Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

Perpustakaan Univeristas Multimedia Nusantara

BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Proses pelaksanaan kerja magang di PT Tokopedia diawali dengan dilakukannya sesi *pre-onboarding* selama 2 minggu yang dilakukan untuk membantu mengenal perusahaan Tokopedia, mengetahui visi dan misi perusahaan, budaya di perusahaan, dan beberapa hal yang perlu dilakukan selama bekerja di Tokopedia. Setelah menyelesaikan masa *pre-onboarding*, proses pelaksanaan kerja magang dilakukan sebagai posisi *Software Engineer Intern* yang berada di dalam naungan divisi *Logistics* Tokopedia. Kerja magang dilakukan dalam tim *Fulfillment Core* dibawah bimbingan Aditya Mili selaku mentor dan *Software Engineer Lead*.

Proses kerja magang di dalam tim *Fulfillment Core* melibatkan pengembangan sistem *backend* dari produk TokoCabang di Tokopedia. Di dalam tim, tidak terdapat perbedaan pekerjaan antara *Software Engineer* maupun *Software Engineer Intern*. Selama melakukan proses kerja magang, koordinasi seringkali dilakukan antara setiap anggota tim dan juga dengan anggota dari tim *Fulfillment External* yang juga berperan dalam pengembangan *backend* produk TokoCabang. Selain itu dalam pengembangan rasio penjualan di TokoCabang, koordinasi dengan *Product Manager*, *Test Engineer* dan *Web Platform Engineer* juga seringkali dilakukan agar hasil rasio penjualan yang ditampilkan ke penjual sesuai dengan yang diharapkan.

3.2 Tugas yang Dilakukan

Selama menjalani proses kerja magang, tugas utama yang dilakukan adalah pengembangan sistem *backend* dari produk TokoCabang di Tokopedia, khususnya dalam membangun fitur rasio penjualan untuk metode pembayaran *Storage Fee* baru di TokoCabang. Teknologi-teknologi yang digunakan dalam pengembangan fitur tersebut diantaranya adalah:

- Bahasa pemrograman Go
- Database PostgreSQL
- GraphQL

- Feature flag

Dalam proses kerja magang, tugas baru akan diberikan setiap satu *sprint*, dimana satu *sprint* berdurasi satu minggu. Untuk setiap tugas yang diberikan, akan diberikan suatu persyaratan yang harus diselesaikan dan akan ditentukan bersama-sama lama waktu pengerjaan tugas tersebut. Ketika dalam tugas yang diberikan terdapat fitur yang berubah atau dikembangkan, maka akan diperlukan untuk membuat *unit tests* dan melakukan testing pada *local*, *staging*, dan *production*.

3.3 Uraian Pelaksanaan Magang

Proses kerja magang untuk membangun fitur rasio penjualan di TokoCabang dilakukan selama kurun waktu 15 minggu. Berikut merupakan uraian kegiatan serta penjelasan setiap tugas-tugas yang dilakukan selama pelaksanaan kerja magang pada Tabel 3.1.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1 - 2	• Menjalani masa *pre-onboarding* • Menjalani acara *#NakaMakeItBetter challenge*
3	• Menjalani masa *onboarding* ke dalam tim • *Setup device* dan mempelajari *tech stack* yang akan digunakan
4 - 7	Melakukan task untuk mempelajari isi dari repository warehouse
8 - 9	• Membuat *table* baru di Database yang digunakan untuk menampung *cluster* suatu produk berdasarkan usia produk • Memperbarui alur pengolahan data produk pada *cron job* untuk mengelompokkan data ke dalam *cluster* usia produk dan di-*insert* ke *table* baru
10 - 11	• Membuat API dan GQL untuk menampilkan data dari *table* yang menampung *cluster* produk berdasarkan usia produk • Membuat fitur *sorting* dari API yang dibuat
12	• Membuat API dan GQL berdasarkan table baru yang telah dibuat untuk menampung hasil perhitungan rasio penjualan setiap produk • Membuat *flow* untuk *shutdown* data dari rasio penjualan sebelumnya untuk setiap pengolahan data produk
13	Membuat API dan GQL berdasarkan table baru yang telah dibuat untuk menampung hasil perhitungan rasio penjualan setiap toko
14	Membuat API untuk menampilkan data *clustering* sebelumnya di dalam bentuk *summary*
15	• Update API dan GQL sebelumnya untuk dapat menerima parameter baru • Membuat GQL untuk menampilkan hasil dari API *summary*

- Minggu pertama dan kedua pada masa kerja magang adalah masa *pre-onboarding* yang digunakan untuk agar peserta kerja magang dapat mengenal perusahaan dengan lebih baik. Peserta kerja magang diperkenalkan dengan visi dan misi perusahaan serta hal-hal yang perlu dilakukan selama menjalani kerja magang. Selain itu, diadakan juga suatu acara bernama *#NakaMakeItBetter challenge*. Acara tersebut mengelompokkan peserta kerja magang dalam kelompok berjumlah 9 orang untuk mempresentasikan suatu ide bisnis yang dapat membantu Tokopedia.
- Minggu ke-3 adalah masa *onboarding* ke dalam masing-masing tim. Masa *onboarding* dilakukan dengan pengenalan ke seluruh anggota tim, penjelasan mengenai *job desc*, dan eskspektasi-ekspektasi kedepannya. Selain itu, minggu ke-3 juga digunakan untuk *setup device* yang diberikan dan mempelajari *tech stack* yang digunakan dalam tim.
- Minggu ke-4 sampai 7 digunakan untuk mempelajari lebih lanjut *repository* yang akan digunakan selama proses kerja magang. Hal tersebut dilakukan dengan menyelesaikan tugas-tugas yang berkaitan dengan eksplorasi *repository*. Tugas yang diberikan pada minggu ke-4 dan ke-6 adalah untuk memperbaiki hasil *query* agar tidak memberikan data yang tidak dibutuhkan dan menghapus *query* yang sudah tidak lagi digunakan. Tugas yang diberikan pada minggu ke-5 adalah memperbaiki *config* yang sudah tidak digunakan. Sedangkan tugas pada minggu ke-7 diberikan untuk mempelajari *module* yang ada serta pengolahan data database.
- Minggu ke-8 dan ke-9 adalah waktu dimulainya ikut serta dalam proyek rasio penjualan. Tugas awal yang diberikan untuk terjun dalam proyek tersebut adalah untuk membuat alur pengelompokkan data ke dalam suatu *cluster* dalam alur pengolahan data produk yang ada dalam *cron job*. Selain itu, *table* database baru juga perlu dibuat agar hasil pengelompokkan data tersebut dapat di-*insert* ke dalam *table* baru.
- Minggu ke-10 sampai ke-11 digunakan untuk membuat API dan GQL yang menampilkan hasil dari pengelompokkan data sebelumnya.Di dalam API yang dibuat diperlukan untuk membuat fitur *filtering* berdasarkan nama produk dan juga *sorting* berdasarkan jumlah produk dalam suatu *cluster*.
- Minggu ke-12 merupakan waktu untuk membuat API dan GQL untuk mendapatkan hasil dari total hasil perhitungan rasio penjualan dari setiap

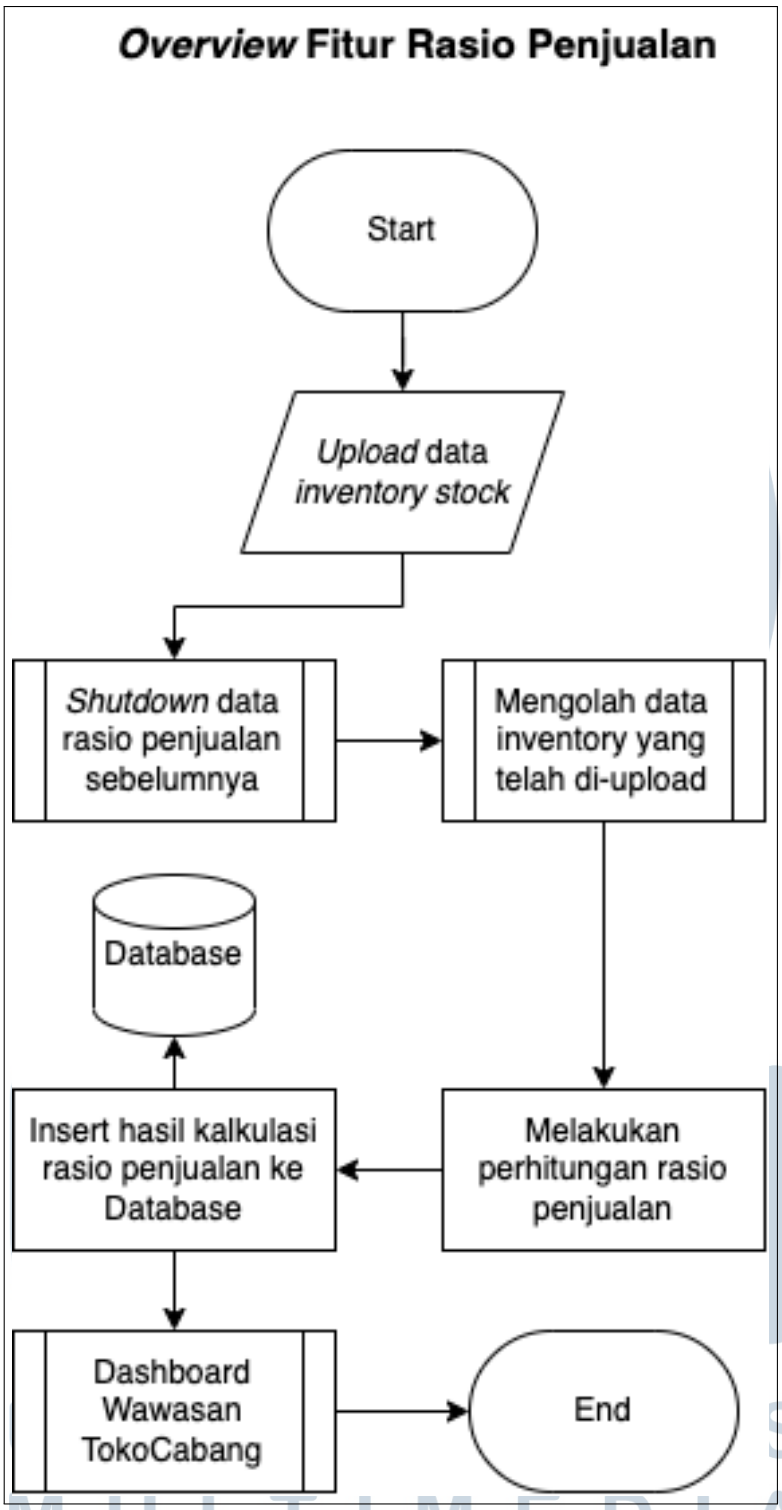
produk yang ada di suatu toko. Hasil dari total perhitungan rasio penjualan ini didapatkan dari suatu *table* baru. Selain membuat GQL, di minggu ke-12 penulis juga turut serta membuat suatu *flow* untuk melakukan *shutdown* data dari rasio penjualan sebelumnya ketika suatu data rasio penjualan baru ingin dibuat.

- Pada minggu ke-13, penulis membuat API serta GQL untuk mendapatkan hasil dari rasio perjualan suatu toko. Sama seperti API dan GQL yang dibuat minggu lalu, perhitungan rasio penjualan per toko ini juga dapat diperoleh dari suatu *table* baru.
- Pada minggu ke-14, waktu digunakan untuk membuat API yang akan menampilkan data *clustering* dalam bentuk *summary*. Dimana *summary* tersebut akan menampung total produk di setiap *cluster* serta label yang berisi deskripsi *cluster* yang digunakan.
- Minggu ke-15 digunakan untuk update beberapa API dan GQL sebelumnya untuk dapat menerima parameter baru. Selain itu, minggu ini juga digunakan untuk membuat GQL berdasarkan API *summary* yang telah dibuat minggu lalu.

3.3.1 Proses Pelaksanaan Kerja Magang

Berikut merupakan suatu diagram alir yang menggambarkan *overview* alur dari fitur rasio penjualan yang akan dibangun pada sistem *backend* TokoCabang.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.1. Flowchart Overview Fitur Rasio Penjualan

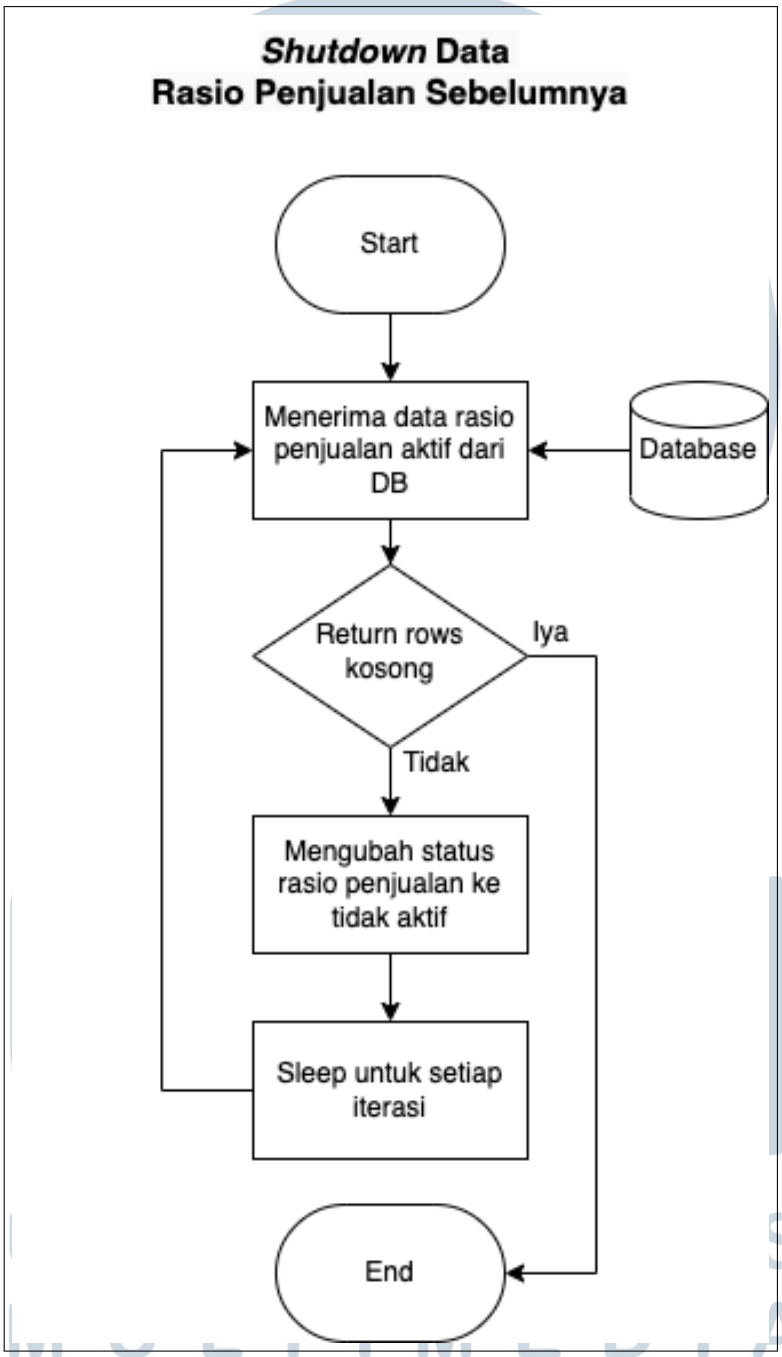
MULTIMEDIA NUSANTARA

Pada gambar 3.1, dapat dilihat bahwa secara garis besar fitur rasio penjualan dimulai ketika terjadi data stok *inventory* di-*upload*. Setelah *upload* dilakukan, maka data dari rasio penjualan sebelumnya yang ada di dalam database akan di non aktifkan karena sudah tidak lagi berlaku setelah data baru diterima. Data *inventory* yang telah diterima tersebut lalu akan diolah untuk dapat digunakan dalam kalkulasi rasio penjualan. Selain itu, data yang telah diolah tersebut akan dikelompokkan ke dalam suatu *cluster* berdasarkan umur dari barang tersebut. Perhitungan rasio penjualan akan dilakukan selanjutnya dan hasil dari perhitungan tersebut akan di-*insert* ke dalam database. Hasil perhitungan yang telah disimpan juga akan ditampilkan ke *seller* TokoCabang dalam bentuk *Dashboard* di halaman Wawasan TokoCabang. Berikut adalah penjeleasan mengenai setiap bagian proses yang ada:

A. *Shutdown* Data Rasio Penjualan Sebelumnya

Agar tetap dapat menyimpan riwayat perhitungan-perhitungan rasio penjualan sebelumnya ke dalam database, maka diperlukan suatu alur yang mendukung suatu *update* terus menerus tanpa menghapus data sebelumnya. Untuk dapat melakukan hal tersebut, maka sistem *backend* yang dibangun akan melakukan *shutdown* ketika data *inventory* stock telah berhasil di-*upload*. Berikut adalah gambaran alur dari *shutdown* data yang dilakukan:

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



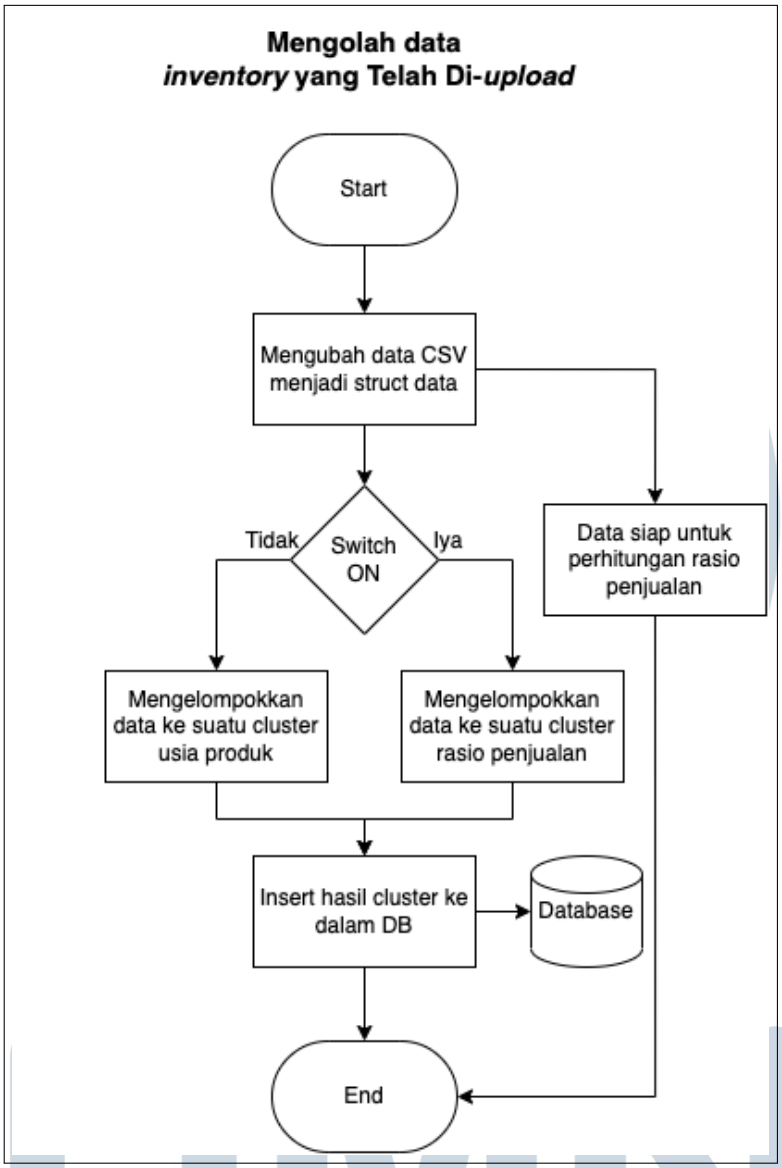
Gambar 3.2. Flowchart Shutdown Data Rasio Penjualan Sebelumnya

Pada gambar 3.2, dapat diketahui bahwa *shutdown* dilakukan dengan menerima data rasio penjualan yang masih aktif di dalam database. Aktif dapat ditandakan dengan status yang ada dalam setiap *rows* data. Jika data yang diperoleh dari database kosong, maka alur dari *shutdown* data selesai. Namun, jika data status aktif terdapat di database maka seluruh data rasio penjualan yang diperoleh tersebut akan segera di-*update* ke status tidak aktif. Untuk menjaga agar *query* yang dilakukan tidak akan memberikan beban yang cukup besar ke database, maka jumlah *rows* yang akan di-*fetch* dan *update* untuk setiap iterasinya akan dibatasi. Selain itu, untuk setiap kali iterasi *shutdown* akan dijalankan juga *sleep* selama beberapa milisecond sebelum dilanjutkan ke iterasi *shutdown* selanjutnya.

B. Mengolah Data *Inventory* yang Telah Di-*upload*

Sebelum dapat digunakan untuk kalkulasi rasio penjualan, data *inventory* yang telah diterima perlu diolah terlebih dahulu. Selain itu, data yang diolah tersebut juga akan digunakan untuk pengelompokkan data berdasarkan *cluster*. Data tersebut akan dikelompokkan ke dalam suatu *cluster* berdasarkan usia produk. Selama pengembangan rasio penjualan masih berjalan, data *cluster* tersebut akan ditampilkan sementara di dalam *dashboard*. Untuk kedepannya, *table* untuk *clustering* usia produk akan digunakan juga untuk pengelompokkan berdasarkan *cluster* rasio penjualan. Berikut merupakan gambaran alur dari pengolahan data *inventory*:

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.3. Flowchart Mengolah Data *Inventory* yang Telah Di-*upload*

UNIVERSITAS MULTIMEDIA NUSANTARA

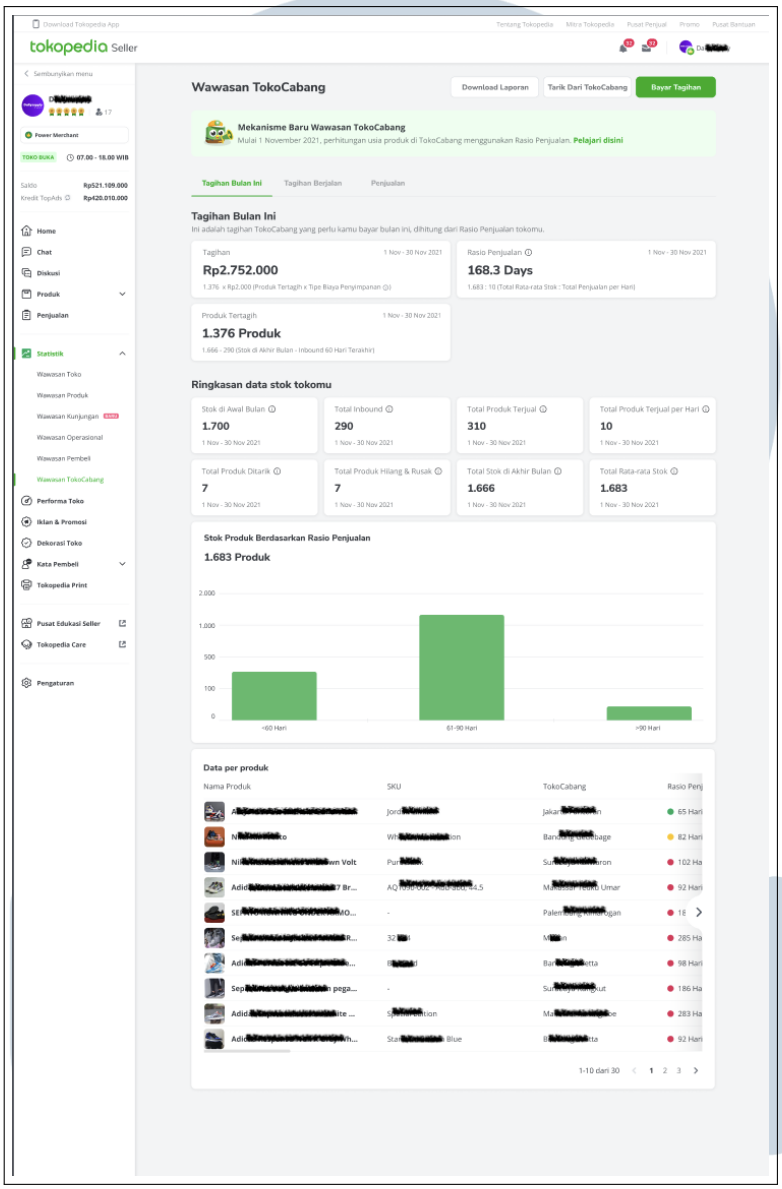
Pada gambar 3.3 dapat dilihat bahwa ketika data *inventory* report diterima, data berupa CSV tersebut akan convert menjadi struct data agar siap untuk digunakan dalam perhitungan rasio penjualan. Selain digunakan untuk perhitungan rasio penjualan, olahan data tersebut juga akan digunakan untuk membuat suatu *cluster*. Selama masih pengembangan sistem rasio penjualan, pengelompokkan data dilakukan berdasarkan *cluster* usia produk. Setelahnya, pengelompokkan akan dilakukan berdasarkan *cluster* rasio penjualan. Agar pengelompokkan dapat diubah dengan mudah, *switch* dengan *feature flag* akan digunakan. Ketika *switch* dalam kondisi *ON* maka pengelompokkan dilakukan berdasarkan rasio penjualan. Sebaliknya, jika *switch* sedang berada dalam kondisi *OFF* maka pengelompokkan tetap dilakukan berdasarkan usia produk.Setelah pengelompokkan berhasil dilakukan, maka hasil *clustering* akan di-*insert* ke dalam database.

C. Dashboard Wawasan TokoCabang

Setelah rasio penjualan telah selesai dihitung dan di-*insert* ke dalam database, maka hasil rasio penjualan tersebut akan ditampilkan dalam *Dashboard* Wawasan TokoCabang. Hasil dari rasio penjualan terbagi menjadi 2 bagian, yaitu rasio penjualan produk dan rasio penjualan toko. Rasio penjualan produk merupakan hasil perhitungan setiap produk yang dimiliki oleh toko tersebut sedangkan rasio penjualan toko merupakan hasil perhitungan dari kumulatif seluruh produk yang dimiliki suatu toko.

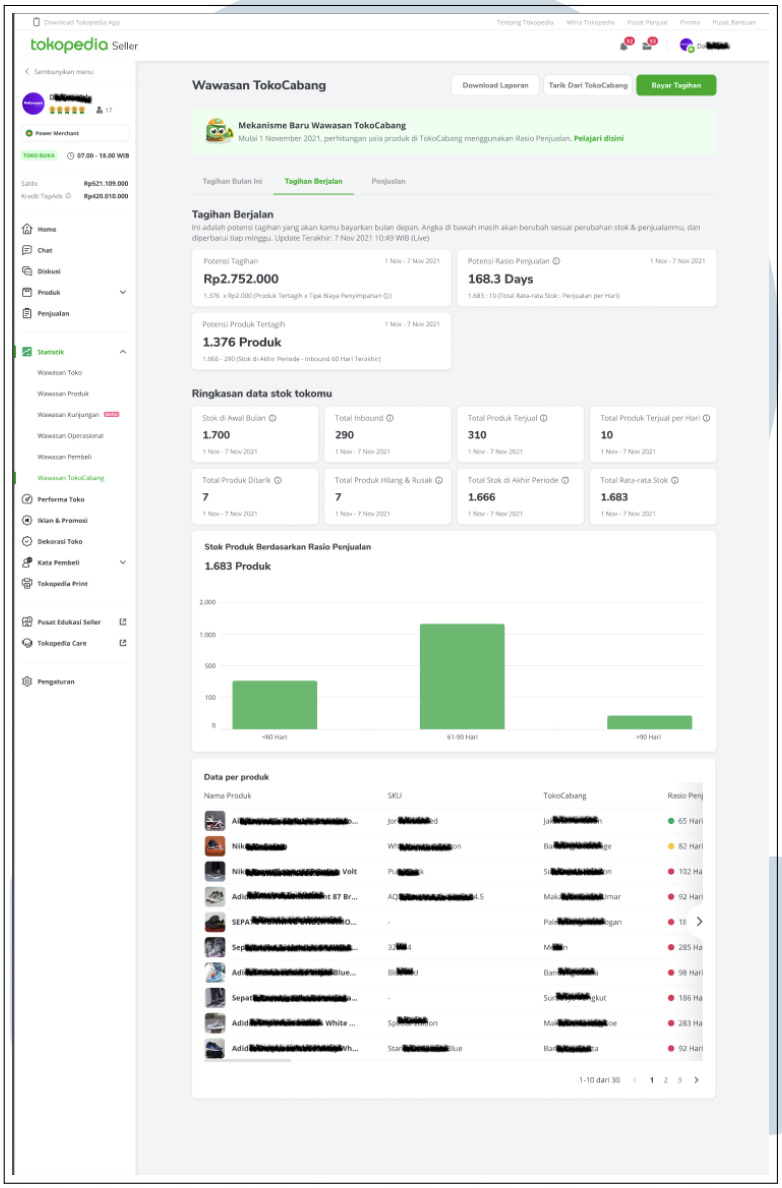
Berikut merupakan design dari halaman *Dashboard* Wawasan TokoCabang yang akan menampilkan hasil rasio penjualan:

UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.4. Design dari Halaman Wawasan TokoCabang Bagian Tagihan Bulan Ini

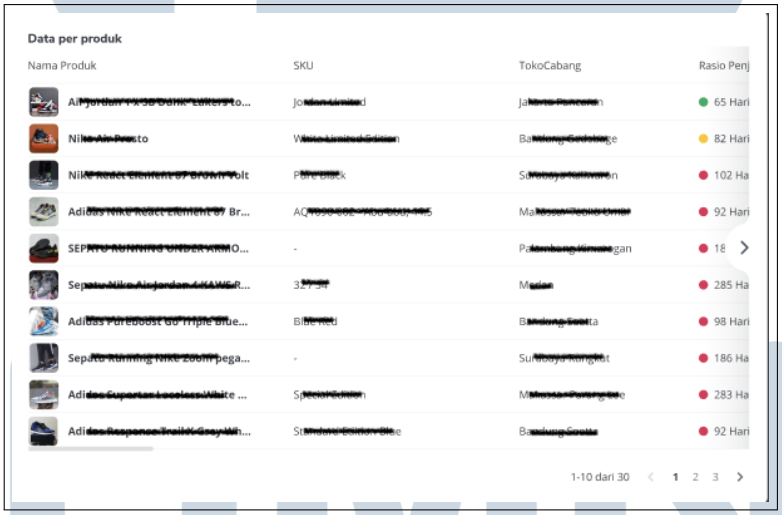
UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.5. Design dari Halaman Wawasan TokoCabang Bagian Tagihan Berjalan

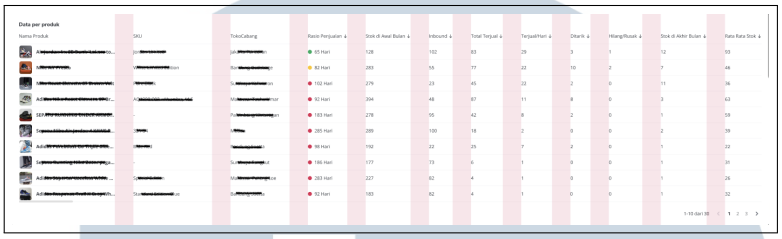
UNIVERSITAS MULTIMEDIA NUSANTARA

Pada gambar 3.4 dan gambar 3.5, dapat diketahui bahwa tampilan dari Wawasan TokoCabang yang berhubungan dengan rasio penjualan akan dibagi menjadi dua bagian. Bagian *tab* tagihan bulan ini yang akan menggunakan rasio penjualan tipe bulanan dan bagian *tab* tagihan berjalan akan menggunakan rasio penjualan tipe mingguan. Hasil dari rasio penjualan produk akan digunakan dalam bagian bawah halaman Wawasan TokoCabang di bagian data per produk. Hasil dari rasio penjualan toko akan digunakan dalam bagian *widget* atas dan ringkasan data stok tokomu. Sedangkan hasil dari *clustering* rasio penjualan produk akan digunakan dalam diagram batang stok produk berdasarkan rasio penjualan. Berikut adalah isi dari bagian-bagian tersebut dengan lebih detail:



Gambar 3.6. Design dari Data Per Produk berdasarkan Hasil Rasio Penjualan Produk

UNIVERSITAS MULTIMEDIA NUSANTARA

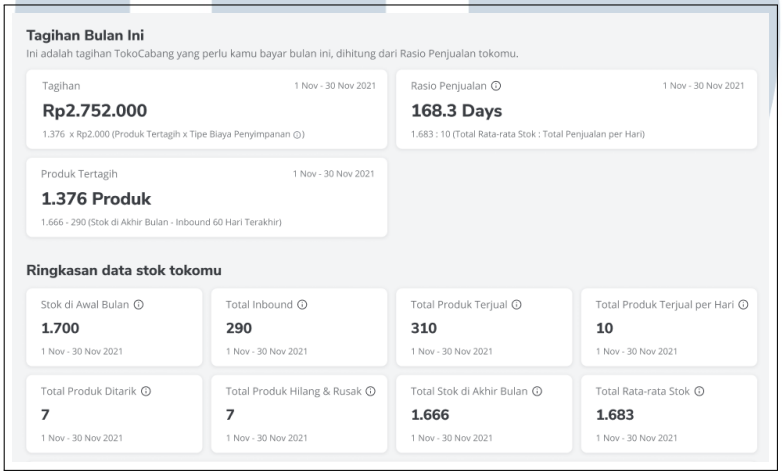


Gambar 3.7. Design dari Data Per Produk berdasarkan Hasil Rasio Penjualan Produk dengan Penuh

Gambar 3.8. Design dari Data Per Produk berdasarkan Hasil Rasio Penjualan Produk dengan Keterangan

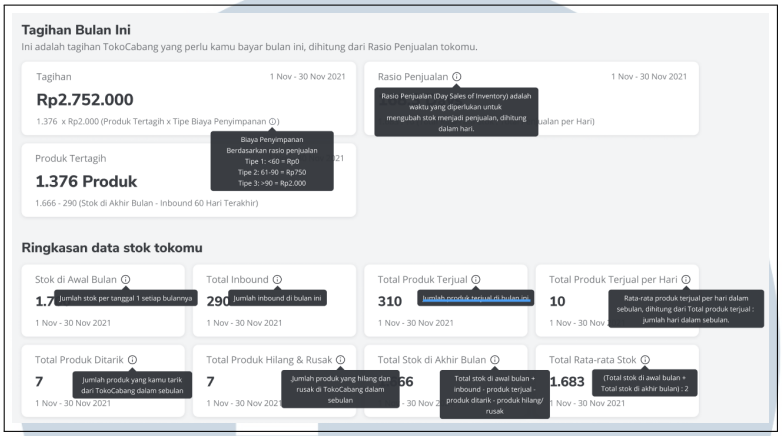
UNIVERSITAS MULTIMEDIA NUSANTARA

Pada gambar 3.6, gambar 3.7, dan gambar 3.8 dapat dilihat bahwa data rasio penjualan produk akan ditampilkan dalam bentuk tabel yang memiliki 12 kolom. Dimana hasil dari tabel tersebut diperoleh melalui hasil *query* GQL. Pada kolom rasio penjualan akan ditambahkan juga suatu indikator berdasarkan tingkat rasio penjualan. Tingkat tersebut dibagi menjadi penjualan produk yang *fast moving*, penjualan produk sedang, dan penjualan produk yang *slow moving*.



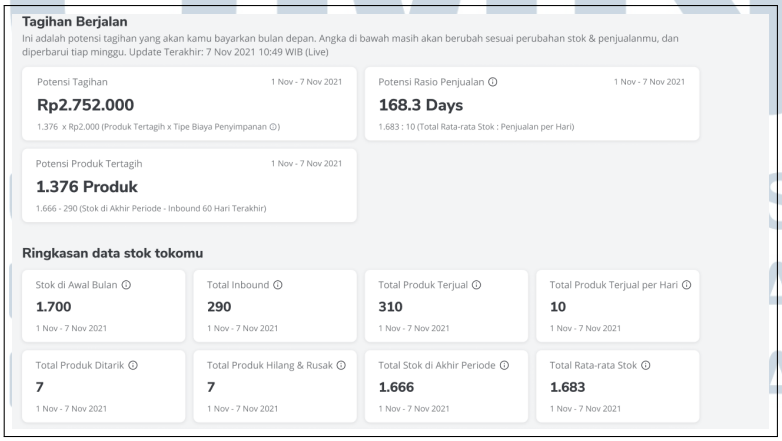
Gambar 3.9. Design dari Tagihan Bulan Ini dan Ringkasan Data Stok Tokomu

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

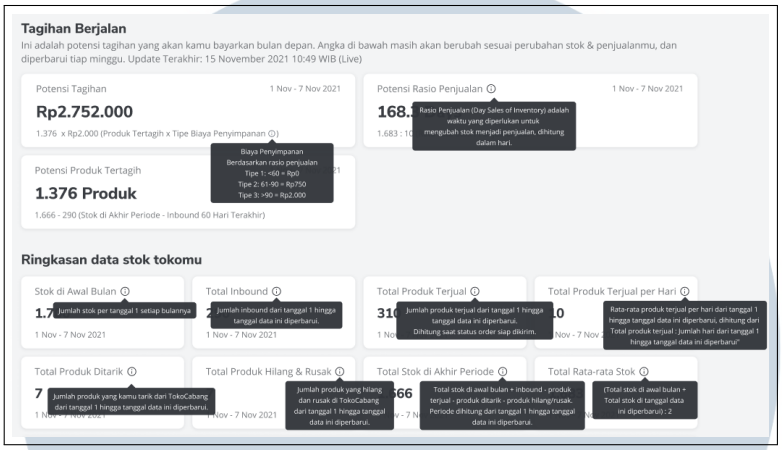


Gambar 3.10. Design dari Tagihan Bulan Ini dan Ringkasan Data Stok Tokomu dengan Keterangan

Pada gambar 3.9 dapat diketahui bahwa hasil dari API *Shop Aging* yang telah di-*query* dengan GQL. Hasil dari setiap *widget* yang ditampilkan merupakan *query-query* berbeda terhadap satu API. Hasil dari *widget* ini menggunakan rasio penjualan toko tipe bulanan. Selain itu, untuk setiap *widget* juga akan memiliki *tooltip* masing-masing seperti gambar 3.10.



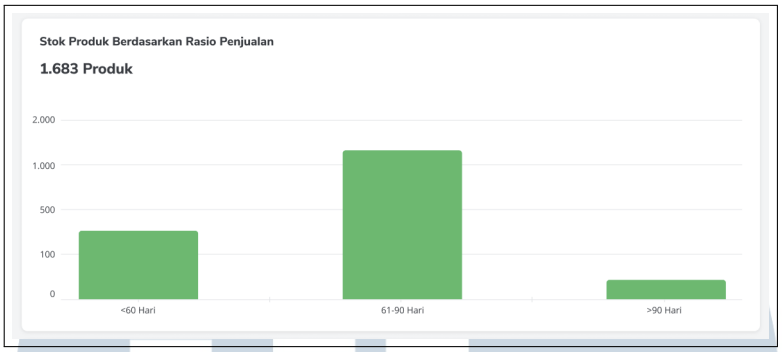
Gambar 3.11. Design dari Tagihan Berjalan dan Ringkasan Data Stok Tokomu



Gambar 3.12. Design dari Tagihan Berjalan dan Ringkasan Data Stok Tokomu dengan Keterangan

Pada gambar 3.11 dan gambar 3.12, dapat dilihat bahwa pada *tab* tagihan berjalan akan menampilkan data yang hampir sama dengan *tab* tagihan bulanan. Namun, *widget* tersebut memiliki nama yang berbeda dengan *tab* sebelumnya. Hasil dari *widget* ini memanfaatkan GQL untuk mendapatkan rasio penjualan toko tipe mingguan. Bagian *widget* ini juga memiliki *tooltip* masing-masing.

UNIVERSITAS MULTIMEDIA NUSANTARA

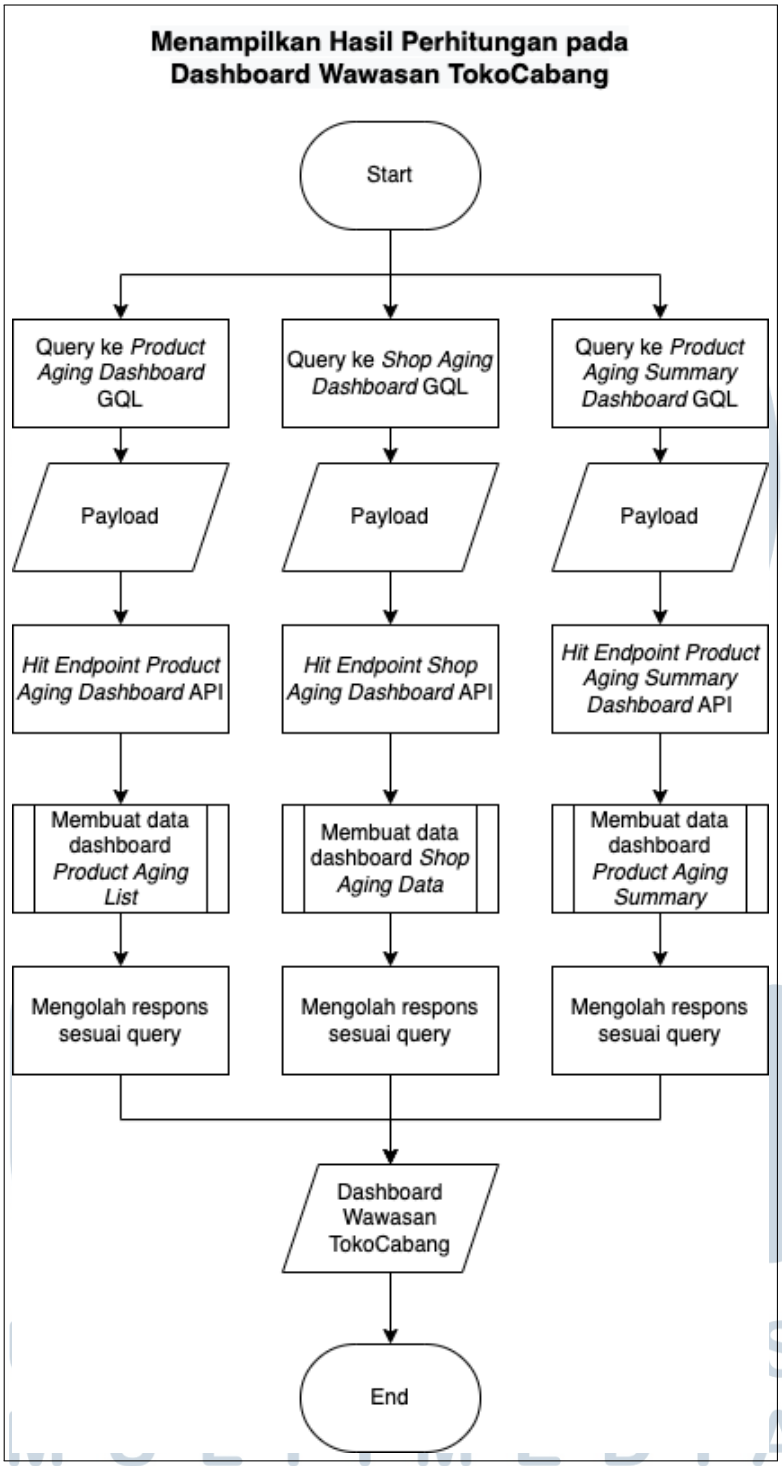


Gambar 3.13. Design dari Stok Produk Berdasarkan Rasio Penjualan

Gambar 3.14. Design dari Stok Produk Berdasarkan Rasio Penjualan dengan Keterangan

Pada gambar 3.13 dan gambar 3.14, dilihat bahwa data rasio penjualan produk akan ditampilkan dalam bentuk *widget* grafik batang. Data dari grafik diperoleh melalui *query* GQL *summary* yang berbentuk *clustering* produk berdasarkan rasio penjualan.

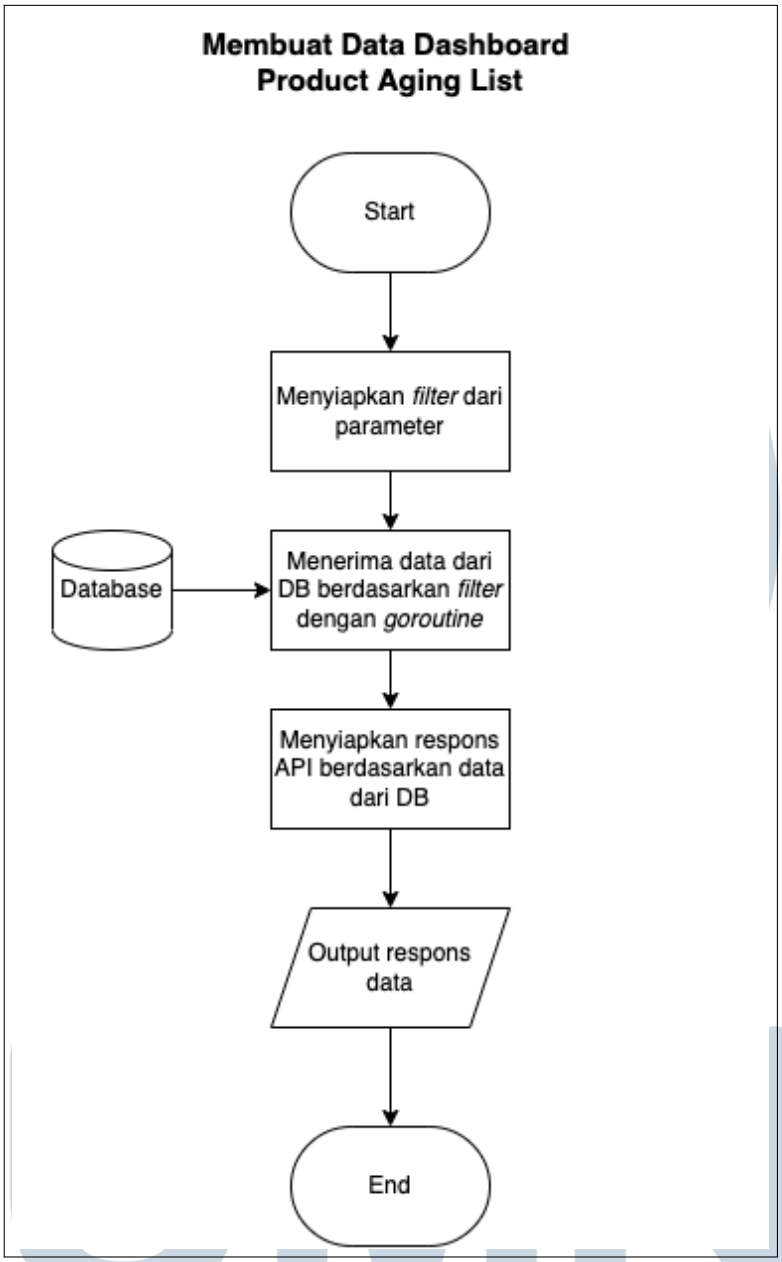
UNIVERSITAS MULTIMEDIA NUSANTARA



Gambar 3.15. Flowchart Hasil Perhitungan pada Dashboard Wawasan TokoCabang

Berdasarkan gambar 3.15, dapat dilihat bahwa untuk menampilkan data rasio penjualan pada *Dashboard* Wawasan TokoCabang seluruh data tersebut perlu untuk di-*query* melalui GQL. Ketika *query* dilakukan, maka GQL akan *hit endpoint* API sesuai dengan GQL yang digunakan. Ketika API berhasil untuk di-*hit*, maka data *dashboard* akan dibuat dan dikirim dalam bentuk respons. Respons tersebut lalu akan diolah sesuai dengan *query* yang digunakan. Dalam *dashboard* Wawasan TokoCabang, tim *frontend* akan melakukan *query* untuk mendapatkan data *Product Aging Dashboard, Shop Aging Dashboard,* dan *Product Aging Summary Dashboard*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.16. Flowchart Membuat Data Dashboard Product Aging List

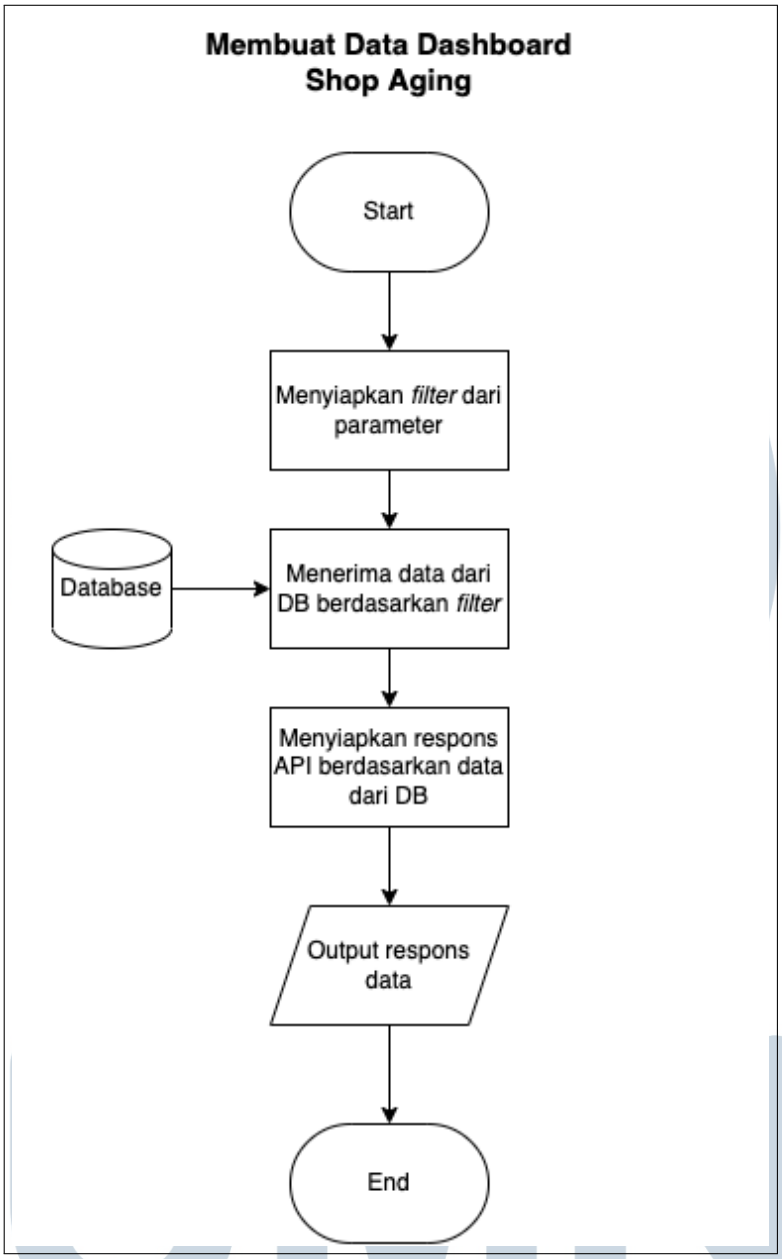
UNIVERSITAS MULTIMEDIA NUSANTARA

Pada gambar 3.16, dapat dilihat bahwa ketika *endpoint Product Aging List* telah berhasil di-*hit* maka akan dibuat suatu *filter* berdasarkan parameter yang ada. *Filter* yang digunakan dalam API ini adalah dalam bentuk *keyword* produk, *page*, *type*, dan *sort*. *Filter keyword* digunakan untuk hanya menampilkan produk yang memiliki nama sesuai dengan *keyword*. Sedangkan *filter page* digunakan untuk

pagination. *Filter type* digunakan untuk menentukan tipe data yang ditampilkan, tipe data yang ada adalah tipe Tagihan Bulan Ini dan Tagihan Berjalan. *Sort* digunakan sebagai index yang menentukan kolom data yang digunakan untuk sortir serta sortir *ascending* atau *descending*.

Setelah *filter* berhasil dibuat, maka data rasio penjualan produk akan diterima dari database berdasarkan hasil *filter* tersebut. Data akan diterima dari database dengan memanfaatkan *goroutine*. *Goroutine* adalah suatu *thread* yang bersifat *lightweight* yang pekerjaannya dikelola oleh runtime Go [4]. *Goroutine* akan membantu pengolahan data yang berukuran besar secara *concurrent* sehingga penjual tidak perlu menunggu respons untuk waktu yang lama walaupun data produk berukuran cukup besar. Data yang diperoleh tersebut lalu akan diubah sesuai dengan format respons dan dikirimkan sebagai output.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

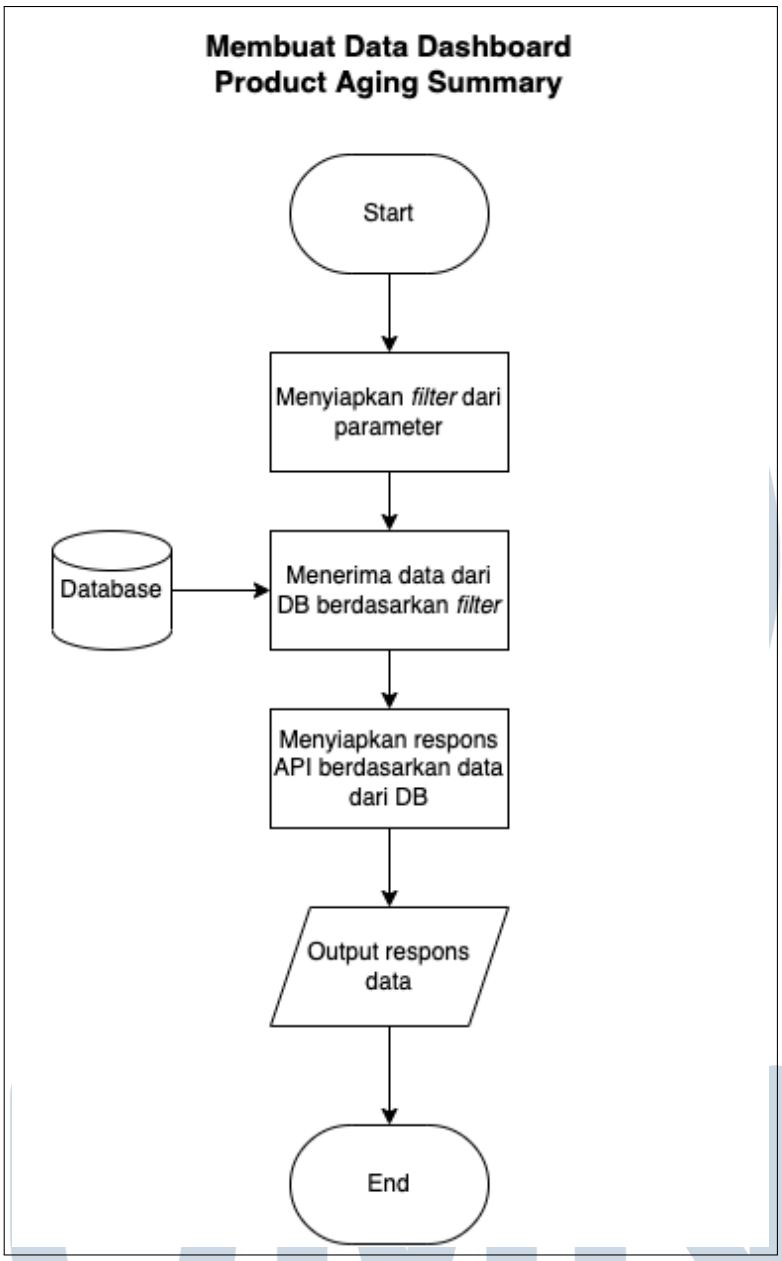


Gambar 3.17. Membuat Data Dashboard Shop Aging

Sama seperti membuat data *dashboard* di untuk rasio penjualan produk, data *dashboard* untuk rasio penjualan toko diawali dengan menyiapkan *filter* berdasarkan parameter sesuai dengan gambar 3.17. *Filter* yang digunakan dalam membuat respons API ini adalah berdasarkan *type* saja. Dimana *filter* tersebut digunakan untuk menentukan tipe data yang ditampilkan. Tipe yang ada dari respons API ini adalah Tagihan Bulan Ini dan Tagihan Berjalan.

Ketika *filter* telah berhasil dibuat, maka data akan data akan diterima dari database berdasarkan *filter*. Berbeda dengan rasio penjualan produk, data yang diterima hanya akan berjumlah satu baris saja dan tidak akan memanfaatkan *goroutine*. Data yang diterima dari database tersebut lalu akan diolah sesuai format respons dan dikirimkan sebagai output.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.18. Membuat Data Dashboard Product Aging Summary

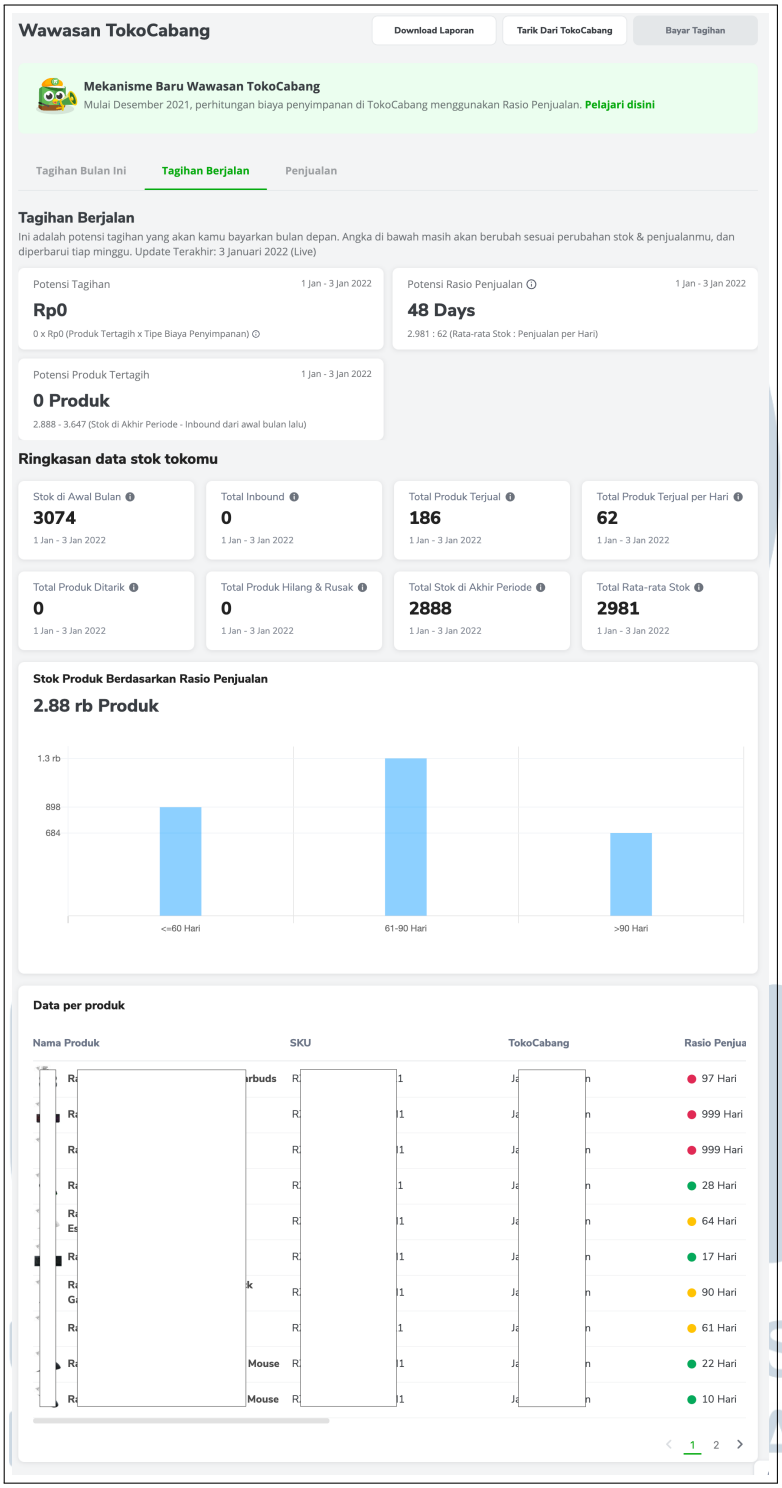
Pada gambar 3.18, dapat diketahui bahwa alur proses yang digunakan dalam pengolahan data *Dashboard Product Aging Summary* tidak jauh seperti pengolahan data rasio penjualan toko sebelumnya. *Filter* yang digunakan juga berupa *type* dengan jenis tipe yang sama, yaitu Tagihan Bulan Ini dan Tagihan Berjalan. Data yang diperoleh juga akan berdasarkan *filter* tersebut dan tidak memanfaatkan *goroutine* karena hanya menerima satu baris data.

Data yang diterima juga akan diolah sesuai format yang dibutuhkan sebagai output.

3.3.2 Bukti Penyelesaian Pekerjaan

Fitur rasio penjualan pada *backend* TokoCabang dapat dinyatakan selesai ketika seluruh pekerjaan yang ada telah berjalan dengan baik. Dimana fitur rasio penjualan akan menghasilkan data melalui hasil *query* GQL yang ditampilkan melalui Dashboard Wawasan TokoCabang. Sehingga seluruh pekerjaan dapat dinyatakan selesai, ketika data rasio penjualan dapat dilihat melalui halaman Dashboard Wawasan TokoCabang.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.19. *Screenshot* dari Halaman Wawasan TokoCabang

Pada gambar 3.19, dapat dilihat berdasarkan hasil *screenshot* bahwa data rasio penjualan berhasil diperoleh dan sudah siap ditampilkan dalam halaman Dashboard Wawasan TokoCabang. Sehingga, dapat disimpulkan bahwa seluruh pekerjaan dalam membangun fitur rasio penjualan sudah berhasil diselesaikan.

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala Selama Proses Kerja Magang

Berikut merupakan kendala yang ditemui oleh penulis selama proses kerja magang:

- Kesulitan untuk berkomunikasi dan berkoordinasi dengan anggota tim serta anggota divisi lain selama menjalani *work from home*
- Mempelajari istilah-istilah baru yang digunakan di dalam tim
- Kesulitan dalam melakukan *testing* karena tidak memiliki banyak akses di *staging* maupun di *production*

3.4.2 Solusi yang Ditemukan

Solusi yang digunakan untuk mengatasi kendala-kendala yang dialami selama proses kerja magang adalah sebagai berikut:

- Menggunakan *google meet* untuk melakukan *meeting* atau diskusi serta selalu aktif untuk *over-communicate* untuk mengurangi terjadi miskomunikasi
- Menanyakan istilah-istilah yang tidak diketahui selama proses kerja magang dengan anggota tim lainnya dan berusaha untuk mencari dokumentasi-dokumentasi yang berkaitan
- Mempelajari cara kerja *middleware* yang digunakan serta meminta bimbingan kepada anggota tim lainnya ketika mengalami kendala dalam *testing*