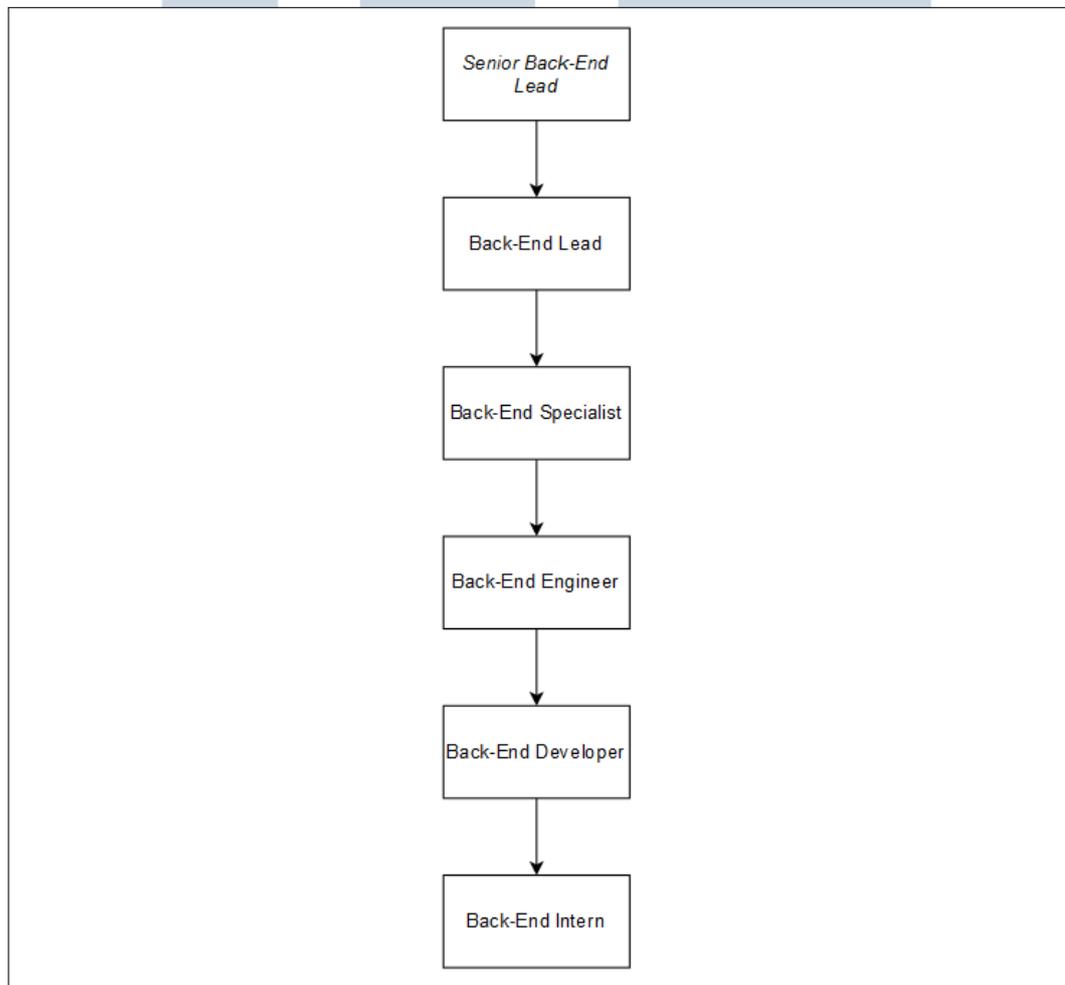


## BAB 3 PELAKSANAAN KERJA MAGANG

### 3.1 Kedudukan dan Koordinasi

Di PT Omni Digitama Internusa, Departemen *Back-End* terbagi menjadi beberapa tingkatan seperti yang tercantum pada Gambar 3.1, yang terdiri dari *Senior Back-End Lead*, *Back-End Lead*, *Back-End Specialist*, *Back-End Engineer*, *Back-End Developer*, dan *Back-End Intern*.



Gambar 3.1. Struktur Organisasi Departemen *Back-End* PT Omni Digitama Internusa

Departemen ini dipimpin oleh Bapak Rusmin Subiakto sebagai *Senior Back-End Lead*. Di bawah kepemimpinan Bapak Rusmin Subiakto, terdapat enam *Back-End Lead*, salah satunya adalah Bapak Christian William selaku *supervisor* selama

praktik pelaksanaan kerja magang. Bapak Christian William memimpin tiga orang *Back-End Engineer* dan satu *Back-End Intern*.

Selama praktik magang, penugasan diberikan dan dipandu oleh Bapak Christian William sebagai *supervisor* dan *Back-End Lead* dari tim. Daftar tugas atau *backlog* diatur dalam satu *platform* terdedikasi yaitu Jira, dan distribusi tanggung jawab diatur langsung oleh *lead* pada masing-masing tim. Penugasan selalu disertai dengan tautan tiket Jira bersamaan dengan sesi *onboarding* oleh Bapak Christian William terkait obyektif yang ingin dicapai pada penugasan tersebut. Pada beberapa kondisi, pelaksanaan *onboarding* diserahkan pada PIC (*Person In Charge*) terkait dengan *repository* atau *codebase* yang akan menjadi bagian penugasan, ataupun kepada *developer* yang pernah mengerjakan hal atau fitur serupa.

Dokumentasi terkait *codebase* ataupun *progress* penugasan dilakukan melalui platform *Notion* yang terdedikasi untuk masing-masing tim. Setiap harinya, Bapak Christian William selaku *lead* akan menyelenggarakan *daily scrum* bersama tim untuk mengumpulkan *progress* terkait hal yang dikerjakan pada hari tersebut untuk keperluan dokumentasi. Selain itu, setiap anggota juga wajib memperbaharui *progress* pekerjaan untuk masing-masing tiket pada satu *interface* yang terdedikasi untuk pemantauan *progress* tiket atau penugasan. Setiap penuntasan tiket atau penugasan, akan dilaksanakan *code review* oleh *lead* ataupun pihak terkait untuk membahas hasil pekerjaan, baik itu dari segi *improvement*, aspek *clean code*, ataupun kesalahan *logic* jika ada.

### 3.2 Tugas yang Dilakukan

Selama enam bulan praktik kerja magang, adapun tugas-tugas yang dilaksanakan sebagai *Back-End Intern* adalah sebagai berikut.

1. Mempelajari Alur Aplikasi.
2. Mempelajari Arsitektur Sistem *Back-End* dan *Design Pattern*.
3. Integrasi Vendor Pengiriman Shipper dengan Fitur *Reorder*.
4. Memperbaiki *Bug Voucher* pada Fitur *Reorder*.
5. Menulis Dokumentasi *Service* dan *Codebase*.
6. Modifikasi *Endpoint Invoice Reorder*.

7. Modifikasi *Endpoint Reorder Bidding* dan *Assignment* untuk *Reorder Per Business Unit*.
8. Memperbaiki *Bug* Penghilangan *Voucher Refund Hasil Pemesanan Ulang* pada Halaman *Checkout*.
9. Memperbaiki *Bug* Kesalahan Perhitungan Parameter Kalkulasi Kurir Shipper Termurah pada Pemesanan Ulang.
10. Menyesuaikan Perhitungan Stok *Reorder* dengan *Endpoint Baru*.
11. Menambahkan *ETA (Estimated Time Arrival)* pada Notifikasi *Reorder*.
12. Melakukan improvisasi pada mekanisme *logging* antar *service* untuk kasus *reorder*.
13. Improvisasi performa *endpoint reorder*.

### 3.3 Uraian Pelaksanaan Magang

Berikut adalah detail terkait kegiatan-kegiatan yang dilakukan selama praktik kerja magang.

Tabel 3.1. Uraian pelaksanaan praktik kerja magang

Minggu ke	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none"> <li>• <i>Onboarding techstack</i></li> <li>• Instalasi dan <i>setup development environment</i></li> <li>• Pembelajaran mandiri terhadap aplikasi dan <i>codebase</i></li> <li>• Aktivasi akses aset <i>development</i> milik perusahaan</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Uraian pelaksanaan praktik kerja magang (lanjutan)

Minggu ke	Pekerjaan yang dilakukan
2	<ul style="list-style-type: none"> <li>• <i>Onboarding</i> penugasan fitur <i>Shipper Phase 2</i> (integrasi kurir <i>shipper</i> termurah untuk <i>reorder bid</i> dan <i>reorder assignation</i>)</li> <li>• Pembelajaran mandiri terhadap flow <i>back-end</i> terkait penugasan</li> <li>• <i>Tracing &amp; debugging</i></li> <li>• <i>Development</i> atau pengembangan fitur <i>Shipper Phase 2</i></li> </ul>
3	<ul style="list-style-type: none"> <li>• <i>Development</i> atau pengembangan fitur <i>Shipper Phase 2</i></li> <li>• <i>API testing</i> dan <i>end-to-end testing</i> hasil <i>development</i> fitur <i>Shipper Phase 2</i></li> </ul>
4	<ul style="list-style-type: none"> <li>• <i>Self-testing</i> hasil <i>development</i> fitur <i>Shipper Phase 2</i></li> <li>• <i>Code review</i></li> <li>• <i>Refactoring</i> dan <i>bug fixing</i></li> </ul>
5	<ul style="list-style-type: none"> <li>• <i>Onboarding</i> penugasan kedua <i>Shipper Phase 2</i> (integrasi kurir <i>shipper</i> termurah untuk fitur <i>reorder manual</i>)</li> <li>• Pendekatan mandiri terkait penugasan kedua</li> </ul>
6	<ul style="list-style-type: none"> <li>• <i>Development</i> atau pengembangan penugasan kedua pada fitur <i>Shipper Phase 2</i></li> <li>• <i>API testing</i> dan <i>end-to-end testing</i> hasil pengembangan penugasan kedua fitur <i>Shipper Phase 2</i></li> <li>• <i>Bug fixing</i> dan <i>refactoring</i> hasil <i>development</i> penugasan kedua fitur <i>Shipper Phase 2</i></li> </ul>
Lanjut pada halaman berikutnya	

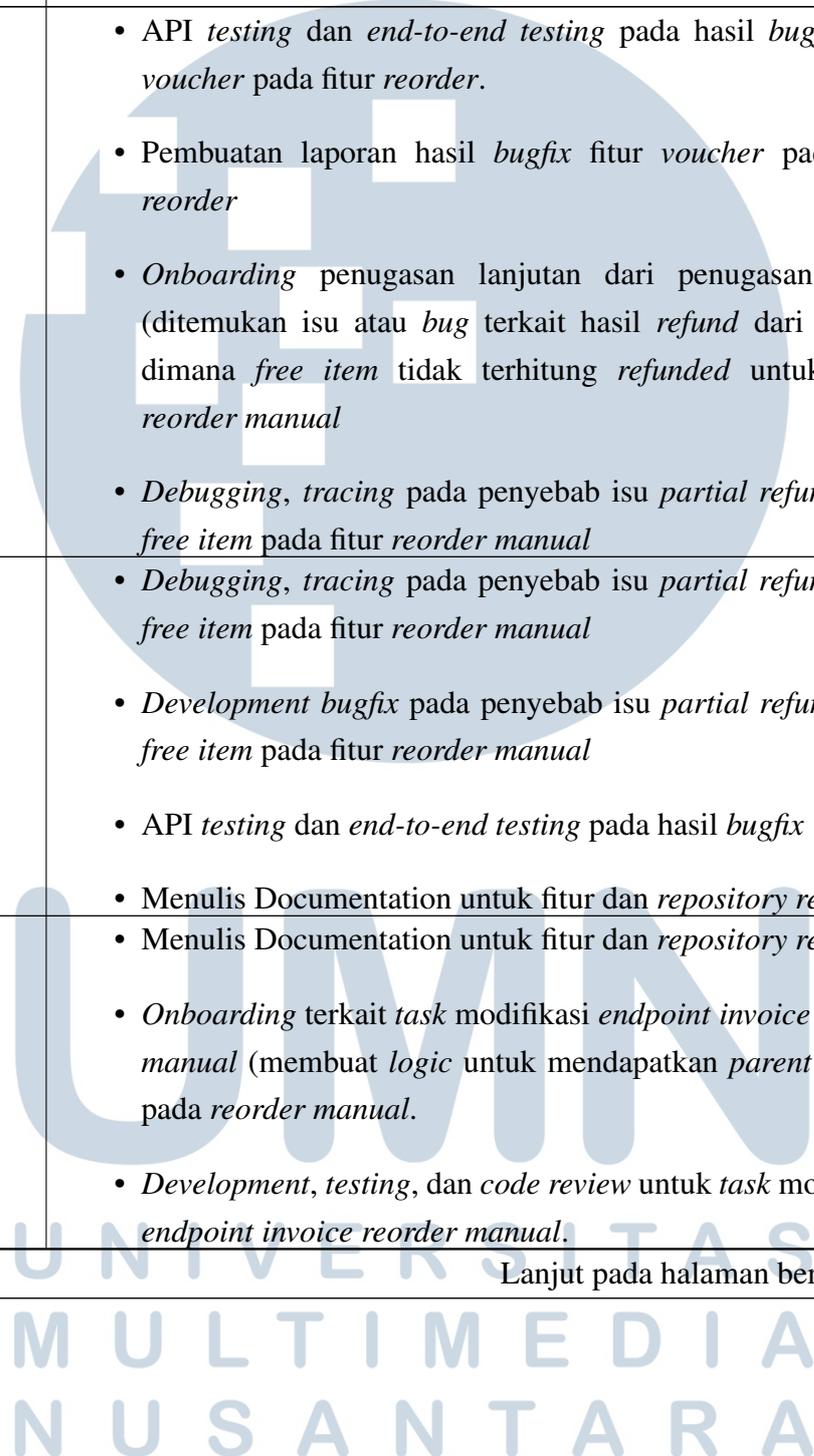
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

Tabel 3.1 Uraian pelaksanaan praktik kerja magang (lanjutan)

Minggu ke	Pekerjaan yang dilakukan
7	<ul style="list-style-type: none"> <li>• <i>Code review</i> penugasan <i>Shipper Phase 2</i></li> <li>• Revisi hasil <i>code review</i></li> <li>• <i>Onboarding</i> penugasan penambahan fitur untuk penugasan pertama dan kedua pada fitur <i>Shipper Phase 2</i> (integrasi untuk skenario <i>parent carrier</i> ruparupa)</li> <li>• <i>Development</i> atau pengembangan tugas tambahan <i>Shipper Phase 2</i></li> </ul>
8	<ul style="list-style-type: none"> <li>• <i>API testing</i> dan <i>end-to-end testing</i> hasil pengembangan penugasan tambahan <i>Shipper Phase 2</i></li> <li>• <i>Code review</i></li> <li>• <i>Bug fixing, refactoring, dan revisi</i> hasil pengembangan penugasan tambahan <i>Shipper Phase 2</i>.</li> <li>• <i>Onboarding</i> penugasan terkait <i>endpoint webhook</i> fitur <i>ownfleet</i></li> <li>• <i>Debugging, tracing, dan penyelesaian</i> penugasan <i>endpoint webhook</i> (tidak ada perubahan karena sudah sesuai ekspektasi)</li> <li>• <i>Onboarding</i> penugasan terkait <i>bug</i> pada fitur <i>voucher</i> pada <i>reorder bid, reorder assignation, dan reorder manual (free item</i> duplikat pada pembelian dua atau lebih barang dengan <i>free item</i> yang memiliki SKU (<i>Stock Keeping Unit</i>) yang sama).</li> <li>• <i>Development</i> terkait penugasan <i>bugfix</i> fitur <i>voucher</i> pada fitur <i>reorder</i>.</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Uraian pelaksanaan praktik kerja magang (lanjutan)

Minggu ke	Pekerjaan yang dilakukan
9	<ul style="list-style-type: none"> <li>• API <i>testing</i> dan <i>end-to-end testing</i> pada hasil <i>bugfix</i> fitur <i>voucher</i> pada fitur <i>reorder</i>.</li> <li>• Pembuatan laporan hasil <i>bugfix</i> fitur <i>voucher</i> pada fitur <i>reorder</i></li> <li>• <i>Onboarding</i> penugasan lanjutan dari penugasan <i>bugfix</i> (ditemukan isu atau <i>bug</i> terkait hasil <i>refund</i> dari <i>reorder</i> dimana <i>free item</i> tidak terhitung <i>refunded</i> untuk kasus <i>reorder manual</i></li> <li>• <i>Debugging, tracing</i> pada penyebab isu <i>partial refund</i> pada <i>free item</i> pada fitur <i>reorder manual</i></li> </ul>
10	<ul style="list-style-type: none"> <li>• <i>Debugging, tracing</i> pada penyebab isu <i>partial refund</i> pada <i>free item</i> pada fitur <i>reorder manual</i></li> <li>• <i>Development bugfix</i> pada penyebab isu <i>partial refund</i> pada <i>free item</i> pada fitur <i>reorder manual</i></li> <li>• API <i>testing</i> dan <i>end-to-end testing</i> pada hasil <i>bugfix</i></li> <li>• Menulis <i>Documentation</i> untuk fitur dan <i>repository reorder</i>.</li> </ul>
11	<ul style="list-style-type: none"> <li>• Menulis <i>Documentation</i> untuk fitur dan <i>repository reorder</i>.</li> <li>• <i>Onboarding</i> terkait <i>task</i> modifikasi <i>endpoint invoice reorder manual</i> (membuat <i>logic</i> untuk mendapatkan <i>parent invoice</i> pada <i>reorder manual</i>.</li> <li>• <i>Development, testing, dan code review</i> untuk <i>task</i> modifikasi <i>endpoint invoice reorder manual</i>.</li> </ul>
Lanjut pada halaman berikutnya	



Tabel 3.1 Uraian pelaksanaan praktik kerja magang (lanjutan)

Minggu ke	Pekerjaan yang dilakukan
12	<ul style="list-style-type: none"> <li>• Menulis Documentation untuk fitur dan <i>repository reorder</i>.</li> <li>• <i>Code review and refactoring</i> untuk <i>task</i> modifikasi <i>endpoint invoice reorder manual</i>.</li> <li>• <i>Onboarding</i> terkait <i>task Reorder</i> otomatis untuk per unit bisnis (membuat <i>logic</i> agar <i>Reorder</i> terjadi hanya untuk <i>store</i> dengan <i>brand</i> yang sama).</li> </ul>
13	<ul style="list-style-type: none"> <li>• <i>Development</i> dan <i>API testing</i> untuk <i>task Reorder</i> unit bisnis.</li> <li>• <i>Onboarding, development, dan testing</i> untuk penugasan <i>bugfix</i> terkait penghilangan <i>voucher refund</i> hasil pemesanan ulang pada halaman <i>checkout</i>.</li> <li>• <i>Onboarding</i> penugasan <i>bugfix</i> parameter item tidak berbayar pada perhitungan kurir Shipper termurah.</li> </ul>
14	<ul style="list-style-type: none"> <li>• <i>Development</i> dan <i>API testing</i> untuk penugasan <i>bugfix</i> parameter perhitungan kurir Shipper termurah.</li> <li>• Menulis Documentation untuk <i>endpoint flow reorder</i>.</li> <li>• <i>Onboarding, Development, and Testing</i> penugasan migrasi perhitungan stok <i>reorder</i>.</li> </ul>
15	<ul style="list-style-type: none"> <li>• <i>Code Review</i> dan revisi penugasan migrasi perhitungan stok <i>reorder</i>.</li> <li>• Mempelajari <i>unit test</i>.</li> <li>• <i>Onboarding</i> dan <i>development</i> penugasan notifikasi ETA (<i>Estimated Arrival Time</i>) pada fitur <i>reorder</i>.</li> </ul>
16	<ul style="list-style-type: none"> <li>• <i>Development, bug-fixing, dan self-testing</i> penugasan notifikasi ETA (<i>Estimated Arrival Time</i>) pada fitur <i>reorder</i>.</li> </ul>
17	<ul style="list-style-type: none"> <li>• <i>Development, bug-fixing, dan self-testing</i> penugasan notifikasi ETA (<i>Estimated Arrival Time</i>) pada fitur <i>reorder</i>.</li> <li>• <i>Onboarding</i> dan <i>development</i> penugasan <i>refactor response</i> pada repositori instalasi barang.</li> </ul>
Lanjut pada halaman berikutnya	

Tabel 3.1 Uraian pelaksanaan praktik kerja magang (lanjutan)

Minggu ke	Pekerjaan yang dilakukan
18	<ul style="list-style-type: none"> <li>• <i>Bug-fixing, tracing, dan self-testing</i> pada integrasi <i>server staging</i> penugasan notifikasi ETA (<i>Estimated Arrival Time</i>) pada fitur <i>reorder</i>.</li> </ul>
19	<ul style="list-style-type: none"> <li>• Revisi <i>improvement</i> perhitungan stok <i>reorder</i>.</li> <li>• Support QA terkait <i>reorder</i> pada proyek <i>revamp</i> tempel stok.</li> <li>• <i>Improvement</i> dan perbaikan <i>branching</i> pada penugasan notifikasi ETA <i>reorder</i>.</li> <li>• <i>Development and testing</i> improvisasi mekanisme <i>logging</i> antara <i>service</i> keranjang dengan <i>service</i> pengiriman.</li> </ul>

### 3.3.1 Mempelajari Alur Aplikasi

Pelaksanaan praktik kerja magang diawali dengan pendekatan dan pembelajaran terkait berbagai pengetahuan dan istilah umum terkait proses bisnis yang berjalan pada aplikasi Ruparupa. Proses pembelajaran didampingi oleh pembimbing lapangan dan alur yang dipelajari secara spesifik adalah terkait dengan sistem atau mekanisme pemesanan ulang atau *reorder*, dimulai dari sisi klien (*customer facing*) hingga sisi *admin* atau pengelola toko.

Fitur pemesanan ulang atau *reorder* merupakan fitur yang berfungsi sebagai mekanisme pengalihan pesanan yang diarahkan pada suatu toko kepada toko lain ketika toko orisinil penerima pesanan tidak dapat memenuhi pesanan produk yang diajukan oleh pelanggan. Lingkup atau *scope* pengalihan pesanan dilakukan pada tingkat produk, sehingga memungkinkan pengelola toko untuk menerima pesanan secara *partial*. Pengalihan pesanan dapat disebabkan oleh berbagai alasan, diantaranya sebagai berikut:

- Produk fisik rusak / cacat.
- Stok fisik pada toko tidak tersedia (tidak sesuai dengan data pada sistem).
- Produk kadaluarsa.
- Toko tidak memiliki jumlah produk yang mencukupi.
- Produk tidak lengkap (khusus untuk produk yang memerlukan instalasi).

- Produk yang dipilih merupakan barang *display* atau pajangan.
- Kesalahan pribadi pelanggan.
- Kesalahan atau alasan pribadi pihak toko.
- Pembatalan langsung oleh *admin internal* RUPARUPA.

Kelanjutan proses konfirmasi pemesanan ulang bergantung pada metode pengiriman dan tipe kurir yang dipilih pada *invoice* yang menjadi referensi atau sumber pemesanan ulang. Untuk metode pengiriman melalui pengambilan langsung oleh pelanggan di toko atau *pickup*, maka pelanggan akan diberikan notifikasi untuk memberikan respon terhadap pemesanan ulang, diantaranya sebagai berikut:

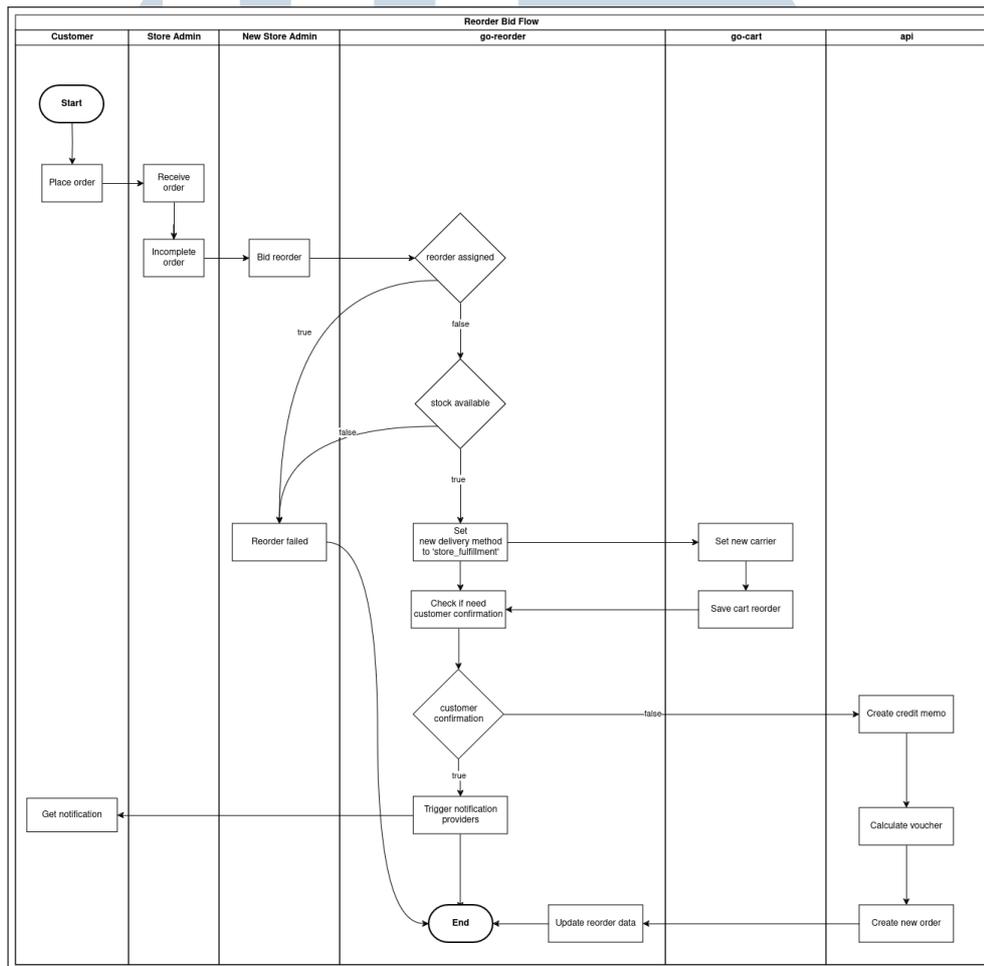
- Pemilihan metode pengiriman *pickup* yang berarti bahwa pelanggan setuju untuk tetap melakukan pengambilan unit di toko penyedia produk yang baru.
- Pemilihan metode pengiriman *delivery* oleh kurir dari toko penyedia produk yang baru.
- Pembatalan pesanan dengan *refund* atau pengembalian dana.

Apabila metode pengiriman awal adalah pengiriman kurir atau *delivery* dengan kurir *non-express (instant* atau *sameday delivery)*, maka pengiriman akan dilakukan dengan menggunakan kurir Shipper (<https://shipper.id>) termurah dengan mempertimbangkan berbagai faktor sebagai parameter perhitungan. Apabila pilihan kurir awal adalah kurir *express* dan jarak pengiriman baru kurang dari 25 KM, maka akan dilakukan perhitungan kurir *express* Shipper termurah. Jika jarak pengiriman melebihi 25 KM, maka akan digunakan kurir Shipper termurah dengan pemetaan tertentu sesuai dengan tipe kurir awal yang dipilih dan melalui notifikasi konfirmasi dari pihak pelanggan terlebih dahulu.

Terdapat tiga mekanisme pemesanan ulang yang diterapkan oleh aplikasi RUPARUPA, diantaranya sebagai berikut:

- *Bidding*, yaitu toko yang tidak dapat memenuhi permintaan pelanggan mempublikasikan detail pesanan dan akan diambil alih oleh toko tercepat. Lingkup atau *scope* untuk publikasi tawaran tidak langsung dibuka untuk seluruh toko yang terdaftar pada sistem, melainkan melalui mekanisme SLA (*Service Level Agreement*). Tawaran akan muncul di level kecamatan (sesuai

dengan kecamatan toko orisinal) selama 15 menit, lalu dibuka untuk level kota 30 menit setelah masa kadaluarsa level kecamatan, dan dibuka untuk level provinsi dua jam setelah masa kadaluarsa level kota. Berikut adalah alur dari *bidding reorder*.

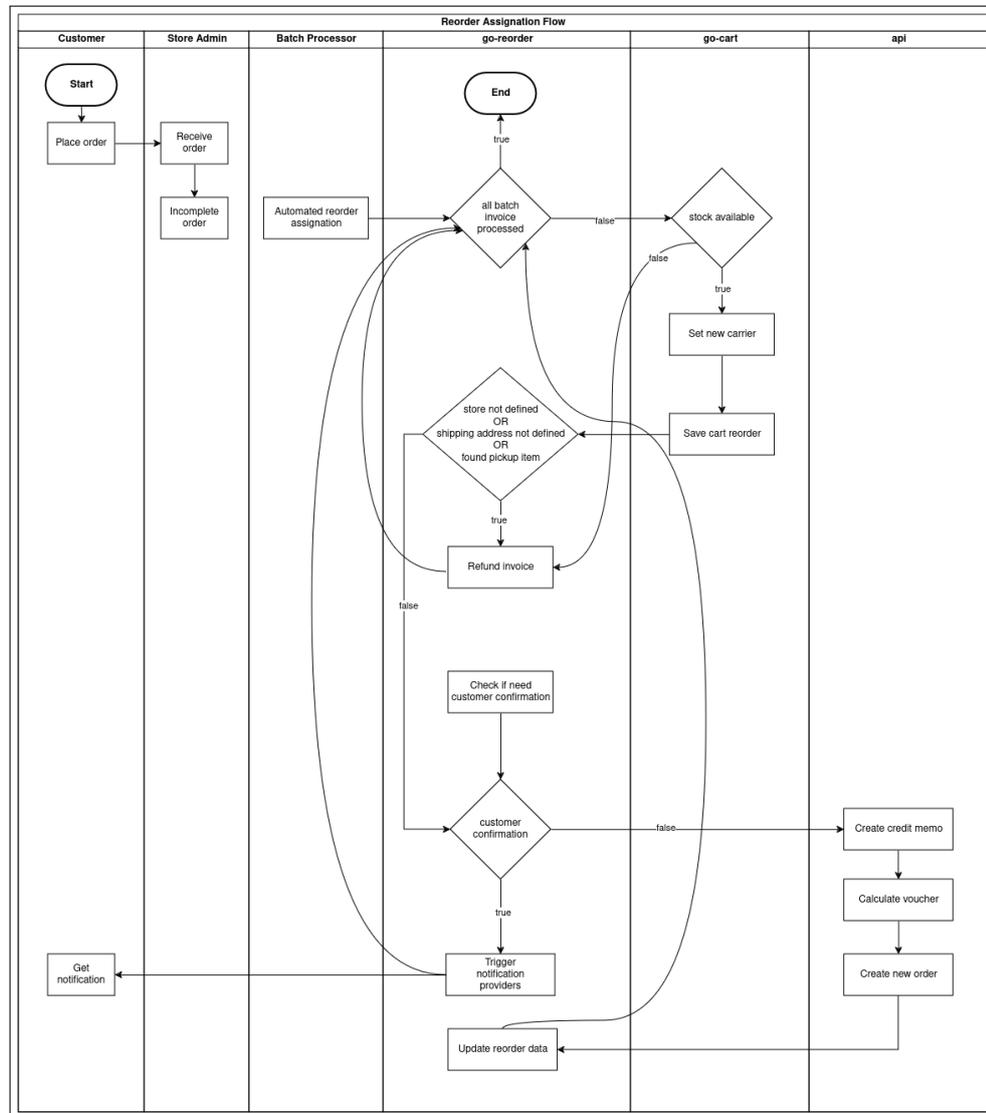


Gambar 3.2. Flow *Bidding Reorder*

Gambar 3.2 menunjukkan alur dari fitur pemesanan ulang dengan mekanisme *bidding*. Proses diawali dengan pengalihan pesanan oleh toko awal dengan membuka tawaran bagi toko lain untuk mengambil alih *invoice* atau barang yang tidak dapat dipenuhi oleh toko sebelumnya. Satu *invoice* tidak boleh di-*bid* oleh lebih dari satu toko, dan stok toko yang melakukan *bidding* juga harus memadai untuk produk tersebut. Metode pengiriman untuk mekanisme *reorder* diubah secara khusus menjadi "*store-fulfillment*", terkecuali untuk metode pengiriman "*pickup*". Setelah itu, logika bisnis akan dieksekusi

dengan melakukan perubahan pada data keranjang dan pemeriksaan terkait kebutuhan untuk meminta konfirmasi *pelanggan*. Pelanggan akan diminta untuk melakukan konfirmasi apabila terjadi salah satu skenario seperti yang tertulis di bawah ini.

- *Invoice* lama dan pengalihan pesanan memiliki metode pengiriman *pickup*.
- Hasil pengalihan pesanan memiliki metode pengiriman *store-fulfillment* dengan tipe kurir *non-express* dan *invoice* lama memiliki metode pesanan *delivery* dengan tipe kurir *express*. Skenario ini dapat terjadi karena terdapat kalkulasi untuk perbandingan pemilihan kurir diantara dengan vendor *Shipper* ataupun kurir 3PL (*Third-Party Logistics*) reguler.
- Hasil pengalihan pesanan memiliki metode pengiriman *store-fulfillment* dengan tipe kurir *express* namun kurir tidak tersedia. Kurir dinyatakan tersedia apabila jarak pengiriman baru tidak lebih dari 25 kilometer dan tidak terjadi *error* ketika melakukan perhitungan *shipping rate* dari *endpoint* milik kurir *express*.
- Data lokasi seperti *geolocation* milik pelanggan tidak ditemukan. Setelah melalui validasi-validasi tersebut, pesanan tersebut diproses menuju pembuatan *order* dan *invoice* baru dengan mengaplikasikan *promo reorder* untuk seluruh barang karena pelanggan telah melakukan pembayaran sebelumnya.
- *Assignment*, yaitu pelaksanaan pengalihan pemesanan secara otomatis dengan parameter-parameter tertentu yang dieksekusi oleh sistem (*batch*) apabila tidak ada respon dari toko manapun, yaitu setelah kadaluarsa SLA level provinsi. *Batch* akan dieksekusi melalui *cron job* setiap 30 menit. Berikut adalah alur dari *reorder assignment*.

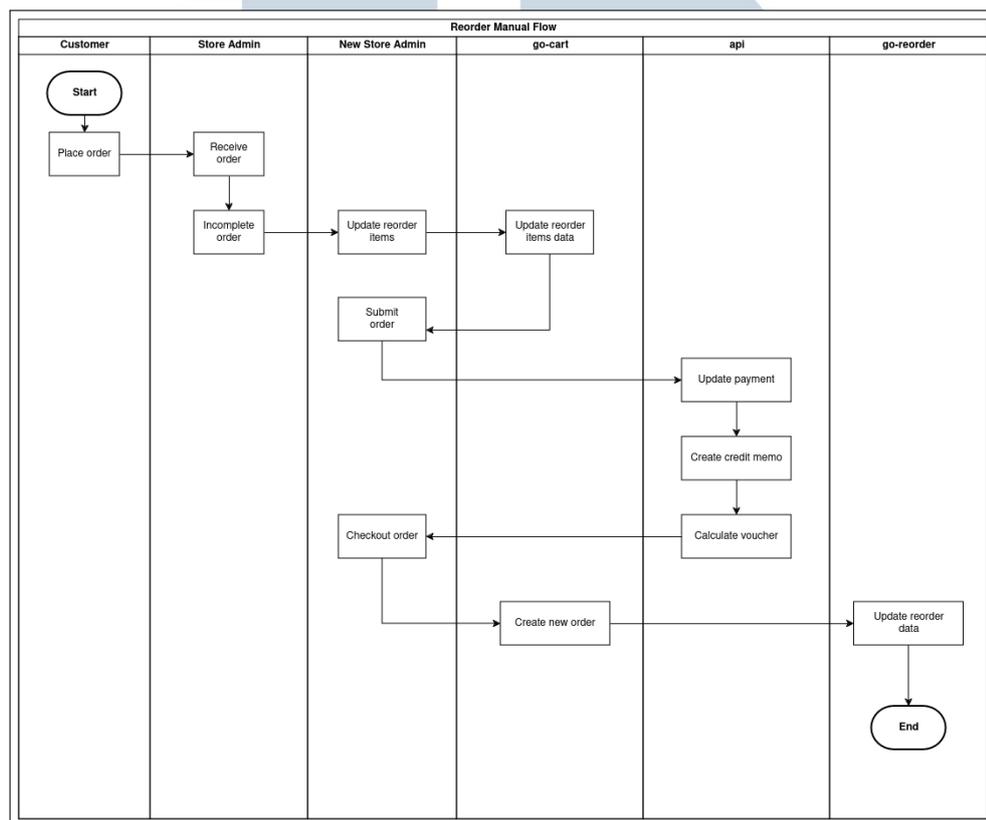


Gambar 3.3. Flow Assignment Reorder

Gambar 3.3 menunjukkan alur dari fitur pemesanan ulang dengan mekanisme *assignment*. *Batch assignment* ini dapat memproses lebih dari satu *invoice* yang masih menunggu proses *reorder*. Setiap *invoice* akan melalui proses pengecekan stok, perhitungan perbandingan penggunaan kurir baru atau lama, dan proses pertimbangan konfirmasi pelanggan. Apabila terjadi kegagalan validasi pada logika bisnis, maka *invoice* tersebut akan di-*refund* secara otomatis. *Refund* pada *invoice* juga dapat terjadi apabila metode pengiriman *invoice* tersebut adalah *pickup*.

- *Manual*, yaitu pelaksanaan *reorder* yang dilaksanakan secara manual oleh

pengelola toko terkait untuk mengarahkan pemetaan toko dan barang secara manual. Pada mekanisme ini, pengelola toko dapat menambahkan atau mengubah daftar barang yang dibeli.



Gambar 3.4. Flow *Manual Reorder*

Gambar 3.4 menunjukkan alur dari fitur pemesanan ulang dengan mekanisme *manual*. Pemesanan ulang ini dilakukan langsung oleh *admin* toko yang bersangkutan untuk manajemen produk yang dibeli, dan dapat secara komunikatif antara pelanggan dengan *admin* toko karena pelanggan dapat mengajukan permintaan untuk mengubah daftar produk yang ingin dibeli. Setelah itu, *admin* toko akan melakukan *checkout* dan pembuatan *order* baru secara manual dan mengajukan pembayaran tambahan dari pelanggan apabila terdapat tambahan produk selama proses *reorder manual*.

### 3.3.2 Mempelajari Arsitektur Sistem Back-End dan Design Pattern

Sistem *Back-End* yang diterapkan adalah *microservice* dengan *berbagai subservice* yang ditulis dengan bahasa *Go* dengan *framework Echo*, serta *Core*

*API / Core Service* dengan *framework Phalcon PHP*. *Endpoint* atau logika bisnis yang dieksekusi pada *Core API* akan langsung dieksekusi pada *service* tersebut, dan menggunakan *API Gateway* untuk logika bisnis yang dieksekusi pada *subservice*.

Komunikasi antar *service* dilakukan melalui dua metode, yaitu *API-Driven* yang melalui *HTTP request*, dan *Event-Driven* yang memanfaatkan *message broker* untuk komunikasi *asynchronous* antar *service*. *Message broker* yang digunakan adalah *NSQ* dan *Amazon SQS (Simple Queue Service)*. Selain itu, terdapat pemrosesan yang bersifat *batch* untuk beberapa logika bisnis tertentu yang biasanya berskala besar yang dieksekusi melalui *scheduler cron* ataupun secara manual.

*Database* yang digunakan adalah *MySQL* dan *MongoDB* sebagai penyimpanan utama, *Redis* sebagai penyimpanan *cache* terdistribusi, dan *Elastic Search* sebagai penyimpanan yang sekaligus digunakan sebagai *search engine*. *MySQL* digunakan untuk penyimpanan data seperti informasi terkait *order*, *invoice*, *kurir*, dan lain-lain. *MongoDB* digunakan untuk menyimpan data keranjang pembelian pelanggan, data stok produk toko, dan data lainnya yang menyimpan berbagai informasi dalam jumlah yang besar dari berbagai konteks yang berbeda. *Redis* digunakan sebagai mekanisme penyimpanan bersifat *caching* untuk data-data yang bersifat statis seperti informasi mengenai metadata lokasi toko. *Elastic Search* digunakan untuk data produk-produk agar proses pencarian produk melalui berbagai parameter oleh pelanggan dapat dieksekusi dengan cepat.

*Design Pattern* yang diterapkan pada sebagian besar *service* memanfaatkan *layered architecture*. Arsitektur ini merupakan pola yang mengisolasi antara proses logika bisnis dan pengambilan data, dan terdiri dari tiga bagian diantaranya sebagai berikut:

- *Presentation Layer*, yaitu *layer* yang mengendalikan validasi input atau *request* serta konversi hasil logika bisnis menjadi *response endpoint*.
- *Service / Business Layer*, yaitu *layer* yang menjadi pusat eksekusi seluruh logika bisnis, seperti operasi pengecekan *conditional*, proses *looping*, lokasi eksekusi *query database*, dan lain sebagainya.
- *Repository / Data Layer*, yaitu *layer* yang terisolasi khusus untuk operasi *CRUD (Create Read Update Delete)* pada *database*.

Implementasi *layered architecture* pada *service* berbasis *Go* memanfaatkan fitur *receiver* dan konsep *dependency injection*. *Dependency Injection* pada konteks ini digunakan untuk menginstansiasi setiap *layer* yang menjadi *dependency*

terhadap *layer* lainnya, seperti *service layer* yang dapat mengakses obyek pada *repository layer* dan instansiasi koneksi *database* sebagai *dependencies* pada *repository layer* yang dapat diakses sebagai atribut dari obyek *repository layer*. Untuk *service* berbasis PHP, *layered architecture* diimplementasikan dengan memanfaatkan *class* atau paradigma OOP (*Object-Oriented Programming*).

### 3.3.3 Integrasi Vendor Pengiriman Shipper dengan Fitur Reorder

Sebelumnya, tidak ada pergantian jenis kurir pada proses *reorder*. Proses pengiriman selalu menggunakan kurir yang berasal dari berbagai *third-party* reguler seperti JNE ataupun *express* seperti *Gojek* dan *Grab*. Pada proyek baru ini, Ruparupa akan mengintegrasikan vendor logistik Shipper dengan tujuan untuk mendapatkan harga pengiriman yang lebih murah.

Tabel 3.2. Pemetaan Vendor Shipper untuk *Reorder* dengan Kurir *Parent* Shipper

Order Parent Carrier	Option Reorder	Action	Result
<i>Pickup</i>	Konfirmasi pelanggan	<i>Delivery</i>	Kurir <i>reguler</i> atau <i>trucking</i> termurah
<i>Express</i> ≤ 25 KM	Otomatis	-	Kurir <i>express</i> termurah
<i>Express</i> > 25 KM	Konfirmasi pelanggan	<i>Delivery</i>	Kurir <i>reguler</i> termurah
Shipper <i>reguler</i>	Otomatis	-	Kurir <i>reguler</i> termurah
Shipper <i>trucking</i>	Otomatis	-	Kurir <i>trucking</i> termurah

Tabel 3.2 menunjukkan pemetaan kurir untuk *reorder* apabila *invoice* asli yang tidak dapat dipenuhi (*parent invoice*) menggunakan vendor Shipper. Pada *task* ini, dilakukan integrasi dengan *endpoint* yang dikembangkan dari tim *shipment* dengan mengambil ID dari hasil kurir termurah untuk disimpan pada data keranjang dan tabel pengiriman, sesuai dengan pemetaan pada tabel 3.2 *Endpoint* tersebut bertugas untuk mengambil data terkait pengiriman, seperti data lokasi pelanggan, lokasi toko, dimensi barang, detail produk sebagai parameter untuk perhitungan harga pada *endpoint* milik vendor Shipper.

Tabel 3.3. Pemetaan Vendor Shipper untuk *Reorder* dengan Kurir *Parent 3PL*

Order Parent Carrier	Option Reorder	Action	Result
<i>Pickup</i>	Konfirmasi pelanggan	<i>Delivery</i>	Kurir <i>reguler</i> atau <i>trucking</i> termurah
<i>Express</i> ≤ 25 KM	Otomatis	-	Tidak berubah
<i>Express</i> > 25 KM	Konfirmasi pelanggan	<i>Delivery</i>	Kurir <i>reguler</i> termurah
<i>Reguler</i>	Otomatis	-	Kurir <i>reguler</i> atau <i>trucking</i> termurah
<i>Trucking</i>	Otomatis	-	Kurir <i>reguler</i> atau <i>trucking</i> termurah

Tabel 3.3 menunjukkan pemetaan kurir untuk *reorder* apabila *invoice* asli yang tidak dapat dipenuhi (*parent invoice*) menggunakan kurir 3PL non-Shipper. Secara *logic*, penyimpanan data ID kurir tidak berbeda dengan *parent* kurir Shipper. Perbedaannya hanya terletak pada pemetaan sesuai dengan kebutuhan bisnis.

Dari segi *development* pribadi, dilakukan perubahan pada *service* yang bertanggung-jawab atas fitur *reorder* dan fitur *keranjang* dengan perubahan sebagai berikut:

- Membuat fungsi untuk melakukan *API call* pada *endpoint* dari *service* pengiriman untuk mendapatkan data kurir Shipper termurah.
- Menyesuaikan data kurir pada keranjang, detail *reorder*, tabel pengiriman, *invoice*, dan lain-lain.
- Menyesuaikan *logic* untuk pertimbangan konfirmasi pelanggan sesuai dengan jenis kurir Shipper yang didapat dan pemetaan yang telah diberikan.

### 3.3.4 Memperbaiki Bug Voucher pada Fitur Reorder

Sistem *voucher* yang diimplementasikan oleh tim *marketing* secara teknikal diaplikasikan kepada setiap data barang yang dibeli pada tabel barang yang di-*order* dengan tipe data string dari struktur JSON (*JavaScript Object Notation*) dari hierarki data yang dibentuk dari proses logika bisnis *marketing*. Proses aplikasi *marketing rule* dapat dilaksanakan secara *end-to-end* oleh tim operasional melalui *dashboard* khusus yang terdedikasi untuk manajemen *marketing rule*.

**Alamat Pengiriman** Pilih Alamat Lain

↑ jakbar  
qweqwe (083831232132)  
qwewqwqw, Angke, Tambora, Kota. Jakarta Barat, DKI Jakarta, 11330

**Dikirim ke Alamat (1 Toko)**

Pengiriman 1  
Jakarta Barat - ACE Grand Paragon Mall

**Krischef Pompa Air Galon Rechargeable - Putih**  
Putih, Dispenser Air Minum  
Rp159.900 x 1  
FREE GIFT  
Proclean 66x43 Cm Lap Chamois Pva  
Gratis: 1(pcs)

**Luggo 20 Inchi Ux-Hybrid Koper Pc Tsa Lock Pearl - Sil...**  
Silver, Tas Trolly  
Rp2.999.000 x 1  
FREE GIFT  
Proclean 66x43 Cm Lap Chamois Pva  
Gratis: 1(pcs)

**Detail Rincian Pembayaran**

Subtotal Harga (4 produk) Rp3.332.700  
Perbedaan Harga Zona Rp0  
Total Ongkir Rp44.000  
Total Diskon Ongkir -Rp44.000  
Promo Produk -Rp173.800

Ajukan Faktur Pajak

**Total Pembayaran** Rp 3.158.900

Pilih Metode Pembayaran

Gambar 3.5. Contoh skenario pemesanan yang menimbulkan *bug* saat dilakukan pemesanan ulang

Gambar 3.5 menunjukkan pengaplikasian sistem *voucher* yang menimbulkan kesalahan pada daftar barang yang di-*reorder* apabila terdapat lebih dari satu produk, dan terdapat paling sedikit dua diantara barang tersebut yang memiliki promo *free item* dengan SKU (*Stock Keeping Unit*) yang sama.

Product	Item Status	Original Price	Price	Qty	Stock Assignment	Subtotal	Discount Amount	Handling Fee	Row Total	Shipment Status
KRISCHIEF POMPA AIR GALON RECHARGEABLE - PUTIH SKU: 10456565	Invoiced (INV270799401)	Rp. 159.900.00	Rp. 159.900.00	1	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 159.900.00	Rp. 0.00	Rp. 0.00	Rp. 159.900.00	New - JNE JTR
LUGGO 20 INCI UX-HYBRID KOPER PC TSA LOCK PEARL - SILVER SKU: 10540022	Invoiced (INV270799401)	Rp. 2.999.000.00	Rp. 2.999.000.00	2	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 5.998.000.00	Rp. 0.00	Rp. 0.00	Rp. 5.998.000.00	New - JNE JTR
PROCLEAN 66X43 CM LAP CHAMOIS PVA SKU: 10085163	Invoiced (INV270799401)	Rp. 86.900.00	Rp. 86.900.00	1	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 86.900.00	Rp. 86.900.00	Rp. 0.00	Rp. 0.00	New - JNE JTR
PROCLEAN 66X43 CM LAP CHAMOIS PVA SKU: 10085163	Invoiced (INV270799401)	Rp. 86.900.00	Rp. 86.900.00	1	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 86.900.00	Rp. 86.900.00	Rp. 0.00	Rp. 0.00	New - JNE JTR
PROCLEAN 66X43 CM LAP CHAMOIS PVA SKU: 10085163	Invoiced (INV270799401)	Rp. 86.900.00	Rp. 86.900.00	2	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 173.800.00	Rp. 173.800.00	Rp. 0.00	Rp. 0.00	New - JNE JTR
PROCLEAN 66X43 CM LAP CHAMOIS PVA SKU: 10085163	Invoiced (INV270799401)	Rp. 86.900.00	Rp. 86.900.00	2	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 173.800.00	Rp. 173.800.00	Rp. 0.00	Rp. 0.00	New - JNE JTR

Gambar 3.6. Contoh hasil pemesanan ulang (*reorder*)

Gambar 3.6 menunjukkan daftar barang setelah dilakukan *reorder*. Jumlah total barang *reorder* dan *order* asli seharusnya tidak mengalami perubahan, namun pada kasus ini barang yang tergolong *free item* hasil dari aplikasi *marketing rule* mengalami duplikasi.



Gambar 3.7. Contoh hasil status *parent invoice* yang di-reorder

Gambar 3.7 menunjukkan status hasil *refund* pada *parent invoice* yang di-reorder. Pada umumnya, proses *reorder* akan mengubah status dari *order* sebelumnya menjadi *full refund* atau *partial refund* sebagai mekanisme untuk invalidasi proses *order* lama. *Full refund* menandakan bahwa seluruh produk pada *order* tersebut dialihkan ke toko lain, dan *partial refund* menandakan bahwa hanya sebagian produk yang dialihkan (*reorder*) ke toko lain. Pada kasus *reorder* dengan *free item*, barang yang tergolong sebagai *free item* tidak akan tercatat pada daftar *refund*, sehingga menyebabkan perubahan status menjadi *partial refund* yang dimana seharusnya produk *parent* dan *free item* memiliki status *refund* yang menjadikan status *full refund* pada *order* referensi.

Setelah dilakukan proses *tracing* dan *debugging* melalui *logs* aplikasi dan dengan bantuan fitur *debugger* pada *text editor* yang digunakan, terdapat celah pada logika bisnis pada *codebase* yang melakukan proses pengolahan data pada data keranjang *reorder* untuk bagian pengolahan ulang *marketing rule* pada setiap barang *free*, seperti perhitungan ulang *subtotal*, *identifier parent* untuk mengidentifikasi SKU yang mengaktifasi *free item*, *flag* untuk *free item*, dan lain-lain. Proses logika bisnis terjadi pada proses penyimpanan keranjang *reorder* baru yang mengaplikasikan *for loop* pada tiga variabel, yaitu variabel *array cart.Items* yang akan menampung data barang yang termasuk dengan daftar SKU dari *marketing rule*, variabel *array orderDetail.Items* yang menampung data seluruh barang, dan variabel *array cartRules* yang menampung daftar *marketing rule* yang teraplikasi pada keranjang. Proses bisnis tersebut tidak melakukan validasi kesamaan SKU pada setiap fase *looping* pada *inner loop* yang dapat tereksekusi sesuai dengan jumlah elemen pada *array* yang menampung seluruh produk.

Solusi yang diterapkan adalah dengan menambahkan variabel *boolean* (*freeItemAssigned*) yang berfungsi sebagai *flag* agar *free item* tidak ditambahkan lebih dari satu kali untuk setiap *marketing rule* per produk yang dibeli (*non free*

item) pada keranjang yang akan digunakan sebagai data *reorder*.

Product	Item Status	Original Price	Price	Qty	Stock Assignment	Subtotal	Discount Amount	Handling Fee	Row Total	Shipment Status
KRISCHEF POMPA AIR GALON RECHARGEABLE - PUTH SKU: 1045655	Invoiced (INV104323683)	Rp. 159.900.00	Rp. 159.900.00	1	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 159.900.00	Rp. 0.00	Rp. 0.00	Rp. 159.900.00	New - JNE JTR
LUUGO 20 INCI UX-HYBRID KOPER PC TSA LOCK PEARL - SILVER SKU: 10540022	Invoiced (INV104323683)	Rp. 2.999.000.00	Rp. 2.999.000.00	2	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 5.998.000.00	Rp. 0.00	Rp. 0.00	Rp. 5.998.000.00	New - JNE JTR
PROCLEAN 66X43 CM LAP CHAMOIS PVA SKU: 10085163	Invoiced (INV104323683)	Rp. 86.900.00	Rp. 86.900.00	1	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 86.900.00	Rp. 86.900.00	Rp. 0.00	Rp. 0.00	New - JNE JTR
PROCLEAN 66X43 CM LAP CHAMOIS PVA SKU: 10085163	Invoiced (INV104323683)	Rp. 86.900.00	Rp. 86.900.00	2	A601 (Jakarta Barat - ACE Living Plaza Puri PX) - Store fulfillment	Rp. 173.800.00	Rp. 173.800.00	Rp. 0.00	Rp. 0.00	New - JNE JTR

Gambar 3.8. Contoh hasil *fix* daftar barang yang di-*reorder*

Gambar 3.8 menunjukkan hasil perbaikan dari sisi *end-user*. Jumlah produk setelah dilakukan proses *reorder* sesuai dengan jumlah produk awal, sesuai dengan ekspektasi dari segi alur bisnis.

Untuk perbaikan isu terkait *partial refund* pada *reorder* dengan *free items*, solusi yang diterapkan adalah menghapuskan validasi pada *codebase* Core API yang melakukan pengecualian pada *free items*, karena hasil proses *tracing* dan *debugging* menunjukkan bahwa validasi tersebut menghilangkan data produk *free* pada *array* produk yang akan digunakan untuk logika bisnis *refund* dan *voucher*.

Invoice #INV803946482   full_refund   27 Feb 2024 10:50:47	
Order #ODI21357736221   full_refund   27 Feb 2024 10:50:47	
Order Date	27 Feb 2024 10:50:05
Order Status	complete
Purchased From	desktop
Placed From IP	172.16.0.244

Gambar 3.9. Contoh hasil *fix* status *parent invoice* yang di-*reorder*

Gambar 3.9 menunjukkan hasil perbaikan dari sisi *end-user*. Status *refund* berubah menjadi *full refund* untuk *reorder* dengan *free items*.

### 3.3.5 Menulis Dokumentasi Service dan Codebase

Selama praktik kerja magang, dilakukan penulisan dokumentasi API (*Application Programming Interface*) untuk *endpoint reorder*, beserta dokumentasi terkait alur logika bisnis yang diterapkan didalamnya. Selain itu, dilakukan juga penulisan terkait gambaran besar dan deskripsi dari segi bisnis terkait fitur *reorder*

beserta dengan istilah-istilah penting yang perlu diketahui terkait dengan *reorder*. Dokumentasi dilakukan pada *platform* Notion.

7. **Create Reorder** : Used to incomplete order so that other store can bid and take over the order.

```
curl --location '██████████' \
--header 'Content-Type: application/json' \
--data '{
  "customer_invoice_id": "██████████"
}'
```

Request body :

███: Customer's invoice number.

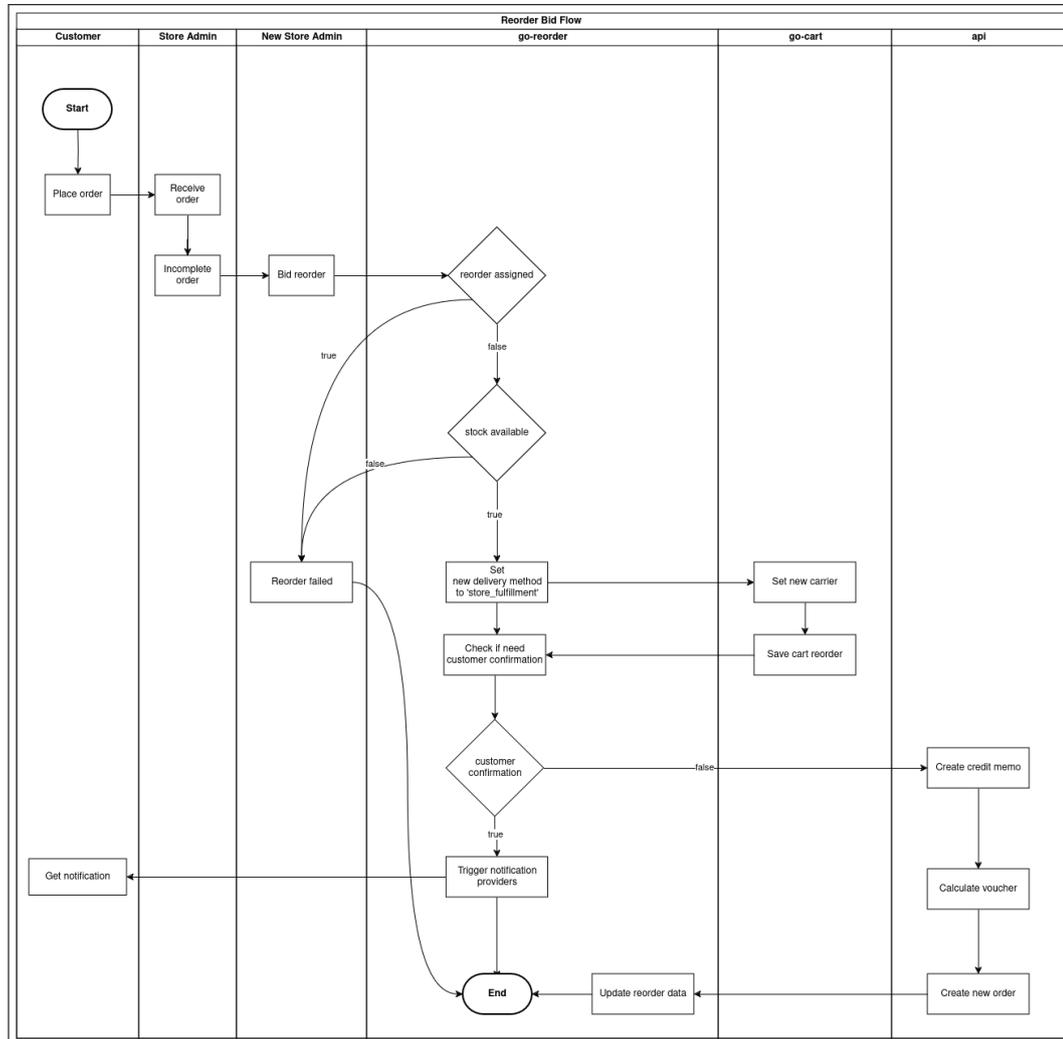
Flow :

- Get invoice data by passing the ███ from table ███ joined along with ███ and ███.
- Validate to not create reorder if order comes from new retail, *Shopee* (███ as order number prefix), or *Tokopedia* (███ as order number prefix).
- Get invoice items from table ███. If invoice item does not exist then return failed reorder creation response.
- Get parent invoice ID by passing ███. Data is retrieved from table ███.
- If the given invoice reorder count is more than 3 times (this create reorder request is included), then create refund for the invoice (reorder is canceled due to reorder count threshold is set to 3). Refund creation is processed on ███.
- If reorder count has not passed the reorder threshold, then check based on the reorder existence of the given ███
  - If reorder already exists and status is not ███, then send reorder has been created error.
  - If reorder is new, then create reorder SLA (*Service Level Agreement*) time for district, city, and province.
    - SLA min time is ███ GMT.
    - SLA max time is ███ GMT.
    - District SLA interval is ███ minutes.
    - City SLA interval is ███ minutes.
    - Province SLA interval is ███ hours.
    - For each region SLA calculation (district, city, province) :

Gambar 3.10. Contoh Dokumentasi API

Gambar 3.10 menunjukkan contoh dokumentasi yang dilakukan untuk setiap *endpoint reorder*. Detail yang dicantumkan meliputi format *cURL*, *request*

*payload*, dan alur logika bisnis yang disertakan dengan kebutuhan bisnis.



Gambar 3.11. Contoh dokumentasi flow bisnis

Gambar 3.11 menunjukkan contoh dokumentasi atau penjelasan terkait garis besar dari salah satu fitur *reorder* dengan menggunakan *swimlane diagram*. Diagram ini digunakan untuk menjelaskan alur komunikasi dari *user* dengan atau antar sistem atau *service* dari *back-end*.

### 3.4 Kendala dan Solusi yang Ditemukan

Kendala yang ditemukan adalah minimnya dokumentasi terkait *repository* beserta alur bisnis secara teknikal untuk berbagai alur bisnis ataupun setiap *endpoint*. Hal ini menimbulkan kesulitan bagi *developer* yang memiliki

penugasan terkait suatu fitur yang belum pernah dikerjakan sebelumnya, sehingga menimbulkan ketergantungan kepada *developer* lain untuk *onboarding*, serta potensi terjadinya kesalahpahaman alur bisnis ketika pendekatan dilakukan secara mandiri. Solusi yang ditemukan adalah dengan menulis dokumentasi *repository* yang tidak hanya mencakup detail *endpoint*, namun menyertakan juga alur baik itu dari segi bisnis maupun secara teknis dalam bentuk diagram.

Selain itu, terdapat inkonsistensi terhadap struktur *response endpoint* pada beberapa *repository* atau *service*, seperti tidak menggunakan *status code* yang sesuai dengan skenario *error* yang dikembalikan, tidak melakukan *passing* pesan *error* dari *endpoint* yang melakukan pemanggilan API (*Application Programming Interface*) dari *service* lain, struktur *response* yang terkadang tidak konsisten sehingga terkadang menyebabkan kesulitan dan inkonsistensi untuk melakukan pengolahan terhadap *response*. Solusi yang ditemukan adalah dengan melakukan *refactoring* agar *status code* yang diberikan sesuai dengan konteks dan memperbaiki beberapa bagian yang tidak konsisten untuk mengikuti satu standar yang telah ditetapkan dari *team lead*.

Mekanisme sistem *logging* pada beberapa skenario komunikasi antar beberapa *service* tidak dilakukan secara konsisten. Secara standar, setiap *log* pemanggilan dan *response* API yang masuk disertakan dengan *request identifier* yang memudahkan proses *debugging* dan *tracing* terhadap *error*, khususnya ketika dilakukan pada *server staging* yang seringkali digunakan oleh lebih dari satu orang yang menyebabkan intensitas *write* pada *log file* meningkat dan tercampur antar satu dengan yang lainnya. Mekanisme ini belum diterapkan secara menyeluruh pada seluruh *repository*. Solusi yang ditemukan adalah mengimplementasikan pengiriman *request identifier* melalui *request header* apabila belum diimplementasikan sebelumnya, serta menambahkan *logic* pada *middleware* untuk melakukan ekstraksi *request identifier* yang dikirimkan dan disimpan pada *request context*.

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