

BAB III

PELAKSANAAN KERJA

3.1 Kedudukan dan Koordinasi

Bagian ini berisi keterangan/informasi mengenai posisi penulis dan alur koordinasi penulis dengan pembimbing lapangan pada saat penggerjaan suatu proyek/pengerjaan.

3.1.1 Kedudukan

Selama program Career Acceleration Program (CAP), penulis menempati posisi sebagai Intern Quality Assurance (QA) yang berada di bawah Departemen Product Operations PT Global Loyalty Indonesia (GLI). Posisi ini merupakan bagian penting dari tim teknologi yang bertanggung jawab secara langsung terhadap kualitas produk digital utama perusahaan, yaitu Alfagift.

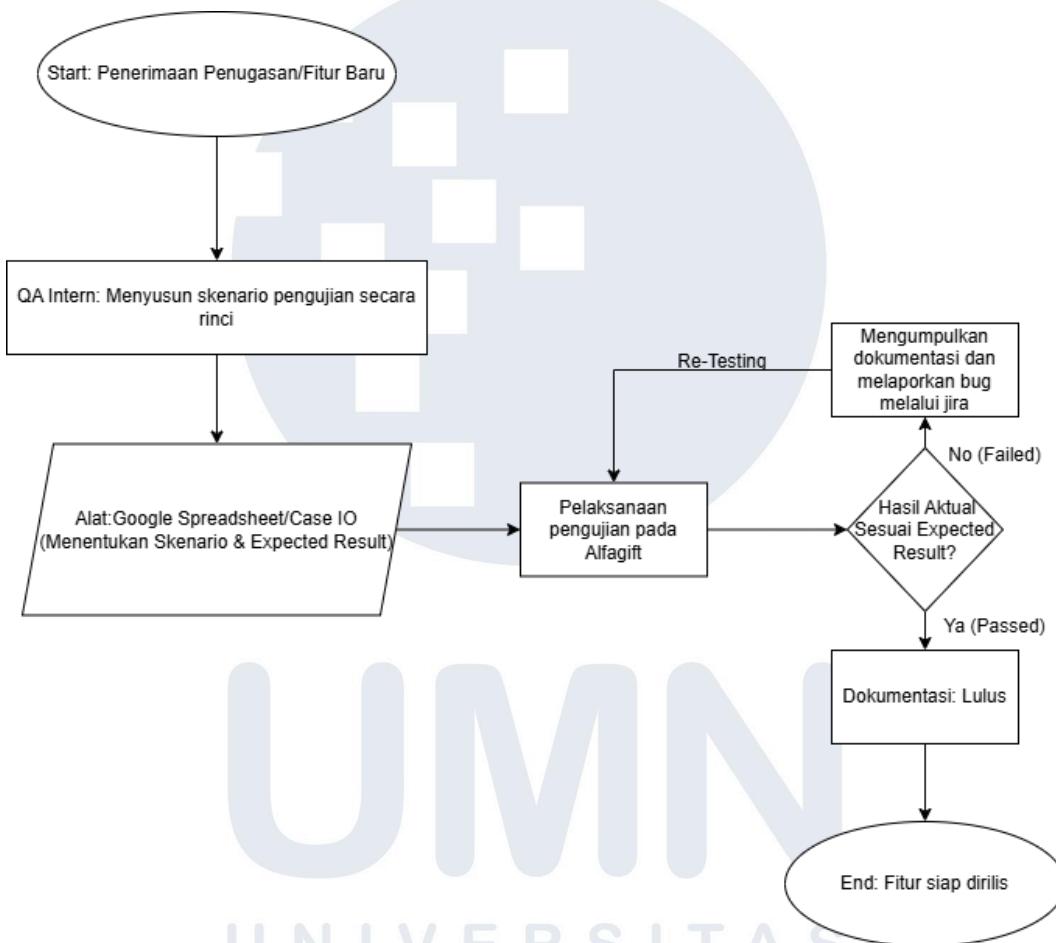
Dalam struktur divisi, peran penulis dikoordinasikan langsung oleh Product Operations Manager, Bapak Benny Chandra, yang memegang tanggung jawab atas keseluruhan operasional dan kelancaran produk digital. Selain itu, penulis juga mendapatkan bimbingan dan arahan teknis dari Mas Budi Haryanto, yang bertindak sebagai mentor atau pembimbing lapangan (pengarah) sehari-hari dalam mengimplementasikan alur kerja QA.

Tanggung jawab utama penulis sebagai Quality Assurance Intern adalah melaksanakan pengujian sistematis pada aplikasi Alfagift. Hal ini mencakup verifikasi fungsionalitas fitur baru dan memastikan sistem bebas dari bug fungsional sebelum rilis resmi ke publik melalui App Store maupun Play Store.

3.1.2 Koordinasi

Koordinasi pekerjaan dan alur pelaksanaan tugas di PT Global Loyalty Indonesia dirancang secara sistematis untuk memastikan komunikasi yang cepat, pelaporan yang akurat, dan penyelesaian masalah yang efisien antar divisi.

A. Alur Pelaksanaan Tugas Utama



Gambar 3.1 Flow Chart Alur Pelaksanaan

Alur pelaksanaan tugas Quality Assurance Intern mengikuti tahapan siklus pengembangan produk (SDLC), yaitu:

- Penerimaan Penugasan: Proyek dimulai ketika penulis menerima penugasan fitur atau produk baru dari supervisor.
- Penyusunan Skenario Pengujian (*Test Scenario*): Sebelum pengujian dimulai, penulis bertanggung jawab menyusun skenario pengujian rinci.

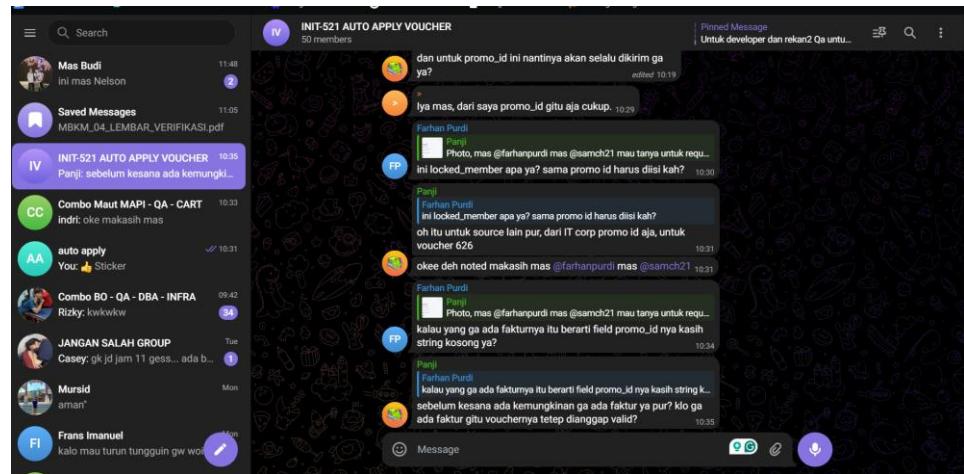
Skenario ini, yang biasanya dibuat menggunakan Google Spreadsheet atau platform manajemen kasus uji seperti Case IO, berfungsi sebagai panduan yang memuat:

- Deskripsi fitur.
- Langkah-langkah pengujian spesifik (*positive* dan *negative scenario*).
- Hasil yang diharapkan (*expected result*).
- Status pengujian (Lulus/Gagal).
- Pelaksanaan Pengujian: Penulis menguji aplikasi Alfagift berdasarkan skenario yang telah disusun. Setiap langkah diverifikasi kesesuaian antara *actual result* dan *expected result*.
- Pelaporan *Bug* dan *Error*: Jika ditemukan ketidaksesuaian (*bug* atau *error*), penulis bertanggung jawab mendokumentasikan dan melaporkannya kepada tim pengembang (*developer*) melalui platform Jira. Laporan ini harus mencakup deskripsi masalah, langkah-langkah untuk mereproduksi *bug*, perbandingan hasil, dan bukti pendukung (seperti tangkapan layar atau video).

B. Mekanisme Komunikasi dan Pertemuan

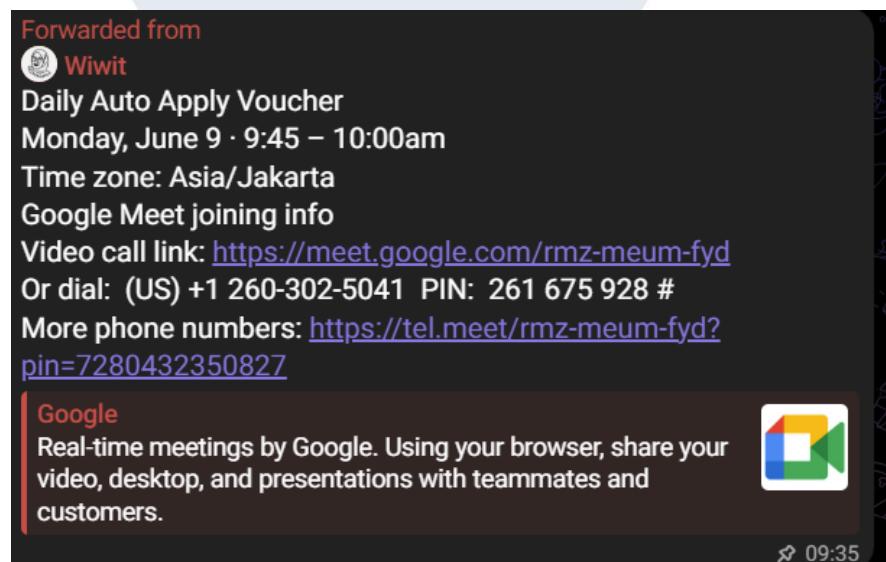
Komunikasi dan koordinasi dalam tim QA, dengan tim *developer*, dan divisi lain dilakukan melalui saluran dan frekuensi yang terstruktur:

- Telegram: Digunakan sebagai saluran komunikasi utama untuk koordinasi harian yang cepat (*real-time*), baik antar sesama anggota tim QA maupun dengan divisi lain yang terlibat dalam proyek.



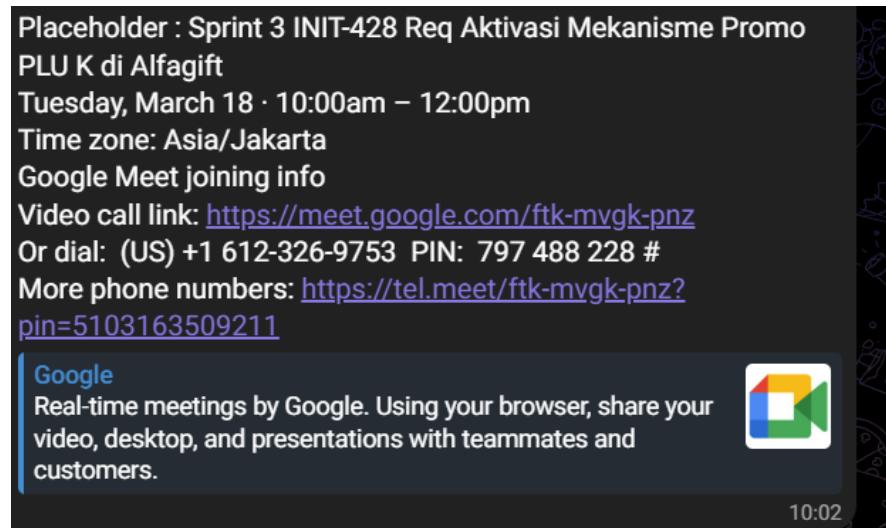
Gambar 3.2 Telegram

- Google Meet (*Daily Update*): Pertemuan rutin harian (pagi hari) yang dipimpin oleh *Project Manager* untuk memberikan pembaruan tentang kemajuan pekerjaan hari sebelumnya, kendala yang dihadapi, rencana kerja hari ini, dan pembahasan penyelesaian *bug* yang ditemukan.



Gambar 3.3 Google meet (Daily update)

- Google Meet (*Sprint Refinement*): Pertemuan ini dilakukan saat proyek baru hendak dimulai atau seiring berjalannya proyek. Fokusnya adalah membahas *System Design* dan *scope* pekerjaan, serta penyempurnaan *backlog* untuk memastikan semua item terdefinisi dengan jelas sebelum diimplementasikan.

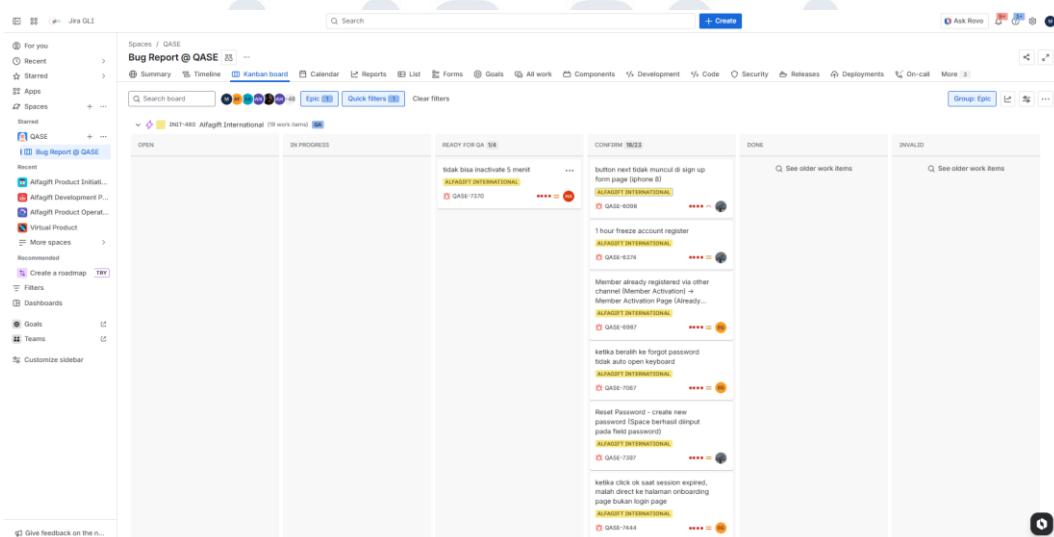


Gambar 3.4 Google meet (Sprint Refinement)

- Jira: Dalam menjaga transparansi dan efisiensi koordinasi antara tim *Quality Assurance* dan tim pengembang, perusahaan menggunakan Jira sebagai platform utama untuk pelacakan *bug (bug tracking)*. Setiap kendala teknis yang ditemukan selama eksekusi skenario uji didokumentasikan ke dalam tiket Jira dan dikelola melalui papan *Kanban* dengan status sebagai berikut:
 - a) Open: Status awal saat sebuah *bug* atau isu baru saja dilaporkan oleh tim QA. Pada tahap ini, tiket berisi deskripsi lengkap mengenai masalah, langkah-langkah reproduksi, serta bukti pendukung seperti tangkapan layar atau rekaman layar.
 - b) In Progress: Tahap di mana tim pengembang (*developer*) sedang melakukan analisis kode dan mengerjakan perbaikan atas *bug* yang dilaporkan. Penulis memantau status ini untuk mengetahui prioritas pengerjaan isu-isu kritis.
 - c) Ready for QA: Status ini menandakan bahwa tim pengembang telah selesai melakukan perbaikan pada lingkungan pengembangan dan kode tersebut telah siap untuk diuji kembali (*retesting*) oleh tim QA. Penulis akan menarik tiket ini untuk divalidasi pada lingkungan *Staging* atau *Beta*.

- d) Confirm: Setelah penulis melakukan pengujian ulang dan memastikan bahwa *bug* tersebut benar-benar telah diperbaiki tanpa menimbulkan isu baru (regresi), tiket dipindahkan ke status *Confirm*. Ini adalah tahap verifikasi akhir sebelum isu dianggap selesai.
- e) Done: Status final yang menandakan bahwa isu telah tuntas sepenuhnya dan fungsionalitas fitur tersebut telah sesuai dengan hasil yang diharapkan (*expected result*).
- f) Invalid: Tiket akan dipindahkan ke status ini apabila isu yang dilaporkan ternyata bukan merupakan kesalahan sistem (*bug*), melainkan perilaku sistem yang memang diharapkan (*expected behavior*), atau isu tersebut tidak dapat diproduksi ulang setelah melalui analisis lebih lanjut.

Melalui mekanisme ini, penulis dapat memastikan bahwa seluruh hambatan teknis, seperti tombol yang tidak muncul pada perangkat tertentu atau kendala pada formulir registrasi akun, terdokumentasi dan tertangani secara sistematis hingga mencapai standar kualitas yang ditetapkan.



Gambar 3.5 Jira

- Glitz Assemble: Kegiatan rutin bulanan yang sering dibawakan oleh *C-Level* perusahaan. Tujuannya adalah memberikan pembaruan umum dari berbagai divisi, membahas pencapaian, menetapkan target, serta menjadi wadah untuk motivasi dan perkenalan karyawan baru.



Gambar 3.6 Glitz Assemble

3.2 Tugas yang Dilakukan

Selama menjalani program *Career Acceleration Program* (CAP) selama enam bulan di PT Global Loyalty Indonesia, penulis mengemban tanggung jawab sebagai *Quality Assurance (QA) Intern*. Dalam kesehariannya, penulis bekerja di bawah arahan Departemen *Product Operations* dengan bimbingan langsung dari Bapak Benny Chandra selaku manajer dan Mas Budi Haryanto sebagai mentor teknis. Fokus utama dari posisi ini adalah memastikan bahwa setiap fitur pada aplikasi Alfagift memiliki kualitas yang optimal dan bebas dari kendala teknis sebelum dilepas ke tangan pengguna luas. Untuk mencapai standar tersebut, penulis menerapkan dua metode pengujian, yaitu manual dan otomatis, yang dipilih berdasarkan urgensi serta kompleksitas setiap proyek yang ditangani.

Siklus kerja penulis biasanya diawali setelah tahap perumusan teknis atau *Sprint Refinement*, di mana penulis menerima rancangan desain sistem (*System Design*) yang berfungsi sebagai panduan utama pengujian. Berdasarkan acuan tersebut, penulis merancang skenario uji yang mendalam menggunakan Google Spreadsheet maupun alat manajemen tes seperti *Case IO*. Dokumen ini menjadi peta jalan yang mencakup detail persyaratan fitur, langkah-langkah pengetesan, hingga hasil akhir yang diharapkan. Secara spesifik, pengujian dibagi menjadi dua ranah: verifikasi antarmuka dan interaksi pengguna (*Front-End*) serta pemeriksaan akurasi data pada sisi *Back-End* melalui sistem *Back Office* perusahaan.

Terhadap fitur-fitur yang menjadi pilar utama aplikasi dan memerlukan pengetesan berulang, penulis menerapkan pengujian otomatis (*automation testing*). Proses ini dikembangkan dengan bahasa pemrograman Java yang didukung oleh *framework* Appium serta perangkat lunak IntelliJ IDEA. Guna menjembatani aspek teknis dengan kebutuhan bisnis, skrip otomatisasi ini ditulis menggunakan format *Behavior-Driven Development* (BDD) dengan bahasa Gherkin yang lebih mudah dipahami secara logika manusia, misalnya melalui perintah "*user will see warning message*". Apabila ditemukan adanya kegagalan sistem, penulis akan mendokumentasikannya secara resmi ke dalam platform Jira untuk diperbaiki oleh tim pengembang.

Sementara itu, untuk fitur-fitur baru yang memiliki variasi skenario sangat beragam, pengujian secara manual menjadi pilihan utama untuk memastikan ketelitian yang lebih mendalam. Jika dalam prosesnya ditemukan *bug*, penulis melakukan pelaporan dan pemantauan perbaikan secara intensif melalui platform Jira dan Spreadsheet. Setelah perbaikan selesai dilakukan, penulis akan melakukan pengujian ulang (*retesting*) untuk menjamin bahwa kendala tersebut benar-benar tuntas. Sebagai tahap akhir yang esensial, penulis menyusun dokumentasi laporan akhir yang mencakup seluruh temuan dan pencapaian selama masa magang. Seluruh perjalanan proyek dan rincian tugas yang dikerjakan penulis selama enam bulan di PT Global Loyalty Indonesia terangkum dalam tabel di bawah ini:

Tabel 3.1 Detail Pekerjaan yang Dilakukan

No	Tugas	Mulai	Selesai
<i>1st Project: Auto Apply</i>			
1.	Pembuatan skenario Manual pada Spreadsheet	05 Agustus 2025	08 Agustus 2025
2.	<i>Testing</i> Skenario Manual	11 Agustus 2025	19 September 2025
<i>2nd Project: Penambahan Kolom Note/Catatan pada Alamat Kirim</i>			
1.	Pembuatan Skenario Automation pada Qase	24 September 2025	30 September 2025
2.	<i>Testing</i> Skenario Manual & Automation	01 Oktober 2025	03 Oktober 2025
<i>3rd Project: Voucher Type A</i>			
1.	Pembuatan Skenario Manual pada Spreadsheet	01 November 2025	08 November 2025
2.	<i>Testing</i> Skenario Manual	08 November 2025	04 Desember 2025

3.3 Uraian Pelaksanaan Kerja

Pada bagian ini, penulis menjabarkan rincian pekerjaan yang dilakukan selama menjalani program magang di PT Global Loyalty Indonesia (GLI). Fokus utama pekerjaan penulis adalah mendukung tim *Product Operations* dalam menjaga stabilitas dan kualitas aplikasi Alfagift melalui serangkaian pengujian sistematis. Penulis terlibat langsung dalam siklus pengembangan produk, mulai dari tahap

analisis dokumen teknis hingga verifikasi akhir sebelum fitur dirilis ke lingkungan produksi.

Secara umum, uraian pelaksanaan kerja penulis mencakup aspek-aspek berikut:

- Analisis Persyaratan Bisnis dan Teknis: Sebelum memulai pengujian, penulis melakukan bedah dokumen *Business Requirement*, *Business Rule*, dan *System Requirement* untuk memahami logika fitur yang akan dikembangkan.
- Perancangan dan Penyusunan Skenario: Penulis merumuskan rencana pengujian yang mencakup skenario fungsional, non-fungsional, serta penanganan kesalahan (*negative testing*) yang dituangkan dalam format *Test Scenario*.
- Eksekusi Pengujian Antarmuka dan Data: Pelaksanaan pengujian dilakukan pada sisi *Front-End* (aplikasi Alfagift) untuk memvalidasi interaksi pengguna, serta sisi *Back-End* melalui sistem *Back Office* (BO) untuk memastikan akurasi data yang dikirim dan diterima oleh server.
- Manajemen Defect dan Pelaporan: Apabila ditemukan ketidaksesuaian antara hasil aktual dengan hasil yang diharapkan, penulis melakukan dokumentasi *bug* secara mendalam ke dalam platform Jira untuk ditindaklanjuti oleh tim pengembang.
- Otomatisasi Pengujian (Automation Testing): Pada proyek-proyek tertentu, penulis mengimplementasikan skrip pengujian otomatis menggunakan bahasa pemrograman Java dan *framework* Appium guna meningkatkan efisiensi proses regresi.

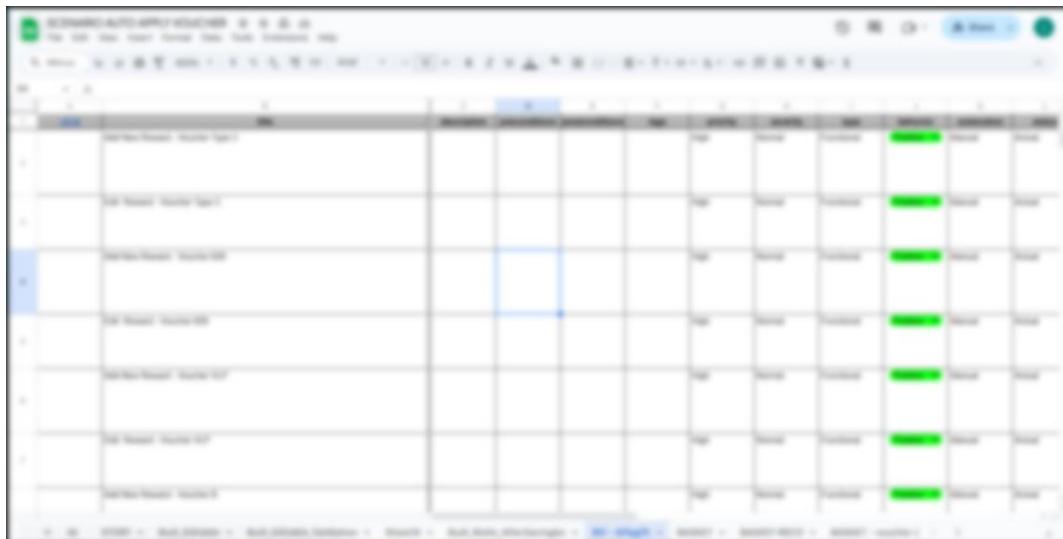
Uraian mendalam mengenai masing-masing proyek, tantangan yang dihadapi, serta kontribusi spesifik penulis akan dijelaskan pada sub-bab berikutnya.

3.3.1 Proses Pelaksanaan

Dalam melaksanakan tugas sebagai *Quality Assurance Intern*, penulis mengikuti alur kerja sistematis yang terintegrasi dengan siklus pengembangan perangkat lunak (*Software Development Life Cycle*) di PT Global Loyalty Indonesia. Secara garis besar, setiap proyek yang ditangani oleh penulis melibatkan tahapan-tahapan utama sebagai berikut:

- Analisis Persyaratan (Requirement Analysis): Proses dimulai dengan membedah dokumen *Business Requirement*, *Business Rule*, dan *System Requirement* untuk memahami secara mendalam tujuan fitur, batasan sistem, serta target pengguna.
- Penyusunan Skenario Pengujian: Penulis merancang skenario pengujian yang komprehensif, mencakup *positive case* untuk alur normal dan *negative case* untuk menguji ketahanan sistem terhadap input yang tidak sesuai. Skenario ini didokumentasikan dalam Google Spreadsheet atau alat manajemen tes yang mencakup deskripsi langkah, hasil yang diharapkan (*expected result*), dan status kelulusan.
- Eksekusi Pengujian (Testing Execution): Penulis melakukan pengujian pada fitur terkait di aplikasi Alfagift (*Front-End*) maupun pengaturan pada sistem *Back Office* (BO). Pengujian dilakukan baik secara manual maupun otomatis menggunakan *framework* Appium dengan bahasa pemrograman Java.
- Manajemen Bug dan Pelaporan: Jika ditemukan ketidaksesuaian antara hasil aktual dengan hasil yang diharapkan, penulis mendokumentasikannya ke dalam platform Jira dengan menyertakan bukti pendukung berupa tangkapan layar atau video untuk kemudian ditindaklanjuti oleh tim pengembang.
- Verifikasi Akhir dan Dokumentasi: Setelah perbaikan dilakukan oleh tim pengembang, penulis melakukan pengujian ulang (*retesting*) untuk memastikan fitur telah stabil sebelum dinyatakan layak untuk dirilis.

3.3.1.1 Auto Apply

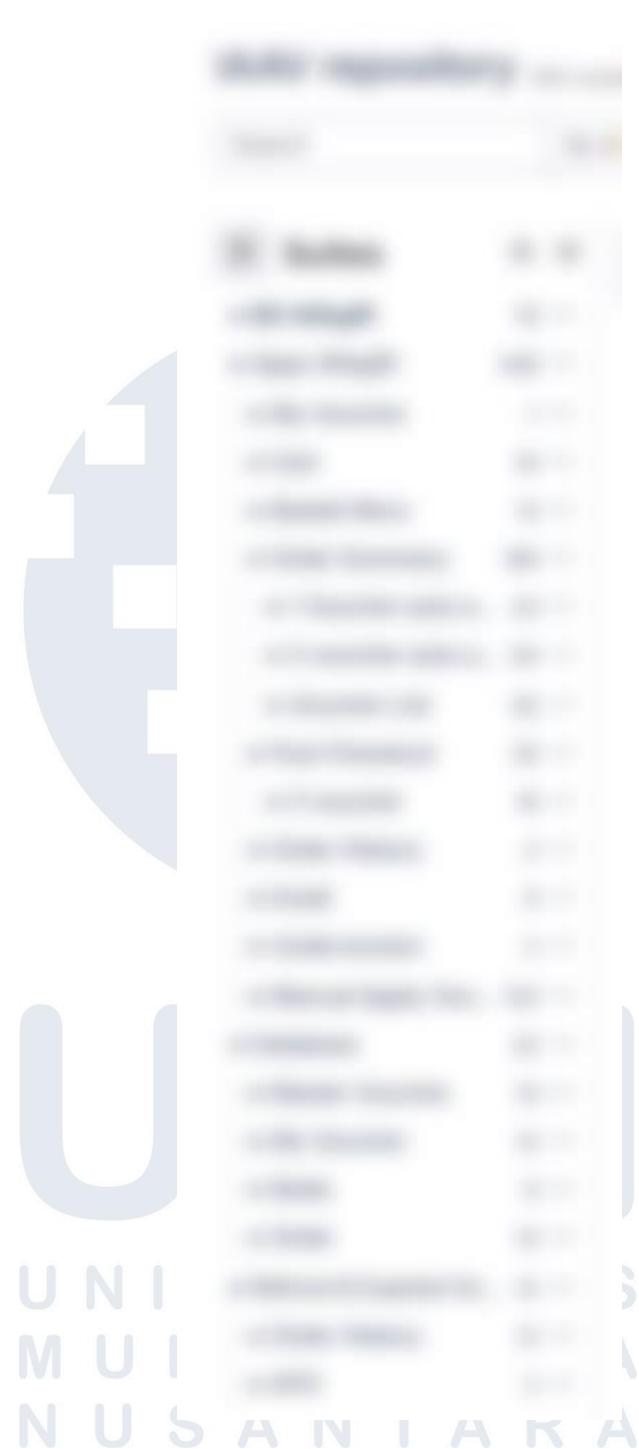


The screenshot shows a software interface with a title bar 'SCENARIO AUTO APPLY VOUCHER'. Below the title bar is a toolbar with various icons. The main area is a grid table with 6 rows and 10 columns. The rows are labeled 'Seluruh Voucher' and the columns are labeled 'Seluruh Voucher'. The first column is highlighted in blue, and the last column is highlighted in green. The grid contains several green checkmarks in the last column, indicating successful scenarios. The interface has a light gray background and a standard Windows-style window frame.

SCENARIO AUTO APPLY VOUCHER									
Seluruh Voucher									
Seluruh Voucher									
Seluruh Voucher									
Seluruh Voucher									
Seluruh Voucher									
Seluruh Voucher									
Seluruh Voucher									
Seluruh Voucher									

Gambar 3.7 Scenario Auto Apply

Proyek pertama yang penulis kerjakan adalah pengujian fitur Auto Apply Voucher. Sederhananya, fitur ini dibuat agar pelanggan Alfagift tidak perlu lagi bingung memilih satu per satu voucher mana yang paling menguntungkan saat ingin membayar belanjaan. Fitur ini akan bekerja secara otomatis untuk menghitung dan memasangkan kombinasi voucher yang memberikan potongan harga paling besar bagi pelanggan.



Gambar 3.8 Suite app alfagift

Suite Apps Alfagift (448 Skenario) Ini merupakan kumpulan pengujian yang berfokus pada pengalaman pengguna di aplikasi mobile. Pembagian detailnya meliputi:

- Order Summary (169 Skenario): Fokus utama fitur Auto Apply berada di sini, di mana sistem harus otomatis memasangkan kombinasi 1 hingga 3 voucher secara tepat saat pengguna berada di halaman ringkasan pesanan.
- Manual Apply Voucher (133 Skenario): Pengujian untuk memastikan bahwa fitur Auto Apply tidak berkonflik ketika pengguna memutuskan untuk memilih voucher secara manual.
- Cart & Basket Reco (94 Skenario): Memvalidasi sinkronisasi voucher sejak produk masuk ke keranjang belanja hingga munculnya rekomendasi basket.
- Post Checkout & Order History (34 Skenario): Memastikan data voucher yang diaplikasikan secara otomatis tercatat dengan benar pada status pesanan setelah pembayaran berhasil.
- Suite BO Alfagift (78 Skenario) Suite ini digunakan untuk memvalidasi fungsi pengujian pada sisi Back Office. Penulis melakukan verifikasi pada pengaturan parameter voucher di sisi server untuk memastikan mesin Auto Apply menerima input data yang akurat dari admin pusat.
- Suite Database (42 Skenario) Pengujian pada lapisan database sangat krusial untuk fitur Auto Apply karena melibatkan banyak tabel data.
- Master Voucher (10 Skenario): Memastikan skema data induk voucher tersimpan dengan benar.
- Redis (8 Skenario): Melakukan pengecekan pada sistem caching untuk menjamin kecepatan kalkulasi otomatis saat pengguna melakukan transaksi.
- Order & My Voucher (24 Skenario): Memvalidasi integritas data transaksi dan status kepemilikan voucher setelah proses otomatisasi berjalan.
- Suite Refund & Expired Voucher (14 Skenario) Skenario ini dirancang untuk menangani kondisi pengecualian, seperti bagaimana sistem Auto Apply merespons jika terdapat voucher yang tiba-tiba kedaluwarsa atau

terjadi proses refund pesanan yang melibatkan voucher otomatis. Hal ini termasuk integrasi dengan sistem APO (Alfamart Personal Order).

Sebelum fitur *Auto Apply* dapat diuji di aplikasi Alfagift, penulis harus memastikan bahwa berbagai jenis voucher telah tersedia di sistem pusat atau *Back Office* (BO). Tahap ini sangat krusial karena sistem *Auto Apply* tidak akan bisa memilih voucher jika data voucher tersebut tidak dibuat dengan parameter yang benar di sisi server.

1Pembuatan dan Pengaturan Berbagai Kategori Voucher Penulis menyusun skenario untuk membuat (*Add New*) dan mengubah (*Edit*) puluhan jenis voucher guna mencakup seluruh kategori produk yang ada di Alfamart. Hal ini dilakukan untuk memastikan mesin pencari voucher otomatis dapat mengenali berbagai kategori, seperti:

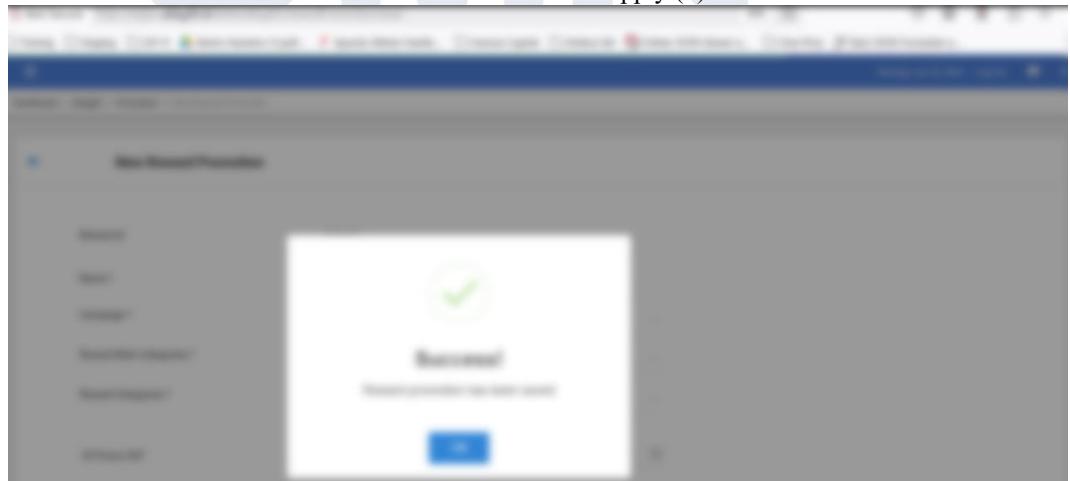
- Voucher Belanja Utama: Meliputi Voucher Tipe C (potongan harga), Voucher 626 (cashback), dan Voucher VLP (loyalitas).
- Voucher Kategori Produk: Menyiapkan voucher khusus untuk barang spesifik seperti Bahan Pokok, Makanan, Minuman, Obat-obatan, hingga Kebutuhan Bayi & Anak.
- Voucher Khusus: Termasuk voucher untuk permainan (*Gamification*), kupon undian, serta voucher khusus member (*Voucher Merchant* atau *Stamp*).



Gambar 3.9 Back office auto apply (1)



Gambar 3.10 Back office auto apply (2)



Gambar 3.11 Back office auto apply (3)

2. Pengujian Integrasi Data dengan Gateway SAT Setelah data voucher dibuat di BO, penulis harus memastikan bahwa sistem Alfagift dapat "meminta" data tersebut ke pusat data Alfamart (Gateway SAT). Penulis melakukan pengetesan teknis (*Get Faktur by Gateway*) untuk memastikan dua kondisi utama:
 - Ketersediaan Data: Memastikan respons dari Gateway menunjukkan bahwa nomor faktur dan kode voucher tersedia dan siap digunakan oleh pelanggan.
 - Validasi Voucher: Mengetes bagaimana respons sistem jika voucher ada namun kodennya tidak tersedia, sehingga sistem *Auto Apply* tahu bahwa voucher tersebut tidak boleh direkomendasikan kepada pelanggan saat itu juga.
3. Proses Uggah Data Massal (*Mass Upload*) Penulis juga melakukan simulasi pengunggahan data voucher dalam jumlah besar (*Upload Voucher Sukses*) untuk setiap kategori utama, mulai dari kategori Voucher Belanja hingga Kupon Undian. Hal ini bertujuan untuk memastikan bahwa ketika perusahaan merilis promo besar-besaran, sistem *Back Office* mampu memproses ribuan voucher sekaligus tanpa ada kesalahan data yang masuk ke aplikasi pelanggan.



Gambar 3.12 Test run

A. Memahami Aturan Main Fitur (*Requirement Analysis*) Sebelum melakukan pengujian, penulis terlebih dahulu mempelajari "aturan main" fitur ini agar tahu apa yang harus dianggap benar atau salah oleh sistem. Beberapa aturan utama yang penulis pelajari antara lain:

- Voucher yang Terpilih: Sistem hanya otomatis memilih voucher jenis tertentu, seperti voucher potongan harga (Tipe C) dan voucher program loyalitas lainnya.
 - Kapan Fitur Bekerja: Voucher akan terpasang otomatis saat pelanggan masuk ke halaman ringkasan pesanan (*order summary*), selama pelanggan belum memilih voucher lain secara manual.
 - Mencari Keuntungan Maksimal: Sistem harus pintar membandingkan voucher. Misalnya, jika ada dua voucher yang potongannya sama, sistem akan mendahulukan voucher yang tanggal kedaluwarsanya paling dekat agar voucher tersebut tidak hangus sia-sia.
 - Batasan Sistem: Fitur ini hanya berlaku untuk belanja lewat aplikasi Alfagift versi terbaru dan tidak berlaku untuk pemesanan lewat WhatsApp atau Website.

B. Proses Pengujian dan Skenario Uji Karena aturan perhitungannya sangat banyak, penulis membuat daftar panjang berisi kondisi-kondisi yang harus dites. Penulis menggunakan sebuah sistem manajemen pengujian untuk mencatat dan memantau setiap langkah. Secara keseluruhan, terdapat 88 skenario yang harus penulis pastikan berjalan tanpa *error*.

Penulis mencoba berbagai kemungkinan, seperti:

1. Skenario Normal: Mencoba belanja barang biasa dan melihat apakah voucher benar-benar terpasang sendiri dengan potongan harga yang benar.
2. Skenario Pengecekan Pesan: Memastikan aplikasi memunculkan tulisan "Voucher terpilih secara otomatis" agar pelanggan tahu bahwa mereka sudah mendapatkan potongan harga.
3. Skenario Batasan: Mencoba menggunakan voucher yang sudah habis kuotanya (limit) untuk memastikan sistem cukup pintar untuk menolak voucher tersebut dan menggantinya dengan voucher lain yang masih aktif.

C. Hasil Akhir Pengujian Proyek ini membutuhkan ketelitian yang sangat tinggi, sehingga penulis menghabiskan waktu total selama 5 minggu 4 hari untuk menyelesaiannya. Berdasarkan catatan pada sistem pengujian (BO Alfagift), berikut adalah hasil akhir dari kerja keras penulis:

- 64 Skenario Lulus (*Passed*): Fitur berjalan dengan sempurna sesuai harapan pelanggan.
- 24 Skenario Dibatalkan (*Cancelled*): Beberapa pengujian dihentikan karena adanya perubahan rencana dari tim pengembang atau kondisi yang sudah tidak relevan lagi saat proses tes berlangsung.

Dengan tingkat keberhasilan mencapai 100% dari rencana kerja yang diproses, penulis memastikan bahwa fitur ini sudah aman dan sangat

membantu memudahkan pelanggan Alfagift dalam berbelanja hemat tanpa repot.

3.3.1.2 Penambahan Kolom Note/Catatan pada Alamat Kirim

A screenshot of a code editor window with multiple tabs open. The tabs are labeled 'index.php', 'index.html', 'index.js', 'index.css', and 'index.js.map'. The main code editor area shows a large block of PHP code. The code includes various functions and classes, such as 'function', 'class', and 'foreach' loops. It also contains comments and several 'use' statements at the top. The code is well-organized with proper indentation and spacing.

Gambar 18. Code Automation (1)

Proyek kedua yang penulis kerjakan adalah pengembangan fitur penambahan kolom catatan atau instruksi khusus untuk kurir pada data alamat pengiriman di aplikasi Alfagift. Latar belakang proyek ini adalah untuk meminimalisir kendala kurir dalam menemukan lokasi pelanggan, yang sering kali disebabkan oleh titik *pinpoint* peta yang kurang akurat atau kondisi lingkungan yang kompleks. Dalam fase awal, penulis melakukan analisis mendalam terhadap dokumen persyaratan teknis yang menetapkan bahwa fitur ini harus memungkinkan pelanggan Alfagift menambahkan catatan opsional pada detail alamat mereka yang nantinya dapat diakses oleh *crew store* melalui sistem APO (*Alfamart Personal Order*). Secara sistem, fitur ini dirancang dengan batasan ketat, yaitu hanya menerima input maksimal 100 karakter dengan dukungan karakter alfanumerik dan

simbol-simbol tertentu seperti tanda baca titik, koma, dan kurung. Batasan ini bertujuan untuk menjaga integritas data pada *database* sekaligus memastikan informasi yang disampaikan tetap ringkas dan relevan bagi kurir, sementara penggunaan fitur di luar pengiriman langsung (seperti *pick up* atau transaksi pihak ketiga) dinyatakan berada di luar lingkup (*out of scope*) penggerjaan.

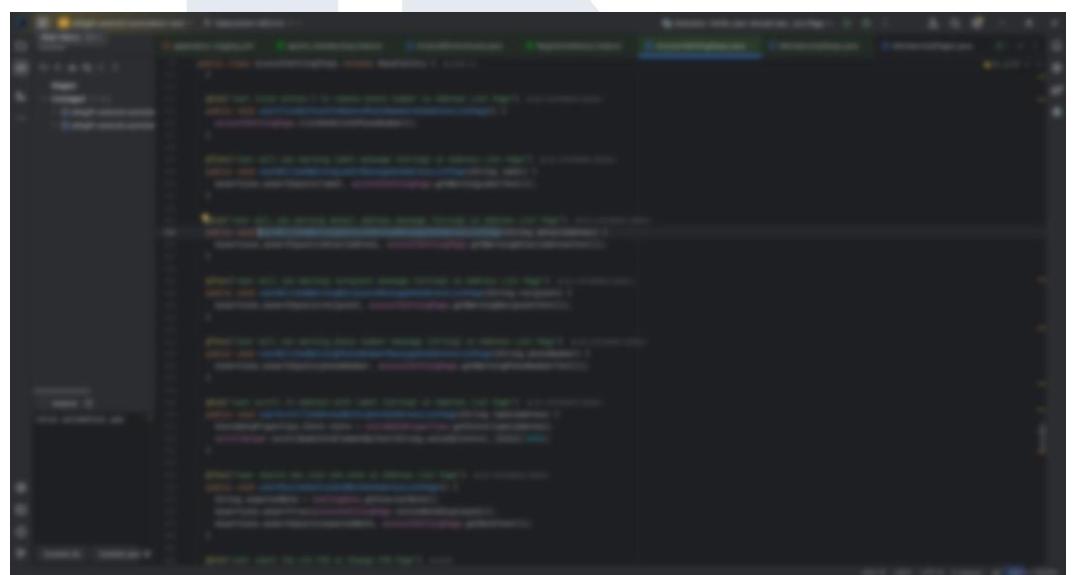


Gambar 3.13 Code Automation (2)

Setelah memahami cakupan bisnis dan teknis, penulis beralih ke tahap perancangan skenario pengujian yang lebih kompleks untuk memastikan fitur ini stabil saat digunakan oleh jutaan pengguna. Penulis menyusun berbagai skenario, mulai dari alur positif di mana pengguna berhasil menyimpan alamat dengan catatan lengkap, hingga skenario negatif untuk menguji ketahanan sistem terhadap input yang tidak valid. Pengujian ini tidak hanya dilakukan secara manual, tetapi juga diimplementasikan ke dalam kerangka kerja otomatisasi (*automation testing*) menggunakan bahasa Gherkin pada berkas RegisterAddress.feature. Dalam skrip tersebut, penulis mendefinisikan langkah-langkah verifikasi yang mendetail, seperti pengecekan apakah pesan peringatan muncul dengan benar ketika

pengguna mengosongkan kolom wajib lainnya saat sedang mencoba menyimpan catatan alamat, serta memastikan bahwa data catatan yang baru saja dimasukkan benar-benar tersimpan dan dapat ditampilkan kembali pada halaman daftar alamat dengan format yang sesuai.

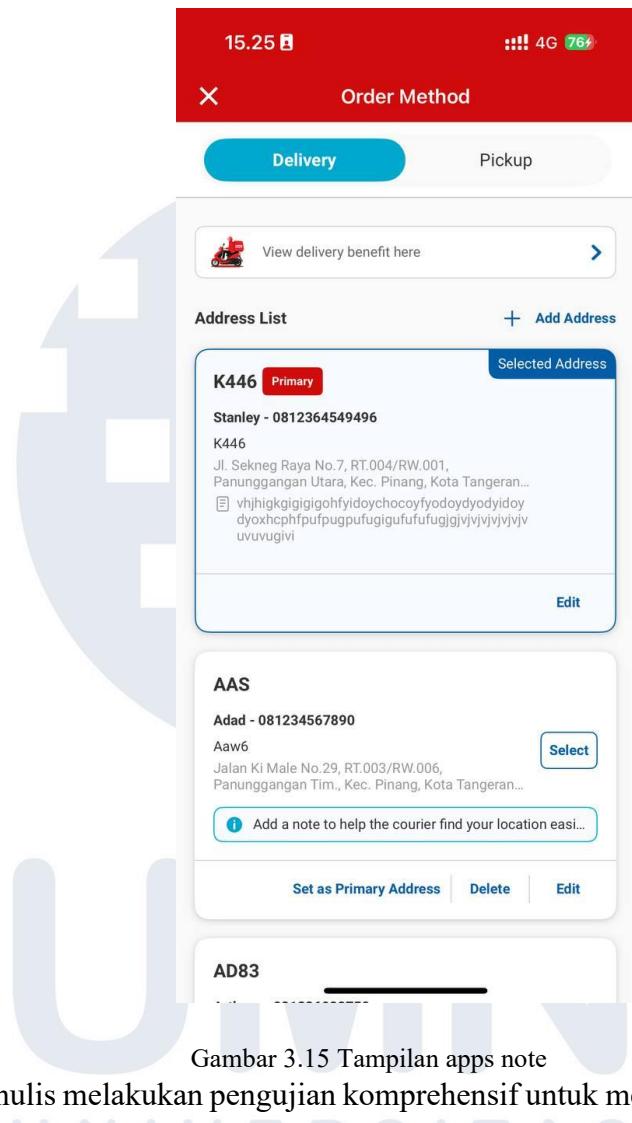
Secara teknis, penulis mengimplementasikan pengujian otomatis ini dengan mengadopsi pola desain *Page Object Model* (POM) untuk memisahkan logika pengujian dari elemen antarmuka.



Gambar 3.14 Code Automation (3)

Pada kelas `AccountSettingSteps`, penulis menuliskan logika pemrograman untuk menangkap elemen-elemen spesifik seperti ikon catatan dan teks instruksi kurir, memastikan bahwa setiap komponen visual muncul sesuai dengan rancangan UI/UX. Lebih lanjut, pada kelas `AccountSettingPage`, penulis mengembangkan metode khusus seperti `isIconNoteDisplayed()` dan `getNoteText()` untuk melakukan validasi otomatis terhadap *actual result* yang diterima dari sistem dibandingkan dengan *expected result* yang telah didefinisikan dalam dokumen teknis. Melalui pendekatan pengujian berlapis ini, penulis berhasil memverifikasi bahwa fitur catatan alamat tidak hanya berfungsi secara fungsional, tetapi juga memiliki validasi sisi klien yang kuat, sehingga dapat memberikan pengalaman belanja

omnichannel yang lebih mulus dan membantu operasional pengiriman menjadi lebih efisien.



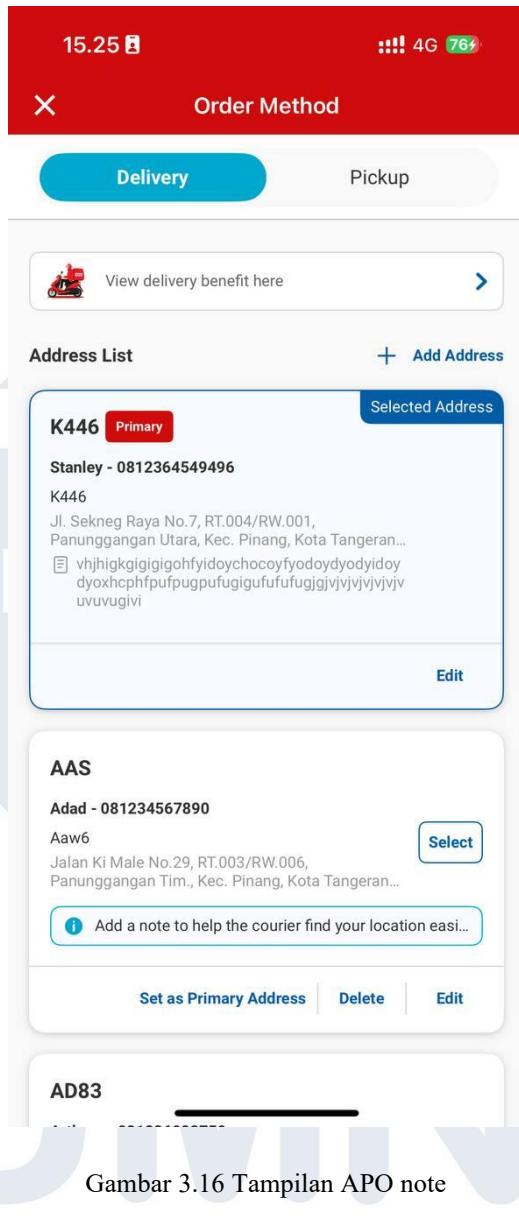
Gambar 3.15 Tampilan apps note

Penulis melakukan pengujian komprehensif untuk memastikan bahwa data catatan yang dimasukkan oleh pelanggan dapat tersampaikan dengan akurat hingga ke tangan kurir di gerai fisik. Pengujian ini melibatkan sinkronisasi dua platform utama, yaitu aplikasi Alfagift (*Front-End*) dan sistem APO (*Back-End Operational*).

Pada tahap ini, penulis memvalidasi fitur dari perspektif pengguna akhir untuk memastikan kemudahan dalam menambahkan instruksi khusus pengiriman. Detail pengujian antarmuka meliputi:

- Fitur Input Catatan: Pada halaman *Address List*, pengguna diberikan opsi "Add a note to help the courier find your location easily" pada setiap entitas alamat yang belum memiliki catatan.
- Visualisasi Catatan: Sistem secara otomatis menampilkan ikon dokumen kecil di samping teks catatan yang telah disimpan, guna memberikan penanda visual yang jelas bagi pelanggan mengenai instruksi yang telah mereka buat.
- Integrasi Peta dan Detail Alamat: Penulis memastikan catatan tersebut tampil berdampingan dengan informasi alamat lengkap, termasuk nama penerima, nomor telepon, dan titik koordinat, guna meminimalisir kesalahan interpretasi data.





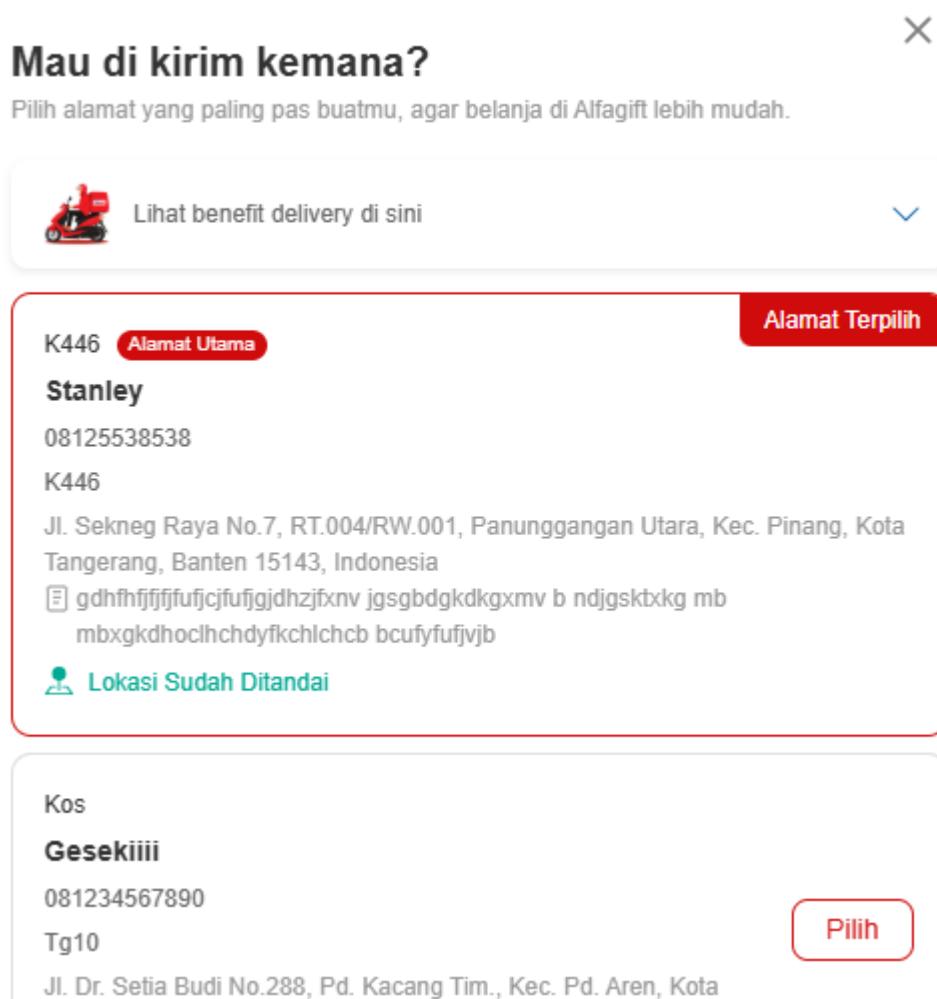
Gambar 3.16 Tampilan APO note

Setelah data tersimpan di sisi pelanggan, penulis melakukan verifikasi pada sistem APO yang digunakan oleh *crew store* untuk memproses pesanan. Tahap ini sangat krusial karena berkaitan langsung dengan efisiensi pengiriman di lapangan:

- Tampilan Detail Pesanan: Pada halaman "Detail Pesanan - Baru", sistem menyediakan kolom khusus bertajuk "Catatan Untuk Kurir" yang terletak tepat di bawah rincian alamat tujuan.
- Aksesibilitas Informasi Kurir: Penulis menguji fungsionalitas menu *dropdown* pada kolom catatan tersebut untuk memastikan instruksi

husus dari pelanggan dapat dibaca dengan jelas oleh personil toko sebelum mereka menekan tombol "Arahkan" atau "Packing".

- Sinkronisasi Status Stok: Selain catatan alamat, penulis juga memvalidasi bahwa sistem APO secara akurat menampilkan informasi stok barang (misalnya "Stock: 0" untuk barang tertentu), sehingga kurir dapat berkoordinasi dengan pelanggan melalui fitur "Chat / Call" jika terdapat kendala stok sebelum proses pengiriman dimulai.



Gambar 3.17 Tampilan WEBCOM note

Selain pengujian pada antarmuka aplikasi dan sistem operasional toko (APO), penulis juga melakukan validasi menyeluruh pada platform **WEBCOM** untuk memastikan integritas data alamat dan catatan

tambahan yang disimpan oleh pelanggan. Pengujian pada WEBCOM berfokus pada bagaimana sistem pusat menerima, menyimpan, dan menyajikan data tersebut untuk kebutuhan administratif.

Berdasarkan hasil *Test Run* pada lingkungan *Staging* untuk modul WEBCOM, penulis telah menyelesaikan serangkaian skenario uji dengan rincian sebagai berikut:

- Verifikasi Struktur Data Alamat: Penulis memastikan bahwa kolom catatan tambahan telah terintegrasi dengan benar dalam skema data alamat pelanggan. Pengujian mencakup pengecekan apakah teks instruksi kurir yang diinput oleh pelanggan pada aplikasi *mobile* tampil secara akurat pada kolom deskripsi alamat di panel admin WEBCOM.
- Validasi Batasan Karakter: Sesuai dengan spesifikasi teknis, penulis memverifikasi bahwa sistem WEBCOM mampu menangani dan membatasi input hingga maksimal 100 karakter alfanumerik. Hal ini diuji untuk mencegah terjadinya pemotongan data (*data truncation*) yang dapat menyebabkan informasi instruksi kurir menjadi tidak lengkap saat diakses oleh sistem lain.
- Sinkronisasi Status Lokasi: Selain data teks, penulis memvalidasi status koordinat yang menyertai alamat tersebut. Pada tampilan WEBCOM, penulis memastikan status "Lokasi Sudah Ditandai" tersinkronisasi dengan benar, sehingga kurir tidak hanya mendapatkan instruksi berupa teks, tetapi juga titik *pinpoint* yang akurat pada peta.
- Pengujian Alur Modifikasi: Penulis melakukan skenario pengeditan alamat melalui aplikasi dan memverifikasi apakah perubahan tersebut terupdate secara *real-time* di WEBCOM. Hal ini krusial untuk memastikan tidak ada redundansi data atau kesalahan informasi ketika pelanggan memperbarui instruksi kurir mereka.

Dengan selesainya pengujian pada sisi WEBCOM, penulis memastikan bahwa siklus data fitur catatan alamat telah tervalidasi secara *end-to-end*, mulai dari penginputan oleh pelanggan,

penyimpanan di pusat data, hingga pemrosesan akhir oleh tim operasional di lapangan.

3.3.1.3 Voucher Type A



Gambar 3.18 Scenario Voucher Type A

Proyek ketiga yang penulis laksanakan adalah pengujian menyeluruh terhadap sistem Voucher Tipe A, sebuah modul promosi inovatif pada aplikasi Alfagift. Berbeda dengan voucher konvensional yang memberikan potongan nominal, Voucher Tipe A dirancang untuk memberikan produk *groceries* secara gratis kepada pelanggan. Pengujian ini memiliki tingkat kerumitan yang tinggi karena melibatkan sinkronisasi *real-time* antara ketersediaan stok barang di toko dengan status validasi voucher di pusat data. Penulis bertanggung jawab memastikan bahwa seluruh alur, mulai dari pendaftaran promo oleh admin hingga diterimanya struk digital oleh pelanggan, berjalan tanpa hambatan teknis.

A. Analisis Mendalam Ketentuan Bisnis dan Logika Sistem Penulis memulai proyek dengan membedah batasan fungsionalitas yang ditetapkan dalam dokumen persyaratan. Fokus utama pengujian diarahkan pada logika ketergantungan produk, di mana transaksi menggunakan Voucher Tipe A wajib disertai minimal satu produk

reguler berbayar; sistem secara otomatis akan memblokir transaksi jika keranjang belanja hanya berisi produk gratis. Dari sisi sistem, voucher ini diidentifikasi melalui kode unik dengan awalan (*prefix*) huruf 'A' sebanyak 12 digit. Penulis juga menganalisis aturan penggabungan voucher (*multi-voucher*), di mana Voucher Tipe A dikategorikan sebagai *Special Hemat* yang dapat dikombinasikan dengan maksimal 3 voucher per transaksi atau 6 voucher per hari, namun dibatasi hanya 1 voucher per faktur untuk ID promo yang sama.

B. Pengujian Konfigurasi pada Sistem *Back Office* (BO) Tahap ini melibatkan pengujian pada sisi manajemen data untuk memastikan tim operasional dapat mendaftarkan kampanye dengan akurat. Penulis melakukan serangkaian pengujian pada modul *Reward*, antara lain:

Pendaftaran Kampanye: Memastikan admin dapat memasukkan kategori *Reward* Tipe A dan memvalidasi bahwa kolom SKP (*Syarat Ketentuan Promo*) bersifat wajib diisi guna mencegah kesalahan data saat promo dirilis.

Manajemen Data Massal: Menguji fitur *mass upload* untuk memastikan ribuan kode voucher dapat masuk ke *database* utama secara *bulky*, serta memverifikasi sistem peringatan melalui email jika terdapat kegagalan pada baris data tertentu.

Integrasi Gateway SAT: Penulis memvalidasi proses *mapping* antara nomor faktur, ID Promo, dan kode produk (PLU) melalui layanan IT Corp agar sistem dapat mengenali produk hadiah apa yang harus diberikan saat kode voucher dimasukkan.

C. Eksekusi Pengujian Antarmuka dan Pengalaman Pengguna (Android & iOS) Setelah data di sisi *Back-End* dinyatakan stabil, penulis beralih ke pengujian pada aplikasi untuk memverifikasi

pengalaman pelanggan. Pengujian ini mencakup skenario yang sangat detail pada halaman keranjang belanja (*basket*):

Otomatisasi Hadiah: Verifikasi bahwa produk hadiah terpilih secara otomatis (*auto-select*) dan masuk ke keranjang segera setelah voucher sukses diaplikasikan.

Sinkronisasi Stok dan Status: Jika stok produk hadiah di toko terkait dinyatakan kosong (*Out of Stock*), penulis memastikan aplikasi mampu menampilkan pesan peringatan dan secara otomatis melepas voucher tersebut agar tidak terjadi kegagalan saat pembayaran.

Logika Pembatalan: Penulis menguji kondisi di mana pelanggan membatalkan pesanan atau menghapus produk hadiah secara manual; sistem harus mampu melakukan *release* voucher kembali ke status aktif sehingga dapat digunakan di transaksi berikutnya.

D. Pengujian Performa dan Integritas Data Akhir Sebagai tahap penutup, penulis terlibat dalam *Stress Test* untuk mengevaluasi ketahanan sistem PAS Promo. Penulis juga memvalidasi data pada struk digital (*E-receipt*), memastikan bahwa nilai produk gratis dicatat sebagai komponen "Hemat Special" atau diskon, sehingga total pembayaran pelanggan sesuai dengan perhitungan yang diharapkan. Pengujian ini menjamin bahwa integrasi antara platform digital Alfagift dan operasional toko fisik tetap konsisten, mendukung strategi *omnichannel* perusahaan secara menyeluruh.



Gambar 3.19 Test run BO voucher type A

Pengujian pada sisi Back Office dilakukan untuk memvalidasi kemampuan sistem dalam mengelola parameter promosi dan data voucher sebelum didistribusikan kepada pelanggan. Berdasarkan dokumen Test Run pada lingkungan Staging, penulis menyelesaikan 34 skenario dengan tingkat kelulusan 100% (Passed). Detail teknis pengujian ini meliputi:

- Manajemen Kategori Reward: Penulis memastikan admin dapat mendaftarkan kampanye dengan akurat melalui modul Reward, termasuk memvalidasi bahwa kolom Syarat Ketentuan Promo (SKP) bersifat wajib diisi guna mencegah terjadinya kesalahan data saat promo dirilis ke publik.
- Validasi Aturan Bisnis dan Parameter: Pengujian ini mencakup verifikasi terhadap kemampuan sistem dalam menyimpan dan memperbarui aturan operasional seperti pembaruan nama kampanye, tanggal mulai dan berakhirnya promo, hingga pengaturan prioritas voucher agar tidak terjadi tumpang tindih antar program promosi.
- Keandalan Pengolahan Data Massal (Mass Upload): Penulis menguji fitur mass upload untuk memastikan ribuan kode voucher unik dengan prefix 'A' dapat masuk ke database utama secara bulky tanpa kegagalan sistem.

- Integrasi Pusat Data (Gateway SAT): Penulis memvalidasi proses pemetaan (mapping) antara nomor faktur, ID Promo, dan kode produk (PLU) melalui layanan IT Corp agar sistem aplikasi dapat mengenali secara otomatis produk hadiah yang harus diberikan kepada pelanggan.



Gambar 3.20 Test run mobile voucher type A

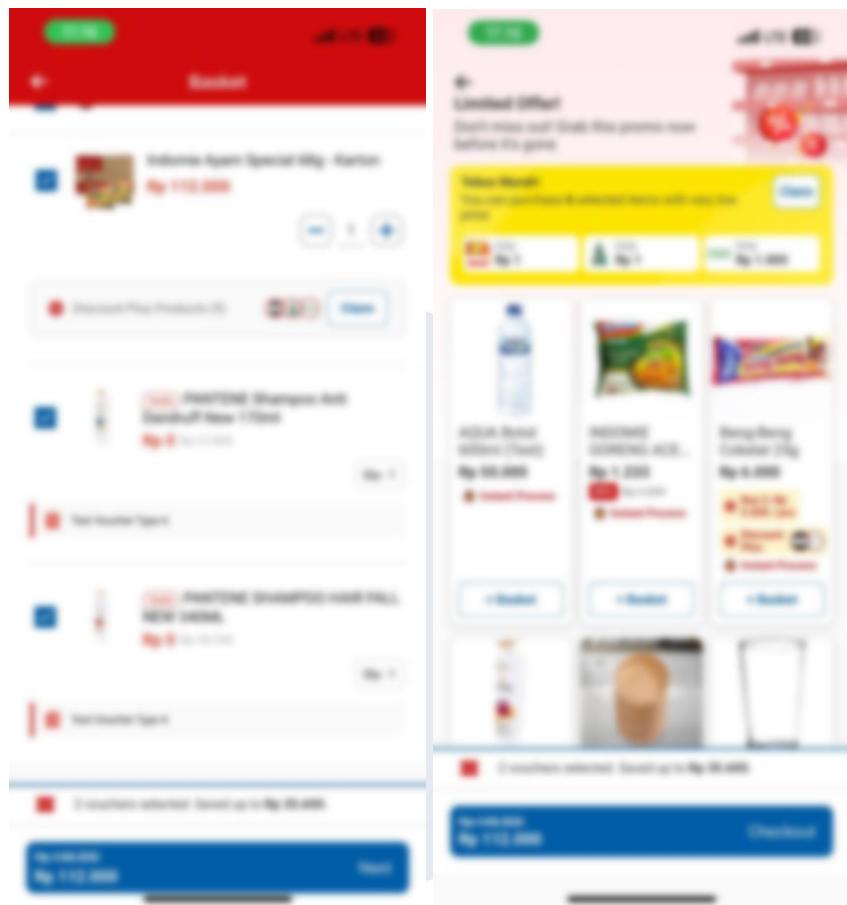
Tahap selanjutnya adalah memvalidasi pengalaman pengguna secara langsung pada perangkat *mobile* guna memastikan integrasi data dari *Back Office* berjalan mulus di tangan konsumen. Pengujian pada sisi aplikasi mencakup **309 skenario** dengan tingkat penyelesaian dan kelulusan **100%**. Rincian pengujian mendalam ini mencakup:

- Manajemen Modul My Voucher: Penulis memverifikasi bahwa berbagai status voucher tampil dengan benar sesuai kondisi aktualnya, mencakup voucher baru, voucher yang akan aktif pada hari berikutnya (H+1), serta voucher yang telah kedaluwarsa atau sudah digunakan.
- Targeting dan Visibilitas: Pengetesan dilakukan untuk memastikan voucher hanya muncul pada akun pelanggan yang menjadi target (*targeted member*) sesuai dengan pengaturan yang telah dibuat pada sistem *Back Office*.
- Logika Penukaran (*Redemption*): Penulis melakukan simulasi penukaran produk hadiah secara gratis untuk berbagai kategori, mulai dari voucher belanja reguler hingga kupon undian, guna memastikan

sistem memberikan manfaat yang sesuai dengan Syarat dan Ketentuan Promo.

- Mekanisme Keranjang Belanja (*Basket*): Validasi krusial dilakukan untuk memastikan produk hadiah terpilih secara otomatis (*auto-select*) dan masuk ke keranjang segera setelah voucher sukses diaplikasikan.
- Sinkronisasi Stok Real-Time: Penulis memastikan aplikasi mampu menampilkan pesan peringatan dan secara otomatis melepas voucher jika stok produk hadiah di toko fisik terkait dinyatakan kosong (*Out of Stock*), guna mencegah kegagalan transaksi di akhir proses.
- Validasi Metode Pengiriman: Pengujian mencakup pengecekan kesesuaian voucher terhadap metode pengiriman yang dipilih pelanggan, baik melalui layanan pengiriman ke rumah (*Delivery*) maupun pengambilan mandiri di gerai (*Pick Up*).





Gambar 3.21 Cart Voucher type A

Berdasarkan dokumentasi hasil pengujian pada antarmuka aplikasi Alfagift, berikut adalah penjelasan teknis mengenai implementasi Voucher Tipe A (Produk Gratis) sebagaimana yang ditampilkan pada kedua foto tersebut:

- Indikator Produk Hadiah: Pada halaman keranjang belanja (*Basket*), produk yang didapatkan secara cuma-cuma melalui voucher ini ditandai dengan label merah bertuliskan "Gratis" di atas nama produk.
- Nominal Transaksi Produk: Harga produk hadiah secara otomatis disesuaikan menjadi Rp 0, sementara harga asli produk tetap dicantumkan dalam format coret sebagai pembanding nilai.
- Keterangan Voucher: Di bawah rincian produk hadiah, sistem menampilkan label informasi "Test Voucher Type A" yang disertai ikon kado merah untuk

menegaskan bahwa produk tersebut merupakan *benefit* dari voucher yang sedang digunakan.

- Fitur Penukaran (*Claim*): Aplikasi menyediakan tombol "Claim" pada bagian *Discount Plus Products* yang memungkinkan pelanggan untuk melihat atau mengambil keuntungan tambahan lainnya yang tersedia.
- Ringkasan Penghematan: Pada bagian bawah layar (atas tombol *Checkout*), terdapat bilah informasi berwarna biru yang mencantumkan jumlah voucher yang diaplikasikan, misalnya "2 vouchers selected", beserta total nilai penghematan yang didapatkan pelanggan, contohnya "Saved up to Rp 35.600".
- Tampilan Penawaran Terbatas: Pada halaman *Limited Offer*, sistem menampilkan fitur "Tebus Murah!" yang memungkinkan pelanggan membeli produk pilihan dengan harga sangat rendah (seperti Rp 1 atau Rp 1.000) sebagai bagian dari ekosistem promosi yang terintegrasi dengan voucher.
- Informasi Stok dan Status: Setiap produk pada halaman penawaran dilengkapi dengan status proses pengiriman, seperti label "Instant Process", guna memberikan kepastian waktu kepada pelanggan sebelum melakukan *Checkout*.

Visualisasi ini membuktikan bahwa sistem telah berhasil memisahkan pencatatan antara produk reguler yang berbayar (seperti Indomie Ayam Special seharga Rp 112.000) dengan produk *benefit* Voucher Tipe A yang bernilai nol.

3.3.1 Kendala yang Ditemukan

Selama pelaksanaan program magang di PT Global Loyalty Indonesia, penulis mengidentifikasi beberapa tantangan dan kendala yang memengaruhi efisiensi serta kelancaran proses penjaminan kualitas. Kendala-kendala tersebut dapat dikategorikan sebagai berikut:

- a) Adaptasi Operasional dan Manajemen Waktu: Kendala awal yang dihadapi adalah lokasi perusahaan yang cukup jauh dari tempat tinggal penulis, sehingga memerlukan penyesuaian intensif dalam manajemen waktu perjalanan agar tetap produktif selama jam operasional kantor. Selain itu, karena penulis menangani tiga proyek besar secara paralel dalam kurun waktu enam bulan, pembagian fokus antara perancangan skenario baru dan pelaksanaan pengujian rutin menjadi tantangan tersendiri.
- b) Kompleksitas Teknis dan Struktur *Database*: Pemahaman terhadap struktur dan intervensi *database* yang kompleks terkadang menimbulkan hambatan dalam memastikan validitas data hasil pengujian. Hal ini diperumit oleh logika bisnis yang sangat detail pada fitur strategis, seperti proyek *Auto Apply Voucher*, yang memerlukan ketelitian ekstra dalam memetakan sisa limit, tanggal kedaluwarsa, dan prioritas voucher.
- c) Hambatan dalam Implementasi *Automation Testing*: Pelaksanaan pengujian otomatis merupakan pengalaman baru bagi penulis, sehingga proses *setup* awal serta penyusunan kode skrip menggunakan *framework* Appium memerlukan waktu belajar yang lebih lama dibanding pengujian manual. Selain itu, pengujian otomatis terkadang terhambat oleh ketidakstabilan lingkungan *staging* atau *beta* yang mengalami gangguan teknis secara mendadak.
- d) Kendala Infrastruktur dan Sinkronisasi Data: Penulis menemukan keterbatasan pada akses jaringan Wi-Fi kantor yang hanya dapat digunakan oleh satu perangkat, sehingga membatasi fleksibilitas saat harus melakukan pengujian pada beberapa gawai sekaligus. Di sisi lain, kendala sinkronisasi data stok secara *real-time* pada proyek Voucher Tipe A juga menjadi tantangan, karena adanya perbedaan data antara sistem pusat dan kondisi fisik di toko yang menghambat penyelesaian skenario pengujian.
- e) Dinamika Perubahan Spesifikasi Proyek: Adanya penyesuaian rencana atau perubahan teknis dari tim pengembang di tengah proses pengujian

menyebabkan beberapa skenario uji yang telah disusun menjadi tidak relevan (*Cancelled*). Hal ini mengharuskan penulis untuk melakukan kalibrasi ulang skenario pengujian secara berulang guna menyesuaikan dengan pembaruan sistem yang ada.

3.3.2 Solusi atas Kendala yang Ditemukan

Untuk mengatasi berbagai kendala yang dihadapi selama masa magang, penulis mengambil langkah-langkah solutif guna menjaga produktivitas dan kualitas hasil pekerjaan. Adapun solusi yang diterapkan adalah sebagai berikut:

- a) Manajemen Waktu dan Skala Prioritas: Menanggapi jarak tempuh yang jauh, penulis melakukan manajemen waktu yang lebih disiplin dengan memulai perjalanan lebih awal untuk memastikan kehadiran tepat waktu. Dalam menangani proyek paralel, penulis menyusun skala prioritas harian menggunakan daftar tugas (*to-do list*) agar fokus antara penyusunan skenario dan eksekusi *testing* tetap seimbang sesuai dengan target rilis perusahaan.
- b) Pendalaman Teknis dan Eksplorasi Mandiri: Guna mengatasi kompleksitas struktur *database* dan logika bisnis pada fitur seperti *Auto Apply*, penulis melakukan studi literatur mandiri serta aktif berdiskusi dengan mentor teknis. Penulis juga memanfaatkan alat bantu seperti DBeaver dan MongoDB secara intensif untuk memverifikasi data secara langsung pada sisi *Back-End*, sehingga validitas pengujian dapat dijamin meskipun logika fitur sangat mendalam.
- c) Optimalisasi Pembelajaran *Automation Testing*: Menghadapi tantangan pada *automation testing*, penulis mengalokasikan waktu khusus untuk mempelajari *framework* Appium dan bahasa pemrograman Java melalui dokumentasi internal perusahaan serta tutorial pendukung. Penulis juga menerapkan pola *Page Object Model* (POM) untuk mempermudah pemeliharaan skrip otomatisasi di masa mendatang.

- d) Pemanfaatan Infrastruktur secara Efisien: Untuk keterbatasan akses jaringan Wi-Fi, penulis melakukan rotasi penggunaan perangkat secara bergantian sesuai dengan prioritas pengujian yang sedang berjalan. Selain itu, terkait kendala sinkronisasi stok pada Voucher Tipe A, penulis melakukan koordinasi intensif dengan tim operasional dan melakukan pengecekan data melalui *Back Office* secara berkala untuk menyelaraskan status stok antara sistem dan aplikasi.
- e) Adaptasi terhadap Dinamika Proyek: Dalam menghadapi perubahan spesifikasi fitur, penulis menerapkan prinsip pengujian yang adaptif dengan melakukan pembaruan dokumen skenario sesegera mungkin setelah mendapatkan informasi pada sesi *Daily Update* atau *Sprint Refinement*. Hal ini memastikan bahwa meskipun terdapat skenario yang dibatalkan, proses pengetesan tetap berjalan pada versi sistem yang paling relevan.
- f) Komunikasi Aktif lintas Divisi: Penulis memanfaatkan saluran komunikasi seperti Telegram dan pertemuan rutin melalui Google Meet untuk melaporkan setiap kendala teknis atau *bug* yang ditemukan secara *real-time*. Koordinasi yang cepat ini memungkinkan hambatan pada lingkungan *staging* atau *beta* dapat segera ditindaklanjuti oleh tim infrastruktur dan pengembang.

