

## BAB III

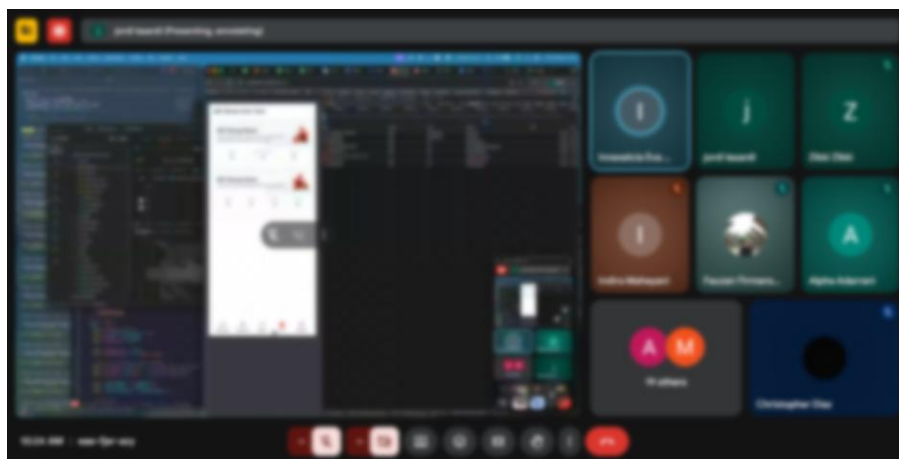
### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

Selama menjalani program magang di PT Global Loyalty Indonesia, mahasiswa menempati posisi sebagai Quality Assurance Intern dan bergabung dalam tim QA yang berada di bawah departemen Product Operation. Tim ini dipimpin oleh Bapak Benny Chandra dan karyawan magang dibimbing langsung oleh mentor, Kak Rani Wardhani. Secara keseluruhan, tim QA terdiri dari 26 anggota, yang mencakup 1 manager sebagai penanggung jawab dan pengarah kegiatan departemen, 16 karyawan QA IT, 3 QA spesialis *automation*, serta 7 karyawan magang. Karyawan magang memiliki fokus utama pada proses *quality assurance* terhadap aplikasi internal, termasuk menyusun skenario pengujian dan melakukan validasi fitur sebelum aplikasi dirilis dan siap digunakan oleh para kru. Skenario pengujian ini memuat langkah-langkah detail, *data input*, *expected result*, serta status pengujian (*passed/failed*) yang menjadi acuan dalam proses *testing*. Pengerjaannya memerlukan pemahaman mendalam terhadap alur dan fungsi fitur. Setelah pengujian dilakukan, temuan seperti *bug* atau *error* dilaporkan kepada tim *developer* untuk dilakukan *fixing* serta *product manager* sebagai bentuk pelaporan. Komunikasi aktif dengan tim menjadi bagian penting dari proses ini, agar kualitas aplikasi tetap terjaga dan dapat memberikan pengalaman pengguna yang optimal.

Koordinasi dan komunikasi dalam tim dilakukan secara fleksibel namun tetap terorganisir dengan baik. Dalam kegiatan sehari-hari, sebagian besar anggota tim menggunakan aplikasi Telegram sebagai sarana komunikasi utama, baik melalui grup tim maupun *chat* pribadi, untuk membahas berbagai isu yang berkaitan dengan *project* secara cepat dan responsif. Selain itu, komunikasi juga difasilitasi melalui rapat rutin dan sesi pembaruan yang dijadwalkan secara berkala menggunakan *platform* Google Meet, dengan tujuan menjaga konsistensi, akuntabilitas, dan sinkronisasi kerja antar anggota tim.

Beberapa bentuk koordinasi yang diterapkan di tim QA PT Global Loyalty Indonesia adalah kegiatan *daily update* dan *weekly meeting* yang dilakukan secara rutin untuk memastikan kelancaran dan keteraturan alur kerja tim. Kegiatan *daily update* diadakan setiap hari kerja, di mana setiap anggota tim menyampaikan perkembangan pekerjaan dari hari sebelumnya serta rencana aktivitas yang akan dijalankan pada hari tersebut. Sesi ini juga berfungsi sebagai wadah untuk membahas kendala atau hambatan (*blocker*) yang dihadapi selama proses pengerjaan *project*, serta melakukan tindak lanjut terhadap *bug* yang telah ditemukan namun belum diperbaiki, guna memastikan setiap isu terus dipantau hingga terselesaikan. Selain itu, tim juga melaksanakan *weekly meeting* setiap hari rabu untuk memantau progres *project* dan memberikan pembaruan capaian kerja dari masing-masing anggota. Dalam pertemuan ini, anggota divisi Quality Assurance menyampaikan laporan singkat mengenai hasil kerja selama sepekan kepada manager tim, mendiskusikan rencana kerja untuk minggu berikutnya, serta membahas permasalahan yang belum terselesaikan. *Weekly meeting* juga menjadi forum bagi manager tim untuk menyampaikan informasi dan pembaruan terkait *project* yang akan datang, sehingga seluruh anggota tetap selaras terhadap tujuan dan prioritas tim.



Gambar 3.1 *Meeting Sprint Project*

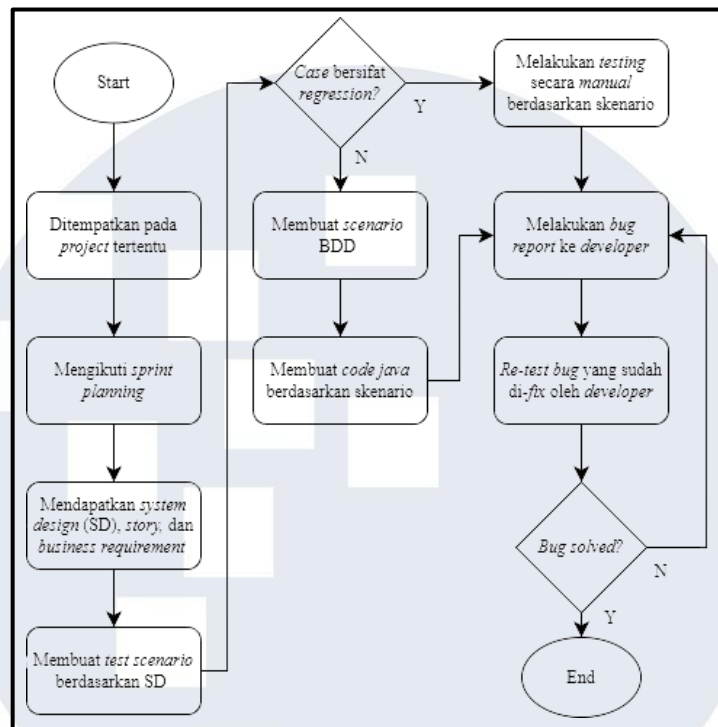
Sementara itu, untuk koordinasi terkait pada *project* tertentu dapat dilakukan melalui sesi *sprint planning*, yang umumnya dilaksanakan setiap kali sebuah *project* dimulai atau memasuki *milestone* tertentu. Gambar 3.1 merupakan contoh

*meeting sprint* pada *project* yang berkaitan dengan aplikasi internal ketika tercapai *milestone* tertentu. *Sprint* dilakukan beberapa kali dalam satu *project*, bergantung pada tingkat kompleksitas dan durasi pengerjaan. Dalam sesi ini, tim membahas sistem desain *project*, termasuk penentuan ruang lingkup pekerjaan untuk setiap divisi yang terlibat. Selain itu, diskusi juga mencakup estimasi effort dan timeline *project*, serta pembahasan teknis seperti penggunaan asset, kebutuhan tambahan, dan penyesuaian terhadap *requirement* yang telah ditetapkan sebelumnya secara terbuka.

### 3.2 Tugas yang Dilakukan

Selama menjalani program magang di PT Global Loyalty Indonesia (GLI), karyawan magang ditempatkan pada divisi Product Operation, tim QA untuk berkontribusi dalam memastikan kualitas aplikasi sebelum dirilis dan siap digunakan. Tujuan utama dari kegiatan ini adalah untuk memahami alur pengujian fitur aplikasi, menyusun skenario pengujian (*test scenario*), serta melakukan validasi terhadap fitur yang sedang dikembangkan.





Gambar 3.2 *Workflow Quality Assurance Diagram*

Selama pelaksanaan kerja magang, karyawan magang mempelajari bagaimana tim QA bekerja, seperti ketika *sprint project* dimulai, pengenalan dengan *system design*, *storylist*, dan beberapa *business requirement* yang perlu diperhatikan jika adanya syarat atau perubahan *flow* sebuah fungsi, serta melakukan pengujian fitur dan pelaporan *bug*. Karyawan magang juga dituntut untuk dapat berdiskusi bersama tim lain, seperti tim *developer* dan tim *design UI/UX* dalam menganani *bug*. Hal ini bertujuan untuk memastikan perbaikan dilakukan dengan tepat dan efisien. Proses tersebut tidak hanya melatih kemampuan komunikasi dan kerja sama tim, tetapi juga memperkuat pemahaman terhadap *workflow* QA yang dinamis seperti Gambar 3.2.

Dalam setiap *project* yang dijalankan, karyawan magang diharapkan memahami dokumen acuan seperti *system design* dan *story list*, mampu menyusun *test scenario* serta *test case* secara rinci berdasarkan desain Figma dan *business requirement*, hingga melakukan *test run* sesuai skenario yang telah ditetapkan. Pengujian yang tergolong repetitif dikategorikan sebagai *regression test*, dimana jenis pengetesan ini dilakukan secara *automation* menggunakan *software* Appium

dan Android Studio. Tabel 3.1 adalah rincian *project* serta detail pekerjaan yang dilakukan selama proses kerja magang:

Tabel 3.1 Rincian *Project* dan Detail Pekerjaan

No.	Nama Kegiatan/ <i>Project</i>	Periode Kerja	Deskripsi Pekerjaan
1.	<i>Setup Software dan Database</i>	4 – 6 Agustus 2025	Menyiapkan <i>environment</i> , <i>software</i> , serta <i>database</i> , termasuk instalasi dan konfigurasi sistem untuk mendukung proses QA
2.	<i>Daily Task Update</i>	Selama <i>project</i> berlangsung	Membuat laporan singkat tentang pekerjaan yang dilakukan setiap hari kerja
3.	<i>Project Dark Store</i> sebagai Hub Tag-I	7 – 12 Agustus 2025	Membuat skenario tambahan untuk modul pesanan dan MAT
		15 – 28 Agustus 2025	Melakukan pengujian terhadap skenario tambahan melalui <i>testrun</i> pada modul terkait secara <i>manual testing</i> dan QA <i>pass</i> tanggal 28 Agustus 2025
4.	<i>Project Tracking Order Live Delivery</i> SAPA	25 Agustus – 3 September 2025	Membuat skenario pengujian untuk penggunaan GPS, modul halaman pesanan untuk proses pengantaran pesanan
		5 September – 9 Oktober 2025	Memulai <i>testrun</i> untuk <i>project</i> ini dan melakukan pengerjaan pada modul serta melakukan pelaporan bug kepada developer. QA <i>pass</i> per tanggal 7/8 Oktober 2025
5.	<i>Project Revamp: Access Role dan Functional</i>	21 September – 29 Oktober 2025	Proses pembuatan skenario berdasarkan modul yang dibagi untuk <i>testing</i> , sekaligus membuat <i>script automation</i> untuk <i>automation testing</i> pada modul regresi <i>homepage</i>
		29 Oktober – 27 November 2025	Melakukan <i>testrun</i> serentak baik <i>automation</i> dan <i>manual testing</i> berdasarkan skenario yang sudah dibuat serta analisis <i>compability system</i> melalui <i>smoke test</i>

### 3.3 Uraian Pelaksanaan Kerja Magang

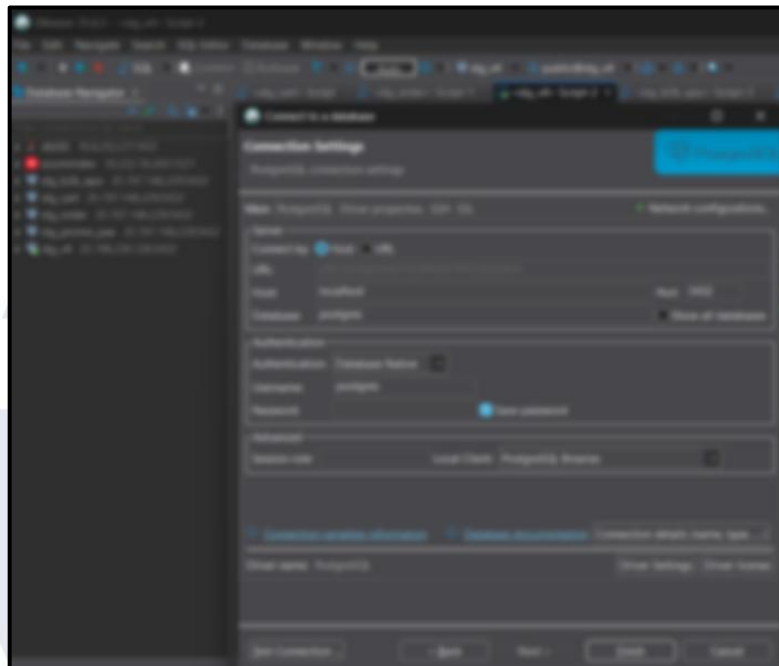
#### 3.3.1 Proses Pelaksanaan

Pada bagian ini akan dibahas penjelasan tentang pekerjaan yang dilakukan secara mendetail selama proses kerja magang di PT Global Loyalty Indonesia, seperti pada tabel 3.1.

##### 3.3.1.1 Setup Software dan Database

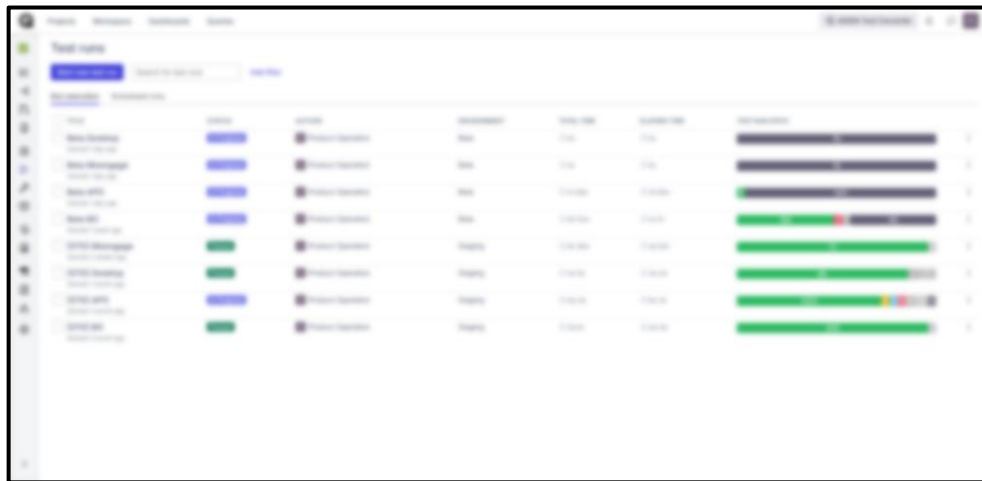
Sebelum terlibat secara langsung dalam pengerjaan project, karyawan magang terlebih dahulu melalui tahapan persiapan berupa *setup software* serta pengenalan terhadap *tools* yang digunakan oleh tim QA dalam aktivitas sehari-hari. Salah satu *tools* utama yang digunakan adalah DBeaver, yaitu sebuah *database management system* (DBMS) yang berfungsi untuk mengakses, memantau, dan memeriksa data pada sistem *internal* perusahaan [15]. Melalui DBeaver, tim QA dapat melakukan pengecekan data secara *real-time* yang digunakan oleh aplikasi, khususnya dalam proses validasi kesesuaian antara *input* dan *output* suatu fitur. Untuk dapat menggunakan *tools* ini, karyawan magang perlu melakukan instalasi aplikasi dan menghubungkannya ke database *internal* perusahaan dengan memasukkan kredensial berupa *username* dan *password* pada host yang telah ditentukan. Tahapan *setup* ini sekaligus menjadi sarana awal dalam memahami struktur data serta skema tabel yang digunakan pada sistem *backend*. Gambar 3.3 menampilkan contoh tampilan proses konfigurasi koneksi database menggunakan DBeaver.





Gambar 3.3 Tampilan *Setup Software Database DBeaver*

Selain itu, terdapat *tools* khusus untuk manajemen *testing* yang digunakan oleh tim QA, yaitu QASE [16]. Melalui *platform* ini, karyawan magang mulai dibiasakan dengan alur kerja pengujian, mulai dari penyusunan *test case*, pelaksanaan *test run*, hingga pencatatan hasil pengujian yang diklasifikasikan ke dalam beberapa status seperti *passed*, *failed*, *cancelled*, maupun *skipped*. Setiap status tersebut memiliki makna tersendiri yang digunakan sebagai indikator keberhasilan fitur atau sebagai sinyal perlunya dilakukan proses *bug fixing*. Selain itu, QASE menyediakan struktur pengelolaan skenario pengujian yang tertata berdasarkan modul aplikasi, sehingga proses dokumentasi dan pelacakan pengujian dapat dilakukan secara konsisten antar project. Gambar 3.4 menampilkan tampilan antarmuka situs *web* QASE yang digunakan sebagai media utama dalam manajemen pengujian.



Gambar 3.4 Tampilan Website QASE

Dalam proses pelaporan *bug*, tim QA memanfaatkan *platform* Jira sebagai media utama untuk mencatat setiap temuan *error* atau kendala yang muncul selama pengujian [17], [18]. Melalui Jira, setiap *bug* didokumentasikan secara terperinci, mulai dari *step to reproduce*, *expected result*, hingga *actual result*, sehingga alur permasalahan dapat dipahami dengan jelas. Karyawan magang juga dibekali pemahaman mengenai cara menyusun laporan *bug* yang terstruktur dan komunikatif agar memudahkan tim *developer* maupun tim *testing operation* dalam melakukan analisis dan perbaikan. Pendokumentasian *bug* yang rapi dan sistematis di Jira menjadi aspek yang krusial, karena seluruh laporan tersebut akan dijadikan referensi utama dalam proses bug fixing, sebagaimana ditunjukkan pada tampilan pada Gambar 3.5.



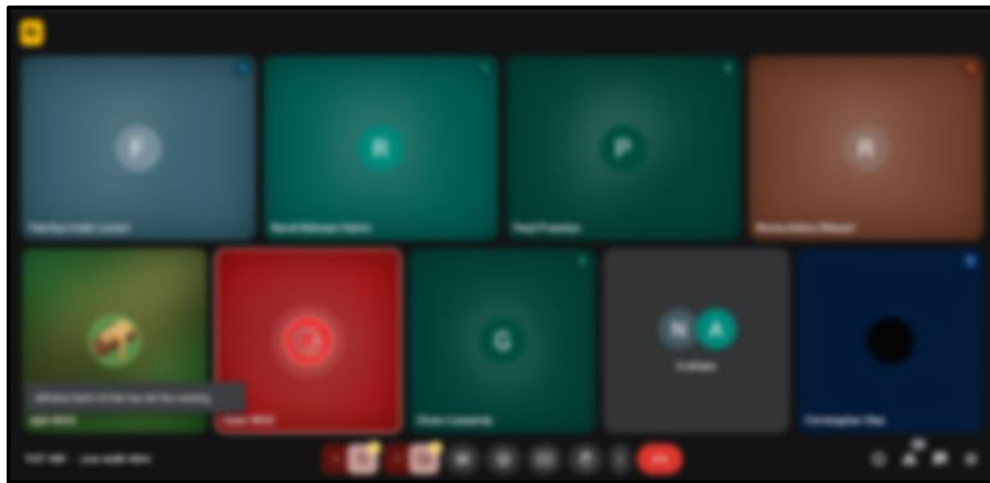


Gambar 3.5 Tampilan *Website* Jira

Selain pengujian dari sisi fungsional, karyawan magang juga diperkenalkan dengan penggunaan *tools* pendukung *system design*, seperti Figma dan Miro. Figma dimanfaatkan sebagai referensi utama *user interface* (UI) untuk setiap fitur yang sedang dikembangkan, sehingga karyawan magang dapat meninjau tampilan aplikasi per halaman secara interaktif sekaligus memahami detail komponen visual yang perlu divalidasi saat pengujian UI. Di sisi lain, Miro digunakan sebagai media kolaboratif untuk melihat dan memahami rancangan UI pada sistem *back-office* yang nantinya digunakan oleh *admin* pada versi *production*. Kehadiran kedua *tools* ini memberikan konteks yang lebih menyeluruh terhadap fitur yang diuji, terutama dalam memastikan bahwa hasil implementasi pada aplikasi telah selaras dengan desain yang direncanakan sejak tahap awal.

#### 3.3.1.2 Daily Task Update

Dalam pelaksanaan *project*, karyawan magang turut menjalani mekanisme pelaporan rutin yang berperan penting dalam menjaga alur kerja tim. Seperti ditunjukkan pada Gambar 3.6, kegiatan ini dilakukan melalui pertemuan *online* setiap pagi menggunakan Google Meet yang dihadiri oleh Project Manager serta seluruh anggota tim terkait. Pada sesi tersebut, setiap anggota tim menyampaikan pembaruan mengenai *progress* pekerjaan yang telah dicapai, rencana tugas yang akan dikerjakan pada hari berjalan, serta hambatan atau kendala yang ditemui. Melalui koordinasi harian ini, komunikasi antaranggota tim dapat terjaga dengan baik, sehingga pelaksanaan *project* tetap selaras dengan jadwal dan *target* yang telah ditetapkan.



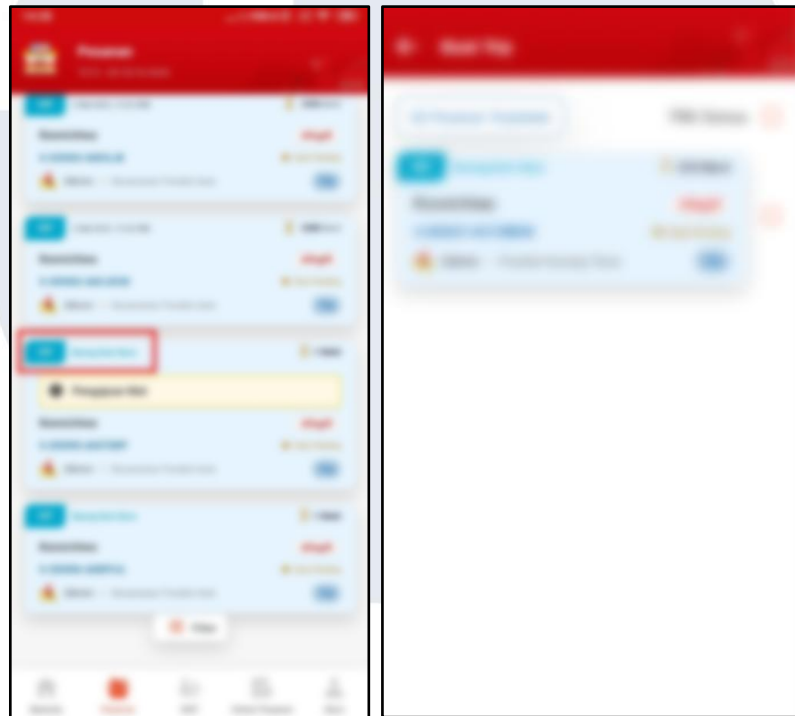
Gambar 3.6 Proses Pelaporan Daily Update

Di luar sesi pertemuan yang membahas perkembangan *project*, karyawan magang juga diwajibkan menyampaikan laporan aktivitas harian kepada manajer tim atau *supervisor* melalui Whatsapp. Laporan tersebut disusun dalam bentuk ringkasan singkat yang mencakup pekerjaan yang telah dilakukan, baik yang berkaitan langsung dengan tugas *project* maupun aktivitas pendukung lainnya, seperti sesi *sharing session* eksplorasi sistem, atau proses pengenalan *tools* baru. Melalui mekanisme pelaporan ini, *supervisor* dapat memantau progres kerja karyawan magang secara individual serta memberikan arahan atau masukan tambahan apabila diperlukan. Praktik pelaporan harian ini selaras dengan pendekatan *Agile* yang menekankan keterbukaan informasi dan komunikasi berkelanjutan, sehingga koordinasi tim dapat berjalan lebih terstruktur dan efisien.

### 3.3.1.3 Project Dark Store Sebagai Hub Tag-I

Setelah proses *setup tools* dan *database* selesai, karyawan magang ditugaskan pada *project Dark Store* Sebagai Hub Tag-I, pada minggu pertama magang *track 2* dimulai. Sebelumnya, aplikasi internal pengiriman merupakan aplikasi milik Alfamart yang digunakan oleh kru SAPA dalam menjalankan operasional pengantaran pesanan pelanggan. Aplikasi ini menjadi bagian dari layanan SAPA (Siap Antar Pesanan

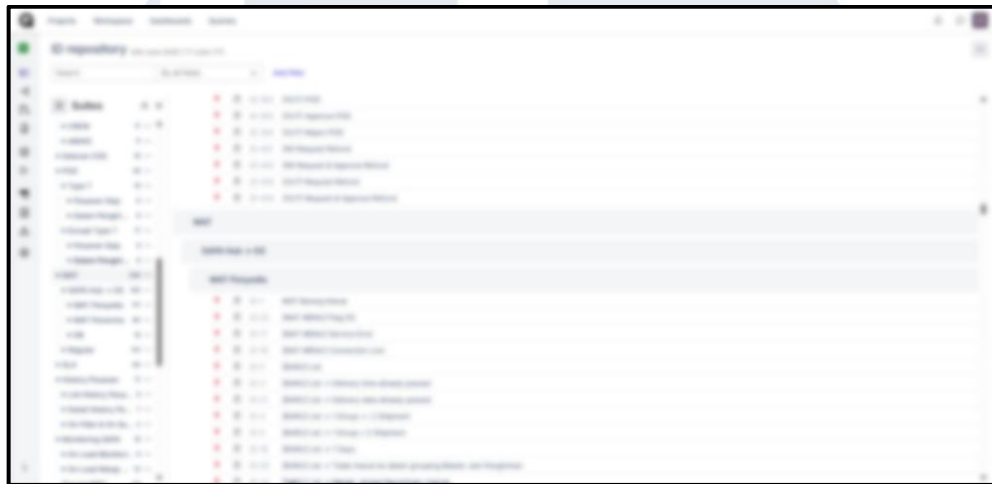
Anda), yaitu layanan pesan antar resmi untuk melayani pesanan produk dari toko Alfamart melalui aplikasi Alfagift dan mengantar pesanan mereka langsung ke lokasi tujuan. Melalui layanan SAPA, Alfamart ingin menciptakan pengalaman belanja yang lebih mudah dan praktis tanpa mengharuskan pelanggan datang ke toko secara langsung.



Gambar 3.7 Pesanan dengan Produk Tag-I (Barang *Dark Store*)

Dalam konteks *Project Dark Store Hub Tag I*, fokus utama pengembangannya adalah meningkatkan efisiensi proses pengantaran serta distribusi produk melalui optimalisasi sistem aplikasi internal. *Project* ini mencakup integrasi titik distribusi yang disebut *Dark Store*, yaitu gudang distribusi yang difungsikan khusus untuk penyimpanan stok sekaligus bertindak sebagai tempat transit pengiriman yang berperan sebagai penghubung dalam sistem distribusi. Melalui penambahan mekanisme *Dark Store* pada aplikasi internal, kru SAPA dapat menerima pesanan, memproses pengiriman, serta memperbarui status pesanan lebih efisien. *Project* yang sudah berlangsung ini dapat membuat pengelolaan pengantaran lebih cepat, sehingga dapat meningkatkan efektivitas proses

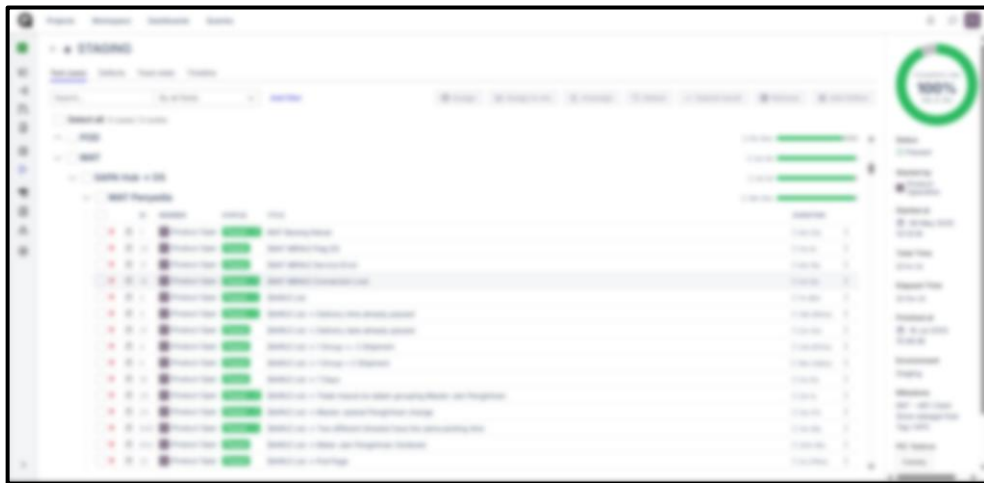
pemenuhan pesanan pelanggan. Gambar 3.7 merupakan contoh daftar pesanan yang akan dikirimkan melalui *Dark Store*, yang akan segera diproses langsung dalam sekali pengantaran sehingga menciptakan efektivitas maksimal dalam pengantaran pesanan.



Gambar 3.8 *Test Scenario* untuk Halaman Pesanan dan MAT

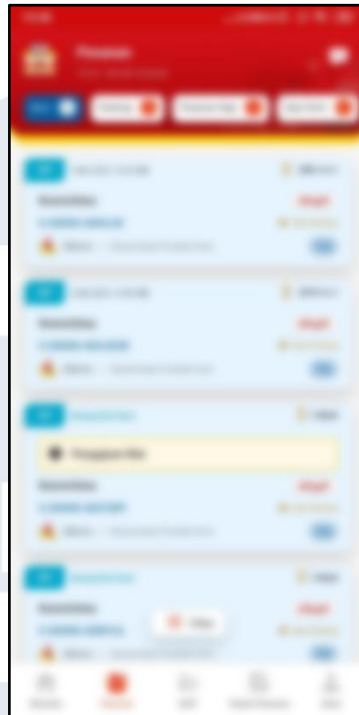
Gambar 3.8 menampilkan salah satu modul *testing* dari repositori *test scenario* yang digunakan dalam *project* ini, di mana seluruh skenario pengujian tersimpan dan dikelola secara terstruktur didalam QASE. Repositori tersebut berisi total 846 *test case* yang berfungsi untuk memastikan setiap fungsi pada sistem berjalan sesuai ekspektasi maupun dapat menangani kondisi yang tidak valid. Seluruh *test case* dalam repositori ini disusun secara sistematis berdasarkan *story list* yang telah dijelaskan pada tahap *sprint planning*. *Story list* tersebut berisi kumpulan kebutuhan fitur serta fungsionalitas yang akan dikembangkan, sehingga menjadi acuan utama dalam proses penyusunan *scenario test* yang terarah dan konsisten dengan *business requirement*.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.9 *Test Run Project Dark Store Sebagai Hub Tag I*

Gambar 3.9 merupakan *test run* yang digunakan dalam proses pengujian aplikasi internal. Dalam tahap ini, setiap *test case* diuji secara menyeluruh untuk memverifikasi aspek fungsionalitas pesanan aplikasi, performa sistem pesanan, serta kompatibilitas aplikasi terhadap berbagai kondisi penggunaan seperti versi android, dll. Selama proses pengujian, karyawan magang turut berperan dalam melakukan *test run* untuk modul List Pesanan dari hub *Dark Store*, MAT (Mutasi Antar Toko) dan proses POD (*Proof of Delivery*) yang dilakukan ketika pesanan tiba. Ketiganya berperan penting dalam pengelolaan serta pelacakan pesanan di dalam sistem aplikasi internal. Melalui pengujian pada kedua modul tersebut, tim QA memastikan bahwa proses perpindahan stok antar toko dan pembaruan status pesanan dapat berjalan dengan lancar, sehingga mendukung kelancaran operasional kru SAPA dalam mengantarkan pesanan pelanggan dengan akurat dan efisien. QA *pass* untuk ini sebenarnya *project* telah dicapai pada bulan Juni. Namun, karena adanya kebutuhan mendesak dari pihak *stakeholder*, dilakukan penyesuaian tambahan yang mengharuskan pengujian ulang pada fitur yang baru dikembangkan. Proses tersebut kembali melalui siklus validasi hingga akhirnya mendapatkan QA *pass* pada tanggal 28 Agustus 2025, dengan total waktu kerja efektif karyawan magang selama 18 hari.

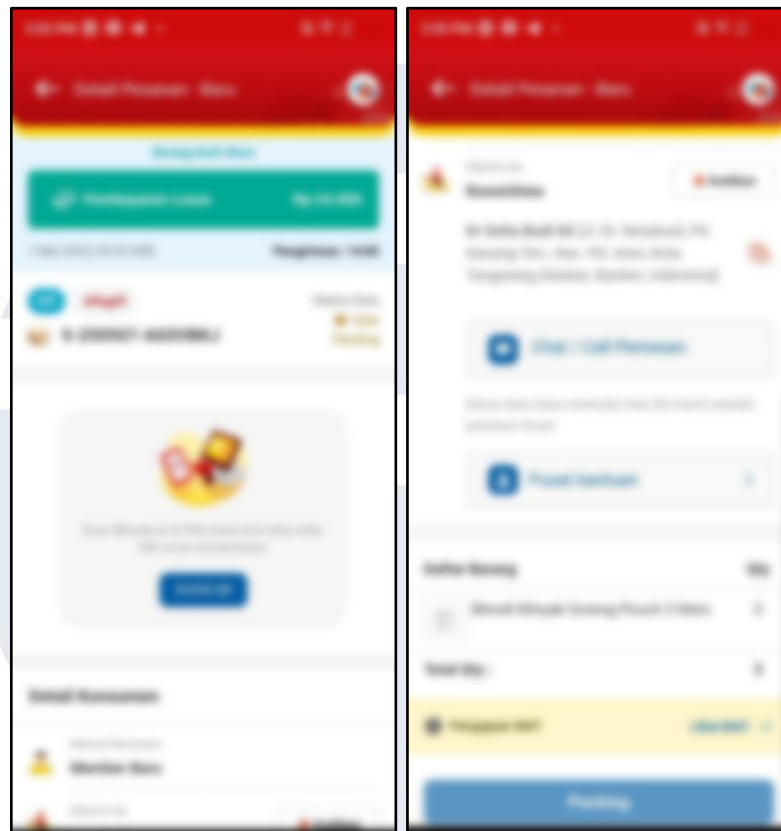


Gambar 3.10 *Testing* Modul Halaman Pesanan, List Order *Dark Store*

Salah satu pembaruan fitur yang diimplementasikan pada proyek aplikasi internal, yaitu penambahan label “Barang Dark Store” yang ditampilkan pada pesanan dari *Dask Store* seperti Gambar 3.10. Label ini muncul di bagian atas setiap *card* pesanan yang terkait, sehingga memudahkan kru SAPA dalam mengenali asal pesanan. Penerapan fitur ini bertujuan untuk membantu proses identifikasi jenis pesanan, terutama bagi kru SAPA yang bertugas menangani pengiriman barang. Dengan adanya label identifikasi ini, sistem dapat memberikan kejelasan yang lebih baik terhadap asal pesanan, sehingga proses sortir, *packing*, dan pengiriman dapat berjalan lebih efisien, terarah, dan meminimalkan potensi kesalahan dalam distribusi.

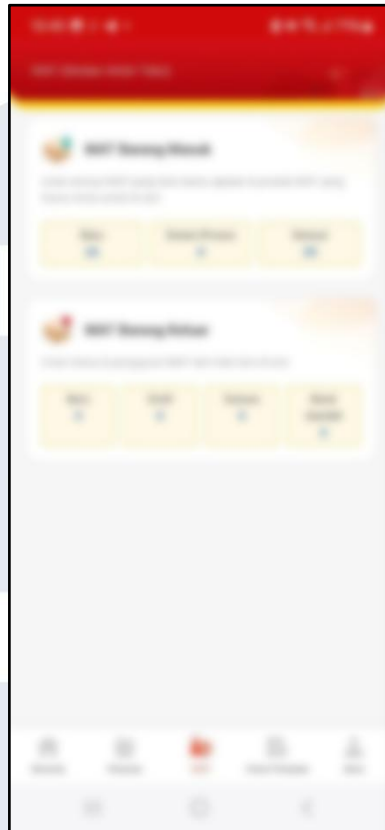
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A





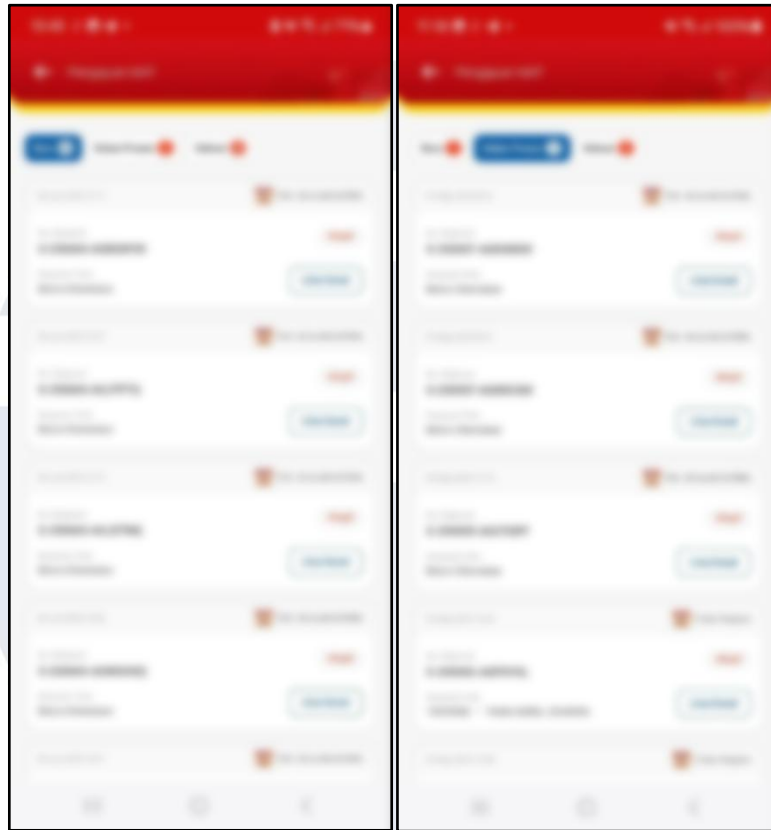
Gambar 3.11 *Testing* Modul Halaman Pesanan, Detail Order

Kru SAPA dapat memproses pesanan *Dark Store* dengan membuka detail pesanan dengan mengeklik salah satu *card* pada list pesanan. Gambar 3.11 merupakan salah satu detail pesanan yang memiliki pengiriman barang dari *Dark Store*. Setiap pesanan yang tiba pada toko SAPA (toko regular) namun berjenis barang *Dark Store* (DS) akan dilakukan proses MAT yang memungkinkan toko SAPA meminta ketersediaan stok dari *Dark Store*. Label pengajuan MAT akan muncul secara otomatis dari sistem setelah *service* membaca jenis pesanan tersebut sebagai *Dark Store*, ditambahkan status baru pada bagian bawah halaman detail pesanan yang ditandai dengan “Pengajuan MAT”. Waktu pemrosesan pesanan, yang disebut sebagai SLA (*service level agreement*) berhenti sementara ketika ajuan MAT sudah dilakukan oleh kru SAPA.



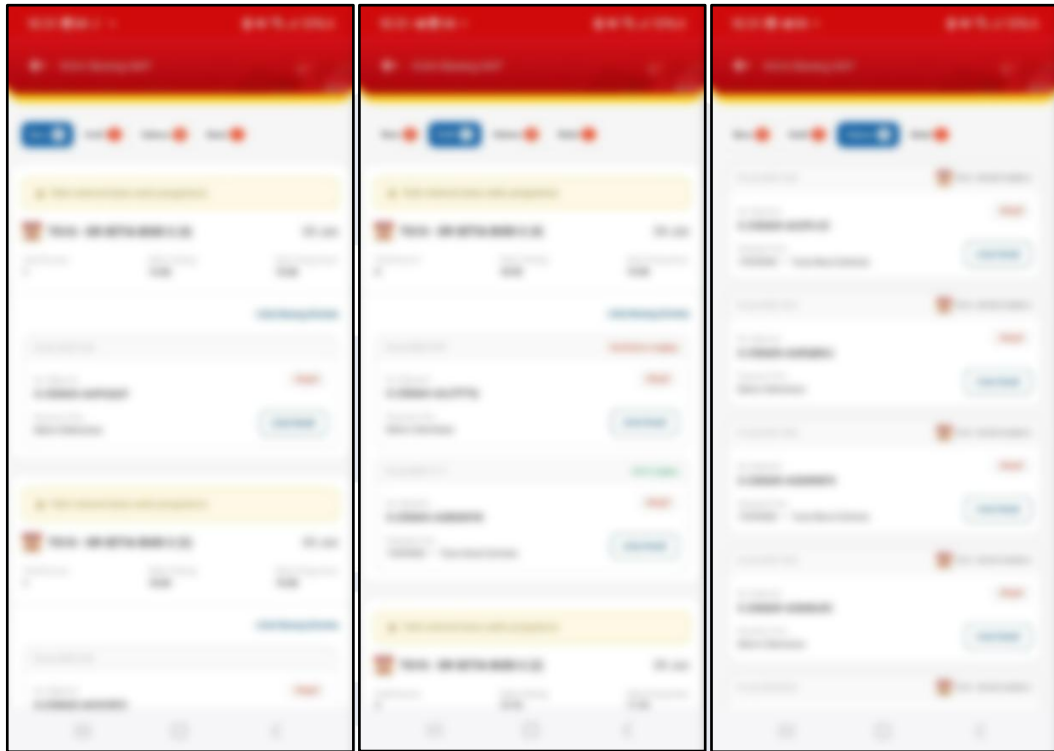
Gambar 3.12 Testing Modul Landing Page MAT

Pesanan *Dark Store* yang secara otomatis masuk ke pengajuan MAT akan menambah jumlah data MAT pada *landing page* MAT seperti pada Gambar 3.12. Sebelumnya, MAT terbagi menjadi 2 tipe, yaitu MAT barang masuk dan MAT barang keluar. Untuk setiap jenis pesanan yang diajukan karena membutuhkan stok lebih melalui MAT, akan tertampil pada data MAT barang masuk dan menambah *counter* pada bagian *card* MAT barang masuk. Sebaliknya, MAT barang keluar berisikan ajuan yang muncul dari toko lain untuk penambahan stok pada toko penerima. Dalam konteks *project* ini, pesanan *Dark Store* yang diajukan MAT oleh sistem juga akan menambah data MAT barang masuk milik toko penerima dan MAT barang keluar pada toko penyedia.



Gambar 3.13 *Testing* Modul Halaman *List MAT* Barang Masuk Per Status

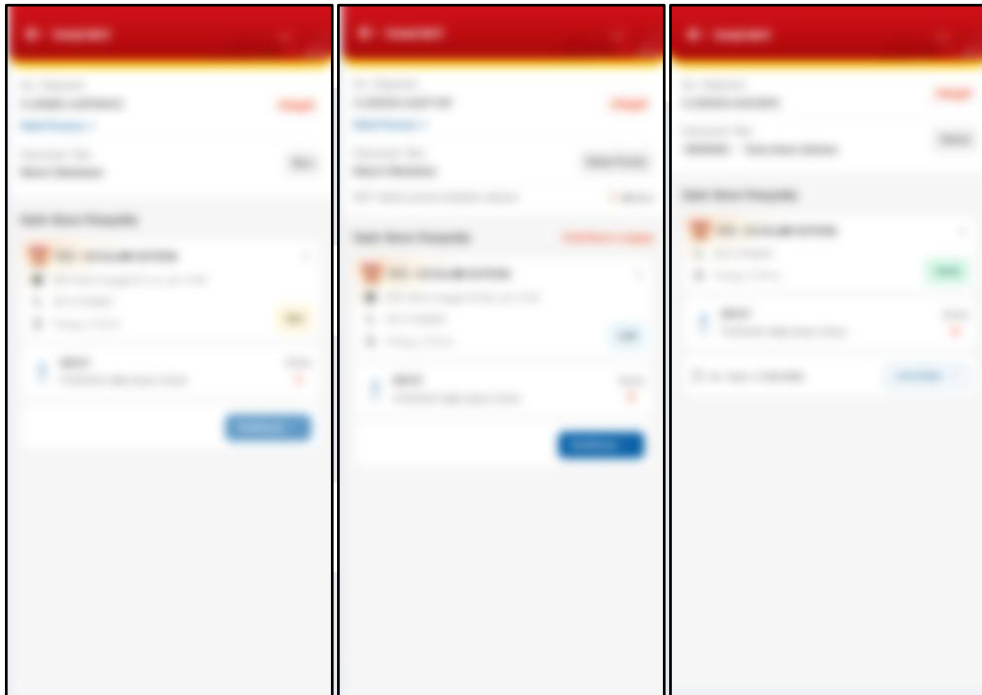
Jika salah satu *card* pada *landing page* MAT diklik, maka halaman akan diarahkan ke halaman *list* MAT seperti pada Gambar 3.13. Terdapat informasi pesanan pada setiap *list* MAT yang tampil sesuai dengan tipe MAT, seperti tanggal diajukannya MAT, nomor pengiriman, toko penyedia, dan karyawan yang mengajukannya. *Project* ini membuat sebuah pembaruan dimana jikaajuan MAT dilakukan oleh sistem maka pada *field* karyawan toko otomatis akan diisi dengan “Belum Ditentukan”, yang nantinya akan terisi bila MAT sudah dalam status “Dalam Proses”, yang merupakan proses dimana kru SAPA dari toko penyedia sedang mengirimkan stok produk yang diminta ke toko penerima. Untuk status selesai akan terjadi ketika barang kiriman MAT sudah diterima oleh toko penerima.



Gambar 3.14 *Testing* Modul Halaman *List* MAT Barang Keluar Per Status

Gambar 3.14 adalah *list* pengajuan MAT yang dilakukan pada toko penerima dari berbagai status ajuan. Pengajuan yang baru terbuat akan masuk pada *tab* status baru, MAT yang sedang dikemas pada status *draft* dan MAT yang berhasil terkirim pada status sukses. Pembaruan pada halaman MAT barang keluar terjadi pada pengelompokkan barang ajuan MAT berdasarkan toko penerima, serta penambahan detail seperti kuantitas permintaan barang, waktu pengemasan dan pengiriman, serta *list* barang yang akan dibawa oleh kru menuju toko penerima. Pada ajuan MAT yang sudah dilakukan draft maka akan muncul nama kru yang bertanggungjawab, beserta informasi kelengkapan pengemasan serta pengiriman barang,

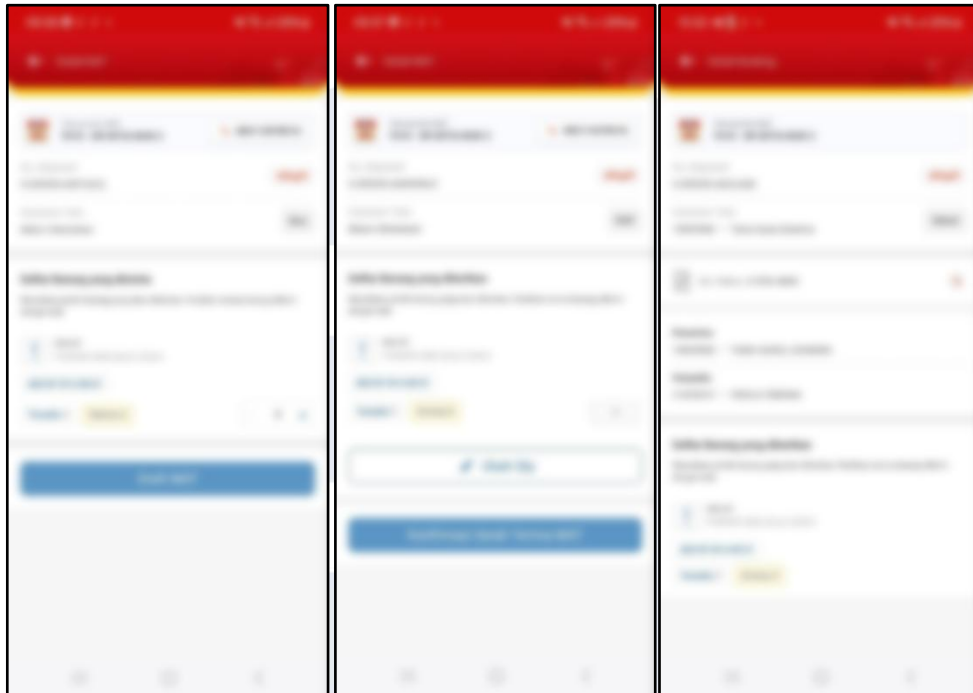
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.15 *Testing* Modul Halaman Detail MAT Barang Masuk

Data terkait detail MAT dapat ditemukan dengan mengeklik salah satu nomor pesanan yang tertera pada *list* MAT sebelumnya. Gambar 3.15 merupakan salah satu contoh detail MAT pada MAT Barang Masuk. Ketika dilakukan pengajuan MAT, sistem akan membuat (*write*) informasi yang akan ditampilkan pada detail MAT seperti nomor pengiriman, kru yang mengajukan, detail toko dimana stok diminta, produk yang diminta beserta dengan kuantitasnya. Pembaruan pada aplikasi internal di *project* ini terjadi pada bagian *labelling* setiap pesanan *dark store*. Kru SAPA dapat mengkonfirmasi ajuan MAT ini ketika barang yang diterima sudah diantarkan ke tempat toko penerima dengan baik. Proses tersebut mengubah status pesanan menjadi selesai.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

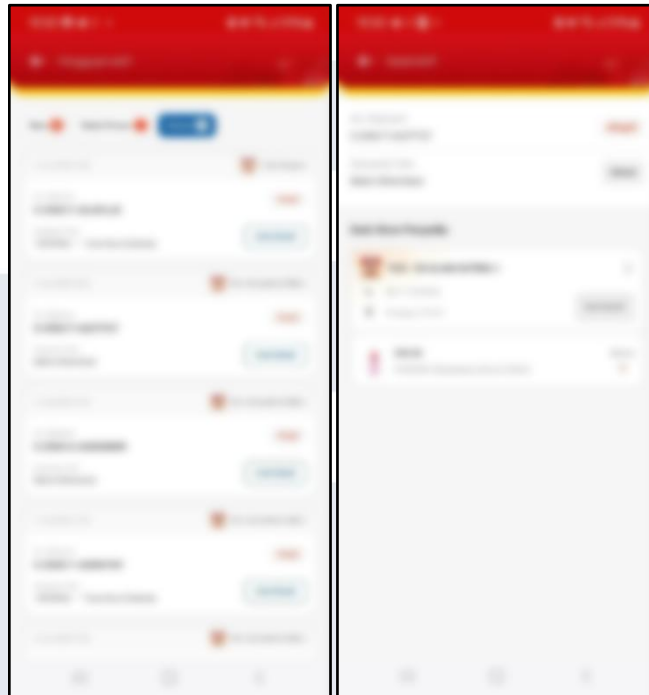


Gambar 3.16 *Testing* Modul Halaman Detail MAT Barang Keluar

Gambar 3.16 merupakan informasi detail mengenai salah satu pesanan yang diajukan kepada MAT barang keluar. Bersamaan dengan MAT barang masuk, sistem akan membuat (*write*) data seperti nomor pengiriman, kru yang mengajukan, detail toko dimana stok diminta, produk yang diminta beserta dengan kuantitasnya. Pada MAT barang keluar, kru SAPA pada toko penyedia dapat melakukan proses MAT ini seperti *packing* dan pengiriman, yang disebut status *drafting*. Saat proses berlangsung, status MAT barang masuk milik kru SAPA pada toko penerima akan berubah menjadi status dalam proses.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

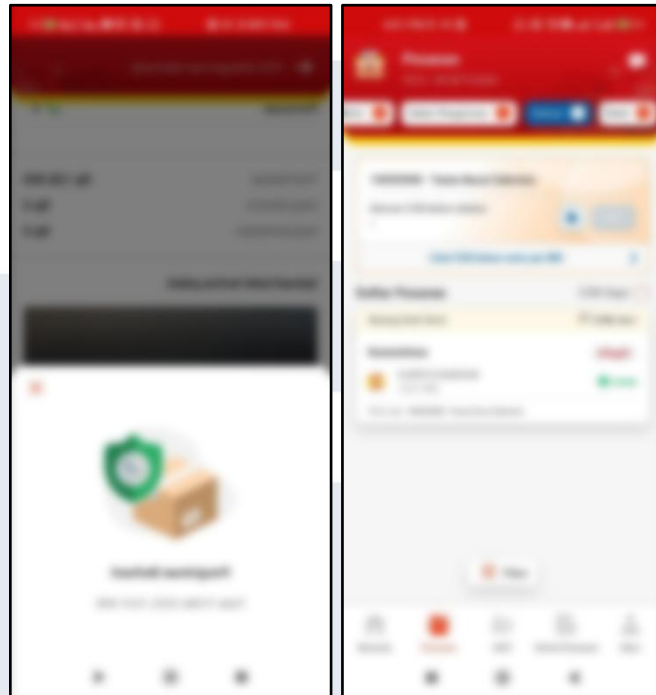




Gambar 3.17 *Testing Mekanisme Auto Cancel Order DS Jika Order > 3 Hari*

Setiap pesanan berjenis *Dark Store* yang tidak diproses selama 3 hari akan secara otomatis dibatalkan oleh sistem. Gambar 3.17 merupakan detail informasi pesanan *Dark Store* yang dibatalkan ajuan MAT-nya oleh sistem. Sama seperti proses pembatalan yang dilakukan oleh kru SAPA, status pesanan diubah menjadi “Batal”, serta status ajuan MAT pesanan DS akan berubah menjadi “Batal Diambil”. Perbedaan proses pembatalan MAT ini terletak pada nama karyawan atau kru SAPA yang tercantum, karena pesanan MAT yang berjenis *Dark Store* ini terbuat secara otomatis, maka *field* tersebut akan tetap kosong baik pada MAT barang masuk maupun keluar.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



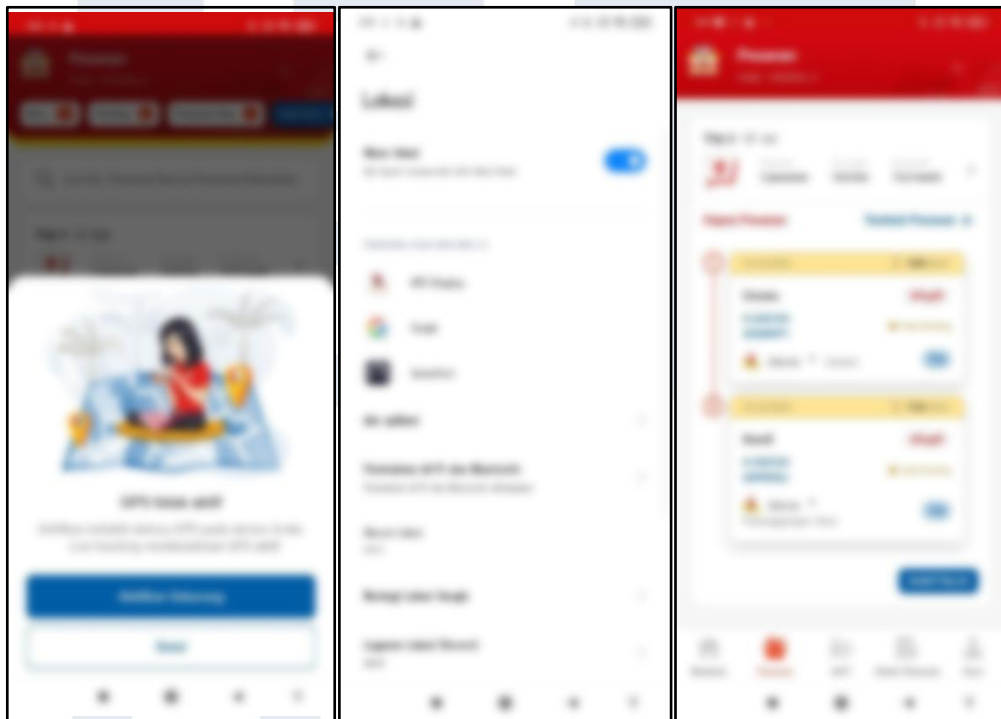
Gambar 3.18 *Testing* Fitur Konfirmasi Pengiriman–POD

Jika ajuan MAT berhasil dilakukan untuk jenis pesanan *Dark Store*, pesanan dapat diproses dan dikirimkan ke alamat tujuan pembeli, dengan melakukan konfirmasi pengiriman POD oleh kru SAPA, seperti pada Gambar 3.18. Kru SAPA dapat mengonfirmasi pesanan tersebut dengan mengisi data penerima dan mengupload bukti penerimaan barang oleh penerima pada aplikasi internal secara langsung. Jika pesanan dilakukan diluar koordinat yang sudah ditentukan maka kru SAPA wajib mengajukan POD terlebih dahulu kepada *Area Coordinator*, yang dapat menentukan apakah ajuan POD tersebut dapat disetujui atau ditolak sesuai bukti yang dikirimkan oleh kru SAPA.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

#### 3.3.1.4 Project Tracking Order Live Delivery SAPA

Pada minggu terakhir bulan Agustus, karyawan magang kembali diberikan penugasan dalam sebuah *project* baru pada aplikasi internal yang berfokus pada pengembangan fitur *tracking order live delivery* pada layanan SAPA. *Project* ini dirancang untuk melakukan proses pemantauan secara langsung terhadap pergerakan kru SAPA selama menjalankan pengantaran pesanan pelanggan, sehingga aktivitas pengiriman dapat diamati secara lebih *real-time* oleh pihak terkait.



Gambar 3.19 Proses *Live Tracking* Ketika Membuat Trip Pesanan

Gambar 3.19 merupakan proses bagaimana *live tracking* bekerja. Dengan adanya fitur ini, setiap *trip* pengiriman yang dilakukan oleh kru akan terekam dan dapat dipantau selama status pesanan berada dalam tahap “Siap Kirim” hingga “Dalam Pengiriman” dan berakhir ketika kru sudah kembali ke toko setelah menyelesaikan seluruh perjalanan pengantaran. Penerapan mekanisme ini tidak hanya bertujuan meningkatkan transparansi proses operasional, tetapi juga membantu meminimalisir potensi kesalahan manusia selama proses pengiriman,

memastikan setiap pesanan ditangani sesuai alur yang telah ditetapkan, serta meningkatkan akurasi pencatatan status pengantaran pada berbagai kondisi, termasuk saat terjadi kendala di lapangan, pengajuan pengiriman ulang, maupun perubahan status pesanan lainnya seperti retur yang tetap tercatat secara sistematis dalam aplikasi.



Gambar 3.20 *Test Scenario* Halaman Pesanan *Live Tracking*

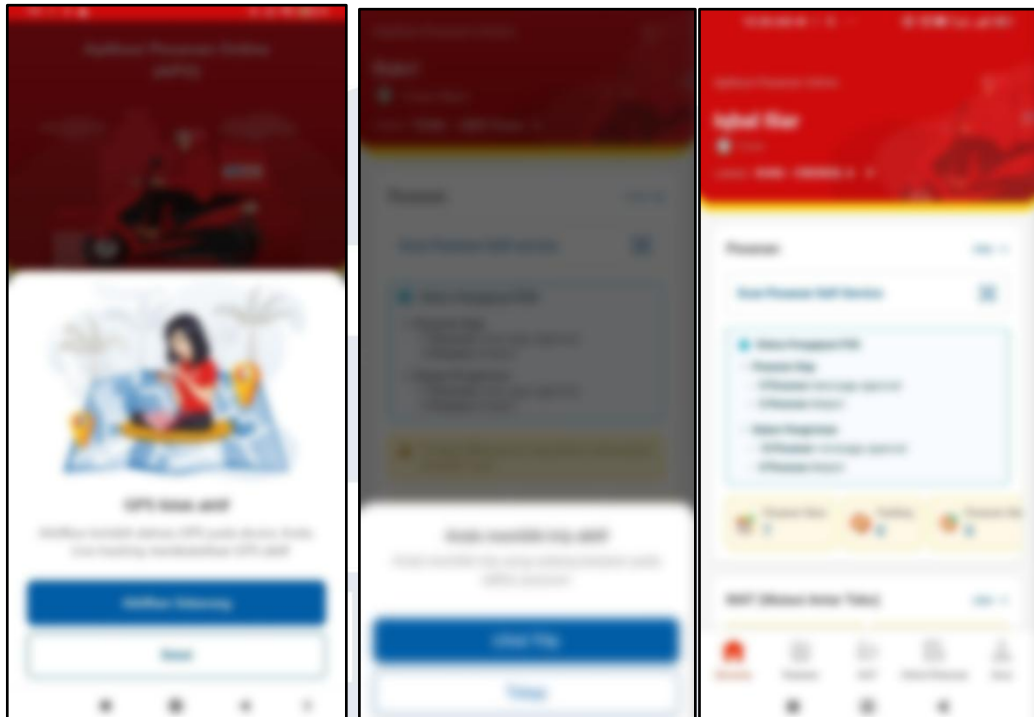
Repository *test scenario* untuk *project* ini dikelola melalui QASE, tempat seluruh *test case* disimpan dan diatur dengan rapi sesuai kebutuhan fitur yang sudah ditentukan pada tahap *sprint planning*. Di dalam repository tersebut terdapat 263 *test case* yang difokuskan pada alur pengantaran pesanan, termasuk modul-modul yang berkaitan dengan halaman pesanan di aplikasi internal. Setiap skenario disusun berdasarkan *business requirement* dan *storylist* yang menjadi acuan pengembangan, sehingga cakupan pengujiannya tetap sesuai dengan fungsionalitas yang harus dipenuhi. Gambar 3.20 menunjukkan salah satu modul yang telah disiapkan untuk kebutuhan *test run*, yang menggambarkan bagaimana keseluruhan skenario pengujian terdokumentasi secara sistematis dan siap dijalankan,



Gambar 3.21 *Test Run Project Tracking Order Live Delivery* SAPA

Gambar 3.21 menampilkan contoh bagian kerja yang dikerjakan pada *project* ini, yaitu proses *test run* untuk modul *login page* yang kini memerlukan status aktif GPS perangkat. Pada tahap ini, karyawan magang memastikan bahwa aplikasi internal mewajibkan GPS aktif sebelum pengguna dapat masuk ke aplikasi, sekaligus memverifikasi bahwa mekanisme tersebut mendukung kebutuhan *live tracking* yang akan berjalan setelah proses *login* berhasil. *Testrun* ini terdiri dari 268 test case, ditambah 5 *test case* tambahan yang muncul akibat regresi, seluruhnya dikelola melalui *platform* manajemen pengujian QASE sebagai *environment* pengujian. Pelaksanaan *testrun* dimulai pada 5 September dan berlangsung hingga 9 Oktober, dengan total 23 hari kerja efektif. Tanggal 9 Oktober juga menjadi tanggal tercapainya *QA pass* untuk *project* ini.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

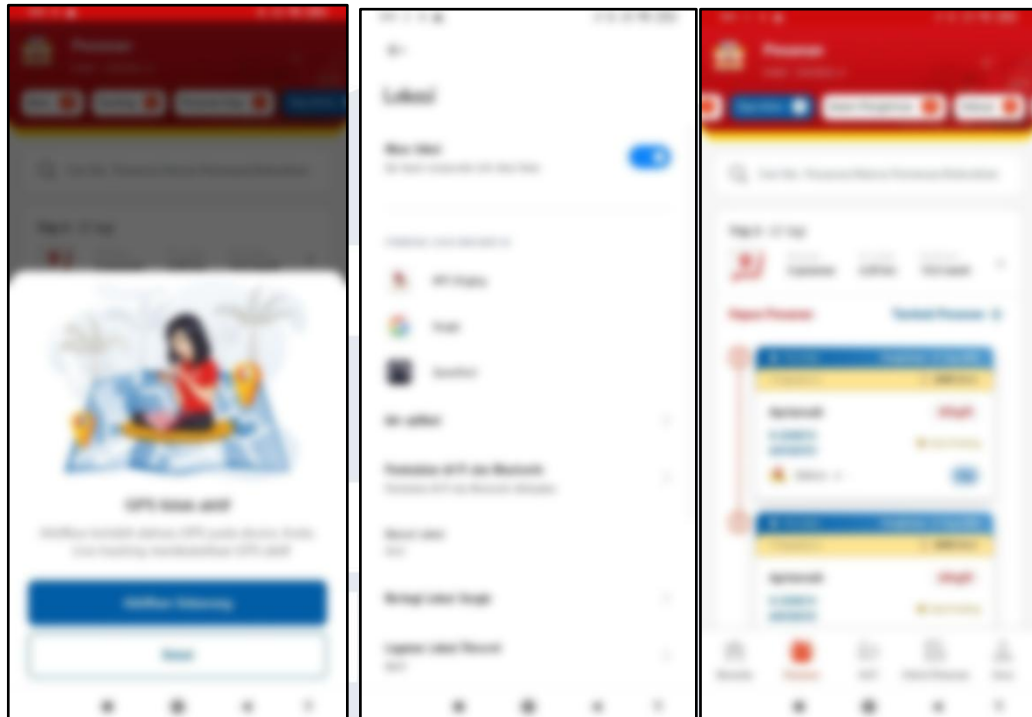


Gambar 3.22 Testing Modul Login, Permission Required GPS, dan Trip Aktif

Gambar 3.22 menunjukkan hasil pengujian pada modul *login* ketika pengguna mencoba masuk ke aplikasi internal dalam kondisi GPS tidak aktif. Pada skenario ini, sistem harus selalu menampilkan permintaan *permission* agar GPS diaktifkan, karena layanan lokasi menjadi komponen penting untuk mendukung fitur *live tracking*. Modul *login* juga diuji untuk memastikan bahwa apabila terdapat sesi aktif di perangkat lain, sistem akan menampilkan *bottomsheet* konfirmasi sebelum melanjutkan proses *login* dan mengakhiri sesi sebelumnya.

Pengujian juga mencakup kondisi ketika kru SAPA masih memiliki *trip* aktif dalam proses pengantaran. Pada situasi tersebut, informasi mengenai *trip* aktif akan dimunculkan pada *homepage* sebagai pengingat setelah kru berhasil *login*. Hal ini menandakan bahwa pesanan tersebut harus diselesaikan terlebih dahulu setelah kru berhasil masuk ke aplikasi. Dengan begitu, sistem tetap menjaga konsistensi proses operasional dan mencegah terjadinya pengabaian terhadap pesanan yang sedang berjalan.

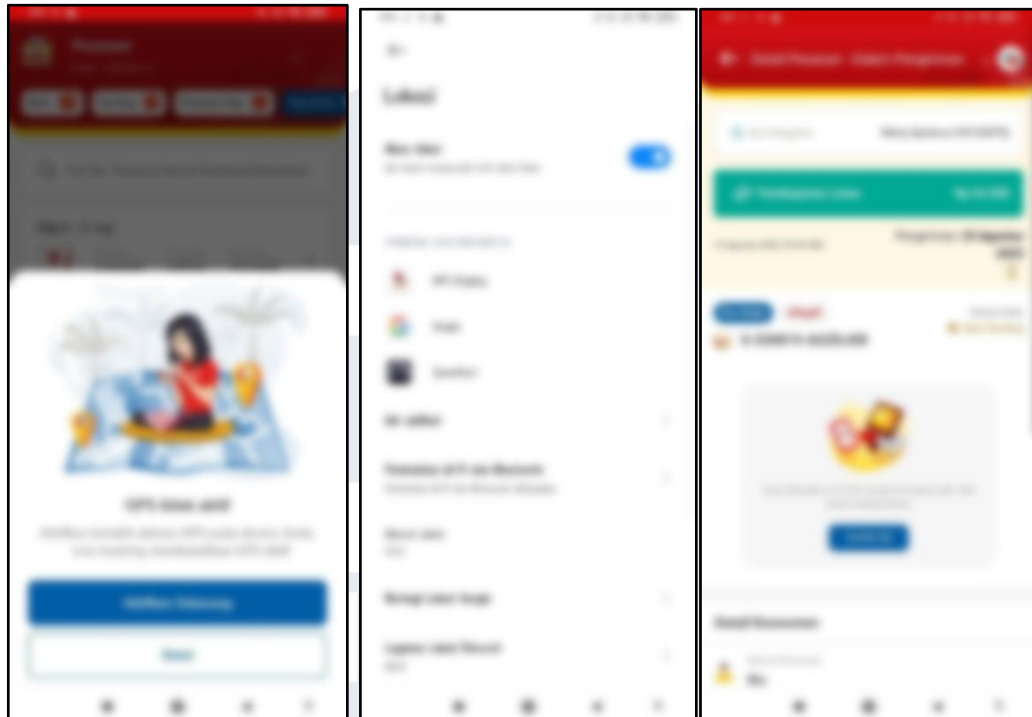




Gambar 3.23 *Testing Modul Pesanan Ketika GPS Off dan Trip Aktif*

Gambar 3.23 menampilkan hasil pengujian terkait mekanisme pengecekan GPS pada halaman pesanan. Berdasarkan *business requirement*, validasi GPS tidak hanya dilakukan pada tahap *login*, tetapi juga kembali diterapkan pada *tab* “Siap Kirim” dan “Dalam Pengiriman”. Kedua *tab* ini berisikan daftar pesanan yang akan atau sedang diproses oleh kru SAPA, sehingga sistem harus memastikan layanan lokasi aktif sebelum pengguna dapat melanjutkan aktivitas operasional di dalamnya. Pengujian ini dilakukan untuk memastikan bahwa fitur *live tracking* tetap berjalan sebagaimana mestinya, karena pelacakan posisi hanya dapat berfungsi apabila GPS dalam kondisi aktif.

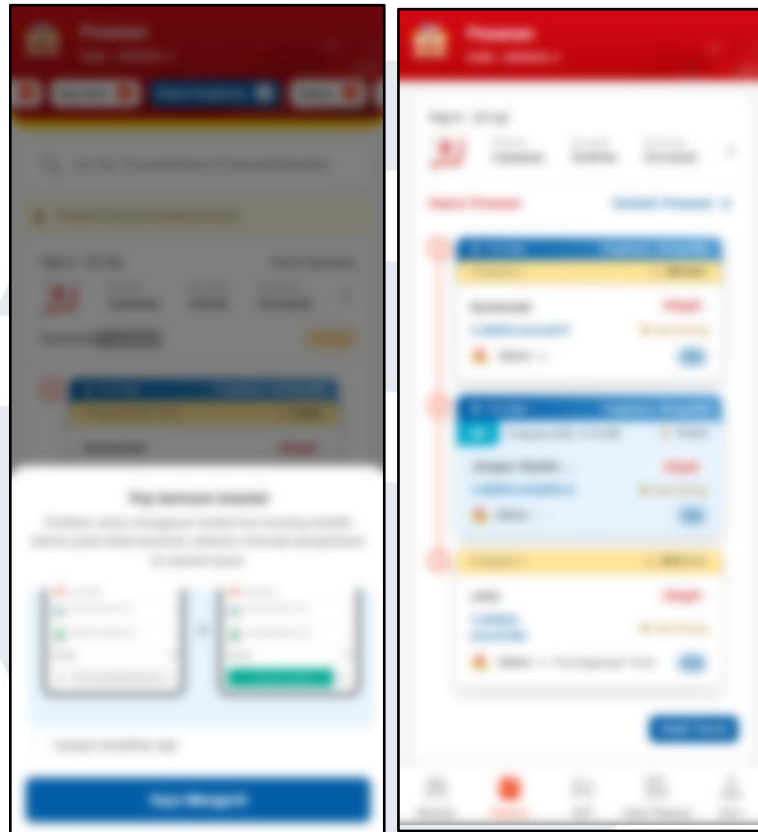
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.24 *Testing* Modul Halaman Detail Pesanan Ketika GPS *Off*

Halaman detail pesanan yang diakses pada *tab* “Siap Kirim” maupun “Dalam Pengiriman” juga menerima validasi GPS seperti ditunjukkan pada Gambar 3.24. Prosesnya kurang lebih sama dengan *permission access* GPS sebelumnya, di mana pengguna harus mengaktifkan GPS terlebih dahulu sebelum dapat membuka halaman detail pesanan. Penerapan validasi ini berfungsi sebagai mekanisme pengendalian akses, karena pada halaman ini terdapat proses pengambilan trip baru, sehingga sistem memastikan bahwa lokasi pengguna telah sesuai sebelum melanjutkan ke tahapan berikutnya.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.25 Testing Mekanisme Baru Pengambilan *Trip* Pesanan

Pada *project tracking order*, ditambahkan sebuah mekanisme baru dalam proses pengambilan *trip* pesanan, sebagaimana ditunjukkan pada Gambar 3.25. Pengambilan *trip* tidak lagi dilakukan melalui satu kali klik tombol, melainkan menggunakan metode *slide confirmation* yang menyerupai mekanisme verifikasi interaktif, sehingga meminimalkan potensi kesalahan *input* yang tidak disengaja. Ketika proses pengambilan *trip* berhasil dilakukan, sistem akan secara otomatis memulai *counter* waktu pengiriman serta memperbarui jumlah pesanan yang sedang dalam status pengantaran, terutama jika kru membawa lebih dari satu pesanan dalam satu waktu. Penambahan tahapan konfirmasi ini bertujuan untuk memastikan bahwa hanya pesanan yang benar-benar telah siap dikirim yang dapat diambil oleh kru, mengingat setiap *trip* yang telah dikonfirmasi akan langsung memicu pengiriman notifikasi kepada pelanggan dan tidak dapat dibatalkan maupun dialihkan ke kru lain.

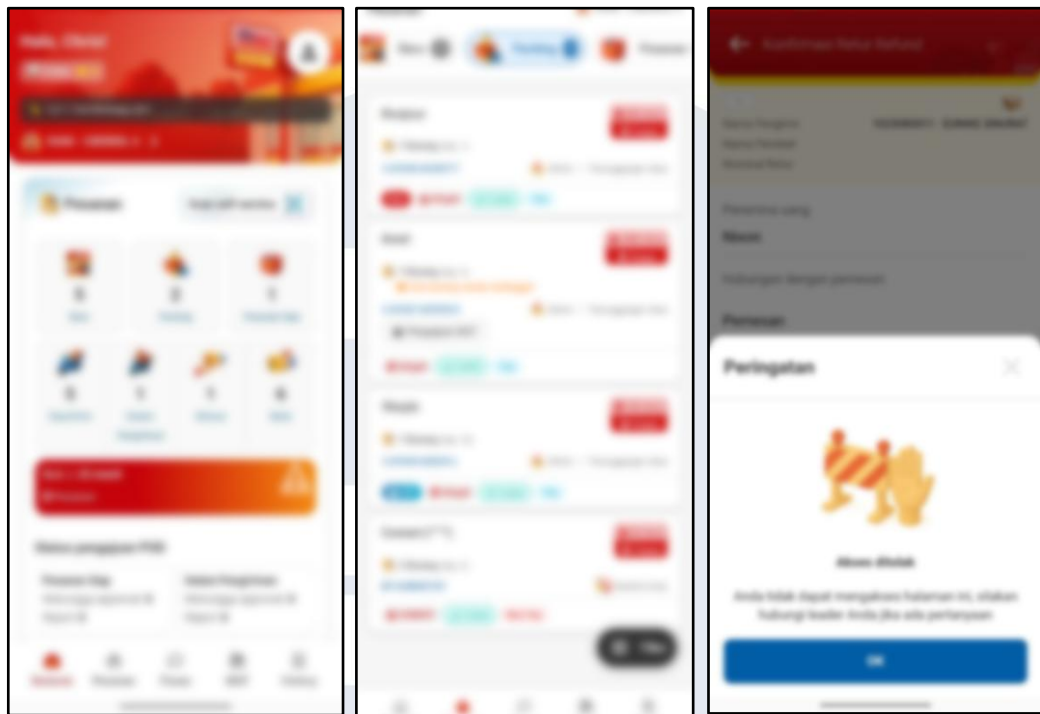
### 3.3.1.5 Project Revamp: Access Role and Functional

Pada minggu ketujuh masa magang, karyawan magang diberikan kesempatan untuk ikut menjadi bagian tim QA untuk *project revamp* aplikasi internal yang melibatkan penambahan fitur *edit* fungsi *role* pada keseluruhan *level user* dan perubahan tampilan yang cukup besar pada halaman *homepage*. Pada awalnya, aplikasi internal pengiriman memiliki 4 *level* utama;

- a. PIC (*Person in Charge*), yang mencakup *role* kru SAPA, kepala toko untuk menjalankan proses operasional toko
- b. AC (*Area Coordinator*), yang berfungsi untuk mengontrol beberapa toko dalam satu area tertentu serta berperan dalam proses approval atas pengajuan operasional toko.
- c. AM (*Area Manager*), yang mengelola banyak toko dalam satu area atau distrik untuk mencapai target melalui kinerja operasional yang baik. Dalam aplikasi internal, AM juga memiliki peran penting dalam proses *refund*.
- d. Super User, merupakan *level* tertinggi yang memiliki akses penuh terhadap seluruh fitur dan fungsi dalam sistem.

Pada *project* ini terdapat penyesuaian akses dalam masing-masing *role* yang dirancang agar pengaturan akses dapat lebih mudah dan juga setiap *role* dapat mengakses modul yang sebelumnya hanya tersedia untuk *role* lain, selama masih berada pada tingkat yang sama atau lebih rendah. Perubahan ini membuat pengaturan akses menjadi lebih fleksibel, tetapi tetap menjaga batasan sesuai struktur hierarki pengguna.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.26 Pembaruan Tampilan pada Aplikasi Internal

Selain dari fungsi akses yang diperbarui, tampilan aplikasi internal juga mengalami perubahan, terutama pada modul *homepage*, halaman pesanan, halaman MAT, aset gambar, dan juga beberapa *handling error* seperti pada Gambar 3.26. Setiap modul yang dapat diakses pada sebuah *role* diatur menggunakan *back office*. Jika terdapat sebuah *role* yang tidak memiliki akses pada modul tertentu maka aplikasi internal akan menampilkan *handling error* bahwa modul yang sedang diakses tidak dapat dibuka, sesuai pengaturan yang disesuaikan dan dapat diberikan jika terdapat keputusan tertentu atau mendesak. Meski demikian, terdapat beberapa *level user* yang hanya memiliki akses khusus pada modul tertentu dan tidak ada *role* lain yang dapat mengaksesnya walaupun sudah diatur menggunakan *back-office*, seperti *role* atasan atau level tertinggi.



Gambar 3.27 *Software Automation*

Pada *project* ini pula karyawan magang diberikan kesempatan untuk melakukan pengujian aplikasi internal sebagai *QA engineer*, yang melakukan *automation testing* menggunakan *software* Appium serta Android Studio sebagai *environment emulator* yang akan digunakan sebagai *media/interface* untuk melakukan pengujian secara otomatis. Gambar 3.27 memperlihatkan *software* yang digunakan dalam pelaksanaan *automation testing* ini. Rangkaian proses pengujian dimulai dari konfigurasi *testing environment*, termasuk penggunaan *emulator* Android sebagai perangkat virtual untuk menjalankan aplikasi melalui Android Studio. Selanjutnya, Appium, sebuah *framework automation* yang berbasis pada arsitektur Selenium, digunakan sebagai *tools* utama untuk menjalankan pengujian otomatis melalui proses identifikasi antarmuka pengguna (*UI element*) [19].

Aplikasi internal yang dirilis oleh *developer* kemudian dipasang dan dijalankan pada *emulator* agar dapat terhubung dan berinteraksi dengan Appium selama proses pengujian berlangsung. Sebelum pengujian dimulai, karyawan magang perlu memastikan bahwa koneksi perangkat telah terdeteksi melalui perintah pada *Command Prompt* (CMD). Nama perangkat yang teridentifikasi selanjutnya digunakan sebagai *parameter* dalam penulisan *script automation*. Setelah perangkat berhasil terkonfigurasi, tahap berikutnya adalah melakukan proses *scripting* awal melalui *automation code file*. Pada bagian ini, karyawan magang mengisi beberapa informasi penting, seperti nama perangkat (UDID), versi sistem operasi, lokasi file APK aplikasi internal, serta pengaturan *driver automation*, misalnya penggunaan UIAutomator. Penyesuaian konfigurasi



ini bertujuan agar *script* dapat dijalankan sesuai dengan *environment* yang telah ditentukan. Kegiatan ini juga memberikan pemahaman luas mengenai konsep bagaimana *automation framework* dikonfigurasi sekaligus cara mengintegrasikan berbagai komponen sistem agar proses pengujian dapat berlangsung secara otomatis. Apabila seluruh pengaturan telah dilakukan dengan benar, *emulator android* akan dapat dijalankan otomatis hingga menampilkan halaman awal, seperti *splash screen* dan *homepage*, layaknya aplikasi yang dibuka melalui *device* asli.

Dalam implementasinya, proses *automation testing* pada *project revamp* aplikasi internal menggunakan pendekatan *Behaviour Driven Development* (BDD), yaitu sebuah metode pengujian yang menggabungkan kebutuhan bisnis dengan aspek teknis ke dalam format skenario yang lebih mudah dipahami oleh berbagai pihak [20]. Pendekatan ini disusun melalui beberapa komponen utama, mulai dari penentuan fitur yang diuji (*feature*), perancangan skenario (*scenario*), penetapan kondisi awal (*given*), aksi yang dilakukan pengguna (*when*), hingga hasil yang diharapkan (*then*). Struktur BDD ini memberikan kemudahan tidak hanya bagi tim QA dalam memahami alur pengujian, tetapi juga memfasilitasi komunikasi dengan pihak non-teknis seperti tim bisnis dan *project manager*. Skenario yang dirancang pada tahap awal tersebut kemudian menjadi fondasi utama dalam *automation testing*, mulai dari implementasi hingga proses validasi. Semua tahap tersebut merupakan penting sehingga *test case* pada modul yang telah ditentukan dapat mulai dieksekusi dalam proses *test run*.

Selain pendekatan BDD, *automation testing* yang dilakukan juga dibangun dengan menggunakan bahasa pemrograman Java sebagai dasar utama dalam penulisan *script* pengujian. Java dipilih karena memiliki stabilitas yang baik, kompatibilitas luas dengan *framework* Appium, serta mendukung berbagai library yang memudahkan proses identifikasi elemen dan eksekusi test case secara otomatis. Penggunaan Java ini membantu

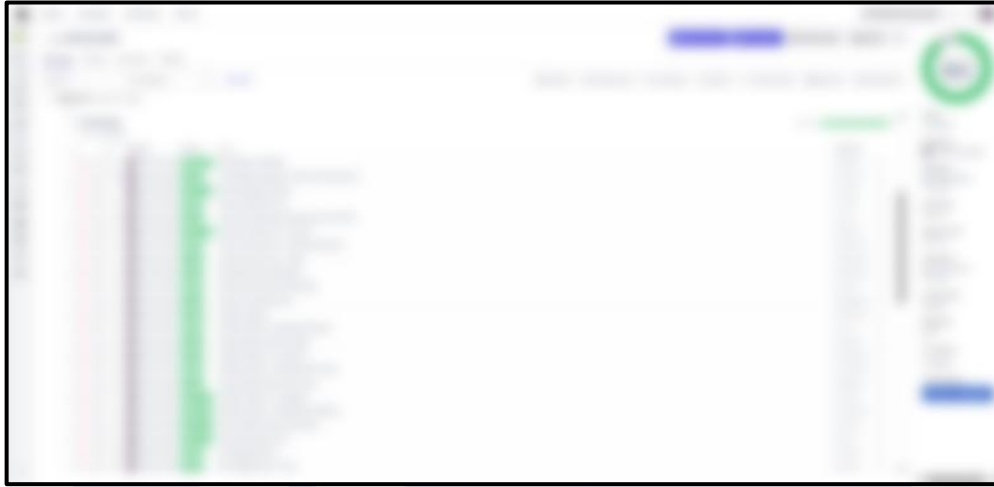
memastikan struktur *automation* lebih terorganisir, mudah di-  
*maintenanec*, dan dapat dikembangkan untuk kebutuhan pengujian lain.



Gambar 3.28 Test Scenario Halaman *Homepage* Setiap Role dan MAT

Gambar 3.28 merupakan *test scenario* dari modul uji untuk halaman *homepage* dari masing-masing *role*. Karyawan magang mendapatkan bagian kerja untuk menguji implementasi desain baru seperti pada halaman *homepage* dan MAT, serta fungsionalitasnya agar berfungsi dengan baik setelah dilakukan migrasi fitur. Terdapat total 1469 *test case* yang terunggah di dalam website manajemen pengujian, QASE. Sesuai diskusi pada tahap *sprint planning*, telah ditentukan sebuah story list yang memuat berbagai kebutuhan fitur dan fungsionalitas yang akan dikembangkan. Daftar inilah yang kemudian dijadikan dasar utama dalam merancang serta menyusun seluruh skenario pengujian di dalam repositori, sehingga pembuatan *test case* dapat menguji kelayakan aplikasi internal sesuai dengan kebutuhan bisnis yang telah ditetapkan.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.29 *Test Run Project Revamp: Access Role and Functional*

Gambar 3.29 menunjukkan modul repositori pengujian yang telah dipersiapkan untuk memasuki tahap *test run* pada proyek *revamp* aplikasi internal. Tahap ini dilakukan setelah seluruh *test scenario* berhasil disusun dan fitur yang dikembangkan telah siap untuk diuji, atau setidaknya telah memiliki fungsi dasar yang dapat divalidasi. Proses pengujian kembali dijalankan dengan mengacu pada kesesuaian antara *test scenario* yang telah ditentukan sebelumnya dan *system design*, guna memastikan bahwa implementasi fitur berjalan sebagaimana mestinya. Dalam proyek ini, terdapat tujuh modul utama yang menjadi fokus pengujian, meliputi halaman *homepage* beserta fitur-fiturnya, halaman pesanan, halaman MAT, halaman riwayat pesanan, hingga halaman akun. Dari keseluruhan modul tersebut, total *test case* yang tersedia mencapai 1.469 *cases*, di mana sebanyak 276 *test case* dijalankan sesuai modul yang diuji oleh karyawan magang. QA *pass* untuk *project* ini terjadi di tanggal 27 November 2025 dengan total waktu kerja efektif selama 43 hari.

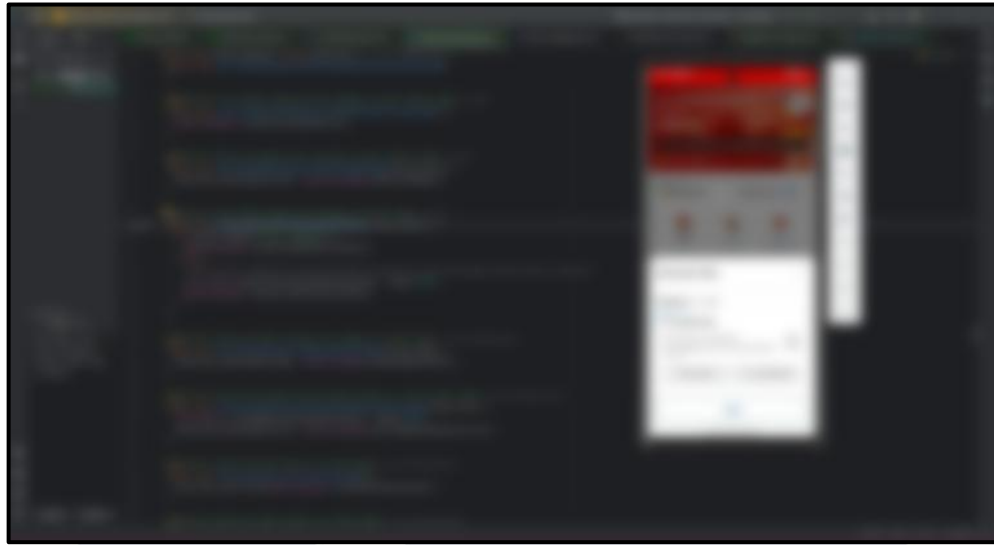
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.30 Pembuatan *Scenario Automation* Modul *Homepage*

Gambar 3.30 adalah proses pembuatan *scenario* menggunakan BDD untuk *automation testing* pada modul *homepage* yang sudah diperbarui dari segi desain dan fungsionalitasnya. Pengujian ini dilakukan secara regresif berdasarkan *role* yang ada. Pada dasarnya setiap *role* dapat melihat fitur yang ada pada *homepage*, namun hanya dibatasi aksesnya sesuai pengaturan masing-masing *role*. Proses ini dilakukan per sub-modul yang ada dengan membuat *scenario* jika terjadi interaksi pada fitur. Seperti contoh, *level user* PIC dapat mengakses halaman pesanan dan melakukan proses operasional toko, sedangkan *level user area manager* tidak dapat mengaksesnya dan aplikasi akan memunculkan *pop-up bottomsheets error* bahwa *role* tersebut tidak memiliki akses terhadap modul terkait. Jika contoh tersebut ingin dibuat dalam bentuk metodologi BDD, digunakanlah bahasa Gherkin untuk menciptakan skenario *Given-When-Then* sesuai contoh sebelumnya.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



Gambar 3.31 *Script Automation Modul Homepage Per Role*

Setelah proses pembuatan *scenario* telah dibuat, maka proses *script writing* untuk *automation testing* dapat dilakukan. Gambar 3.31 adalah proses *scripting* yang berbasis Java pada modul *homepage* untuk pengujian sub-modul informasi toko. Pada contoh tersebut, sub-modul informasi toko diuji dari interaksi *button* yang memunculkan *bottomsheet* detail informasi, hingga *clickable button* membuka aplikasi Google Maps untuk melihat letak geografis toko. Pada saat *script* dieksekusi, *emulator* akan menjalankan setiap *action* sesuai dengan *step* yang telah didefinisikan di dalam kode secara berurutan. Setelah seluruh proses pengujian selesai, sistem akan menampilkan hasil akhir berupa ringkasan performa pengujian (*output*), yang mencakup tingkat penyelesaian (*completion rate*), jumlah skenario yang berhasil dijalankan (*passed test case*), skenario yang gagal (*failed test case*), serta persentase keberhasilan keseluruhan (*success rate*).

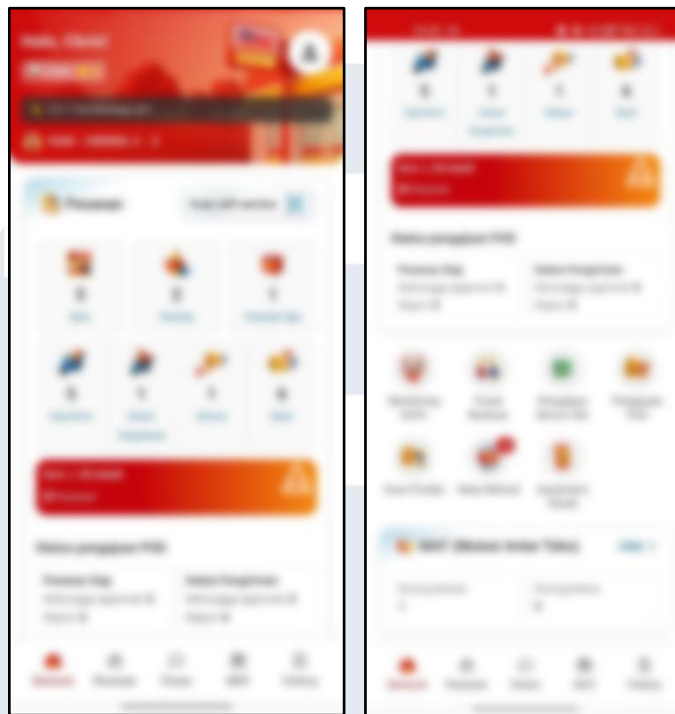
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

Tabel 3.2 *Output Automation Testing Project Revamp APO*

<i>Date (DD/MM)</i>	<i>Total Scenario.</i>	<i>Completion Rate</i>	<i>Passed Test Case</i>	<i>Failed Test Case</i>	<i>Success Rate</i>	<i>Issues</i>
15/10	159 <i>Scenario</i>	100%	139 cases	20 cases	87,19%	8 bugs
21/10	159 <i>Scenario</i>	100%	146 cases	13 cases	92,04%	5 bugs
24/10	159 <i>Scenario</i>	100%	152 cases	7 cases	95,62%	4 bugs
28/10	159	100%	159 cases	0 case	100%	-

Pada tanggal 15 Oktober, seluruh 159 *test scenario* untuk *automation* berhasil dijalankan namun masih ditemukan 20 *failed test case* sehingga *success rate* berada di angka 87,19%. Perbaikan kemudian dilakukan pada siklus berikutnya, yang dilakukan pada pengujian tanggal 21 Oktober setelah *bug* dilakukan *fix*, dengan penurunan jumlah *failed case* menjadi 13 dan peningkatan *success rate* hingga 92,04%. *Progress* yang lebih stabil terlihat pada tanggal 24 Oktober, di mana hanya tersisa 7 *failed test case* dan *success rate* meningkat menjadi 95,62%. Fase tercapainya *automation pass* pada tanggal 28 Oktober, seluruh 159 *test scenario* dapat dieksekusi dengan hasil *passed* tanpa adanya kegagalan, sehingga *success rate* mencapai 100%.

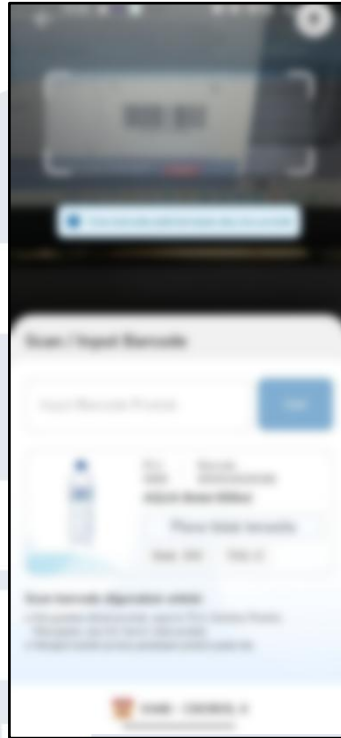




Gambar 3.32 Testing Revamp Modul Homepage Per Role

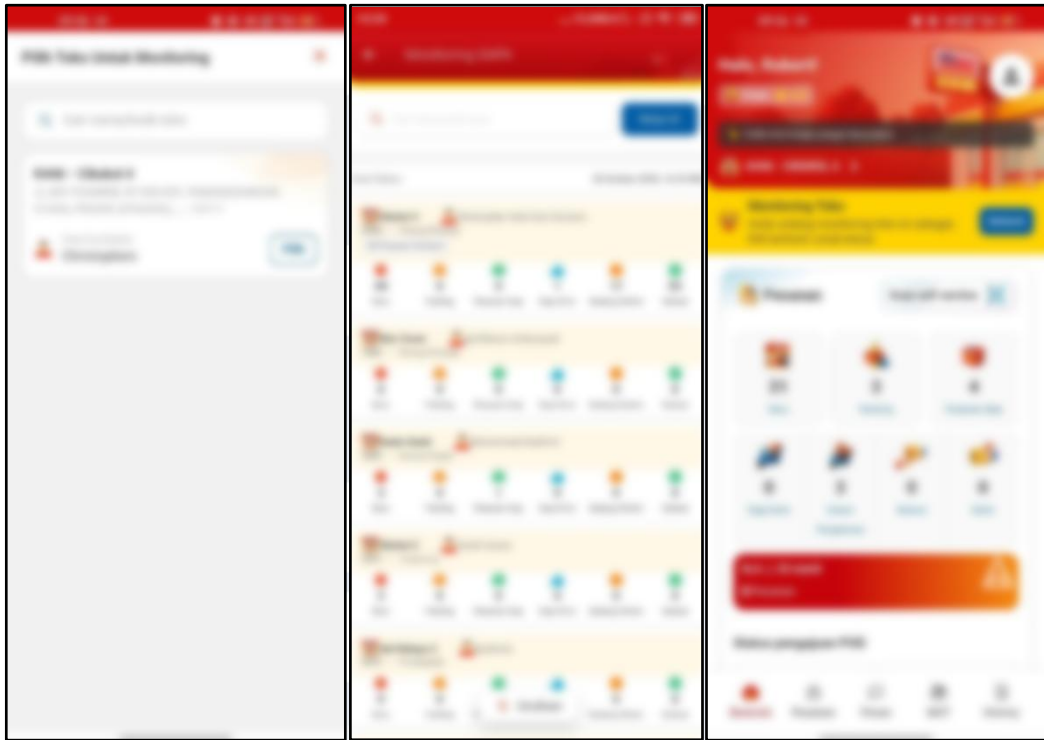
Setelah *script automation* selesai disusun, Gambar 3.32 menampilkan salah satu objek pengujian, yaitu halaman *homepage*. Pada halaman ini terdapat beberapa submodul utama, seperti *section* pesanan yang dilengkapi dengan *icon* berdasarkan status pesanan, *section* menu, serta MAT. *Automation testing* dilakukan secara menyeluruh untuk memastikan setiap fitur dan *button* dapat berfungsi dengan semestinya, baik dalam menampilkan informasi atau peringatan jika tidak memiliki akses melalui *bottomsheet* maupun mengarahkan pengguna ke tampilan halaman terkait. Selain itu, pengujian juga mencakup fungsi *navigation bar*, untuk memastikan bahwa setiap menu seperti *order*, *chat*, MAT, dan *history* pesanan dapat mengarahkan pengguna ke halaman yang sesuai.

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



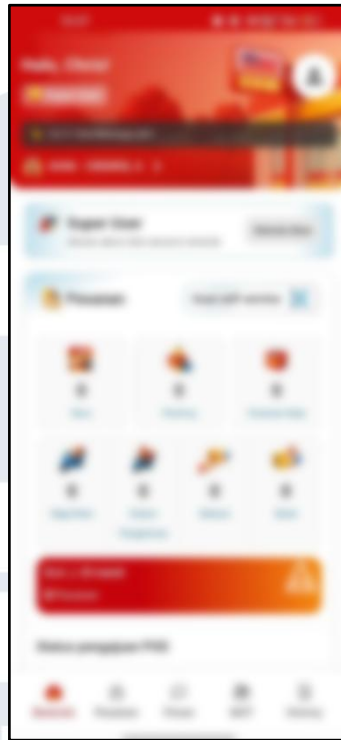
Gambar 3.33 *Testing Revamp* Fitur Scan Produk Packing

Pada tahap berikutnya pada modul yang sama, dilakukan juga pengujian terhadap fitur yang mengalami pembaruan, salah satunya adalah fitur *scan produk packing* seperti yang ditunjukkan pada Gambar 3.33. Fitur ini dirancang untuk memudahkan kru SAPA dalam mengidentifikasi SKU dari produk yang akan dikemas, dengan menampilkan informasi pendukung sehingga proses *packing* dapat berjalan lebih lancar dan pembaruan stok dapat dilakukan secara lebih *seamless*. Pengujian pada fitur ini dilakukan secara manual dengan memerlukan pemberian izin aplikasi untuk mengakses kamera pada perangkat guna melakukan pemindaian barcode produk. Melalui proses tersebut, aplikasi akan menampilkan informasi terkait produk yang dipindai, seperti nama produk, PLU, serta sisa stok yang tersedia di toko tersebut.



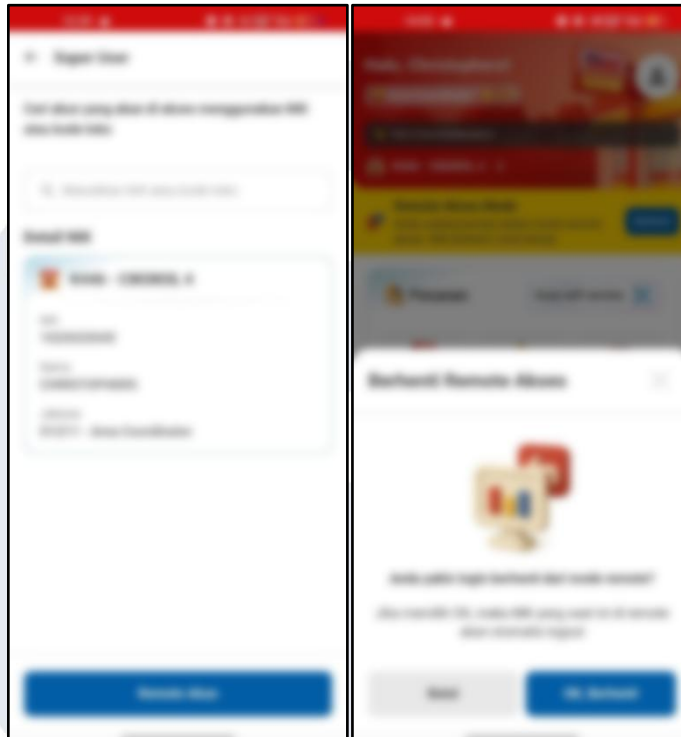
Gambar 3.34 *Testing Revamp Modul Tampilan Homepage Monitoring Toko*

Pada modul *homepage* juga terdapat fitur monitoring toko yang hanya dapat diakses oleh *user* dengan *level* Area Coordinator ke atas. Seperti ditunjukkan pada Gambar 3.34, pengujian dilakukan untuk memastikan proses pengambilan data toko yang akan dimonitor sudah sesuai dengan daftar kepala toko pada wilayah terkait. Pengujian ini turut mempertimbangkan perubahan data yang perlu divalidasi langsung melalui *service* menggunakan Postman dan Redis, sehingga alur pengecekan difokuskan pada kesesuaian data yang ditarik oleh sistem. Setelah toko dipilih, sistem diuji kembali dalam menampilkan data pesanan yang sedang berjalan di toko tersebut, termasuk informasi ringkasan yang muncul sebelum proses monitoring diaktifkan. Ketika sesi monitoring berlangsung, tampilan *homepage* akan menampilkan label berwarna kuning pada bagian *header* sebagai penanda bahwa akun sedang berada dalam mode monitoring menggunakan informasi *user* pada *level* PIC yang dipilih.



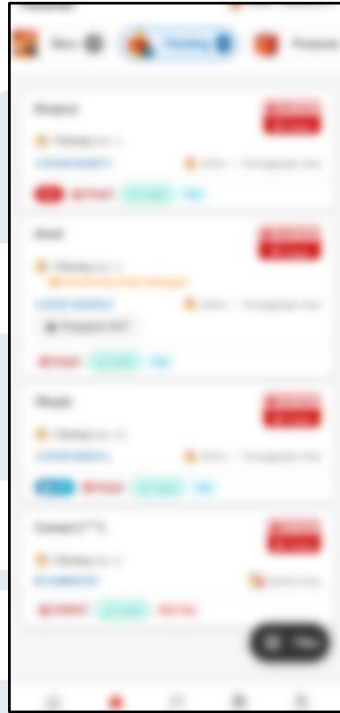
Gambar 3.35 *Testing Revamp Modul Homepage Untuk Role Super User*

Selain fitur monitoring toko, pengguna dengan *level* Super User memiliki akses yang lebih luas melalui fungsi *remote account*. Berbeda dengan mode monitoring yang hanya bersifat observasional, fitur ini memberikan akses bagi *user* untuk melakukan tindakan operasional seperti memproses pesanan yang sedang berjalan. Gambar 3.35 menunjukkan tampilan *homepage* ketika sistem diakses menggunakan akun Super User. Salah satu ciri utamanya ditandai dengan adanya sebuah *section* khusus untuk melakukan *remote account* yang terletak di bagian bawah, tepat setelah *header* pada *homepage*. Keberadaan *section* ini menjadi indikator bahwa akun memiliki hak akses lanjutan untuk melakukan intervensi langsung pada aktivitas operasional di dalam sistem.



Gambar 3.36 Testing Revamp Fitur Remote Account Melalui Super User

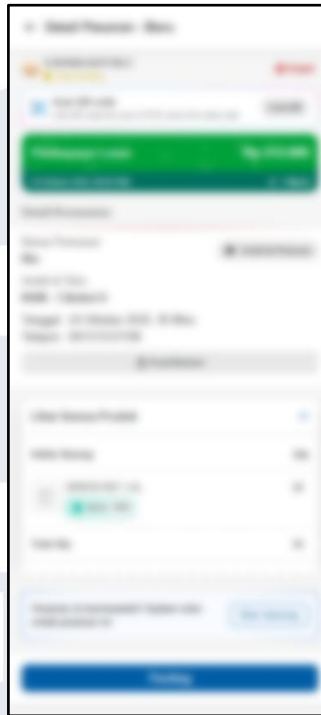
Gambar 3.36 menampilkan alur proses ketika fitur *remote account* dijalankan. Proses ini dimulai dengan memilih akun yang akan di-remote melalui pencarian berdasarkan NIK atau kode toko. Setelah data dimasukkan, sistem akan menampilkan daftar nama kru yang terdaftar pada toko tersebut atau hasil yang sesuai dengan NIK yang diinput. Ketika tombol *remote account* diklik, pengguna akan langsung diarahkan kembali ke halaman *homepage* dengan *session* yang telah berpindah ke akun tujuan. Sama seperti mode *monitoring*, pada tampilan *homepage* akan muncul label berwarna kuning sebagai penanda bahwa *session* sedang berada dalam kondisi *remote account*. Proses ini dapat dihentikan sewaktu-waktu dengan menekan tombol “Berhenti” yang tersedia pada label tersebut.



Gambar 3.37 *Testing Revamp* Modul Halaman Pesanan

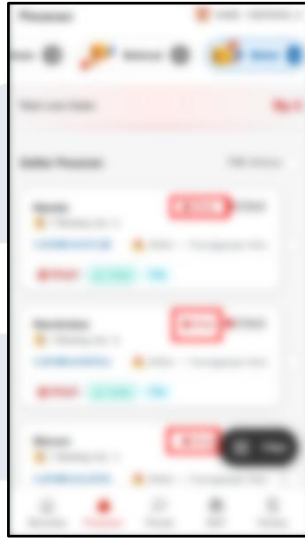
Gambar 3.37 adalah tampilan *revamp* pada modul halaman pesanan, yang menjadi inti operasional pada aplikasi internal. Pada tahap ini, karyawan magang memastikan bahwa desain UI yang ditampilkan telah sesuai dengan *system design* yang ditetapkan, serta alur kerja dalam memproses pesanan tetap berjalan *as-is* seperti pada sistem sebelumnya. Beberapa pembaruan yang diuji meliputi penambahan *icon* pada setiap status pesanan di bagian tab atas, penambahan informasi mengenai toko yang sedang diakses, penampilan label jenis pesanan, hingga perubahan posisi informasi SLA yang dirancang ulang untuk meningkatkan *readability* dan memperkuat *visual hierarchy* pada tampilan. Selain itu, dilakukan pula pengujian terhadap fungsi *filter* pesanan untuk memastikan proses penyaringan data berjalan dengan akurat sesuai kriteria yang dipilih.





Gambar 3.38 *Testing Revamp* Modul Halaman Detail Pesanan

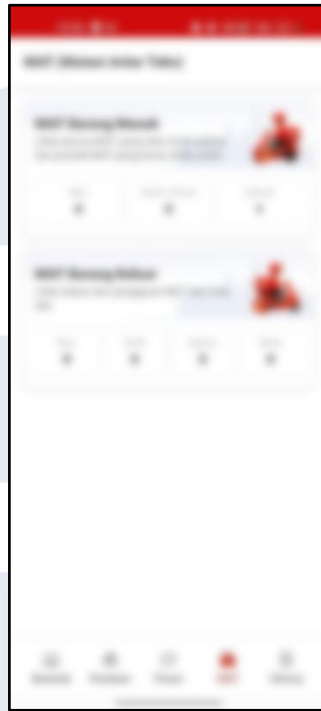
List pesanan yang ditampilkan pada halaman sebelumnya dapat diklik dan akan *direct* aplikasi secara langsung ke halaman detail pesanan yang sesuai, seperti ditunjukkan pada Gambar 3.38. Pada tampilan ini, terdapat perubahan visual yang cukup signifikan, terutama pada bagian *scan* QR pesanan yang dirancang lebih ringkas dan efisien sehingga *layout composition* antarelemen menjadi lebih terstruktur dan optimal dalam pemanfaatan ruang. Selain itu, informasi detail konsumen serta ringkasan pembayaran ditampilkan dalam format yang lebih sederhana dan mudah dipahami. Pembaruan lainnya terlihat pada desain *container* daftar produk yang dipesan, yang kini dibuat lebih tersegmentasi dalam bentuk *card-based layout*, sehingga jumlah serta perbedaan jenis produk dalam satu pesanan dapat terlihat dengan lebih jelas ketika item yang ditampilkan cukup banyak.



Gambar 3.39 *Testing* Modul Halaman Pesanan Status Batal Kasus *Refund*

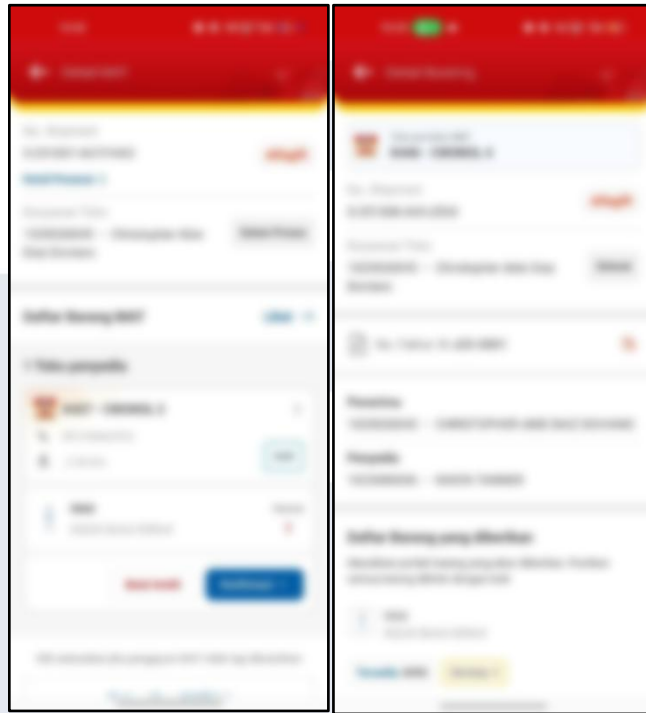
Gambar 3.39 menampilkan beberapa pembaruan pada tampilan *list* pesanan ketika suatu pesanan masuk ke dalam proses retur. Pesanan yang diretur secara otomatis akan ditandai dengan label baru “Retur” dan dipindahkan ke dalam tab status “Batal”. Perubahan ini memberikan kejelasan visual bagi pengguna dalam membedakan pesanan yang masih aktif dengan pesanan yang sudah tidak dapat diproses lebih lanjut. Setiap pesanan yang tercatat pada tab “Batal” kemudian dapat dikategorikan sebagai *loss sales*, sehingga mendukung proses pencatatan dan evaluasi performa finansial toko. Informasi tersebut membantu pihak terkait untuk mengetahui potensi kerugian yang timbul akibat pesanan yang tidak berhasil diselesaikan. Setelah ajukan retur disetujui oleh Area Manager, baik karena kendala operasional maupun permintaan pelanggan, pesanan tersebut secara sistem tidak lagi dapat diproses kembali untuk tahap pengiriman.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



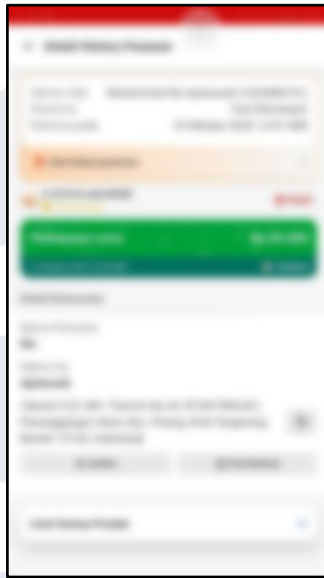
Gambar 3.40 *Testing Revamp Modul Landing Page MAT*

Gambar 3.40 menampilkan perubahan tampilan pada *landing page* MAT yang ikut mengalami penyesuaian dalam *project revamp* aplikasi internal. Dari sisi fungsi dan alur penggunaan, halaman ini masih berjalan dengan mekanisme yang sama seperti versi sebelumnya, namun kini dirancang dengan desain yang lebih simpel, ringan, dan menyatu dengan tema visual utama aplikasi internal. Perubahan yang cukup terlihat ada pada bagian *description* serta *icon* untuk MAT barang masuk dan MAT barang keluar yang didesain ulang agar terlihat lebih ringkas, konsisten, dan mudah dikenali. Penataan elemen juga dibuat lebih rapi, mulai dari peletakan *counter* MAT per status, ukuran *icon*, hingga pemilihan warna yang lebih *clean*, sehingga tampilan halaman terasa lebih nyaman untuk dilihat. Secara tidak langsung, perubahan ini membantu kru SAPA, dalam memahami fungsi halaman dengan lebih cepat tanpa harus membaca terlalu banyak informasi.



Gambar 3.41 *Testing* Perubahan Informasi dan Label Status Detail MAT

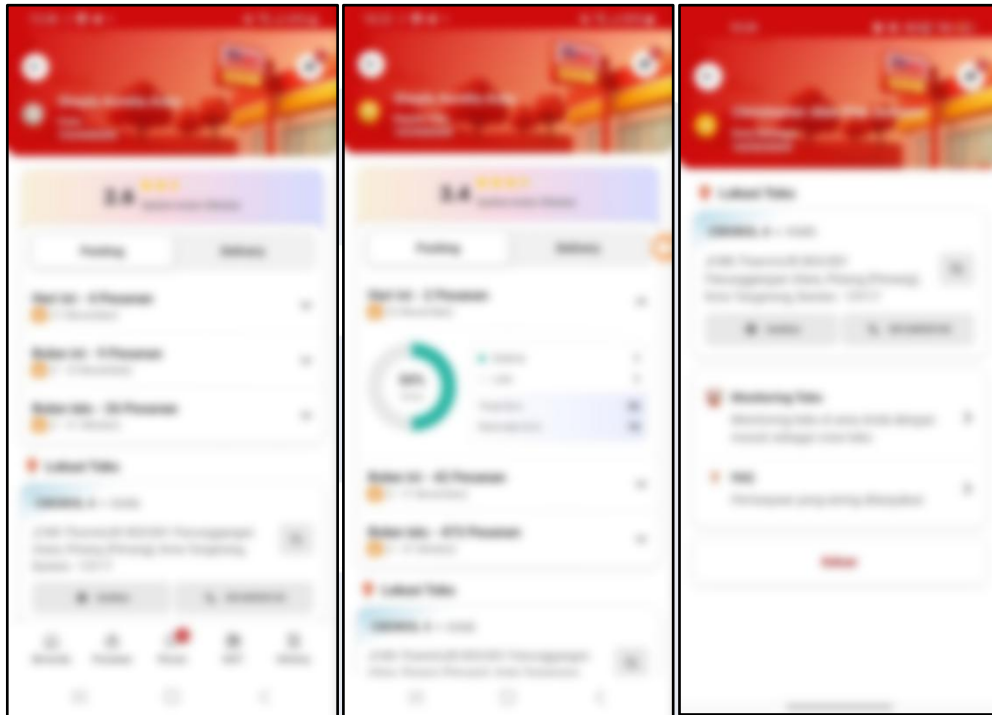
Perubahan yang cukup signifikan juga terjadi pada halaman detail MAT, khususnya pada bagian informasi MAT serta label status di setiap tahapannya. Pada Gambar 3.41 ditunjukkan tampilan ketika MAT berada pada status awal, di mana sistem kini menampilkan label berwarna biru yang menyesuaikan dengan kondisi MAT, seperti draft atau dalam pengiriman. Selain itu, informasi pihak penerima dan penyedia yang sebelumnya masih berupa placeholder kini dibuat lebih dinamis, karena akan langsung menampilkan nama kru yang bersangkutan apabila pesanan termasuk dalam kategori barang *dark store*. Pembaruan lain yang ikut diuji pada halaman ini adalah adanya fitur batal ambil untuk MAT barang masuk, yang masih dapat dilakukan selama status MAT barang keluar masih berada pada tahap *draft*. Penyesuaian tersebut memberikan fleksibilitas lebih dalam proses mutasi barang, sekaligus membantu mengurangi potensi kesalahan pengambilan sebelum proses benar-benar dilanjutkan ke tahap berikutnya.



Gambar 3.42 *Testing Revamp Modul Detail History Pesanan*

Gambar 3.42 menampilkan halaman *detail history* pesanan yang telah mengalami pembaruan tampilan. Pada bagian atas halaman terdapat penambahan *section* khusus untuk informasi penerimaan barang. *Section* ini memuat beberapa detail penting seperti nama kru yang menyerahkan pesanan, nama penerima, serta waktu penerimaan barang berlangsung. Selain itu, juga disediakan sebuah tombol untuk mengecek lokasi saat proses penerimaan barang dilakukan, sehingga informasi yang ditampilkan menjadi lebih transparan dan mudah diverifikasi. Sementara itu, untuk modul dan informasi lainnya, secara umum masih memiliki struktur yang serupa dengan halaman detail pesanan sebelumnya, sehingga pengguna tetap dapat memahami alur informasi tanpa harus beradaptasi terlalu jauh dengan tampilan baru.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A



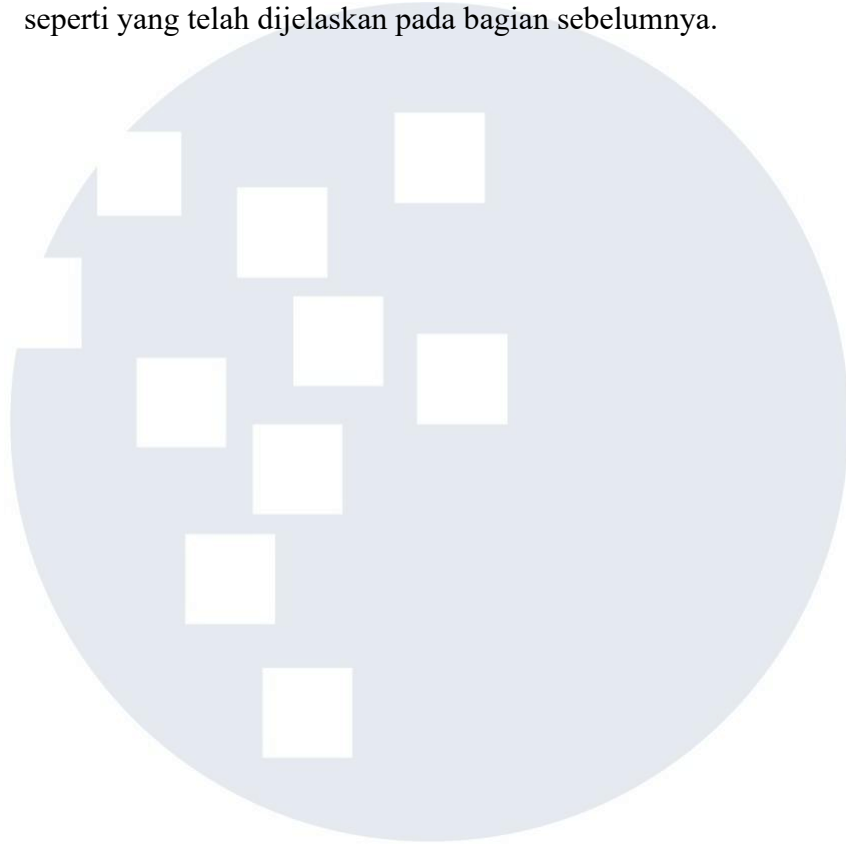
Gambar 3.43 *Testing Modul Halaman Akun Per Role*

Gambar 3.43 menampilkan halaman akun yang telah di-*revamp* pada masing-masing *role*. Perubahan yang diterapkan meliputi pembaruan *icon role* serta informasi terkait role, seperti nama dan jabatan, yang kini dibuat selaras dengan perubahan pada halaman *homepage*. Secara umum, halaman akun pada aplikasi internal menggunakan satu *template header* yang memuat informasi *role* sebelumnya, serta beberapa *section* utama, yaitu *section* lokasi toko, *section* FAQ, dan *button* logout. Pada *level role* tertentu, seperti PIC, ditambahkan fitur sistem *rating* yang digunakan untuk mengevaluasi aktivitas *packing* dan *delivery*. Di dalam *section rating* ini juga tersedia grafik *donut* yang menampilkan jumlah pesanan yang diproses tepat waktu (*ontime*) dan terlambat (*late*), dengan pembagian data berdasarkan periode bulanan.

Sementara itu, untuk *level role* Area Coordinator ke atas, *section rating* tidak ditampilkan. Sebagai penggantinya, *user* pada *level* ini mendapatkan fitur *monitoring* toko, yang sebelumnya juga tersedia pada



halaman *homepage*, dengan alur penggunaan (*flow*) yang tetap sama seperti yang telah dijelaskan pada bagian sebelumnya.



# UMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 3.3.2 Kendala yang Ditemukan

Berikut ini adalah beberapa kendala utama yang dihadapi karyawan magang selama melaksanakan magang di PT Global Loyalty Indonesia:

1. Sulit mengejar pemahaman terhadap alur sistem dan dokumentasi *project*: Sistem yang digunakan telah berjalan cukup lama dan kompleks, dan terkadang pada *project* baru yang diangkat terdapat keterkaitan dengan *project* sebelumnya. Hal ini membuat karyawan magang membutuhkan waktu lebih lama untuk memahami alur kerja dan relasi antar fitur dalam aplikasi.
2. Kesulitan menggunakan *tools* teknis seperti Solr dan Swagger: Di awal magang, penggunaan *tools* tersebut menjadi tantangan karena belum familiar dengan cara kerja yang kurang umum maupun pengujian dengan melakukan intervensi tertentu. Kendala ini terkadang sempat menghambat proses *testing* pada beberapa *project*.
3. Kendala istilah khusus dalam pembahasan *project*: Proses koordinasi yang mayoritas dilakukan secara *online* seringkali mengalami miskomunikasi bagi karyawan magang, terutama ketika membahas *bug* atau kendala teknis. Perbedaan istilah dan kurangnya visualisasi langsung membuat lemahnya pemahaman karyawan magang dan perlunya bertanya lebih lanjut.

### 3.3.3 Solusi atas Kendala yang Ditemukan

Untuk mengatasi kendala yang muncul selama masa magang, dilakukan beberapa langkah penyelesaian sebagai berikut:

1. Mempelajari dokumentasi internal dan melakukan *sharing knowledge* dengan mentor: Untuk memahami alur sistem yang belum terdokumentasi secara lengkap, karyawan magang aktif mencatat ulang alur berdasarkan hasil observasi dan materi dari sprint meeting. Selain itu, sesi diskusi dengan QA senior seperti mentor, serta developer dilakukan untuk memperjelas bagian-bagian sistem yang belum dipahami.
2. Belajar mandiri penggunaan *tools* melalui praktik langsung dan dokumentasi: Dalam menghadapi kesulitan menggunakan *tools* seperti

Solr dan Swagger, karyawan magang mengalokasikan waktu tambahan untuk membaca dokumentasi dan mencoba menggunakan *tools* tersebut dalam contoh kasus. Hal ini membantu dalam mempercepat proses adaptasi terhadap *tools* yang digunakan tim QA.

3. Mengambil inisiatif dalam komunikasi dan meminta klarifikasi langsung saat diperlukan: Ketika terjadi hambatan dalam komunikasi, karyawan magang berinisiatif menghubungi rekan kerja secara personal untuk meminta penjelasan tambahan. Selain itu, penggunaan *screenshot* ketika melakukan inisiatif diskusi terbukti efektif untuk memperjelas konteks permasalahan.

