

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama proses *intern* di PT. Paramadaksa Teknologi Nusantara (NexSOFT) berlangsung, penulis diposisikan pada departemen *Development* yang berperan dalam mengembangkan sebuah aplikasi. Tugas yang diberikan kepada penulis adalah melakukan pengembangan ulang terhadap beberapa aplikasi yang sebelumnya sudah pernah dibuat oleh perusahaan. Lama waktu pengerjaan yang diberikan terhadap masing-masing proyek bervariasi. Ada yang berkisar selama 34 hari dan ada juga yang berkisar selama 35 hari. Proses pengembangan aplikasi dilaksanakan secara mandiri, baik pada sisi *frontend*, *backend*, maupun *database* yang diperlukan oleh aplikasi tersebut.

Proses koordinasi antara *mentor* dan anggota magang dilakukan melalui aplikasi *WhatsApp*. Serta setiap seminggu sekali pada hari yang berbeda-beda, setiap anggota magang akan diminta untuk melakukan *show progress* melalui *Google Meet*. Tujuannya adalah untuk membahas mengenai pekerjaan yang telah mereka lakukan selama satu minggu tersebut serta kendala yang sedang mereka hadapi dalam proses pengerjaan. Namun apabila mereka ingin melakukan *show progress* di luar dari jadwal yang telah disusun maka mereka dapat menghubungi *mentor* satu hari sebelum *show progress* akan dilaksanakan.

3.2 Tugas yang Dilakukan

Minggu pertama proses *intern* penulis diawali dengan pengenalan antara masing-masing anggota magang dengan *mentor*, mengenali sejarah dan nilai-nilai dasar PT. Paramadaksa Teknologi Nusantara (NexSOFT), serta mempelajari alur proses bisnis dan cara pemakaian produk yang dimiliki oleh perusahaan. Pada minggu kedua, anggota magang mulai mempelajari materi sesuai dengan departemen mereka masing-masing. Pada departemen *development*, penulis mempelajari materi dasar mengenai pemrograman dengan berbasis Java. Lalu penulis diminta untuk mengerjakan sebuah *test* yang mencakup keseluruhan materi yang telah diajarkan sebagai bentuk penilaian dan untuk menutupi sesi pada minggu tersebut.

Pada minggu ketiga, penulis melanjutkan pembelajaran materi ke jenjang

yang lebih kompleks. Pada setiap materi tersebut disertakan juga sebuah *test case* yang harus dikerjakan oleh masing-masing anggota magang. Lalu sama seperti minggu sebelumnya, setiap anggota magang juga harus mengerjakan sebuah *test* yang mencakup keseluruhan materi yang telah diajarkan pada minggu tersebut.

Pada minggu ke-empat, penulis diberikan tugas untuk membuat sebuah *game* berjudul *Roll the Dice* dengan menggunakan bahasa pemrograman Java. *Game* yang dibuat harus dapat dijalankan dalam dua jenis media, yaitu pada *terminal* komputer dan juga dalam bentuk aplikasi. Di minggu tersebut, penulis telah menyelesaikan pembuatan *game* pada media *terminal* yang memuat fitur bermain dan menampilkan skor permainan, serta membuat beberapa tampilan *game* untuk media *Android*. Pada minggu ke-lima, penulis berhasil dalam menyelesaikan pembuatan *game* di media *Android* dan melakukan presentasi terhadap *game* yang telah dibuat kepada pembimbingnya.

Dengan tema yang serupa untuk tugas selanjutnya, penulis diminta untuk membuat sebuah *game* yang berjudul *Rock Paper Scissor* dengan menggunakan bahasa pemrograman Java. *Game* ini juga harus dapat dimainkan dalam dua jenis media, yaitu *terminal* komputer dan dalam bentuk sebuah aplikasi. Pada minggu ke-enam, penulis telah menyelesaikan pembuatan *game* untuk media *terminal* yang memiliki fitur bermain secara *single* dan *multi player* serta menampilkan skor tertinggi dari keseluruhan permainan. Lalu pada minggu ke-tujuh, penulis telah membuat beberapa halaman dan fungsi sudah dapat dimanfaatkan pada media *Android*. Di minggu ke-delapan, penulis berhasil menyelesaikan pembuatan *game* pada media *Android* dengan fitur yang sama seperti pada media *terminal* dan melakukan presentasi kepada pembimbingnya.

Pada minggu ke-sembilan, penulis diberikan sebuah projek baru yaitu melakukan pengembangan ulang terhadap aplikasi Gromart yang merupakan salah satu produk karya PT. Paramadaksa Teknologi Nusantara (NexSOFT). Di minggu tersebut, penulis mempelajari lebih dalam mengenai kebutuhan dan cara kerja aplikasi yang akan dibuat. Lalu dilanjutkan dengan membuat *mockup* aplikasi, baik dari sisi tampilan, *backend*, maupun *database* yang akan digunakan. Pada minggu ke-sepuluh, penulis melakukan revisi terhadap *mockup* yang telah dibuat dan mulai mengerjakan beberapa halaman dari aplikasi Gromart, seperti *login* dan *register*. Pada minggu ke-sebelas, penulis melanjutkan pemrograman dengan memperbaiki beberapa hal pada halaman *login* dan *register*, membuat halaman daftar penjual, daftar pembeli, dan daftar produk untuk pengguna dengan peran administrasi, membuat fitur *logout*, membuat *header* yang ditampilkan pada setiap halaman dalam ap-

likasi, serta membuat halaman daftar produk dan fitur tambah produk untuk pengguna dengan peran penjual.

Pada minggu ke-dua belas, penulis membuat fitur untuk mengubah data penjual dan pembeli untuk *user* dengan peran administrasi, serta melakukan perbaikan pada beberapa halaman. Pada minggu ke-tiga belas, penulis membuat fitur *edit* dan *delete* produk serta halaman *report* untuk pengguna dengan peran penjual, membuat fitur untuk melihat detail dari produk untuk pengguna dengan peran administrasi, serta membuat halaman daftar produk, halaman keranjang, fitur *checkout*, dan fitur *add to cart* untuk pengguna dengan peran pembeli. Pada minggu ke-empat belas, penulis berhasil membuat fitur *edit* kuantitas dan *delete* produk di halaman keranjang, fitur pembayaran, serta halaman *order list* milik pengguna dengan peran pembeli. Lalu pada minggu ke-lima belas, penulis menyelesaikan aplikasi Gromart dengan membuat halaman detail dari pesanan untuk pengguna dengan peran pembeli dan memperbaiki beberapa kekurangan serta kesalahan pada aplikasi.

Penulis memulai pengerjaan proyek baru yang berjudul ND6 pada minggu ke-enam belas yang diawali dengan perancangan desain aplikasi. Minggu ke-tujuh belas dilalui oleh penulis dengan memperbaiki pembuatan *mockup* dan melaksanakan pemrograman aplikasi. Pada minggu ke-delapan belas, penulis melanjutkan pemrograman aplikasi. Pada minggu ke-sembilan belas, penulis mulai menyelesaikan beberapa fitur yang ada pada aplikasi. Di minggu ke-dua puluh, penulis melakukan perbaikan pada beberapa fitur dan UI dari aplikasi. Pada minggu ke-dua puluh satu, penulis melanjutkan perbaikan pada beberapa bagian aplikasi dan pada minggu ke-dua puluh dua penulis selesai dalam memperbaiki tampilan aplikasi.

U M N
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 3.1. Tabel kegiatan magang selama 800 jam kerja

Minggu ke-	Kegiatan yang dilakukan
1	Mempelajari nilai dasar dan proses bisnis perusahaan
2	Mempelajari bahasa pemrograman berbasis Java
3	Melanjutkan pembelajaran bahasa pemrograman berbasis Java
4	Membuat permainan <i>Roll the Dice</i> pada media <i>terminal</i>
5	Membuat permainan <i>Roll the Dice</i> pada media Android
6	Membuat permainan <i>Rock Paper Scissor</i> pada media <i>terminal</i>
7	Membuat permainan <i>Rock Paper Scissor</i> pada media Android
8	Menyelesaikan pembuatan permainan <i>Rock Paper Scissor</i> pada media Android
9	Membuat <i>mockup</i> dari aplikasi Gromart
10	Memperbaiki <i>mockup</i> dan membuat halaman <i>login</i> serta <i>register</i> dari aplikasi
11	Membuat berbagai halaman dan fitur untuk administrasi dan penjual
12	Memperbaiki beberapa halaman dan membuat fitur <i>edit</i> untuk administrasi
13	Membuat berbagai macam halaman dan fitur untuk administrasi dan pembeli
14	Membuat berbagai halaman dan fitur untuk pembeli
15	Menyelesaikan pembuatan aplikasi Gromart
16	Membuat <i>mockup</i> dari aplikasi ND6
17	Memperbaiki <i>mockup</i> dan membuat halaman <i>homepage</i> dan <i>master product</i> dari aplikasi
18	Membuat halaman <i>monitoring stock</i> dan <i>stock detail</i> dari aplikasi
19	Menyelesaikan fitur pada beberapa halaman yang sebelumnya belum dapat dikerjakan
20	Melakukan perbaikan pada fitur dan UI dari aplikasi
21	Memperbaiki beberapa bagian pada aplikasi yang masih memiliki <i>error</i>
22	Memperbaiki tampilan dari keseluruhan aplikasi

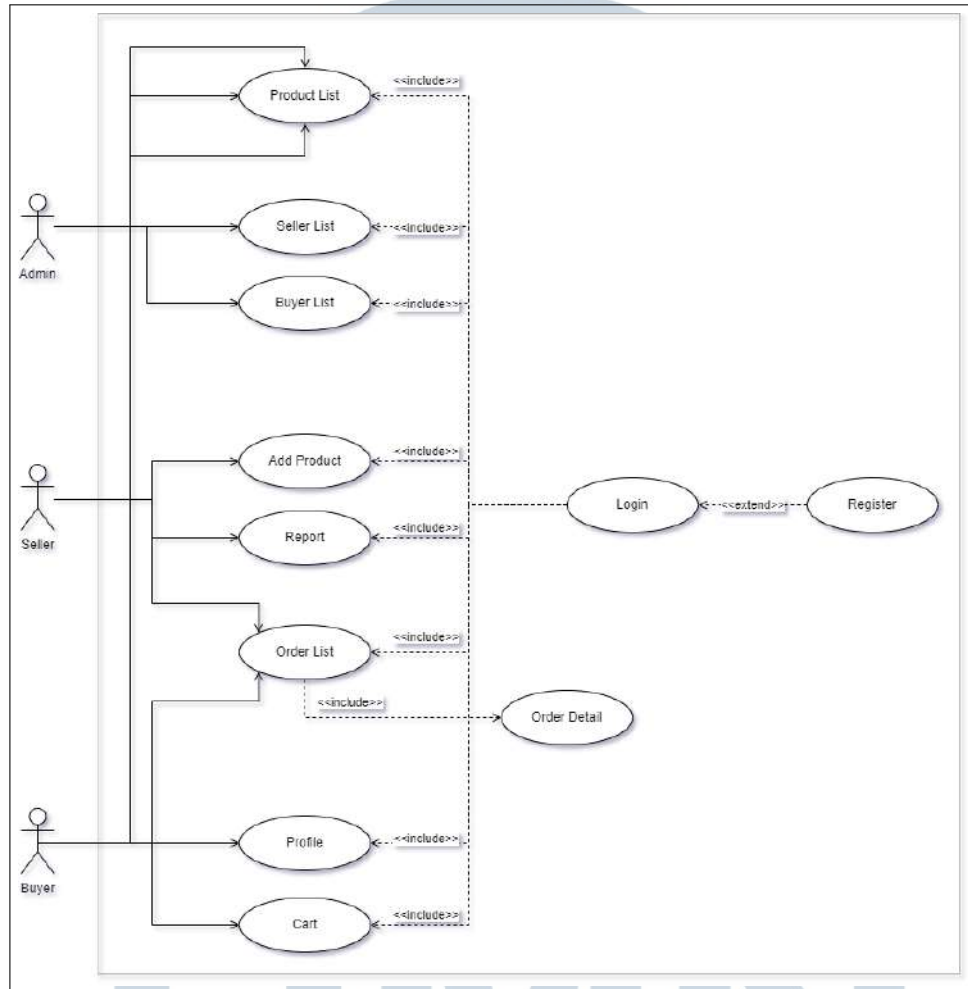
3.3 Gromart

Gromart merupakan sebuah aplikasi pemesanan barang yang diperuntukkan khusus untuk perusahaan dan para pengusaha yang membuka usaha jual beli barang. Aplikasi ini membutuhkan 3 jenis pengguna agar dapat dimanfaatkan, yaitu perusahaan yang berperan dalam menjual produknya sebagai admin, agen dari suatu perusahaan yang berperan dalam menyelesaikan transaksi yang ada sebagai *seller*, serta pengguna yang ingin melakukan pemesanan barang untuk kebutuhannya sebagai *buyer*. Seorang *buyer* yang ingin melakukan pemesanan dapat melihat daftar-daftar produk yang dimiliki oleh perusahaan melalui aplikasi ini. Apabila pesanan telah selesai dibuat, maka *buyer* juga dapat langsung melakukan pembayaran terhadap masing-masing produk yang mereka pesan. Pesanan dan pembayaran tersebut akan langsung dialihkan kepada setiap *seller* yang bertanggung jawab atas produk-produk yang dibeli oleh *buyer*. Pesanan yang tidak memiliki kendala dalam prosesnya akan diterima oleh *seller* dan diproses secara langsung.

Dalam aplikasi ini, seorang admin berperan dalam melakukan pengawasan terhadap keseluruhan data yang ada, seperti daftar penjual, pembeli, maupun produk. Selain melakukan pengawasan, admin juga dapat mengubah data yang dimiliki oleh *seller* maupun *buyer* sesuai dengan kebutuhan mereka. Sedangkan untuk *seller* yang merupakan agen dari perusahaan, mereka berperan dalam mengawasi setiap produk yang telah mereka buat, melakukan penambahan produk apabila terdapat produk baru, mengecek kembali dan menerima pesanan dari seorang *buyer* apabila dapat diterima, serta melaporkan hasil penjualan ketika diperlukan. *Buyer* yang merupakan konsumen dari aplikasi ini berperan dalam melakukan pemesanan barang dari suatu perusahaan dan melakukan pembayaran terhadap setiap produk yang mereka beli.

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 Use Case



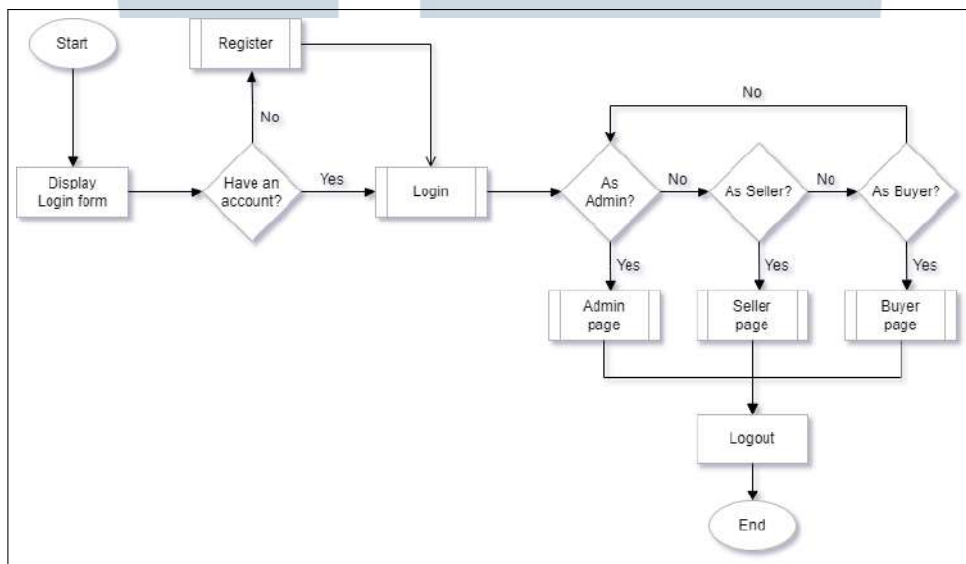
Gambar 3.1. Use case aplikasi Gromart

Berdasarkan pada gambar 3.1 yang merupakan *use case* dari aplikasi Gromart, terdapat tiga jenis pengguna yang berbeda di dalam aplikasi ini antara lain admin, *seller* atau penjual, dan *buyer* atau pembeli. Seorang admin dapat mengakses halaman *product list*, *seller list*, dan *buyer list* karena mereka berperan dalam melakukan pengawasan dan perubahan data ketika diperlukan. Seorang *seller* yang berperan dalam menambahkan produk baru, menerima pesanan, dan membuat laporan dapat mengakses halaman *product list*, *add product*, *order list*, serta *report*. Sedangkan seorang *buyer* dapat mengakses halaman *product list*, *cart*, *order list*, *order detail*, serta *profile*. Halaman-halaman tersebut dapat digunakan oleh *buyer* untuk memesan barang, mengecek kembali pesanan yang telah dibuat, melakukan

pembayaran, ataupun mengubah data profil dari akun yang digunakan. Namun sebelum seorang pengguna dapat mengakses aplikasi secara lebih lanjut, mereka harus melakukan *login* melalui halaman yang telah disediakan. Apabila mereka belum memiliki akun, maka mereka harus melakukan registrasi terlebih dahulu melalui halaman *register*.

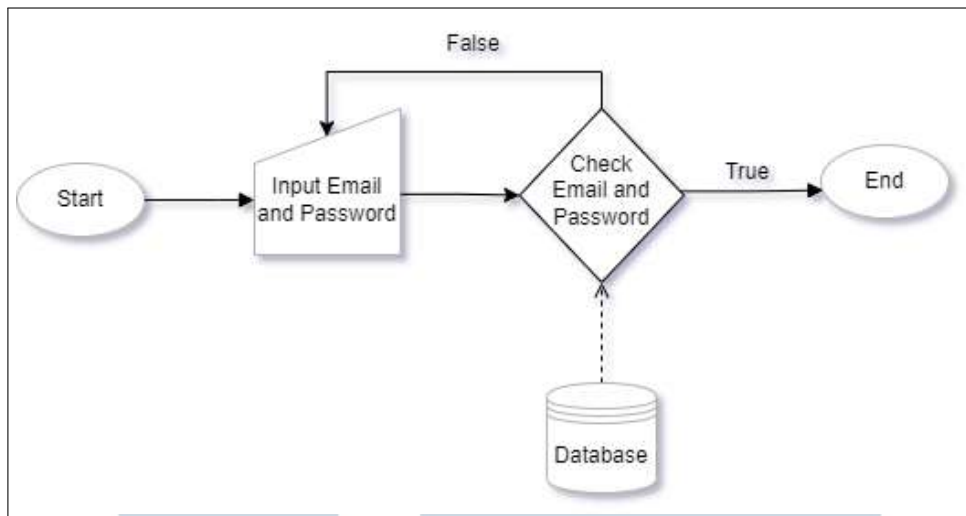
3.3.2 Flowchart

Berikut merupakan diagram *flowchart* dari aplikasi Gromart, yang terdiri atas program utama dan fitur-fitur yang dapat dimanfaatkan oleh setiap pengguna.



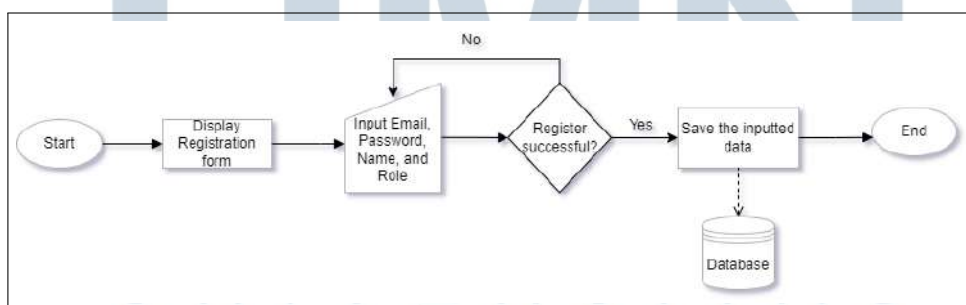
Gambar 3.2. Flowchart utama aplikasi Gromart

Gambar 3.2 merupakan alur dari aplikasi Gromart secara keseluruhan. Pada saat pertama kali melakukan akses ke aplikasi, tampilan *login* akan dimunculkan pada layar pengguna. Apabila pengguna belum memiliki akun pada aplikasi tersebut, maka mereka harus melakukan registrasi dengan menuju ke halaman *register*. Selesai melakukan registrasi, mereka akan langsung diarahkan kembali menuju halaman *login*. Setelah berhasil melakukan *login* dengan menggunakan akun yang telah dibuat, pengguna dapat memasuki halaman-halaman lainnya sesuai dengan *role* atau peran yang mereka pilih pada saat melakukan registrasi. *Role* dibagi menjadi 3 jenis, yaitu admin, *seller*, dan *buyer*. Aplikasi akan dihentikan ketika pengguna memilih fitur *logout* yang terdapat pada setiap halaman dalam aplikasi.



Gambar 3.3. Flowchart Login

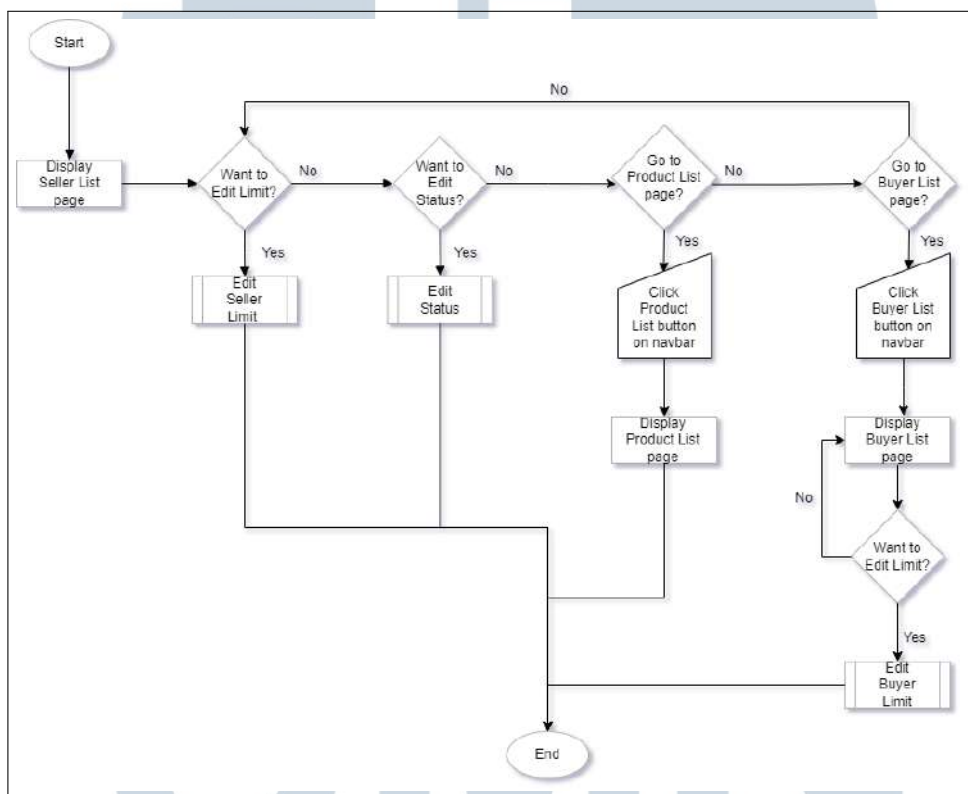
Gambar 3.3 merupakan alur dari proses *login* aplikasi Gromart. Saat memasuki halaman *login*, pengguna akan diminta untuk memasukkan *email* dan *password* dari akun yang sudah pernah dibuat. Lalu program akan melakukan pengecekan dengan memanfaatkan *database*. Hal ini dilakukan untuk mengetahui apakah *email* dan *password* yang dimasukkan sudah terdaftar serta sesuai atau belum. Apabila *email* ataupun *password* yang dimasukkan tidak sesuai atau belum terdaftar, maka pesan *error* yang menandakan bahwa proses *login* gagal dilakukan akan muncul pada layar pengguna.



Gambar 3.4. Flowchart Registrasi

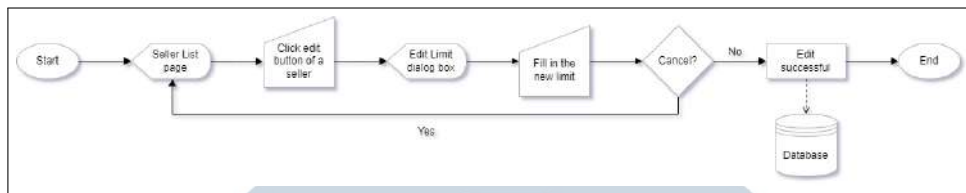
Gambar 3.4 merupakan alur dari proses registrasi aplikasi Gromart. Saat memasuki halaman *register*, pengguna akan diminta untuk memasukkan berbagai macam data yang diperlukan untuk melakukan registrasi. Data-data tersebut terdiri atas *email*, *password*, nama, serta *role* pengguna. *Email* dan *password* merupakan data yang diperlukan untuk melakukan

login. Sedangkan *role* yang hanya terdiri atas pilihan *seller* dan *buyer*, merupakan data yang digunakan untuk menentukan halaman-halaman apa saja yang dapat dikunjungi oleh pengguna tersebut. Khusus untuk pengguna dengan *role* admin, mereka hanya dapat melakukan registrasi melalui *database* secara langsung. Apabila pengguna telah memasukkan data-data yang diperlukan dengan benar, maka akun akan terbuat dan disimpan di *database*. Bila proses *register* gagal, maka pengguna akan diminta untuk memasukkan kembali data-data dengan benar.



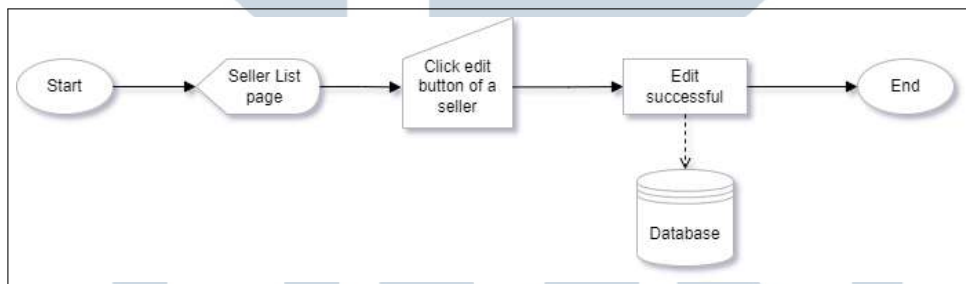
Gambar 3.5. Flowchart Admin

Gambar 3.5 merupakan alur dari proses-proses yang dapat dilakukan oleh seorang admin. Halaman pertama yang akan dilihat oleh admin adalah halaman *seller list*. Pada halaman tersebut, admin dapat melakukan perubahan *limit* atau *status* dari suatu *seller*. Selain halaman *seller list*, admin juga dapat mengakses halaman *product list* dan *buyer list* melalui tombol yang terdapat pada *navbar* halaman tersebut. Admin hanya dapat melihat produk-produk yang dijual oleh perusahaan melalui halaman *product list* sedangkan pada halaman *buyer list* admin dapat melakukan perubahan *limit* pada suatu *buyer*.



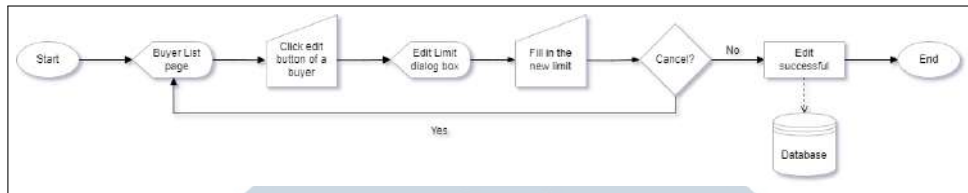
Gambar 3.6. Flowchart edit limit seller oleh Admin

Gambar 3.6 merupakan alur dari proses *edit limit seller* yang dapat dilakukan oleh admin. Proses ini diawali dengan menekan tombol *edit limit* pada salah satu *seller* yang terdapat di halaman *seller list*. Aksi tersebut akan memunculkan sebuah *dialog box* yang berisikan nilai *limit* milik *seller* tersebut. Dengan memasukkan nilai baru dan memilih untuk menerima, maka proses pengubahan *limit seller* telah berhasil dan nilai baru tersebut akan disimpan ke dalam *database*. Namun bila admin memilih untuk menolak melakukan pengubahan, maka pengguna akan dialihkan kembali ke halaman *seller list*.



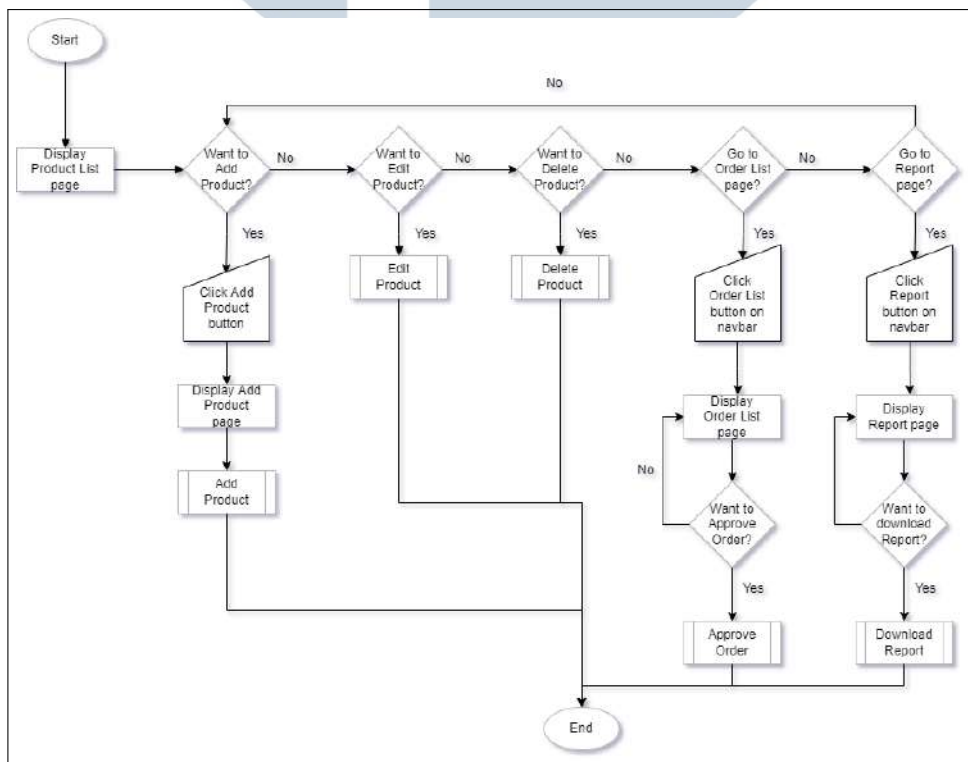
Gambar 3.7. Flowchart edit status seller oleh Admin

Gambar 3.7 merupakan alur dari proses *edit status seller* yang dapat dilakukan oleh admin. Proses ini dijalankan dengan menekan tombol *edit status* pada salah satu *seller* yang terdapat di halaman *seller list*. Status *seller* yang pada awalnya adalah *requested* akan berubah menjadi *active*, dan apabila admin menekan lagi tombol tersebut maka status *seller* yang pada awalnya adalah *active* akan berubah menjadi *inactive*. Perubahan data tersebut akan disimpan ke *database*. Proses ini tidak memiliki *decision symbol* atau penanggulangan terhadap kesalahan di dalamnya karena proses ini tidak memerlukan pengguna untuk memasukkan data secara mandiri.



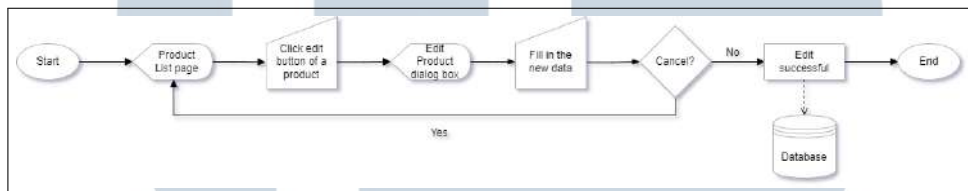
Gambar 3.8. Flowchart edit limit buyer oleh Admin

Gambar 3.8 merupakan alur dari proses *edit limit buyer* yang dapat dilakukan oleh admin. Proses ini diawali dengan menekan tombol *edit limit* pada salah satu *buyer* yang terdapat di halaman *buyer list*. Aksi tersebut akan memunculkan sebuah *dialog box* yang berisikan nilai *credit limit* dan *invoice limit* milik *buyer* tersebut. Dengan memasukkan nilai baru dan memilih untuk menerima, maka proses pengubahan *limit buyer* telah berhasil dan nilai baru tersebut akan disimpan ke dalam *database*. Namun bila admin memilih untuk menolak melakukan pengubahan, maka pengguna akan dialihkan kembali ke halaman *buyer list*.



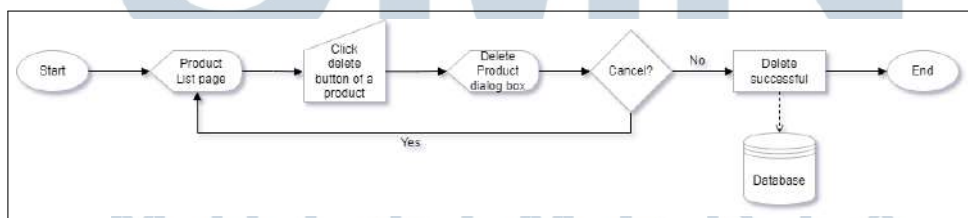
Gambar 3.9. Flowchart Seller

Gambar 3.9 merupakan alur dari proses-proses yang dapat dilakukan oleh seorang *seller*. Halaman pertama yang akan tampil pada layar pengguna adalah halaman *product list*. Melalui halaman tersebut, *seller* dapat melakukan penambahan produk, mengubah data dari suatu produk, serta menghapus produk. *Seller* juga dapat mengakses ke halaman-halaman lainnya melalui tombol yang terdapat pada *navbar* halaman tersebut, seperti halaman *order list* dan *report*. Pada halaman *order list*, *seller* dapat melakukan proses *approve order*. Sedangkan pada halaman *report*, *seller* dapat melakukan proses *download report*.



Gambar 3.10. Flowchart edit product oleh Seller

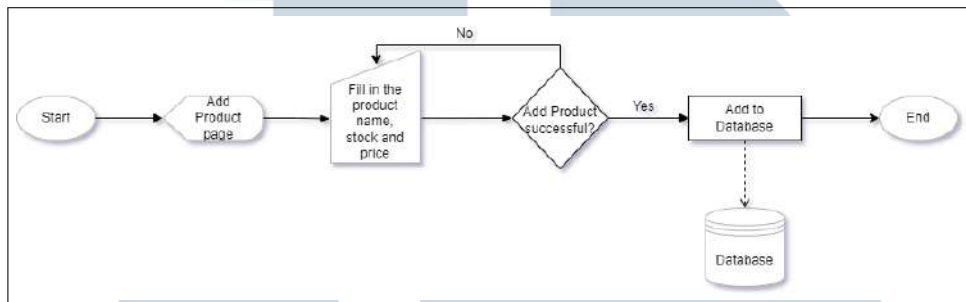
Gambar 3.10 merupakan alur dari proses *edit product* yang dapat dilakukan oleh *seller*. Proses ini diawali dengan menekan tombol *edit* pada salah satu produk yang terdapat di halaman *product list*. Aksi tersebut akan memunculkan sebuah *dialog box* yang berisikan id, nama, status, kuantitas, harga dan tanggal pembuatan produk tersebut. Dengan memasukkan nilai baru dan memilih untuk menerima, maka proses pengubahan nilai produk telah berhasil dan nilai baru tersebut akan disimpan ke dalam *database*. Namun bila *seller* memilih untuk menolak melakukan pengubahan, maka pengguna akan dialihkan kembali ke halaman *product list*.



Gambar 3.11. Flowchart delete product oleh Seller

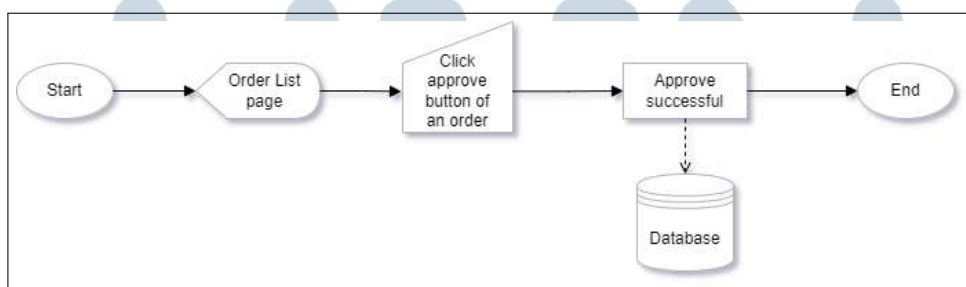
Gambar 3.11 merupakan alur dari proses *delete product* yang dapat dilakukan oleh *seller*. Proses ini diawali dengan menekan tombol *delete* pada salah satu produk yang terdapat di halaman *product list*. Aksi tersebut akan memunculkan sebuah *dialog box* yang menanyakan apakah pengguna

yakin ingin menghapus produk tersebut. Dengan menjawab iya, maka produk akan dihapus dari *database*. Namun bila *seller* memilih untuk menolak, maka pengguna akan dialihkan kembali ke halaman *product list* dan proses *delete* akan dibatalkan.



Gambar 3.12. Flowchart add product oleh Seller

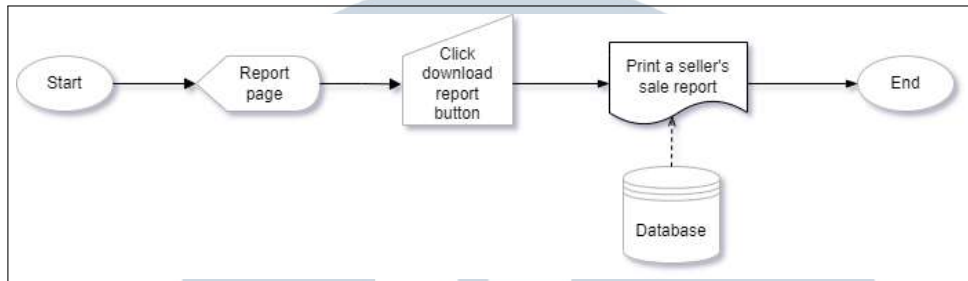
Gambar 3.12 merupakan alur dari proses *add product* yang dapat dilakukan oleh *seller*. Proses ini dilaksanakan dengan mengisi data pada halaman *add product*. Data-data tersebut terdiri atas nama produk, jumlah, serta harganya. Setelah memasukkan data dengan benar dan sesuai kebutuhan, maka produk tersebut akan disimpan ke dalam *database*. Bila proses *add product* gagal, maka *seller* akan diminta untuk memasukkan kembali data-data dengan benar.



Gambar 3.13. Flowchart approve order oleh Seller

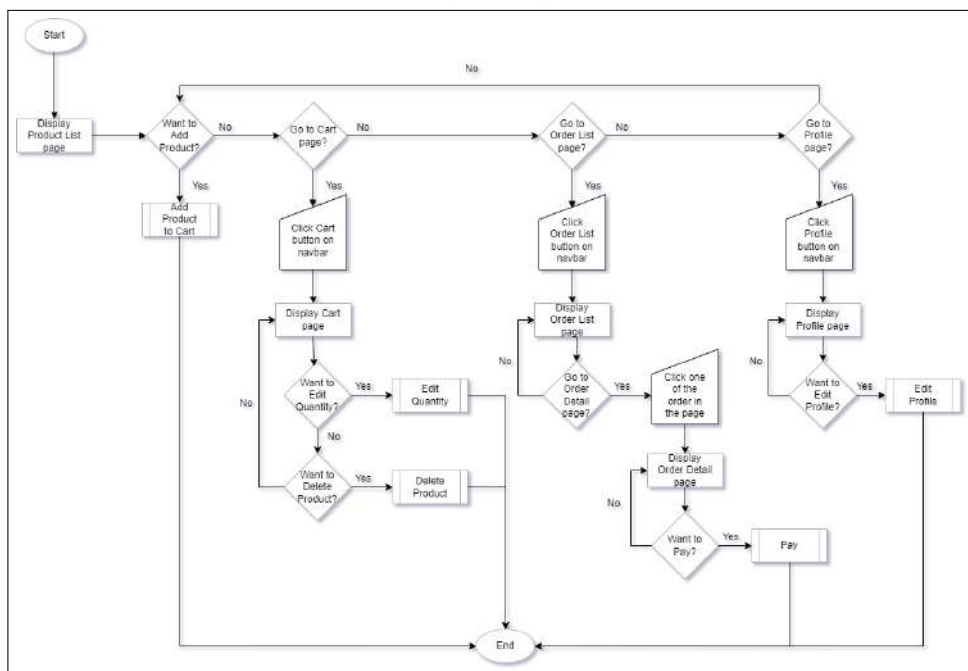
Gambar 3.13 merupakan alur dari proses *approve order* yang dapat dilakukan oleh *seller*. Proses ini dijalankan dengan menekan tombol *approve* pada salah satu *order* yang terdapat di halaman *order list*. Status *order* yang pada awalnya adalah *requested* akan berubah menjadi *accepted*. Lalu perubahan data tersebut akan disimpan ke *database*. Proses ini tidak memiliki *decision symbol* atau penanggulangan terhadap kesalahan di

dalamnya karena proses ini tidak memerlukan pengguna untuk memasukkan data secara mandiri.



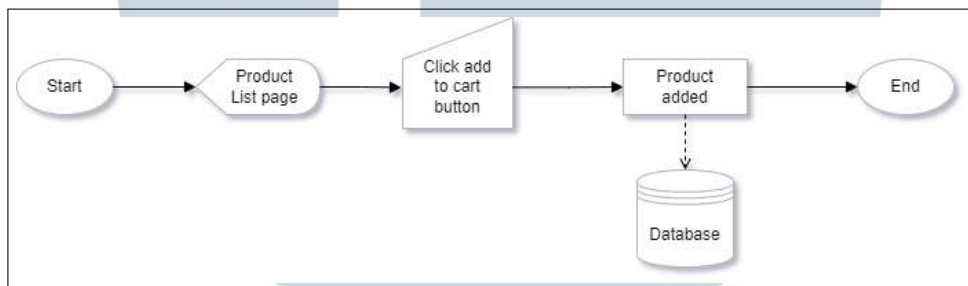
Gambar 3.14. Flowchart download report oleh Seller

Gambar 3.14 merupakan alur dari proses *download report* yang dapat dilakukan oleh *seller*. Proses ini dilaksanakan dengan menekan tombol *download* pada halaman *report*. *Report* yang berisikan data penjualan *seller* tersebut mendapatkan datanya melalui *database*. Proses ini tidak memiliki *decision symbol* atau penanggulangan terhadap kesalahan di dalamnya karena proses ini tidak memerlukan pengguna untuk memasukkan data secara mandiri.



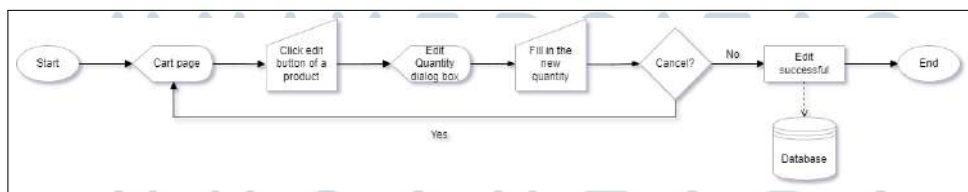
Gambar 3.15. Flowchart Buyer

Gambar 3.15 merupakan alur dari proses-proses yang dapat dilakukan oleh seorang *buyer*. Halaman pertama yang akan dilihat oleh pengguna adalah halaman *product list*. Melalui halaman ini, *buyer* dapat melakukan proses *add product to cart*. Selain itu, *buyer* dapat mengakses ke halaman-halaman lainnya melalui tombol yang terdapat pada *navbar* halaman tersebut, seperti halaman *cart*, *order list*, serta *profile*. Pada halaman *cart*, *buyer* bisa melakukan proses *edit quantity* dan *delete product*. Sedangkan pada halaman *order list*, *buyer* bisa melakukan akses ke halaman *order detail* dan melakukan proses *pay* pada halaman tersebut. Serta pada halaman *profile*, *buyer* bisa melakukan proses *edit profile*.



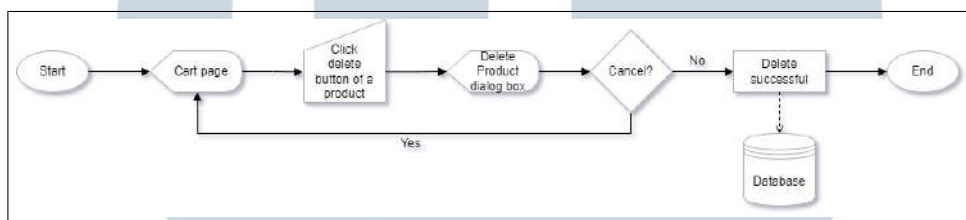
Gambar 3.16. Flowchart add product to cart oleh Buyer

Gambar 3.16 merupakan alur dari proses *add product to cart* yang dapat dilakukan oleh *buyer*. Proses ini dijalankan dengan menekan tombol *add to cart* pada salah satu produk yang terdapat di halaman *product list*. Setelah berhasil menekan tombol tersebut, produk akan disimpan ke dalam keranjang yang terdapat di *database* dan dapat digunakan ketika diperlukan. Proses ini tidak memiliki *decision symbol* atau penanggulangan terhadap kesalahan di dalamnya karena proses ini tidak memerlukan pengguna untuk memasukkan data secara mandiri.



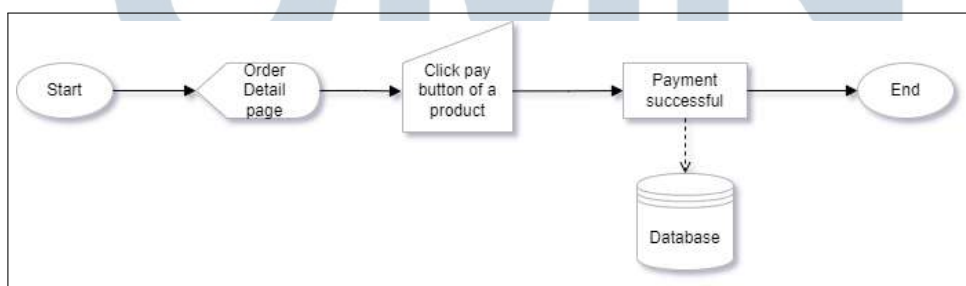
Gambar 3.17. Flowchart edit quantity product oleh Buyer

Gambar 3.17 merupakan alur dari proses *edit quantity product* yang dapat dilakukan oleh *buyer*. Proses ini diawali dengan menekan tombol *edit* pada salah satu produk yang terdapat di halaman *cart*. Aksi tersebut akan memunculkan sebuah *dialog box* yang berisikan nilai *quantity* dari produk tersebut. Dengan memasukkan nilai baru dan memilih untuk menerima, maka proses pengubahan *quantity product* telah berhasil dan nilai baru tersebut akan disimpan ke dalam *database*. Namun bila *buyer* memilih untuk menolak melakukan perubahan, maka pengguna akan dialihkan kembali ke halaman *cart*.



Gambar 3.18. Flowchart delete product oleh Buyer

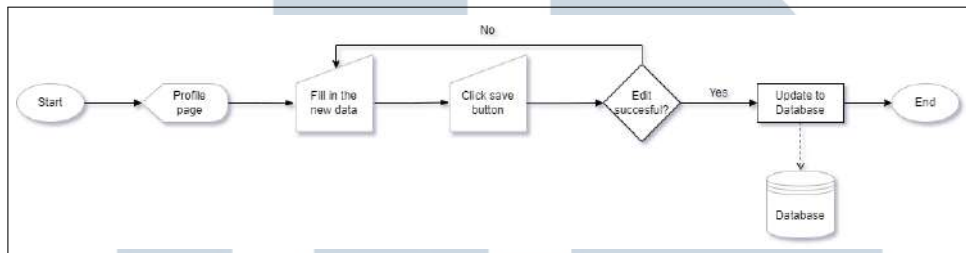
Gambar 3.18 merupakan alur dari proses *delete product* yang dapat dilakukan oleh *buyer*. Proses ini diawali dengan menekan tombol *delete* pada salah satu produk yang terdapat di halaman *cart*. Aksi tersebut akan memunculkan sebuah *dialog box* yang menanyakan apakah pengguna yakin ingin menghapus produk tersebut. Dengan menjawab iya, maka produk akan dihapus dari *database*. Namun bila *buyer* memilih untuk menolak, maka pengguna akan dialihkan kembali ke halaman *cart* dan proses *delete* akan dibatalkan.



Gambar 3.19. Flowchart pay oleh Buyer

Gambar 3.19 merupakan alur dari proses *pay* yang dapat dilakukan oleh *buyer*. Proses ini dijalankan dengan menekan tombol *pay* pada salah satu *order* yang terdapat di halaman *order detail*. Status *order* yang pada

awalnya adalah *unpaid* akan berubah menjadi *requested*. Lalu perubahan data tersebut akan disimpan ke *database*. Proses ini tidak memiliki *decision symbol* atau penanggulangan terhadap kesalahan di dalamnya karena proses ini tidak memerlukan pengguna untuk memasukkan data secara mandiri.

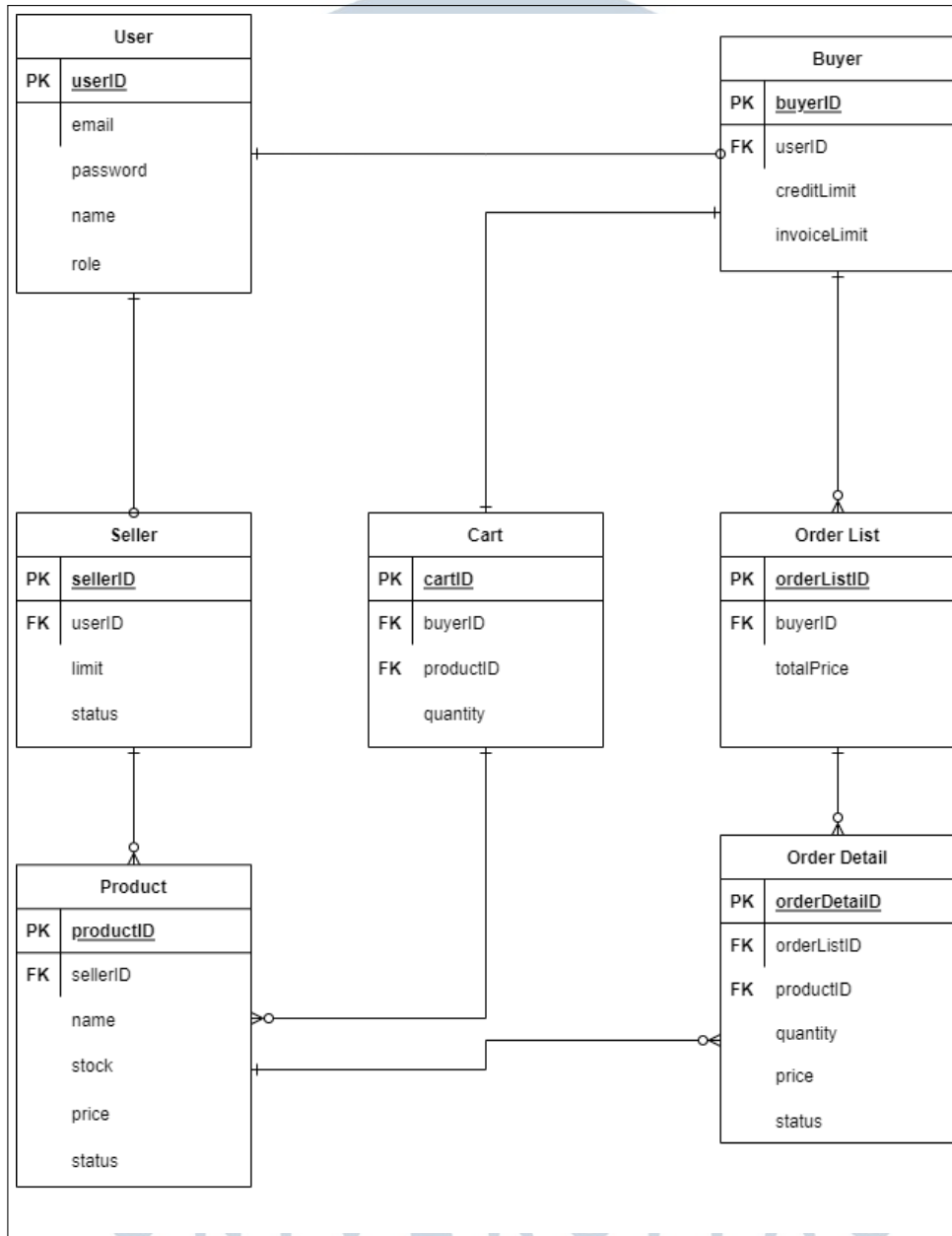


Gambar 3.20. Flowchart edit profile oleh Buyer

Gambar 3.20 merupakan alur dari proses *edit profile* yang dapat dilakukan oleh *buyer*. Proses ini dilaksanakan dengan memasukkan data-data baru pada halaman *profile*. Data-data tersebut terdiri atas *email*, *password* lama, *password* baru, dan konfirmasi *password* baru. Setelah yakin atas data-data yang telah dimasukkan, pengguna dapat menyelesaikan proses *edit* dengan menekan tombol *save* dan data-data tersebut akan secara otomatis tersimpan ke dalam *database*. Bila proses *edit profile* gagal, maka *buyer* akan diminta untuk memasukkan kembali data-data dengan benar.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.3.3 Database Schema dan Tabel



Gambar 3.21. Database schema aplikasi Gromart

Gambar 3.21 merupakan diagram *database schema* yang digunakan pada aplikasi Gromart. Tabel *user* merupakan tempat penyimpanan data setiap pengguna yang telah melakukan registrasi dan akan digunakan ketika ingin melakukan proses *login*. Tabel ini memiliki kaitan dengan tabel *seller* dan *buyer* karena masing-masing *seller* dan *buyer* memiliki data tambahan

yang hanya dimiliki oleh pengguna dengan *role* tersebut. Sehingga pengguna dengan *role* admin tidak memiliki tabel tersendiri karena *role* tersebut tidak memerlukan data tambahan.

Tabel *seller* memiliki kaitan dengan tabel *product* karena setiap *seller* memiliki tanggungjawab terhadap yang mereka kelola. Sedangkan seorang *buyer* mampu menyimpan barang pesannya ke dalam keranjang sehingga pengguna tersebut memerlukan tabel *cart* yang terhubung ke tabel *product*. Selain untuk menyimpan barang, *buyer* juga membutuhkan tabel yang dapat digunakan untuk menyimpan seluruh pesanan yang sudah pernah dilakukan dan juga detail-detail dari pesanan tersebut. Sehingga dibuatlah tabel *order list* yang memiliki hubungan dengan tabel *order detail*, di mana tabel *order detail* juga memiliki hubungan dengan tabel *product*. Tujuan dari penghubungan tabel *cart* dan *order detail* kepada tabel *product* adalah agar kedua tabel tersebut dapat menggunakan data-data yang terdapat pada tabel *product* dan untuk menghindari terjadinya proses duplikasi data.

Tabel 3.2. Tabel User

Name	Type	Null	Default	Extra	Links to
userId	int(11)	No	None	AUTO INCREMENT	
email	varchar(255)	No	None		
password	varchar(255)	No	None		
name	varchar(255)	No	None		
role	varchar(255)	No	None		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.2 merupakan tabel *user* yang berisikan data-data milik pengguna yang telah berhasil melakukan registrasi dan data pada tabel ini juga dapat digunakan sebagai verifikasi ketika pengguna hendak mengakses proses *login*. Data-data pada tabel ini terdiri atas *userId* dengan tipe *integer*, dan *email*, *password*, nama, *role* dengan tipe *varchar*. Data *password* yang tersimpan di dalam tabel ini sudah mengalami proses enkripsi dengan bantuan bahasa pemrograman Javascript. Sedangkan *createdAt* dan *updatedAt* dengan tipe *date* merupakan data yang terdapat pada setiap tabel karena tabel-tabel ini dibuat dengan menggunakan MySQL. Seluruh data pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data ter-

sebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *userId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

Tabel 3.3. Tabel Seller

Name	Type	Null	Default	Extra	Links to
sellerId	int(11)	No	None	AUTO INCREMENT	
userId	int(11)	Yes	NULL		User.userId
limit	int(11)	No	10		
status	varchar(255)	No	Requested		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.3 merupakan tabel milik pengguna dengan *role seller* yang telah berhasil melakukan registrasi. Tabel ini dibuat karena *seller* memiliki data-data tambahan yang tidak dimiliki oleh pengguna lainnya, seperti *limit* dengan tipe *integer* dan status dari *seller* tersebut dengan tipe *varchar*. Data-data lainnya yang terdapat pada tabel ini adalah *sellerId* dan *userId* dengan tipe *integer* di mana *userId* merupakan sebuah *foreign key* yang berasal dari tabel *user*. Dengan menggunakan *foreign key* tersebut, pengguna dapat mengakses seluruh data yang terdapat pada tabel *user*. Pada tabel ini, data *limit* dan status memiliki nilai awalnya masing-masing yaitu berupa angka 10 dan tulisan *requested*. Nilai tersebut akan secara langsung dimasukkan ketika seorang pengguna terdaftar di dalam tabel ini. Selain pada data *userId*, data lainnya pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data tersebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *sellerId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

Tabel 3.4. Tabel Buyer

Name	Type	Null	Default	Extra	Links to
buyerId	int(11)	No	None	AUTO INCREMENT	
userId	int(11)	Yes	NULL		User.userId
creditLimit	int(11)	No	100000		
invoiceLimit	int(11)	No	10		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.4 merupakan tabel milik pengguna dengan *role buyer* yang telah berhasil melakukan registrasi. Tabel ini dibuat karena *buyer* memiliki data-data tambahan yang tidak dimiliki oleh pengguna lainnya, seperti *creditLimit* dan *invoiceLimit* dengan tipe *integer*. Data-data lainnya yang terdapat pada tabel ini adalah *buyerId* dan *userId* dengan tipe *integer* di mana *userId* merupakan sebuah *foreign key* yang berasal dari tabel *user*. Dengan menggunakan *foreign key* tersebut, pengguna dapat mengakses seluruh data yang terdapat pada tabel *user*. Pada tabel ini, data *creditLimit* dan *invoiceLimit* memiliki nilai awalnya masing-masing yaitu berupa angka 100000 dan angka 10. Nilai tersebut akan secara langsung dimasukkan ketika seorang pengguna terdaftar di dalam tabel ini. Selain pada data *userId*, data lainnya pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data tersebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *buyerId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tabel 3.5. Tabel Product

Name	Type	Null	Default	Extra	Links to
productId	int(11)	No	None	AUTO INCREMENT	
sellerId	int(11)	Yes	NULL		Seller.sellerId
productName	varchar(255)	No	None		
stock	int(11)	No	None		
price	int(11)	No	None		
status	varchar(255)	No	Active		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.5 merupakan tabel yang berisikan data-data *product* yang dibuat oleh setiap *seller*. Data-data pada tabel ini terdiri atas *productId*, *sellerId*, *stock*, dan *price* dengan tipe *integer*, serta *productName* dan status dengan tipe *varchar*. Data *sellerId* merupakan sebuah *foreign key* yang berasal dari tabel *seller* sehingga pengguna dapat memanfaatkan data-data yang terdapat pada tabel *seller* melalui tabel ini. Pada tabel ini, hanya data status yang memiliki nilai awal yaitu berupa tulisan *active*. Nilai tersebut akan secara langsung dimasukkan ketika suatu *product* terdaftar di dalam tabel ini. Selain pada data *sellerId*, data lainnya pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data tersebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *productId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

Tabel 3.6. Tabel Cart

Name	Type	Null	Default	Extra	Links to
cartId	int(11)	No	None	AUTO INCREMENT	
buyerId	int(11)	Yes	NULL		Buyer.buyerId
productId	int(11)	Yes	NULL		Product.productId
quantity	int(11)	No	1		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.6 merupakan tabel yang berisikan produk-produk yang terdapat dalam *cart* atau keranjang suatu *buyer*. Data-data pada tabel ini ter-

diri atas *cartId*, *buyerId*, *productId*, dan *quantity* dengan tipe *integer*. Data *buyerId* dan *productId* merupakan sebuah *foreign key* yang berasal dari tabel *buyer* dan *product* sehingga pengguna dapat memanfaatkan data-data yang terdapat pada kedua tabel tersebut melalui tabel *cart*. Pada tabel ini, hanya data *quantity* yang memiliki nilai awal yaitu berupa angka 1. Nilai tersebut akan secara langsung dimasukkan ketika suatu produk terdaftar di dalam tabel ini. Selain pada data *buyerId* dan *productId*, data lainnya pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data tersebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *cartId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

Tabel 3.7. Tabel Order List

Name	Type	Null	Default	Extra	Links to
orderListId	int(11)	No	None	AUTO INCREMENT	
buyerId	int(11)	Yes	NULL		Buyer.buyerId
totalPrice	int(11)	No	None		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.7 merupakan tabel *order list* yang berisikan pesanan-pesanan yang sudah pernah dilakukan oleh suatu *buyer*. Data-data pada tabel ini terdiri atas *orderListId*, *buyerId*, dan *totalPrice* dengan tipe *integer*. Data *buyerId* merupakan sebuah *foreign key* yang berasal dari tabel *buyer* sehingga pengguna dapat memanfaatkan data-data yang terdapat pada tabel tersebut melalui tabel *order list*. Selain pada data *buyerId*, data lainnya pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data tersebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *orderListId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

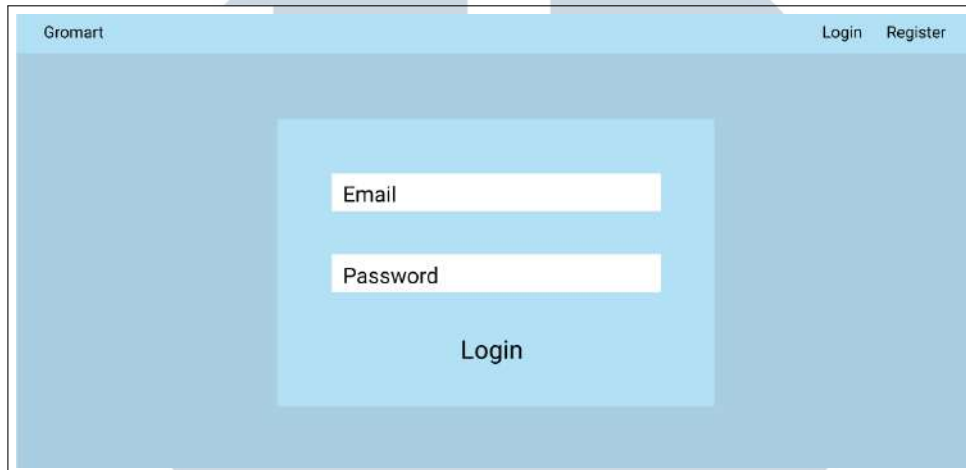
Tabel 3.8. Tabel Order Detail

Name	Type	Null	Default	Extra	Links to
orderDetailId	int(11)	No	None	AUTO INCREMENT	
orderListId	int(11)	Yes	NULL		OrderList.orderListId
productId	int(11)	Yes	NULL		Product.productId
quantity	int(11)	No	None		
price	int(11)	No	None		
status	varchar(255)	No	Unpaid		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.8 merupakan tabel detail dari setiap pesanan yang telah dilakukan oleh *buyer*. Sehingga dalam suatu pesanan atau *order list* akan memiliki banyak macam produk yang dapat dilihat melalui tabel ini. Data-data pada tabel ini terdiri atas *orderDetailId*, *orderListId*, *productId*, *quantity*, dan *price* dengan tipe *integer*, serta status dengan tipe *varchar*. *Price* pada tabel ini berbeda dengan *price* yang terdapat pada tabel lainnya karena *price* di sini merupakan *subtotal* dari suatu produk yang dipesan oleh *buyer*. Data *orderListId* dan *productId* merupakan sebuah *foreign key* yang berasal dari tabel *order list* dan *product* sehingga pengguna dapat memanfaatkan data-data yang terdapat pada kedua tabel tersebut melalui tabel *order detail*. Pada tabel ini, hanya data status yang memiliki nilai awal yaitu berupa tulisan *unpaid*. Nilai tersebut akan secara langsung dimasukkan ketika suatu *product* terdaftar di dalam tabel ini. Selain pada data *orderListId* dan *productId*, data lainnya pada tabel ini harus memiliki nilai atau tidak boleh bersifat *null* karena setiap data tersebut akan dimanfaatkan oleh suatu proses. Serta khusus untuk data *orderListId* yang memiliki nilai *auto increment* berarti nilai dari data ini akan terus bertambah satu secara otomatis tanpa harus dimasukkan oleh pengguna.

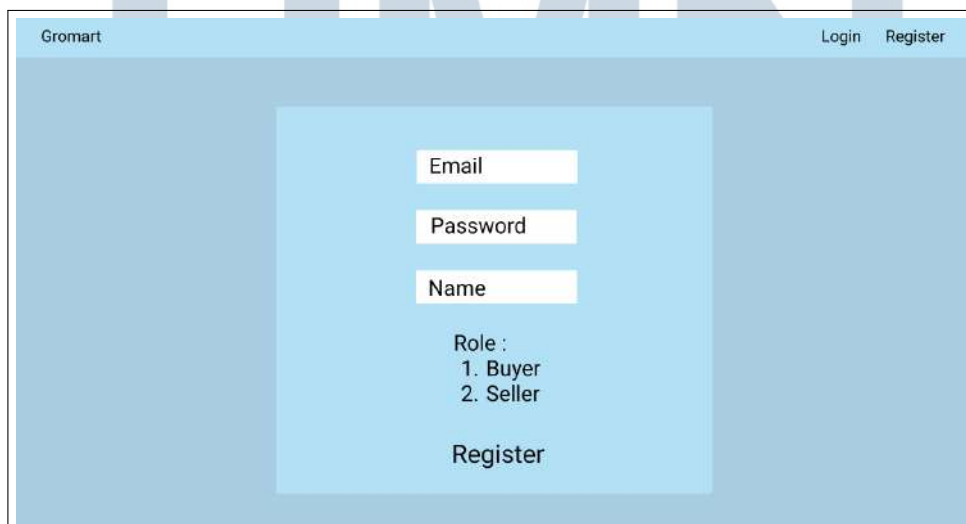
3.3.4 Rancangan Desain Aplikasi

Berikut adalah rancangan desain dari aplikasi Gromart yang disusun dengan menggunakan aplikasi Figma.



Gambar 3.22. Halaman Login

Gambar 3.22 merupakan rancangan desain halaman *login* yang harus dilakukan oleh setiap pengguna sebelum dapat memanfaatkan aplikasi ini secara lebih lanjut. Apabila pengguna ingin melakukan registrasi, maka mereka dapat mengunjungi halaman tersebut melalui tombol yang terdapat di *navbar* halaman tersebut.



Gambar 3.23. Halaman Register

Gambar 3.23 merupakan rancangan desain halaman registrasi yang harus dilalui oleh para pengguna dengan *role seller* dan *buyer* ketika mereka belum memiliki akun dalam aplikasi ini. Setelah selesai melakukan proses registrasi, pengguna akan langsung dialihkan kembali ke halaman *login*.

ID	Name	Register Date	Limit	Active Product	Status
1	Seller1	2021-11-12	10	0	Requested
2	Seller2	2021-11-12	10	0	Active
3	Seller3	2021-11-12	10	0	Inactive

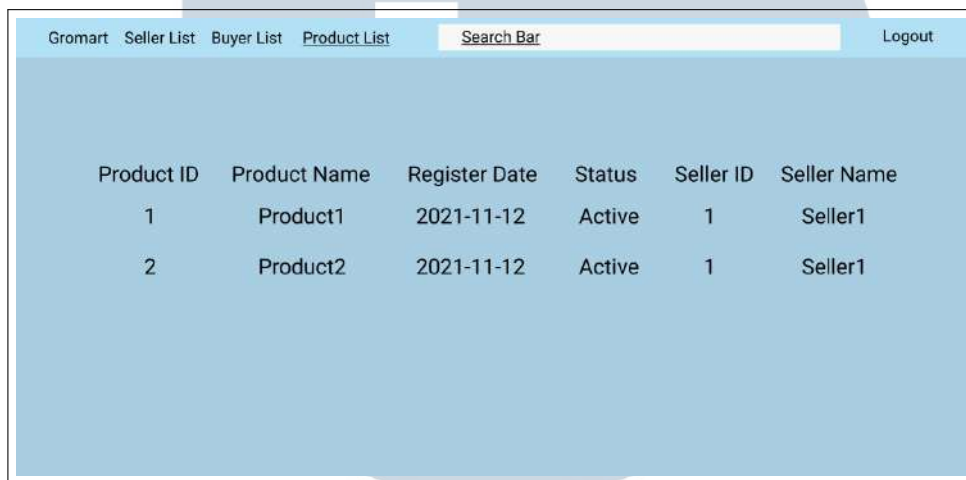
Gambar 3.24. Halaman Seller List Admin

Gambar 3.24 merupakan rancangan desain halaman *seller list* yang dapat diakses oleh admin. Melalui halaman ini, admin dapat melihat seluruh daftar *seller* yang sudah melakukan registrasi dan mengubah data dari suatu *seller* apabila dibutuhkan. Serta pada bagian *navbar* dari halaman ini, admin dapat mengakses halaman-halaman lainnya, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.

ID	Name	Credit Limit	Remaining Credit	Invoice Limit	Remaining Invoice
1	Buyer1	10000	100000	10	10
2	Buyer2	10000	100000	10	10

Gambar 3.25. Halaman Buyer List Admin

Gambar 3.25 merupakan rancangan desain halaman *buyer list* yang dapat diakses oleh admin. Melalui halaman ini, admin dapat melihat seluruh daftar *buyer* yang sudah melakukan registrasi dan mengubah data dari suatu *buyer* apabila dibutuhkan. Sama seperti pada halaman sebelumnya, di halaman ini admin dapat mengakses halaman-halaman lainnya melalui *navbar*, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.



Product ID	Product Name	Register Date	Status	Seller ID	Seller Name
1	Product1	2021-11-12	Active	1	Seller1
2	Product2	2021-11-12	Active	1	Seller1

Gambar 3.26. Halaman Product List Admin

Gambar 3.26 merupakan rancangan desain halaman *product list* yang dapat diakses oleh admin. Melalui halaman ini, admin dapat melihat seluruh daftar yang terdapat pada aplikasi tersebut dan juga dapat melihat nama-nama *seller* yang bertanggung jawab atas suatu . Sama seperti pada halaman sebelumnya, di halaman ini admin dapat mengakses halaman-halaman lainnya melalui *navbar*, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.27. Halaman Product List Seller

Gambar 3.27 merupakan rancangan desain halaman *product list* yang dapat diakses oleh *seller*. Melalui halaman ini, pengguna dapat mengakses halaman *add product*, mengubah nilai dari suatu produk, serta menghapus produk tersebut. Serta pada bagian *navbar* dari halaman ini, pengguna dapat mengakses halaman-halaman lainnya, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.



Gambar 3.28. Halaman Order List Seller

Gambar 3.28 merupakan rancangan desain halaman *order list* yang dapat diakses oleh *seller*. Melalui halaman ini, pengguna dapat melihat seluruh pesanan terhadap produk yang dipegang oleh *seller* tersebut. Serta pengguna juga dapat menerima pesanan yang ada apabila merasa pesanan terse-

but memang dapat dikirimkan. Sama seperti pada halaman sebelumnya, di halaman ini pengguna dapat mengakses halaman-halaman lainnya melalui *navbar*, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.



Product Name	Order Date	Quantity	Price	Status
Kiko	2021-11-12	1	5000	Accepted

Gambar 3.29. Halaman Report Seller

Gambar 3.29 merupakan rancangan desain halaman *report* yang dapat diakses oleh *seller*. Melalui halaman ini, pengguna dapat melihat seluruh pesanan yang telah diterima oleh *seller* pada halaman *order list* sebelumnya. Selain melihat daftar pesanan, pengguna juga dapat mencetak data-data tersebut dalam bentuk laporan dengan format xls atau Microsoft Excel. Sama seperti pada halaman sebelumnya, di halaman ini pengguna dapat mengakses halaman-halaman lainnya melalui *navbar* serta keluar dari aplikasi melalui tombol *logout*.

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. Halaman Product List Buyer

Gambar 3.30 merupakan rancangan desain halaman *product list* yang dapat diakses oleh *buyer*. Melalui halaman ini, pengguna dapat melihat produk-produk yang dapat dibeli dan memasukkan produk yang ingin dibeli ke dalam keranjang pribadi mereka dengan menekan tombol *add to cart*. Serta pada bagian *navbar* dari halaman ini, pengguna dapat mengakses halaman-halaman lainnya, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.



Gambar 3.31. Halaman Cart Buyer

Gambar 3.31 merupakan rancangan desain halaman *cart* yang dapat diakses oleh *buyer*. Melalui halaman ini, pengguna dapat melihat produk-produk yang terdapat pada keranjang milik pengguna tersebut. Selain melihat, pengguna juga dapat mengubah kuantitas atau jumlah produk yang ingin

dibeli serta dapat menghapus produk tersebut dari keranjang. Apabila sudah merasa yakin terhadap barang yang akan dibeli, pengguna dapat melanjutkan transaksi dengan menekan tombol *checkout*. Sama seperti pada halaman sebelumnya, di halaman ini pengguna dapat mengakses halaman-halaman lainnya melalui *navbar*, melakukan proses *searching* dari data-data yang ada di halaman tersebut, serta keluar dari aplikasi melalui tombol *logout*.



ID	Total Price	Order Date	
1	5000	2021-11-12	Detail

Gambar 3.32. Halaman Order List Buyer

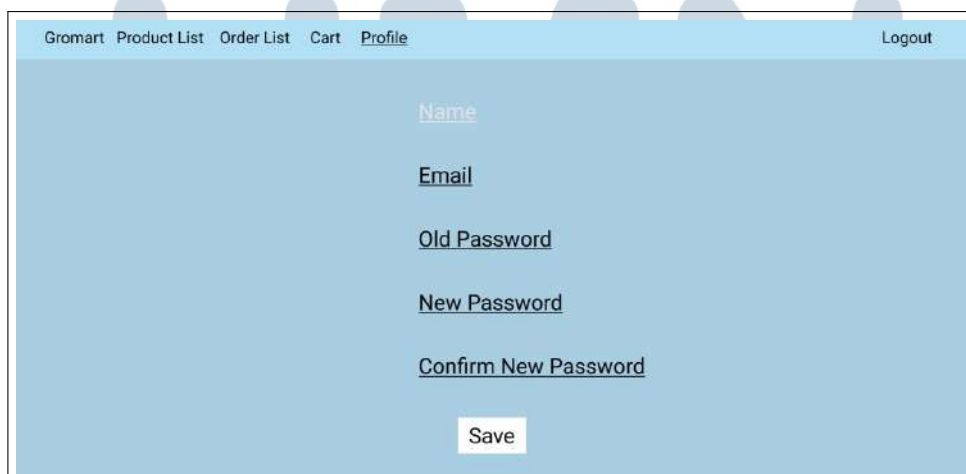
Gambar 3.32 merupakan rancangan desain halaman *order list* yang dapat diakses oleh *buyer*. Melalui halaman ini, pengguna dapat melihat beragam pesanan yang telah dilakukan oleh pengguna tersebut. Pengguna juga dapat melihat isi dari setiap pesanan yang ada dengan menekan tombol detail yang terdapat di sebelah kanan pesanan. Sama seperti pada halaman sebelumnya, di halaman ini pengguna dapat mengakses halaman-halaman lainnya melalui *navbar* serta keluar dari aplikasi melalui tombol *logout*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.33. Halaman Order Detail Buyer

Gambar 3.33 merupakan rancangan desain halaman *order detail* yang dapat diakses oleh *buyer*. Melalui halaman ini, pengguna dapat melihat produk-produk apa saja yang terdapat dalam suatu pesanan. Selain itu, pengguna juga dapat melakukan pembayaran terhadap produk yang mereka pesan melalui halaman ini dan pembayaran dilakukan terhadap setiap produk bukan setiap pesanan. Sama seperti pada halaman sebelumnya, di halaman ini pengguna dapat mengakses halaman-halaman lainnya melalui *navbar* serta keluar dari aplikasi melalui tombol *logout*.



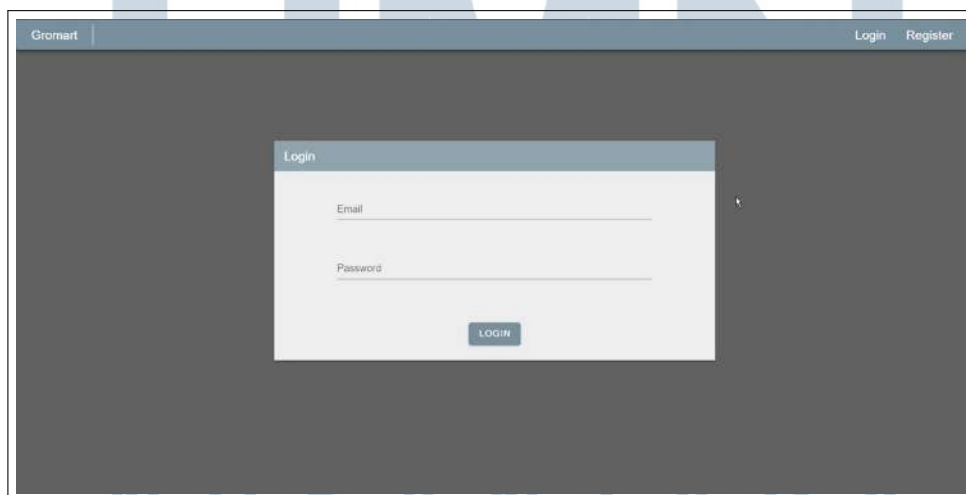
Gambar 3.34. Halaman Profile Buyer

Gambar 3.34 merupakan rancangan desain halaman *profile* yang dapat diakses oleh *buyer*. Pada halaman ini, pengguna dapat melihat nama dan *email* yang digunakan oleh akun tersebut. Pengguna juga dapat mengubah

email maupun *password* dari akun tersebut dengan memasukkan *password* lama, *password* baru, dan kembali memasukkan *password* baru sebagai bentuk konfirmasi. Sama seperti pada halaman sebelumnya, di halaman ini pengguna dapat mengakses halaman-halaman lainnya melalui *navbar* serta keluar dari aplikasi melalui tombol *logout*.

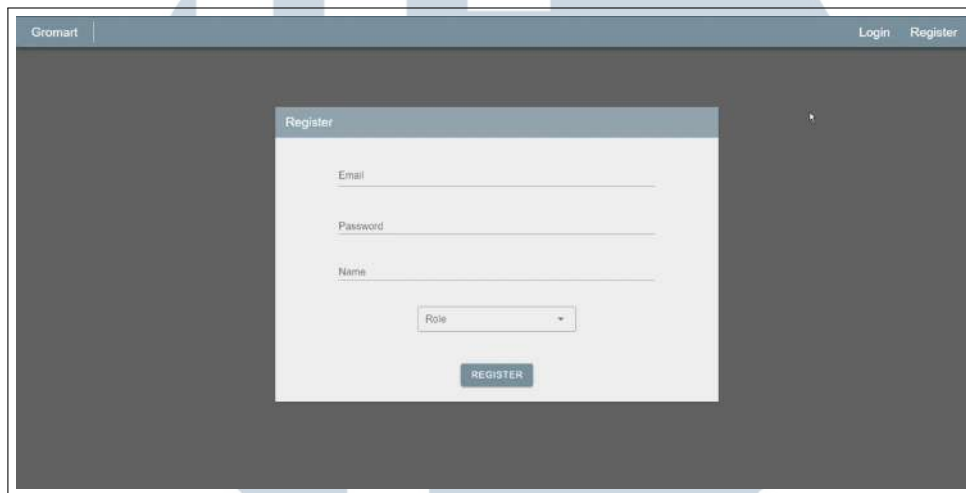
3.3.5 Implementasi

Berikut merupakan hasil akhir dari pembuatan aplikasi Gromart yang dibuat sesuai dengan rancangan desain aplikasi dan *requirement* dari pembimbing lapangan, di mana beberapa contoh dari *requirement* tersebut adalah diperlukannya halaman *login* yang dapat digunakan oleh admin, *seller* dan *buyer*, admin dapat melakukan *approve seller* dan mengatur jumlah produk yang dapat dibuat oleh *seller*, *seller* bisa melakukan *add*, *edit* dan *delete* produk, *seller* memiliki menu *report* yang bisa di *export* dalam bentuk *.csv* / *.pdf* / *.xls*, *buyer* dapat mengubah data profil atau *password* di menu *profile*, *buyer* dapat melakukan *checkout* ketika stok barang masih tersedia dan *buyer* masih memiliki sisa *credit limit* dan *invoice limit*, memastikan agar menu yang ada hanya dapat diakses oleh pengguna dengan hak otoritas yang sesuai, dan masih banyak lainnya. *Requirement* yang telah disebutkan juga masih mengalami perubahan ketika proses pengerjaan aplikasi berlangsung.



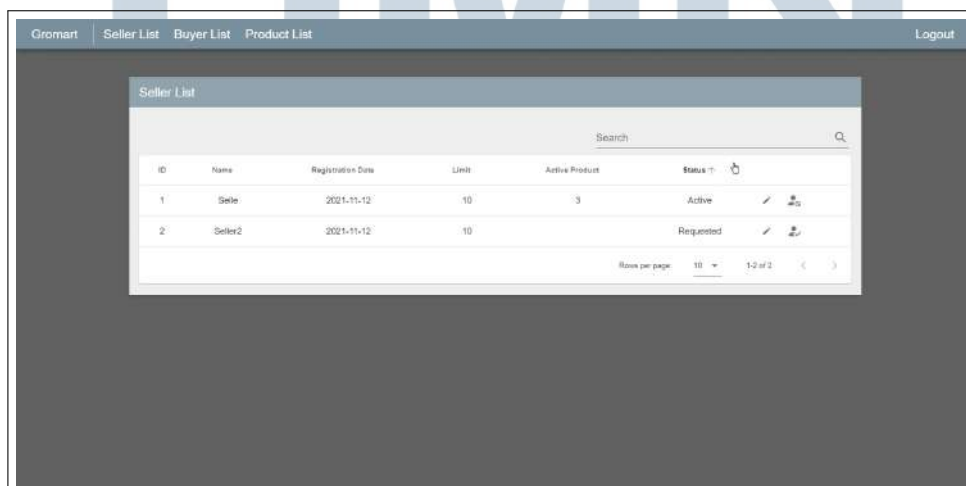
Gambar 3.35. Halaman Login

Gambar 3.35 merupakan hasil akhir dari halaman *login* yang dibuat sesuai dengan rancangan desain. Pengguna akan diminta untuk memasukkan *email* dan *password* untuk melakukan *login*. Halaman ini juga memiliki *navbar* di mana pengguna dapat berpindah dari halaman *login* menuju registrasi dan sebaliknya.



Gambar 3.36. Halaman Register

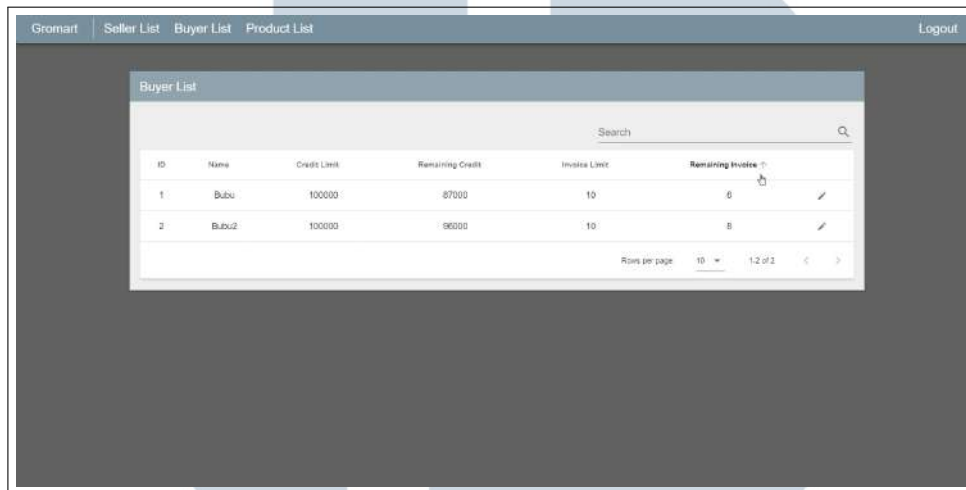
Gambar 3.36 merupakan hasil akhir dari halaman *register* yang dibuat sesuai dengan rancangan desain. Pengguna akan diminta untuk memasukkan nama, *email*, *password* dan *role* dari pengguna agar dapat melakukan registrasi. Halaman ini juga memiliki *navbar* di mana pengguna dapat berpindah dari halaman *register* menuju *login* dan sebaliknya.



ID	Name	Registration Date	Limit	Active Product	Status
1	Selle	2021-11-12	10	3	Active
2	Seller2	2021-11-12	10		Requested

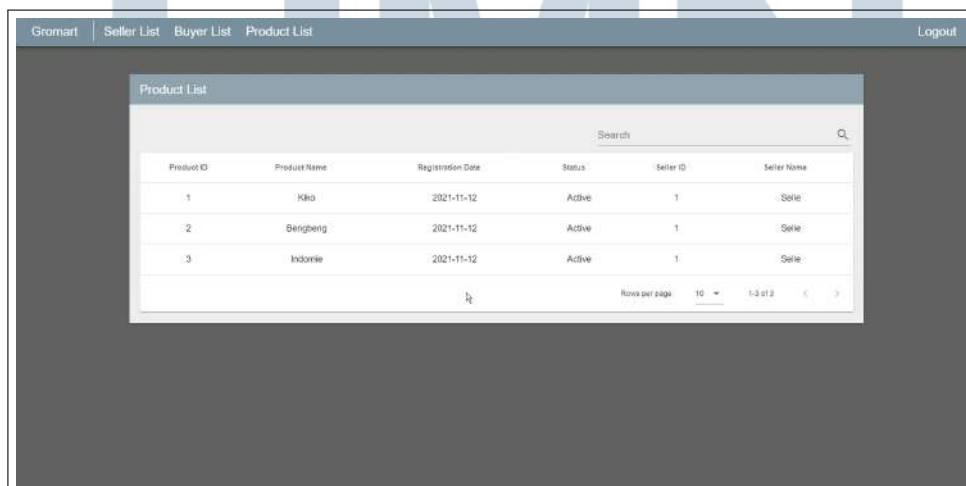
Gambar 3.37. Halaman Seller List Admin

Gambar 3.37 merupakan hasil akhir dari halaman *seller list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.



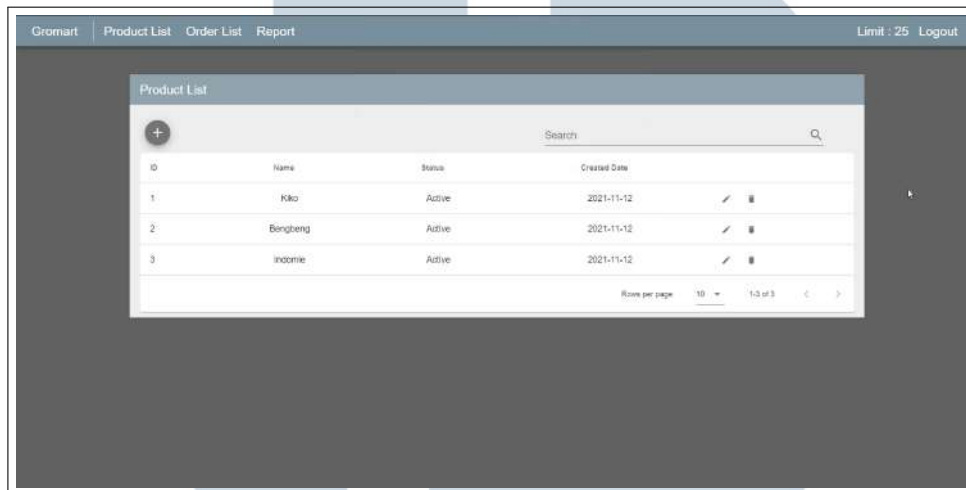
Gambar 3.38. Halaman Buyer List Admin

Gambar 3.38 merupakan hasil akhir dari halaman *buyer list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.



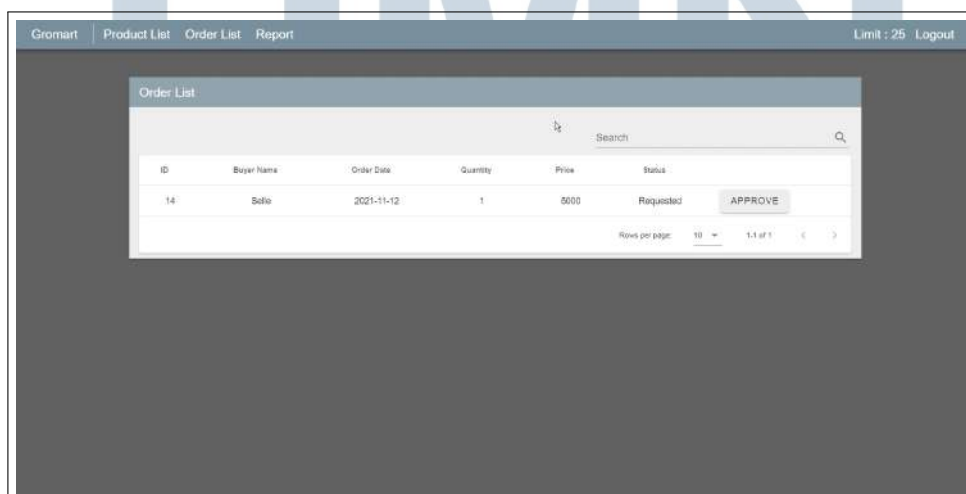
Gambar 3.39. Halaman Product List Admin

Gambar 3.39 merupakan hasil akhir dari halaman *product list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.



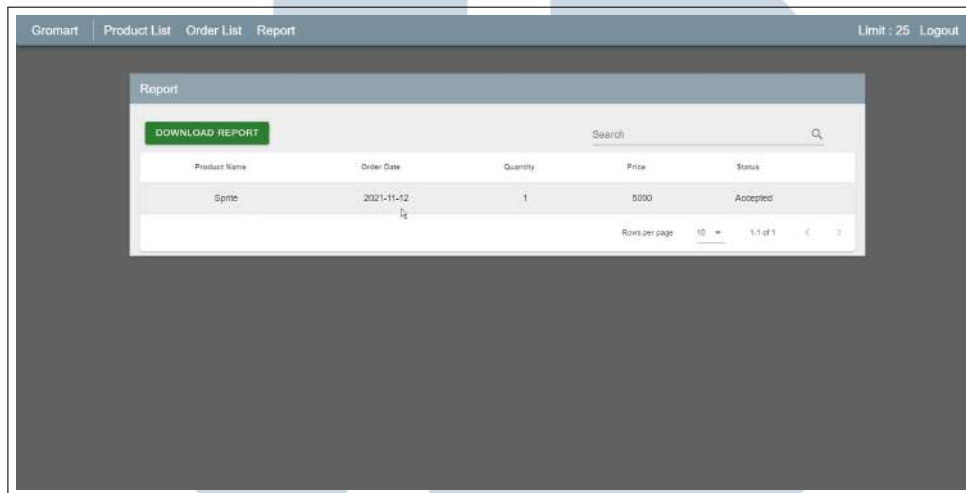
Gambar 3.40. Halaman Product List Seller

Gambar 3.40 merupakan hasil akhir dari halaman *product list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



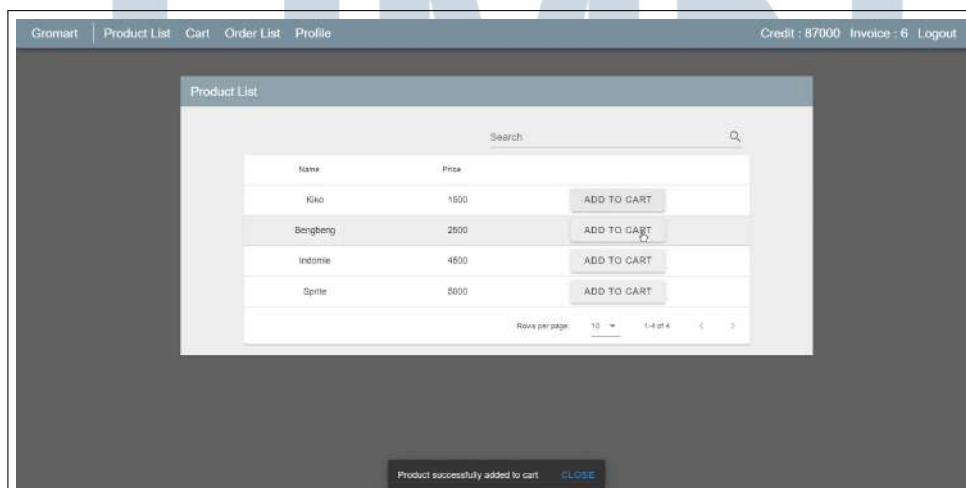
Gambar 3.41. Halaman Order List Seller

Gambar 3.41 merupakan hasil akhir dari halaman *order list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



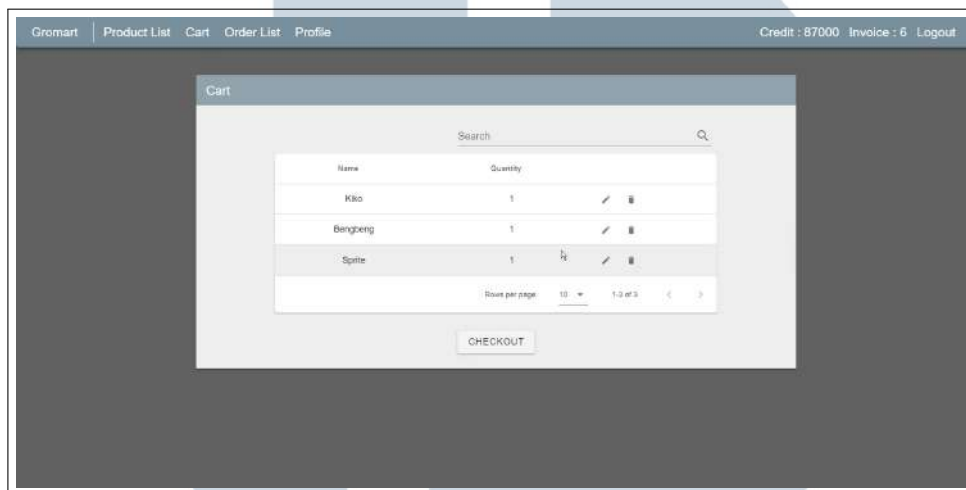
Gambar 3.42. Halaman Report Seller

Gambar 3.42 merupakan hasil akhir dari halaman *report* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



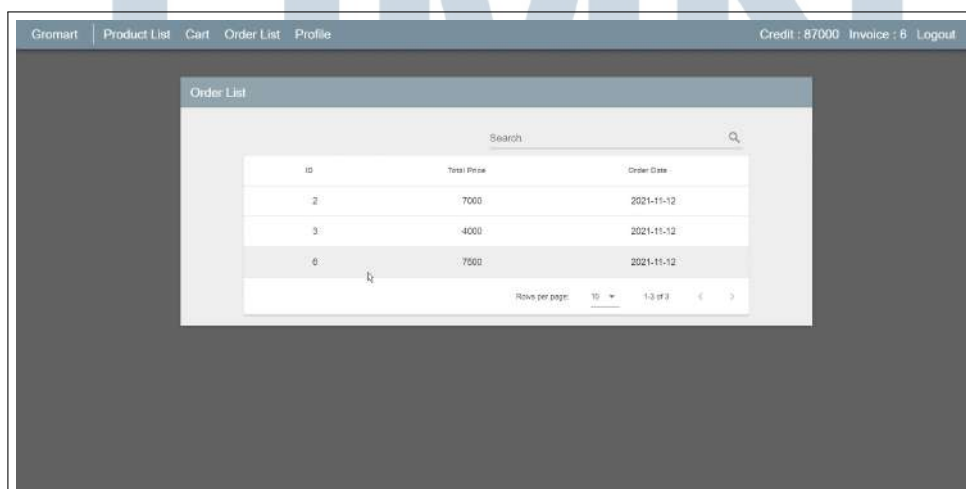
Gambar 3.43. Halaman Product List Buyer

Gambar 3.43 merupakan hasil akhir dari halaman *product list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



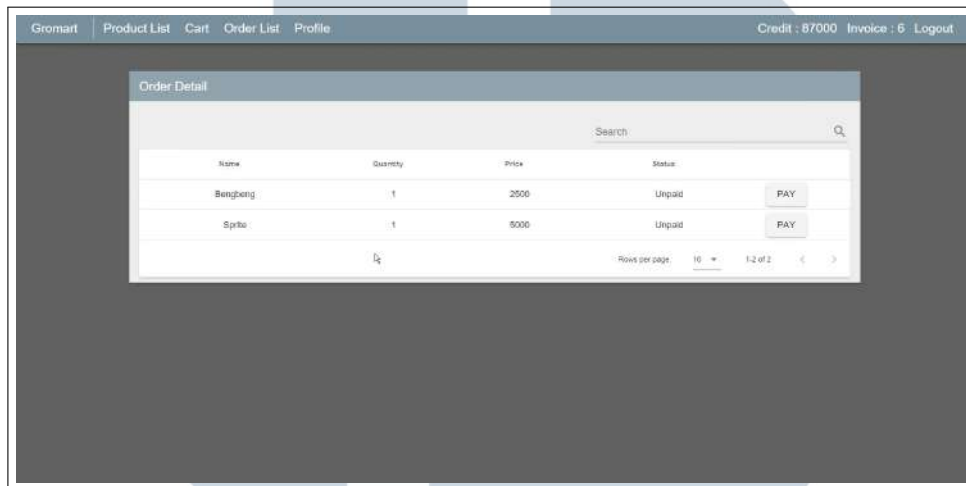
Gambar 3.44. Halaman Cart Buyer

Gambar 3.44 merupakan hasil akhir dari halaman *cart* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



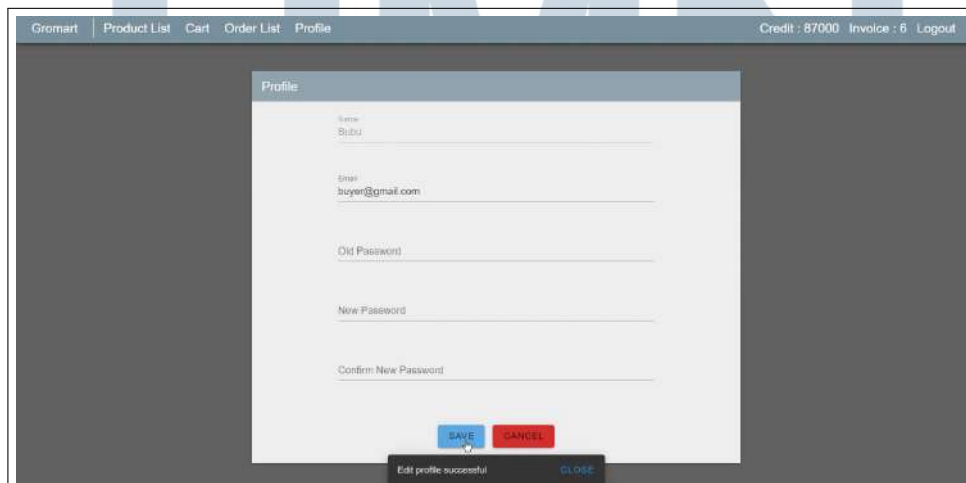
Gambar 3.45. Halaman Order List Buyer

Gambar 3.45 merupakan hasil akhir dari halaman *order list* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



Gambar 3.46. Halaman Order Detail Buyer

Gambar 3.46 merupakan hasil akhir dari halaman *order detail* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur, serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.



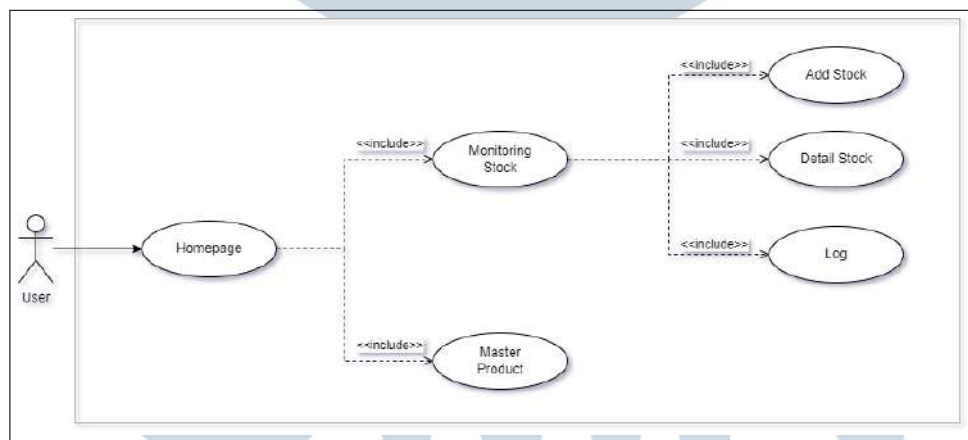
Gambar 3.47. Halaman Profile Buyer

Gambar 3.47 merupakan hasil akhir dari halaman *profile* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data serta data milik pengguna yang ditampilkan pada bagian *navbar* aplikasi.

3.4 ND6

ND6 atau Nex Distribution 6 merupakan sebuah aplikasi transaksi yang dibuat untuk para pengusaha yang membuka usaha jual beli barang. Melalui aplikasi ini, pengguna dapat melakukan pemesanan ataupun menjual barang yang mereka miliki dalam jumlah banyak maupun sedikit. Mereka juga dapat menentukan barang-barang apa saja yang ingin mereka jual dalam usaha mereka. Sehingga dapat dikatakan bahwa aplikasi ini memiliki cara kerja yang hampir sama dengan aplikasi Gromart.

3.4.1 Use Case



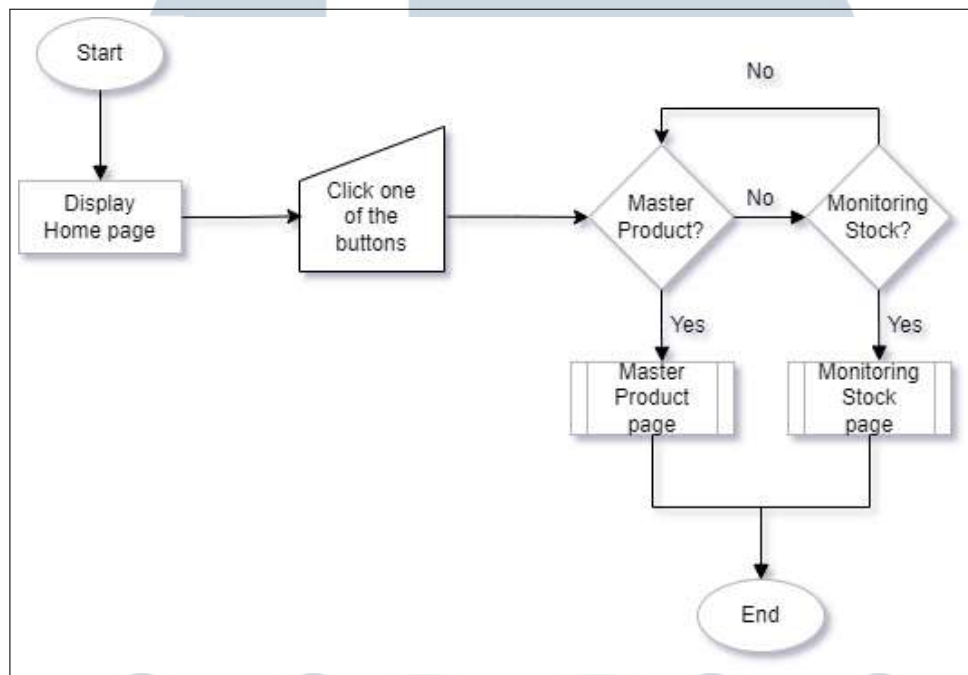
Gambar 3.48. Use case aplikasi ND6

Berdasarkan pada gambar 3.48 yang merupakan *use case* dari aplikasi ND6, hanya terdapat satu jenis pengguna di dalam aplikasi ini. Pada saat pertama kali mengakses aplikasi, pengguna akan diarahkan ke halaman *homepage* dimana pengguna dapat memilih dua halaman yang dapat dimasuki. Halaman *master product* merupakan tempat untuk menentukan produk-produk apa saja yang dapat dijual pengguna atau dibeli oleh konsumen. Sedangkan halaman *monitoring stock* merupakan tempat dimana suatu pengguna dapat melakukan pemesanan ataupun penjualan barang, meli-

hat detail dari pesanan yang ada, serta melihat hasil transaksi yang telah dilakukan dalam kurun waktu tertentu.

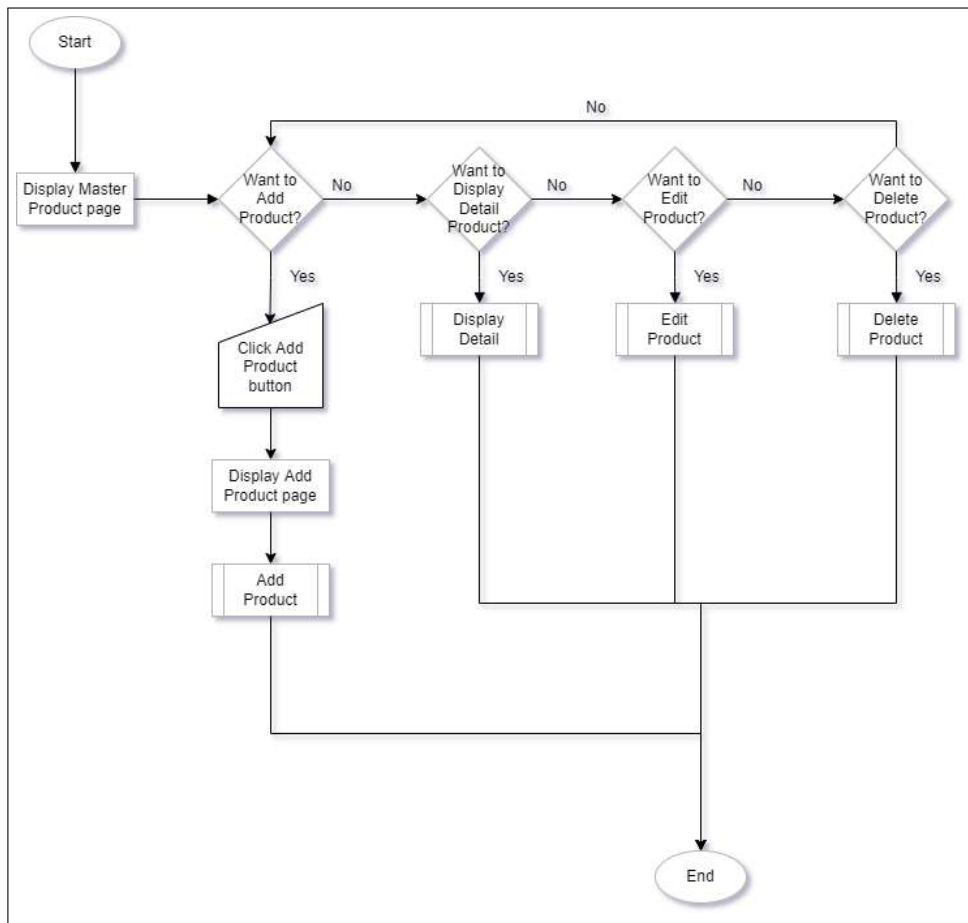
3.4.2 Flowchart

Berikut merupakan diagram *flowchart* dari aplikasi ND6, yang terdiri atas program utama dan fitur-fitur yang dapat dimanfaatkan oleh pengguna.



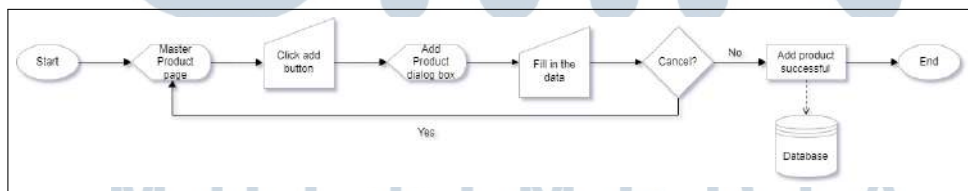
Gambar 3.49. Flowchart utama aplikasi ND6

Gambar 3.49 merupakan alur dari aplikasi ND6 secara keseluruhan. Pada saat pertama melakukan akses ke aplikasi, halaman *homepage* akan dimunculkan pada layar pengguna. Halaman tersebut menyediakan dua jenis tombol yang dapat dipilih oleh pengguna. Dengan menekan tombol *master product* maka pengguna akan diarahkan menuju halaman *master product* dan apabila pengguna menekan tombol *monitoring stock* maka pengguna akan diarahkan menuju halaman *monitoring stock*.



Gambar 3.50. Flowchart Master Product

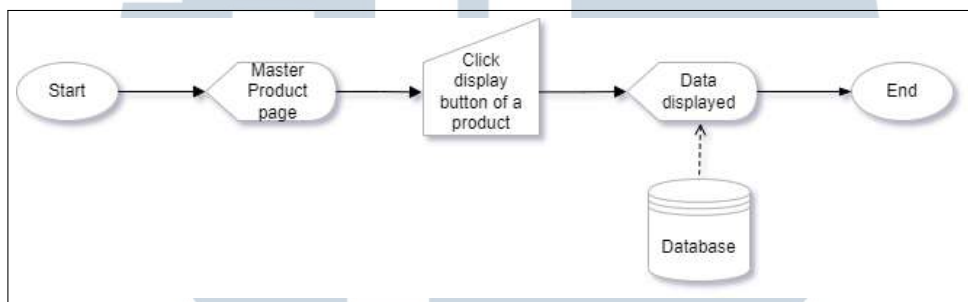
Gambar 3.50 merupakan alur dari proses-proses yang dapat dilakukan oleh pengguna pada halaman *master product*. Pada halaman tersebut, pengguna bisa melakukan *add product*, *display detail product*, *edit product*, serta *delete product*.



Gambar 3.51. Flowchart add product pada Master Product

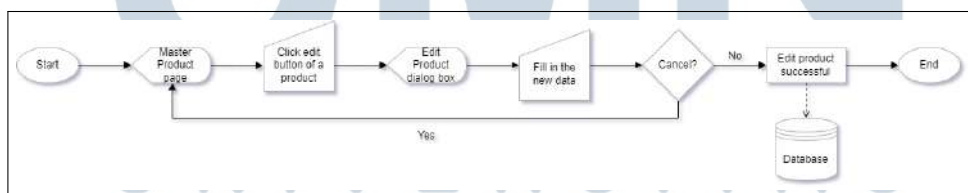
Gambar 3.51 merupakan alur dari proses *add product* yang dapat dilakukan oleh pengguna pada halaman *master product*. Dengan menekan tombol *add product* yang terdapat pada halaman tersebut, *dialog box* dari *add product* akan muncul pada layar pengguna. Pengguna akan diminta

untuk memasukkan data-data yang diperlukan agar dapat menyelesaikan proses *add product* yang berupa nama produk dan catatan atau *description* yang ingin disertakan terhadap suatu produk. Produk akan berhasil dibuat ketika pengguna telah memasukkan data-data yang diperlukan dan menyetujui pembuatan produk tersebut. Namun bila pengguna memilih untuk menolak melakukan pembuatan produk, maka proses *add product* akan dibatalkan dan pengguna akan dialihkan kembali ke halaman *master product*. Produk yang berhasil dibuat akan disimpan ke dalam *database*.



Gambar 3.52. Flowchart display detail product pada Master Product

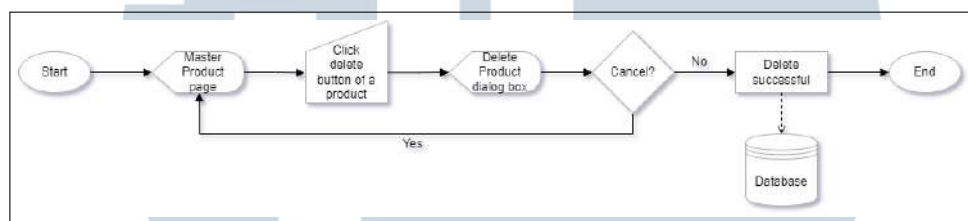
Gambar 3.52 merupakan alur dari proses *display* detail suatu produk yang dapat dilakukan pada halaman *master product*. Proses ini dapat dilakukan dengan menekan tombol *detail* pada salah satu produk yang terdapat di tabel *product list*. Aksi tersebut akan menampilkan data-data yang dimiliki oleh suatu produk. Proses ini tidak memiliki *decision symbol* atau penanggulangan terhadap kesalahan di dalamnya karena proses ini tidak memerlukan pengguna untuk memasukkan data secara mandiri.



Gambar 3.53. Flowchart edit product pada Master Product

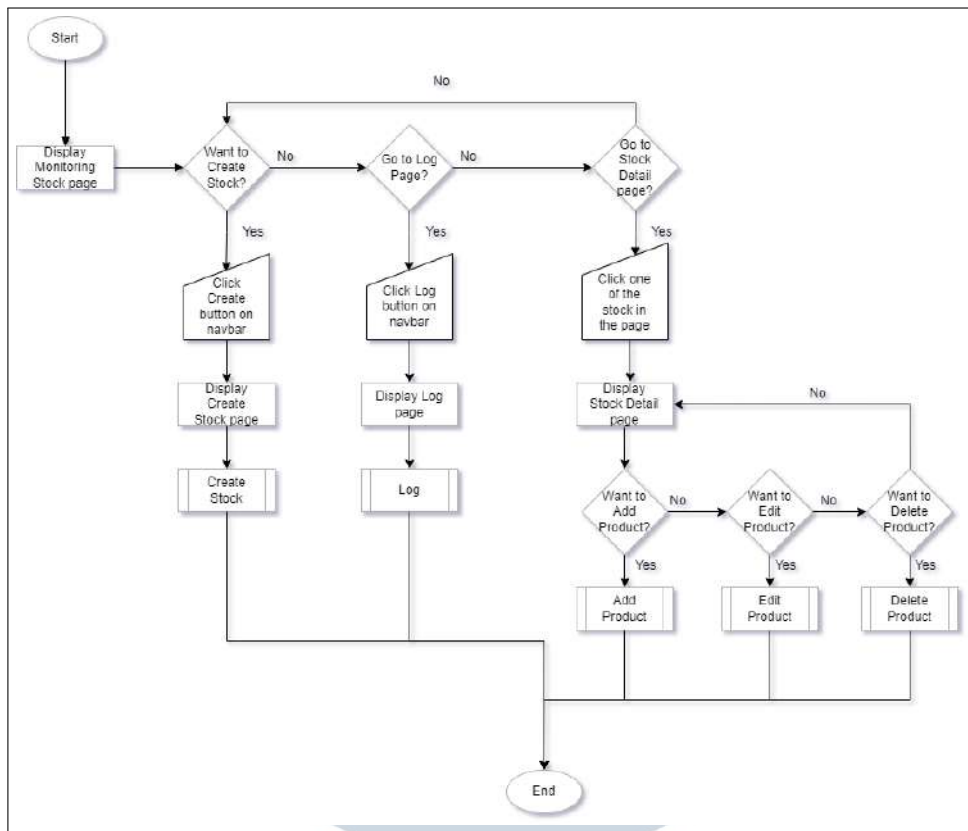
Gambar 3.53 merupakan alur dari proses *edit product* yang dapat dilakukan pada halaman *master product*. Proses ini dapat dilakukan dengan menekan tombol *edit* pada salah satu produk yang terdapat di tabel *product list*. Aksi tersebut akan memunculkan sebuah *dialog box* yang berisikan id,

nama, deskripsi, tanggal kedaluwarsa, kuantitas, serta harga dari suatu produk. Namun fitur ini hanya dapat digunakan untuk mengubah nilai pada nama produk dan deskripsi. Setelah mengubah nilai kedua data tersebut dan memilih untuk menerima melakukan perubahan, proses *edit* akan selesai dijalankan dan nilai baru yang telah dimasukkan akan disimpan ke dalam *database*. Namun bila pengguna memilih untuk menolak melakukan perubahan, maka proses *edit* akan dibatalkan dan pengguna akan dialihkan kembali ke halaman *master product*.



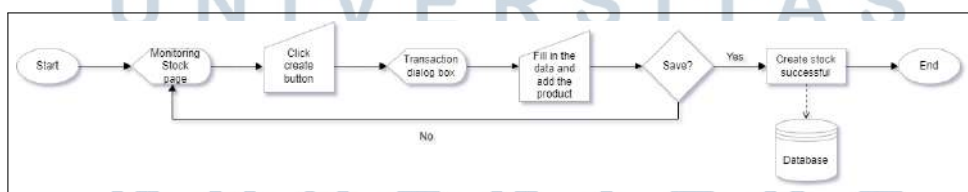
Gambar 3.54. Flowchart delete product pada Master Product

Gambar 3.54 merupakan alur dari proses *delete product* yang dapat dilakukan pada halaman *master product*. Proses ini dapat dilakukan dengan menekan tombol *delete* pada salah satu produk yang terdapat di tabel *product list*. Aksi tersebut akan memunculkan sebuah *dialog box* yang menanyakan apakah pengguna yakin untuk menghapus produk tersebut. Dengan menjawab iya, maka produk akan dihapus dari *database*. Namun bila pengguna memilih untuk menolak, maka pengguna akan dialihkan kembali ke halaman *master product* dan proses *delete* akan dibatalkan.



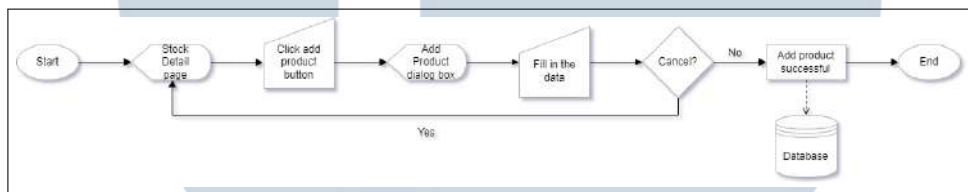
Gambar 3.55. Flowchart Monitoring Stock

Gambar 3.55 merupakan alur dari proses-proses yang dapat dilakukan oleh pengguna pada halaman *monitoring stock*. Pada halaman tersebut, pengguna bisa membuat sebuah dokumen stok baru dan bisa mengakses ke halaman *log* dengan menekan tombol yang terdapat pada *navbar* halaman tersebut. Selain itu, pengguna juga bisa mengakses halaman *stock detail* dengan menekan salah satu dokumen stok yang terdapat pada halaman tersebut. Pada halaman *stock detail*, pengguna bisa melakukan *add product*, *edit product* dan *delete product*.



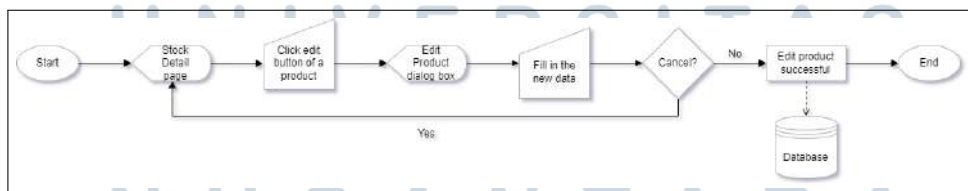
Gambar 3.56. Flowchart create stock

Gambar 3.56 merupakan alur dari proses *create stock* yang dapat diakses oleh pengguna melalui halaman *monitoring stock*. Dengan menekan tombol *create* yang terdapat pada *navbar* halaman tersebut, sebuah *dialog box* akan muncul pada layar pengguna. Pertama-tama, pengguna akan diminta untuk memasukkan data dari dokumen tersebut seperti id, deskripsi, dan status dari dokumen. Lalu pengguna dapat melanjutkan proses dengan mengisi produk-produk yang ingin dipesan atau dijual melalui tombol *add product* pada halaman tersebut. Setelah selesai memasukkan produk-produk yang dibutuhkan dan menekan tombol *save*, dokumen *stock* beserta produk-produk yang terdapat di dalamnya akan disimpan ke dalam *database*.



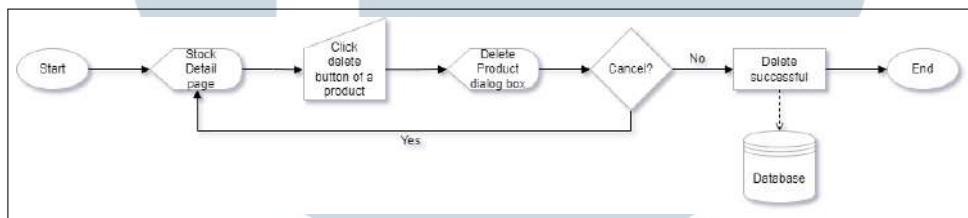
Gambar 3.57. Flowchart add product pada Stock Detail

Gambar 3.57 merupakan alur dari proses *add product* yang dapat dilakukan oleh pengguna pada halaman *stock detail*. Dengan menekan tombol *add product* yang terdapat pada halaman tersebut, *dialog box* dari *add product* akan muncul pada layar pengguna. Pengguna akan diminta untuk memasukkan data-data yang diperlukan sesuai dengan status dari setiap dokumen agar dapat menyelesaikan proses *add product*. Setelah memasukkan data dan menyetujui pembuatan produk, data dari produk tersebut akan disimpan ke dalam *database*. Namun bila pengguna memilih untuk menolak melakukan pembuatan produk, maka proses *add product* akan dibatalkan dan pengguna akan dialihkan kembali ke halaman *stock detail*.



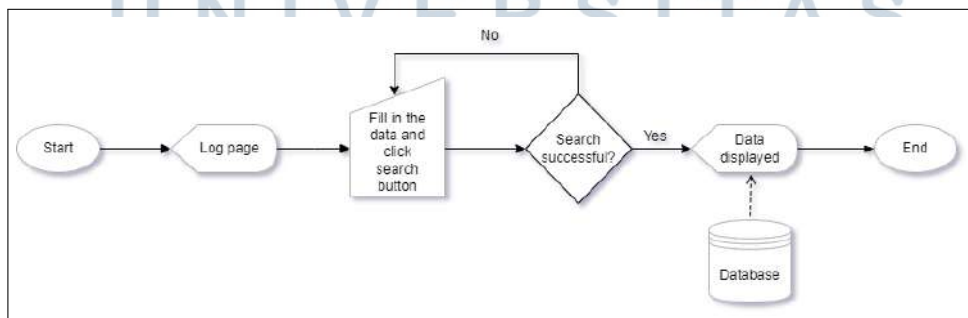
Gambar 3.58. Flowchart edit product pada Stock Detail

Gambar 3.58 merupakan alur dari proses *edit product* yang dapat dilakukan pada halaman *stock detail*. Proses ini dapat dilakukan dengan menekan tombol *edit* pada salah satu produk yang terdapat di tabel *transaction detail*. Aksi tersebut akan memunculkan sebuah *dialog box* yang berisikan id, nama, tanggal kedaluwarsa, kuantitas, serta harga dari suatu produk. Namun fitur ini hanya dapat digunakan untuk mengubah nilai pada kuantitas produk. Setelah mengubah nilai dari data tersebut dan menekan tombol *save*, proses *edit* akan selesai dijalankan dan nilai baru yang telah dimasukkan akan disimpan ke dalam *database*. Namun bila pengguna memilih untuk menolak melakukan perubahan, maka proses *edit* akan dibatalkan dan pengguna akan dialihkan kembali ke halaman *stock detail*.



Gambar 3.59. Flowchart delete product pada Stock Detail

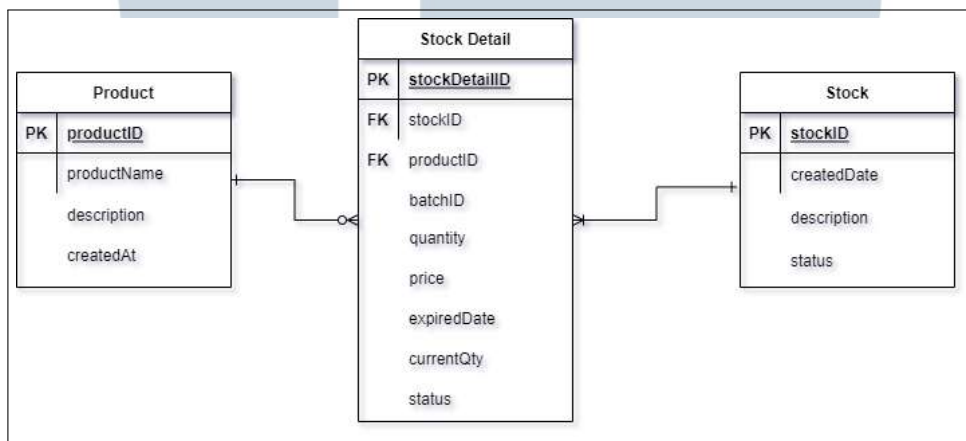
Gambar 3.59 merupakan alur dari proses *delete product* yang dapat dilakukan pada halaman *stock detail*. Proses ini dapat dilakukan dengan menekan tombol *delete* pada salah satu produk yang terdapat di tabel *transaction detail*. Aksi tersebut akan memunculkan sebuah *dialog box* yang menanyakan apakah pengguna yakin untuk menghapus produk tersebut. Dengan menjawab iya, maka produk akan dihapus dari *database*. Namun bila pengguna memilih untuk menolak, maka pengguna akan dialihkan kembali ke halaman *stock detail* dan proses *delete* akan dibatalkan.



Gambar 3.60. Flowchart Log

Gambar 3.60 merupakan alur dari proses *log* aplikasi ND6. Proses ini dapat dijalankan dengan mengisi data-data yang diperlukan pada halaman *log*, seperti id dari suatu produk serta rentang waktu yang dibagi menjadi *start date* dan *end date*. Setelah mengisi data dan menekan tombol *search*, data-data yang tersimpan pada *database* akan ditampilkan pada halaman aplikasi sesuai dengan data yang dimasukkan oleh pengguna. Bila proses *search* gagal, maka pengguna akan diminta untuk memasukkan kembali data-data yang ada dengan benar.

3.4.3 Database Schema dan Tabel



Gambar 3.61. Database schema aplikasi ND6

Gambar 3.61 merupakan diagram *database schema* yang digunakan pada aplikasi ND6. Tabel *product* merupakan tempat penyimpanan data dari seluruh produk yang dapat digunakan pada aplikasi. Setiap kali pengguna membuat sebuah produk di halaman *master product*, data-data tersebut akan disimpan ke dalam tabel *product*. Sedangkan tabel *stock* digunakan untuk menyimpan dokumen-dokumen pemesanan atau penjualan yang telah dilakukan oleh pengguna dan detail dari setiap dokumen tersebut akan disimpan pada tabel *stock detail*. Sehingga tabel *stock detail* menyimpan data-data yang lebih rinci, seperti kuantitas barang, kuantitas barang pada saat itu, harga barang, dan sebagainya.

Tabel 3.9. Tabel Product

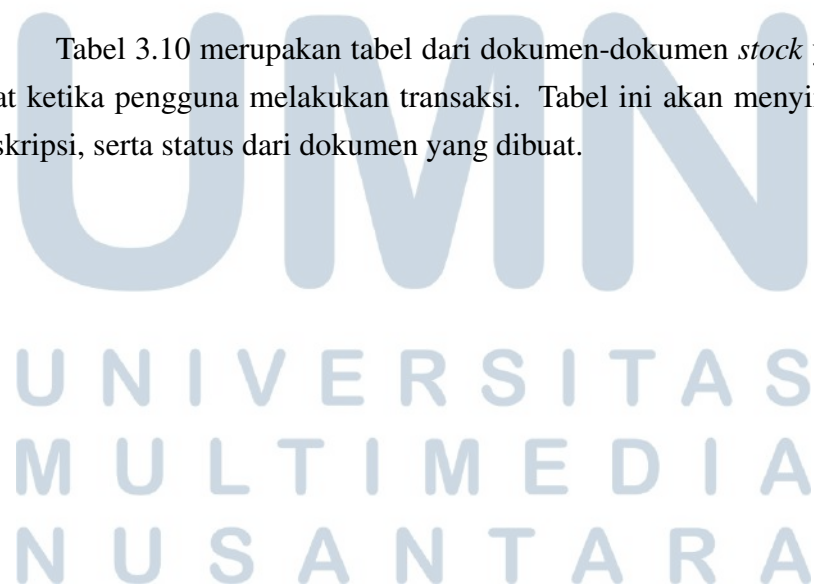
Name	Type	Null	Default	Extra	Links to
productId	int(11)	No	None	AUTO INCREMENT	
productName	varchar(255)	No	None		
description	varchar(255)	No	None		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.9 merupakan tabel dari *product* yang telah dibuat pengguna. Tabel ini akan menyimpan id, nama, serta deskripsi yang telah diberikan terhadap suatu produk ketika proses pembuatan.

Tabel 3.10. Tabel Stock

Name	Type	Null	Default	Extra	Links to
stockId	int(11)	No	None	AUTO INCREMENT	
description	varchar(255)	No	None		
status	varchar(255)	No	None		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.10 merupakan tabel dari dokumen-dokumen *stock* yang ter- buat ketika pengguna melakukan transaksi. Tabel ini akan menyimpan id, deskripsi, serta status dari dokumen yang dibuat.



Tabel 3.11. Tabel Stock Detail

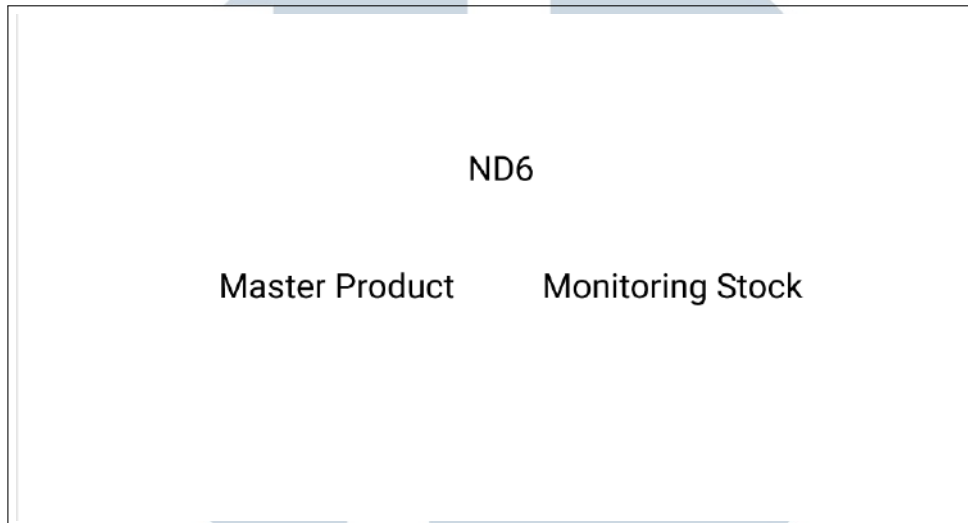
Name	Type	Null	Default	Extra	Links to
stockDetailId	int(11)	No	None	AUTO INCREMENT	
stockId	int(11)	No	None		Stock.stockId
productId	int(11)	No	None		Product.productId
batchId	int(11)	Yes	None		
expiredDate	date	Yes	None		
quantity	int(11)	Yes	None		
price	int(11)	Yes	None		
currentQty	int(11)	Yes	None		
status	varchar(255)	Yes	None		
createdAt	date	No	None		
updatedAt	date	No	None		

Tabel 3.11 merupakan tabel dari *stock detail* yang terdapat pada masing-masing dokumen. Tabel ini akan menyimpan id, *batchId*, tanggal kedaluwarsa, kuantitas, harga, kuantitas pada saat itu, serta status dari suatu produk. Perbedaan antara kuantitas dan kuantitas pada saat itu adalah kuantitas merupakan jumlah barang yang dipesan oleh pengusaha pada pesanan tersebut, sedangkan kuantitas pada saat itu merupakan banyaknya barang yang dimiliki oleh pengusaha pada saat itu juga. Selain itu, di dalam tabel ini juga terdapat *stockId* dan *productId* yang merupakan *foreign key* dari tabel *stock* dan *product*. Sehingga melalui *foreign key* tersebut, tabel *stock detail* dapat menggunakan data dari kedua tabel yang ada.

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

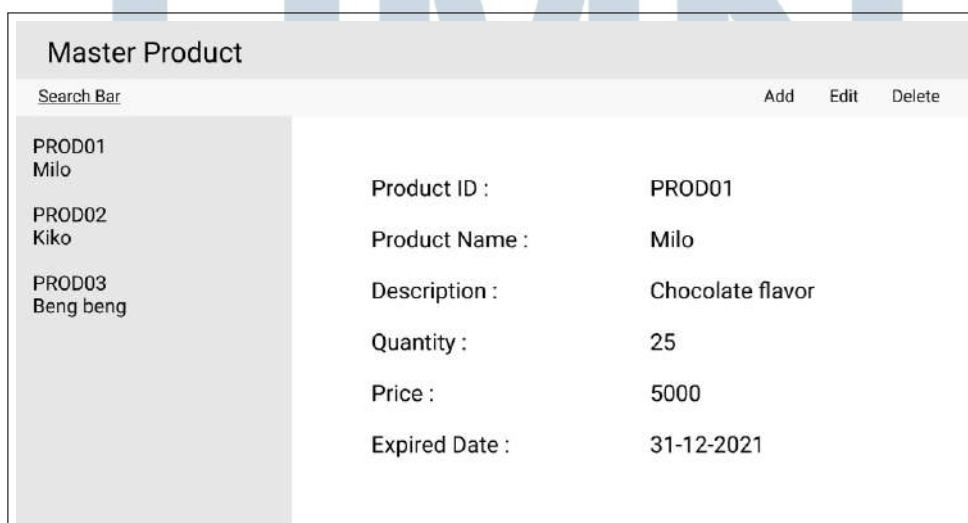
3.4.4 Rancangan Desain Aplikasi

Berikut adalah rancangan desain dari aplikasi ND6 yang disusun dengan menggunakan aplikasi Figma.



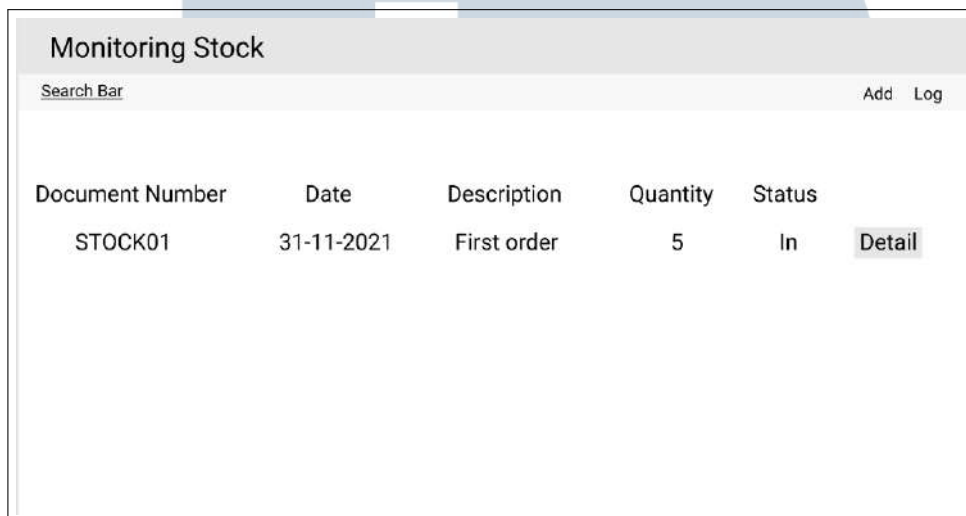
Gambar 3.62. Halaman Homepage

Gambar 3.62 merupakan rancangan desain halaman *homepage* yang akan diakses oleh seluruh pengguna ketika pertama kali membuka aplikasi ini. Melalui halaman ini, pengguna dapat mengakses ke halaman-halaman lainnya seperti *master product* dan *monitoring stock*.



Gambar 3.63. Halaman Master Product

Gambar 3.63 merupakan rancangan desain dari halaman *master product*. Melalui halaman ini, pengguna dapat melihat seluruh produk yang mereka kelola, melihat detail dari suatu produk, menambahkan produk baru, mengubah nilai dari suatu produk, serta menghapus produk. Produk yang telah dibuat pada halaman ini akan dapat diakses oleh halaman-halaman lainnya. Serta terdapat fitur *search bar* yang dapat digunakan untuk mempermudah dalam mencari suatu produk.



Document Number	Date	Description	Quantity	Status	
STOCK01	31-11-2021	First order	5	In	Detail

Gambar 3.64. Halaman Monitoring Stock

Gambar 3.64 merupakan rancangan desain dari halaman *monitoring stock*. Melalui halaman ini, pengguna dapat melihat seluruh dokumen pesanan yang telah dilakukan, melihat detail dari suatu dokumen, membuat dokumen baru, dan melihat transaksi yang telah terjadi terhadap suatu produk dalam kurun waktu tertentu.

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

Monitoring Stock

Transaction Header

Document Number STOCK01	Date 2021-12-01
Description First products	Type in

Transaction Total

Type of Items 3	Quantity 5	Amount 21000
--------------------	---------------	-----------------

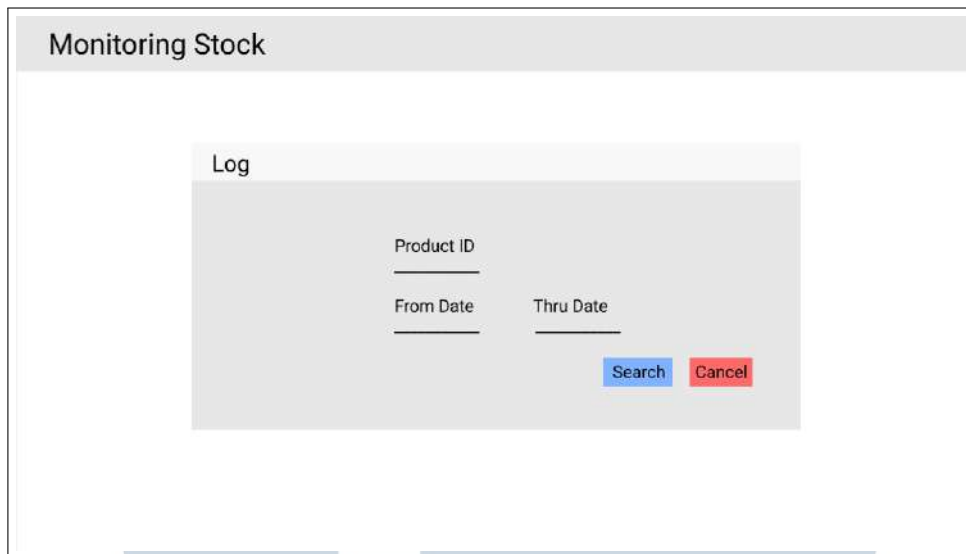
Transaction Detail

ID	Name	Expired Date	Quantity	Price	
PROD01	Milo	31-12-2021	3	5000	<input type="button" value="Add Product"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
PROD02	Kiko	31-12-2021	1	2500	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
PROD03	Beng beng	31-12-2021	1	3500	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

Gambar 3.65. Halaman Stock Detail

Gambar 3.65 merupakan rancangan desain halaman *stock detail* dari suatu dokumen. Detail dari dokumen dibagi menjadi 3 bagian yaitu *transaction header* yang berisikan nomor, deskripsi, tanggal pembuatan, serta tipe dari dokumen. Sedangkan bagian *transaction total* berisikan jumlah dari jenis barang, banyak barang, dan total harga dari produk-produk yang terdapat dalam dokumen. Bagian *transaction detail* berisikan data dari setiap produk yang ada di dalam dokumen tersebut. Serta melalui halaman ini pengguna dapat menambahkan produk baru terhadap suatu dokumen, mengubah kuantitas, serta menghapus produk dari dokumen.

UNIVERSITAS
 MULTIMEDIA
 NUSANTARA

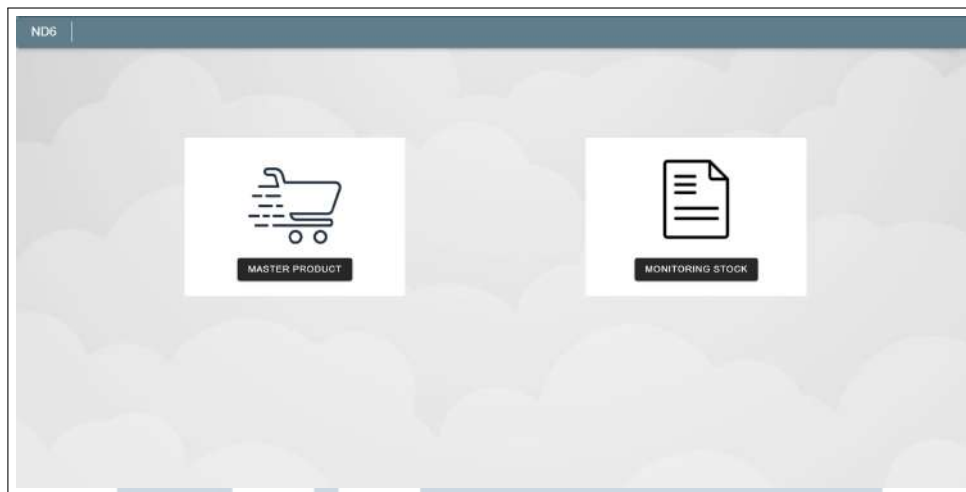


Gambar 3.66. Halaman Log

Gambar 3.66 merupakan rancangan desain dari halaman *log*. Melalui halaman ini, pengguna dapat melihat transaksi yang telah terjadi terhadap suatu produk dalam kurun waktu tertentu. Apabila pengguna tidak memasukkan *product id* yang terdapat pada halaman tersebut, maka nantinya aplikasi akan menampilkan hasil transaksi dari seluruh produk yang dikelola oleh pengguna dalam kurun waktu yang telah ditentukan.

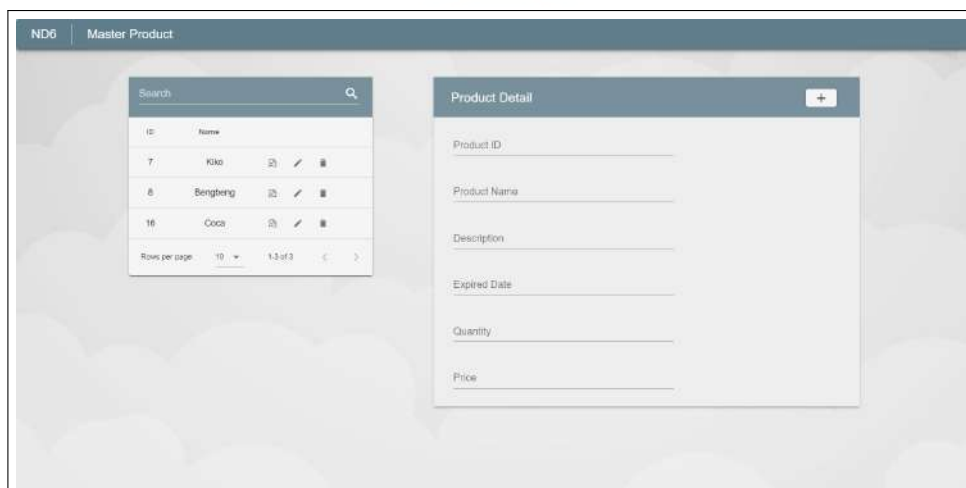
3.4.5 Implementasi

Berikut merupakan hasil akhir dari pembuatan aplikasi ND6 yang dibuat sesuai dengan rancangan desain aplikasi dan *requirement* dari pembimbing lapangan, di mana beberapa contoh dari *requirement* tersebut adalah tampilan dari halaman *master product* harus mengandung *field* id produk, nama produk, deskripsi produk, harga, dan tanggal *expire*, membuat tombol *add*, *edit* dan *delete* pada halaman tersebut, serta membuat *field search* untuk mencari produk sesuai dengan input pengguna. Sedangkan beberapa ketentuan untuk halaman *monitoring stock* adalah *header* tabel harus berisikan nomor dokumen, tanggal dokumen dibuat, deskripsi dokumen, dan total nominal stok, terdapat tombol *log* untuk melihat *history* keluar masuk stok produk, tampilan dari detail dokumen stok terdiri atas *transaction header*, *transaction total*, dan *transaction detail*, dan masih banyak lainnya.



Gambar 3.67. Halaman Homepage

Gambar 3.67 merupakan hasil akhir dari halaman *homepage* yang dibuat sesuai dengan rancangan desain. Pengguna akan diminta untuk menekan salah satu tombol yang ada agar dapat mengakses ke halaman-halaman berikutnya. Untuk saat ini, hanya terdapat dua halaman yang disediakan yaitu halaman *master product* dan *monitoring stock*.



Gambar 3.68. Halaman Master Product

Gambar 3.68 merupakan hasil akhir dari halaman *master product* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.

Document ID	Date	Description	Quantity	Status
152	2021-12-30	Test	2	In
151	2021-12-30	Beginning	1	Beginning
150	2021-12-30	Test	2	In
149	2021-12-30	Beginning		Beginning
148	2021-12-30	Beginning		Beginning
147	2021-12-30	Pesanan baru	2	Out
146	2021-12-30	Test	2	In
123	2021-12-29	Beginning		Beginning
122	2021-12-29	Beginning	1	Beginning
121	2021-12-29	Beginning	1	Beginning

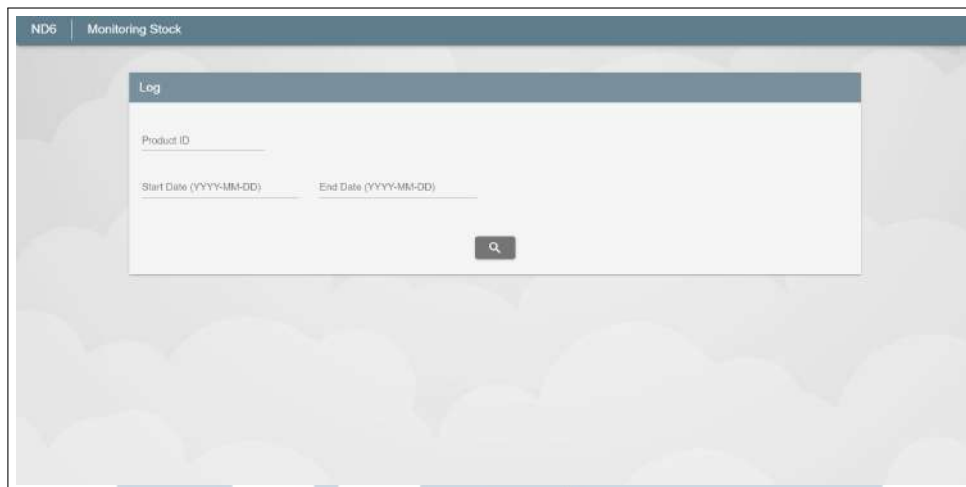
Gambar 3.69. Halaman Monitoring Stock

Gambar 3.69 merupakan hasil akhir dari halaman *monitoring stock* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, tata letak fitur serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.

Transaction Header				
Document Number: 152				
Description: Test		Status: In		
Transaction Total				
Type of Item: 2	Quantity: 200	Amount: 10000		
Transaction Detail				
ID	Name	Expired Date	Quantity	Price
7	Kilo	2022-10-12	100	5000
8	Bengkok	2022-10-09	100	5000

Gambar 3.70. Halaman Stock Detail

Gambar 3.70 merupakan hasil akhir dari halaman *stock detail* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data, data yang disajikan serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.



Gambar 3.71. Halaman Log

Gambar 3.71 merupakan hasil akhir dari halaman *log* yang dibuat sesuai dengan rancangan desain. Perbedaannya hanya terdapat pada cara penyajian data serta bentuk tombol yang berubah agar lebih nyaman untuk dilihat dan digunakan.

3.5 Kendala dan Solusi yang Ditemukan

Kendala yang didapat selama proses magang adalah sebagai berikut.

1. Pembuatan proyek dilaksanakan dengan menggunakan bahasa pemrograman JavaScript dan dengan bantuan *library* VueJS serta ExpressJS. Sehingga penulis mendapatkan kendala karena bahasa pemrograman dan *library* yang digunakan tidaklah sama dengan apa yang diajarkan selama proses perkuliahan. Solusi yang dilakukan oleh penulis adalah dengan mempelajari bahasa tersebut melalui dokumentasi-dokumentasi yang ada.
2. Peran penulis adalah sebagai *full stack developer* sehingga penulis harus memahami alur kerja aplikasi dan kebutuhan-kebutuhan yang diperlukan, serta mampu membuat *frontend*, *backend*, dan *database* dari aplikasi secara individu. Solusi yang penulis lakukan adalah dengan bertanya-tanya lebih banyak kepada *mentor* apabila menemukan kendala terhadap alur kerja aplikasi dan dengan belajar lebih banyak sehingga dapat membuat aplikasi secara utuh meskipun dikerjakan sendirian.