seperti *daily meeting* serta *sprint planning*. Dari situ, pemahaman terhadap konteks proyek, prioritas kerja, serta ekspektasi tim bisa diperoleh secara langsung.

4. Lebih Teliti dan Disiplin

Kesadaran akan pentingnya ketelitian dan kedisiplinan mendorong dilakukannya evaluasi terhadap kesalahan sebelumnya. Dengan mulai menerapkan *checklist test case* pribadi, melakukan verifikasi hasil pengujian secara dua kali (*double check*), dan memanfaatkan *tools* dokumentasi pengujian secara lebih maksimal. Perbaikan kebiasaan ini secara langsung berdampak pada meningkatnya kualitas hasil uji dan berkurangnya revisi.

5. Koordinasi dengan Atasan

Dalam menghadapi keterbatasan akses terhadap sistem atau lingkungan tertentu selama proses pengujian, tim QA mengambil langkah proaktif dengan secara aktif mengajukan permintaan akses tambahan kepada pihak atau tim yang berwenang, seperti tim infrastruktur, *security*, maupun pengelola sistem terkait. Setiap permintaan akses selalu disertai dengan penjelasan konteks dan urgensi kebutuhan, termasuk penjabaran skenario pengujian yang tidak dapat dilakukan tanpa hak akses tertentu.

Meskipun keterbatasan akses tidak sepenuhnya dapat dihindari, strategi ini terbukti membantu meminimalisir hambatan teknis dalam proses *testing*. QA tetap dapat melanjutkan sebagian besar skenario pengujian dengan alternatif atau pendekatan *workaround* yang telah disepakati bersama tim teknis terkait. Dengan demikian, proses pengujian tetap berjalan efektif dan sesuai dengan timeline pengembangan produk, tanpa mengorbankan akurasi maupun kelengkapan hasil uji.

