

BAB 3

METODOLOGI PENELITIAN

3.1 Studi Literatur

Dalam tahap studi literatur, dilakukan eksplorasi dan telaah terhadap berbagai sumber referensi yang mendukung dasar teori dalam penelitian ini. Referensi yang dikaji meliputi jurnal ilmiah, artikel, maupun buku yang berkaitan erat dengan pengembangan aplikasi penjualan berbasis *Android*. Fokus utama dari kajian ini adalah menelusuri teori-teori serta penelitian terdahulu yang relevan, khususnya dalam konteks pengembangan sistem informasi penjualan, digitalisasi proses bisnis distribusi [18], dan metode *Rapid Application Development (RAD)* sebagai pendekatan pengembangan perangkat lunak [3].

Selain membahas aspek konseptual, studi ini juga mencakup kajian terhadap teknologi yang digunakan dalam pengembangan sistem, seperti *Laravel* versi 12 sebagai *framework backend*, serta *React Native* sebagai *framework frontend*. Pemilihan teknologi ini dianalisis berdasarkan efisiensi waktu pengembangan, skalabilitas, kemudahan integrasi antar komponen sistem, serta dukungan terhadap kebutuhan lintas *platform* [6].

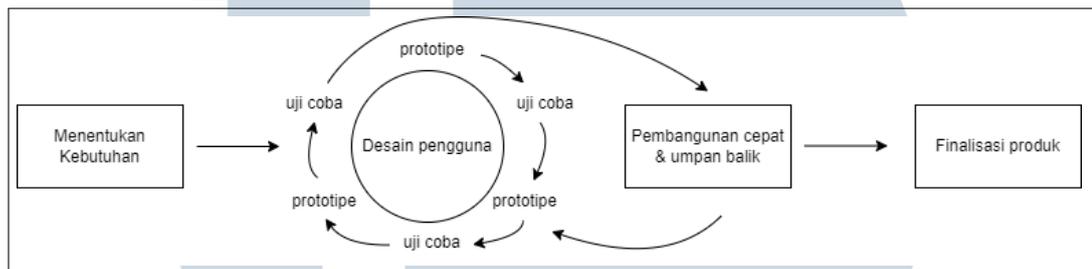
Di samping itu, dipelajari juga teori dan praktik terkait pengujian *usability (usability testing)*, sebagai metode evaluasi untuk mengukur efektivitas dan kemudahan penggunaan sistem oleh *user* akhir. Evaluasi *usability* menjadi bagian penting dari proses pengembangan karena berdampak langsung terhadap keberhasilan implementasi sistem di lapangan [16]. Kajian literatur ini menjadi dasar dalam penyusunan kerangka konseptual, perumusan pendekatan teknis, serta pengambilan keputusan metodologi dalam proses perancangan dan implementasi aplikasi pada lingkungan operasional PT Pacific Metalindo Perkasa.

3.2 Perancangan Aplikasi

Penelitian ini menggunakan metode *Rapid Application Development (RAD)* sebagai kerangka utama dalam proses pengembangan sistem. *RAD*

merupakan pendekatan pengembangan perangkat lunak yang menekankan pada kecepatan pengembangan melalui proses iteratif yang melibatkan pengguna secara aktif.

Dalam metode ini, proses pengembangan terdiri dari beberapa tahapan iteratif yang mempercepat validasi kebutuhan sistem dan perbaikan desain berdasarkan umpan balik pengguna.



Gambar 3.1. Tahapan *Rapid Application Development* (RAD)

3.2.1 Define Project Requirements (Pendefinisian Kebutuhan)

Tahapan awal ini bertujuan untuk mengidentifikasi kebutuhan sistem melalui diskusi dan wawancara langsung dengan *stakeholder*, khususnya pemilik bisnis. Informasi yang diperoleh menjadi fondasi awal dalam proses perancangan sistem dan akan digunakan untuk memastikan bahwa sistem yang dikembangkan sesuai dengan tujuan utama.

3.2.2 User Design (Desain Iteratif oleh Pengguna)

Pada tahap ini, dilakukan proses *prototyping* secara iteratif, di mana perancang dan pengguna bekerja sama dalam merancang antarmuka sistem, menguji rancangan, serta memberikan umpan balik. Iterasi dilakukan sebanyak tiga kali, yang menjadi langkah penyempurnaan desain sebelum masuk ke tahap pembangunan sistem utama. Tahap ini mencakup:

- Pembuatan *prototype* awal
- Uji coba antarmuka
- Revisi desain berdasarkan *feedback* sebanyak tiga kali

3.2.3 Rapid Construction and Feedback (Pengembangan Cepat dan Umpan Balik)

Setelah desain dikonfirmasi, pengembangan sistem dilakukan secara modular dan bertahap. Proses ini bersifat cepat namun tetap terbuka terhadap *feedback* dari pengguna. Setiap modul yang selesai diuji dan disempurnakan berdasarkan masukan dari pengguna sebelum diintegrasikan secara penuh ke dalam sistem utama.

3.2.4 Final Implementation (Implementasi Akhir)

Tahap akhir adalah implementasi sistem ke lingkungan operasional ataupun sebagai tahapan *cutover*. Setelah sistem selesai dikembangkan dan diuji secara menyeluruh, sistem akan diimplementasikan untuk digunakan secara penuh oleh pengguna akhir. *Final testing* dilakukan untuk memastikan sistem telah memenuhi seluruh kebutuhan bisnis yang telah didefinisikan pada tahap awal.



3.2.5 Perbandingan Metode RAD dan Agile

Melalui tabel 3.1, ditampilkan perbandingan antara metode RAD dengan metode Agile.

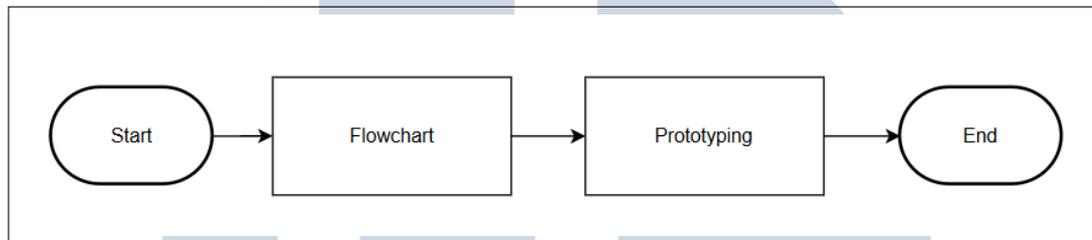
Tabel 3.1. Perbandingan Metode *Rapid Application Development* (RAD) dan *Agile*

RAD (<i>Rapid Application Development</i>)	Agile Development
Fokus terhadap <i>prototyping</i> yang cepat dan iterasi berdasarkan <i>feedback</i> langsung dari <i>user</i>	Fokus pada iterasi melalui <i>sprint</i> terstruktur, dengan <i>feedback</i> rutin dari pengguna
Keterlibatan pengguna sangat intensif sepanjang proses pengembangan	Keterlibatan pengguna dilakukan secara rutin melalui <i>sprint review</i> dan <i>backlog grooming</i>
Iterasi cepat berbasis prototipe yang dapat diuji langsung	Iterasi disusun dalam <i>sprint regular</i> (1–4 minggu) dengan hasil produk yang siap pakai
Sangat fleksibel terhadap perubahan namun rentan perluasan diluar <i>scope</i> jika tidak dikontrol	Adaptif terhadap perubahan dengan pengelolaan yang lebih terstruktur melalui <i>backlog</i>
Kualitas bergantung pada pengujian prototipe	Kualitas dijaga melalui integrasi berkelanjutan dan pengujian di setiap <i>sprint</i>
Cocok untuk pengembangan proyek dengan batas waktu yang pendek	Cocok untuk proyek berskala besar, kompleks, dan produk yang terus berkembang

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 Perancangan Sistem

Gambar 3.2. Menggambarkan *flowchart* terkait tahapan yang dilakukan pada perancangan sistem.



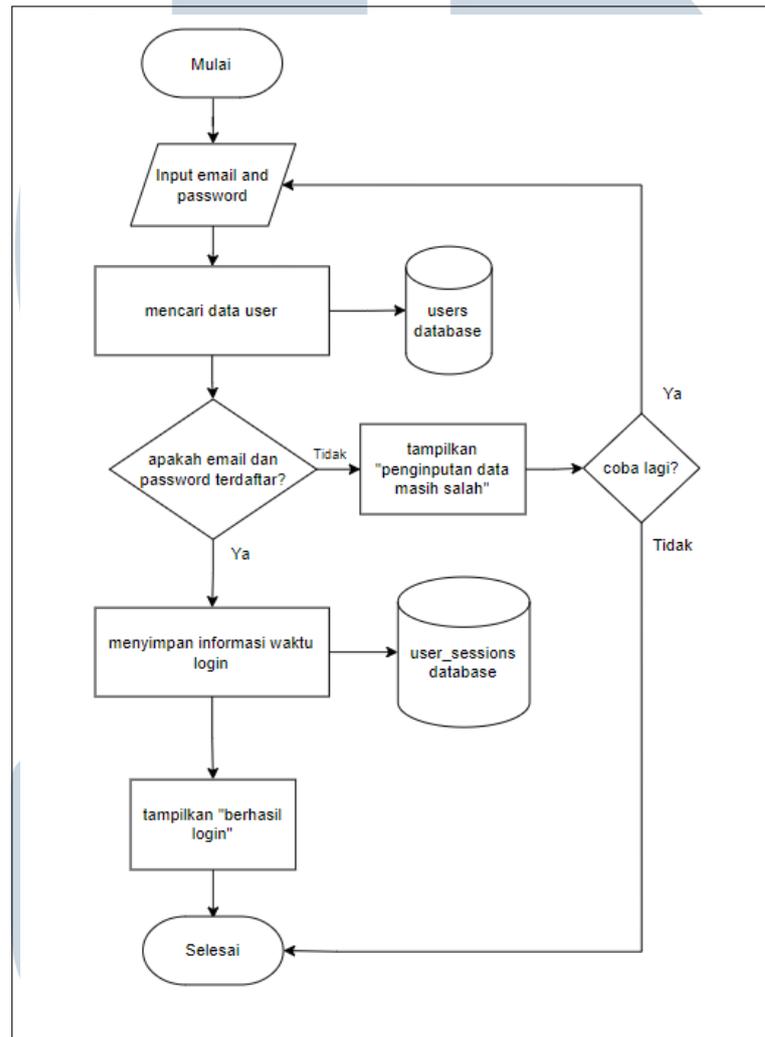
Gambar 3.2. Tahapan perancangan sistem

Pada tahap perancangan sistem, langkah pertama yang dilakukan adalah membuat dan menggambarkan alur kerja sistem melalui pembuatan *flowchart*. *Flowchart* berfungsi sebagai representasi visual dari proses bisnis yang mempermudah proses pengembangan dan pemangku kepentingan dalam memahami tahapan logis yang akan dijalankan oleh sistem. Alat bantu yang digunakan dalam proses ini adalah situs *web draw.io* [19], yang memungkinkan pembuatan diagram alur dengan simbol-simbol standar seperti terminator, proses, *input/output*, dan *decision*, sesuai dengan prinsip dasar pemodelan algoritma [7]. Setelah struktur alur sistem tergambar dengan baik, langkah selanjutnya adalah menyusun rancangan antarmuka *user (interface)* melalui proses *prototyping*. Prototipe dirancang menggunakan *Figma* [20] sebagai *platform* desain visual yang mempermudah kolaborasi dan dokumentasi rancangan antar tim. Proses perancangan ini tidak hanya mencakup tampilan visual, tetapi juga mempertimbangkan kebutuhan *user* yang berbeda berdasarkan klasifikasi peran, seperti *sales*, *telemarketing*, dan pemilik perusahaan. Seperti yang dijelaskan dalam penelitian terdahulu, proses *prototyping* merupakan bagian penting dalam metodologi *Rapid Application Development* karena memungkinkan pengujian dan umpan balik yang cepat sebelum tahap implementasi sistem dimulai [21]. Oleh karena itu, tahapan ini menjadi dasar penting untuk memastikan sistem dibangun sesuai dengan ekspektasi *user* dan berjalan secara efisien dalam proses bisnis yang telah ditentukan.

3.3.1 Flowchart

(a) Perancangan fitur *Login*

Gambar 3.3. Menggambarkan *flowchart* terkait fitur *Login*.

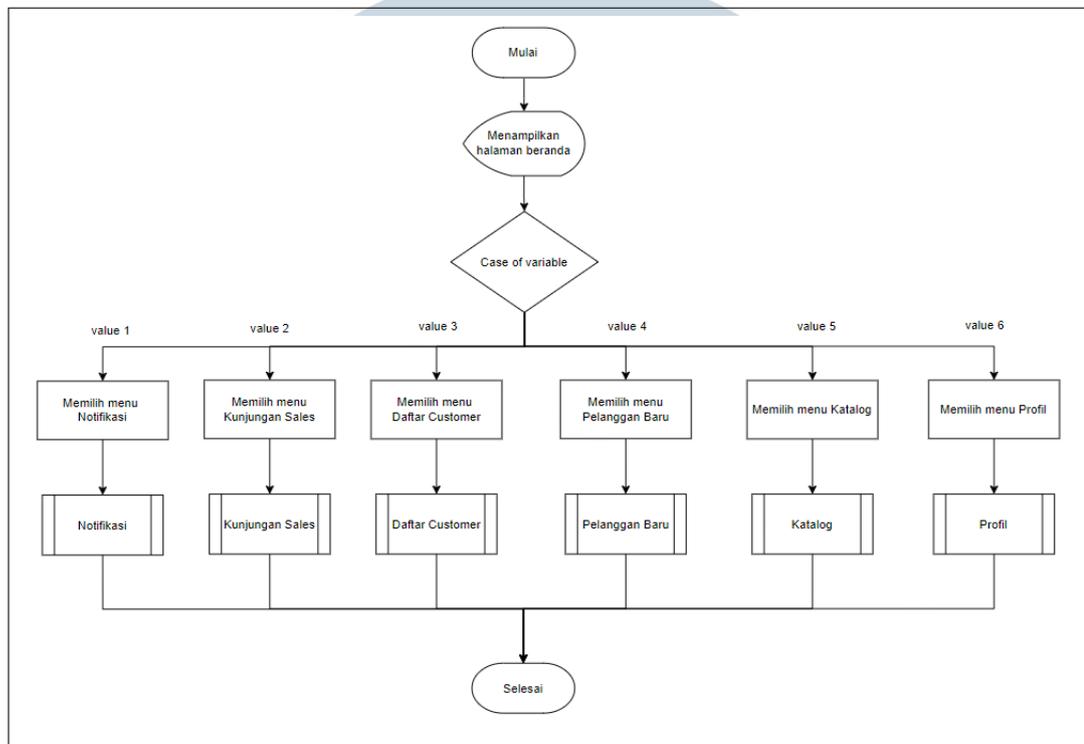


Gambar 3.3. *Flowchart* fitur *Login*

Saat pertama kali, *user* akan langsung ternavigasi ke halaman *Login*. Pada halaman ini, *user* perlu melakukan *input* pada bagian *email* dan *password* yang sudah di registrasikan melalui *backend*. Ketika *user credential valid*, maka *user* tersebut akan dinavigasi ke halaman selanjutnya yaitu Beranda.

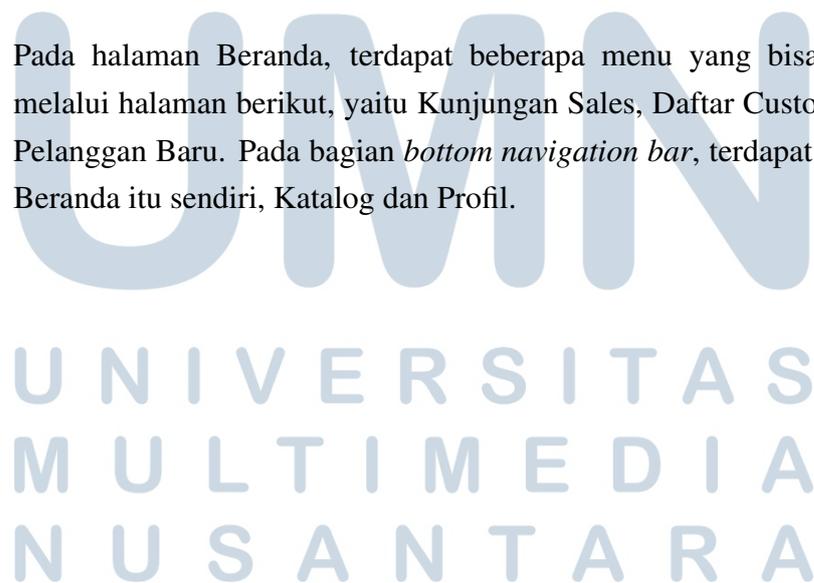
(b) Perancangan fitur yang ada pada Navigation Bar

Gambar 3.4. Menggambarkan *flowchart* terkait halaman Beranda.



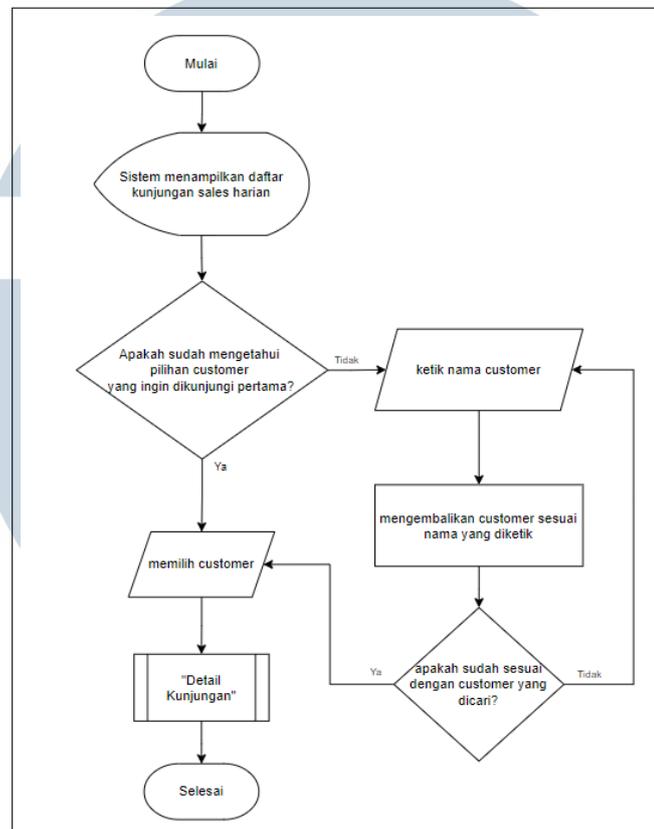
Gambar 3.4. *Flowchart* halaman beranda

Pada halaman Beranda, terdapat beberapa menu yang bisa diakses melalui halaman berikut, yaitu Kunjungan Sales, Daftar Customer, dan Pelanggan Baru. Pada bagian *bottom navigation bar*, terdapat halaman Beranda itu sendiri, Katalog dan Profil.



(c) Perancangan fitur Kunjungan Sales

Gambar 3.5. Menggambarkan *flowchart* terkait fitur Kunjungan Sales.

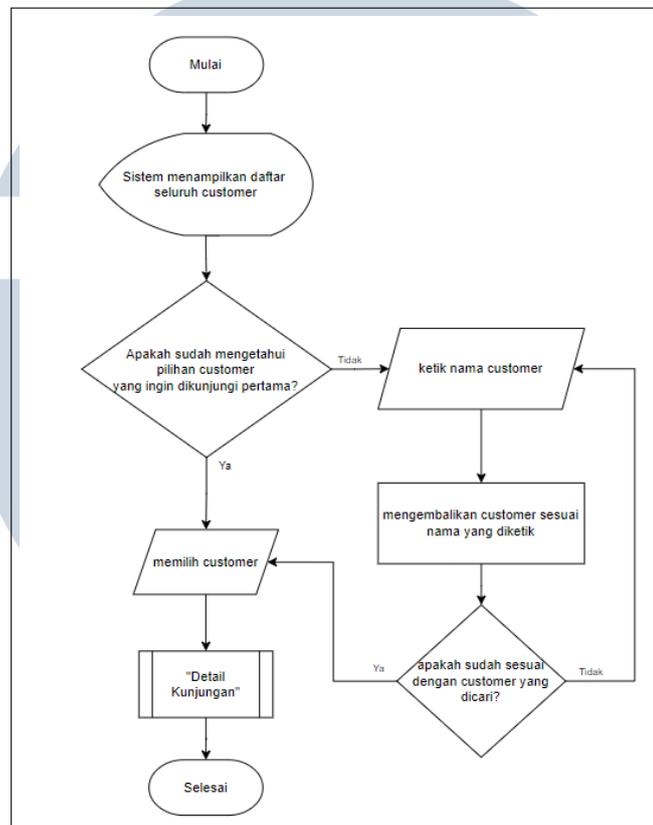


Gambar 3.5. *Flowchart* Kunjungan sales

Pada awal tampilan fitur Kunjungan Sales, sistem menampilkan *list* toko yang perlu di kunjungi oleh *sales* pada hari tersebut. Jika *sales* terdapat kebingungan dan perlu untuk mencari spesifik toko yang ingin dikunjungi, *sales* dapat menggunakan *search bar* yang ada pada halaman tersebut, dengan menggunakan nama toko sebagai *keyword* pencarian. Jika toko yang dicari oleh *user* terdapat pada *database*, maka sistem akan mengembalikan permintaan *user* dengan menampilkan toko yang sesuai dengan pencarian *user*.

(d) Perancangan fitur Daftar *Customer*

Gambar 3.6. Menggambarkan *flowchart* terkait fitur Daftar *Customer*

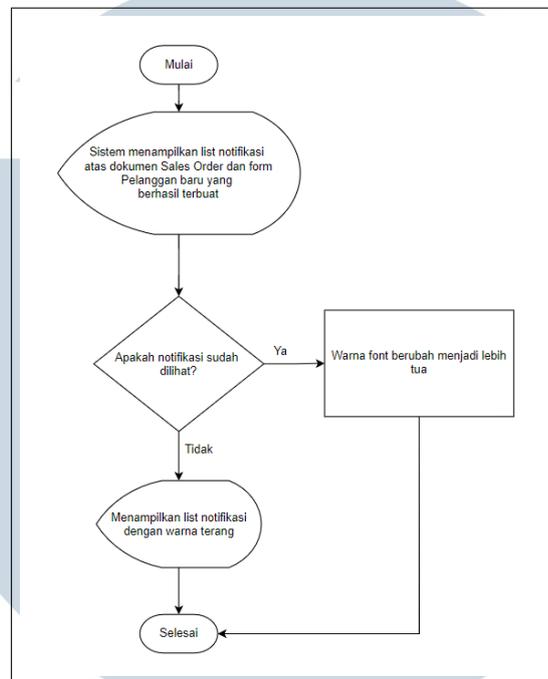


Gambar 3.6. *Flowchart* Daftar *Customer*

Fitur Daftar *Customer* mempunyai alur yang sama dengan fitur Kunjungan *Sales*, tetapi terdapat satu hal yang berbeda yaitu Kunjungan *Sales* menampilkan *list* toko yang perlu dikunjungi harian, sedangkan Daftar *Customer* menampilkan *list* dari semua toko yang berada di luar jadwal harian tersebut, sehingga memaksimalkan kerja *user* agar tetap dapat membuat dokumen *sales order* terhadap toko yang tidak ada dalam *list* kunjungannya. Pada awal tampilan fitur Daftar *Kunjungan*, sistem menampilkan *list* toko yang ada pada data *Backend*. Jika *sales* perlu untuk mencari toko yang ingin dikunjungi, *sales* dapat menggunakan *search bar* yang ada pada halaman tersebut, dengan menggunakan nama toko sebagai *keyword* pencarian.

(e) Perancangan fitur Notifikasi

Gambar 3.7. Menggambarkan *flowchart* terkait fitur Notifikasi.



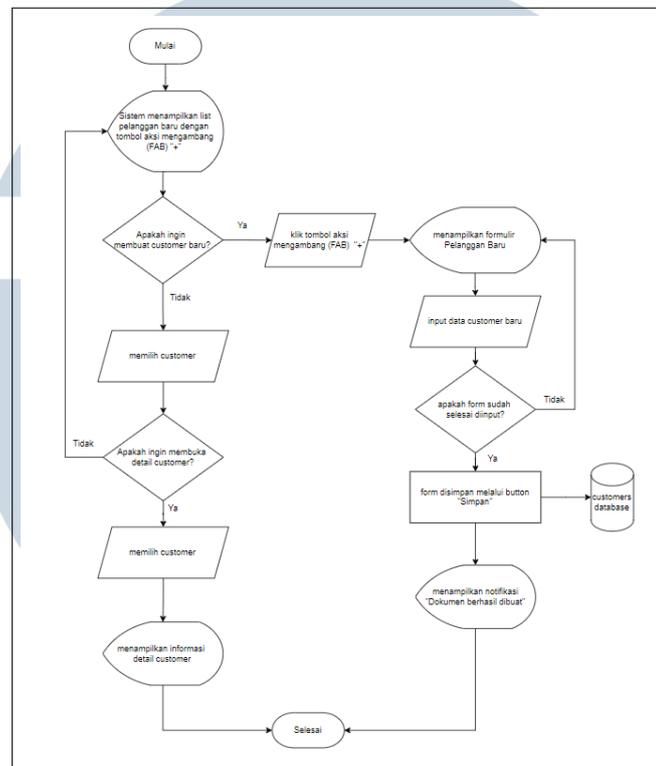
Gambar 3.7. *Flowchart* Notifikasi

Pada halaman fitur notifikasi, sistem menampilkan notifikasi atas dokumen *Sales Order* atau dokumen *form* Pelanggan Baru yang telah berhasil dibuat pada hari tersebut. *Font* dan *card* pada halaman notifikasi dapat berubah warna menjadi lebih gelap jika notifikasi tersebut telah terbuka atau terbaca oleh *user*.

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

(f) Perancangan fitur Pelanggan Baru

Gambar 3.8. Menggambarkan *flowchart* terkait fitur Pelanggan Baru.



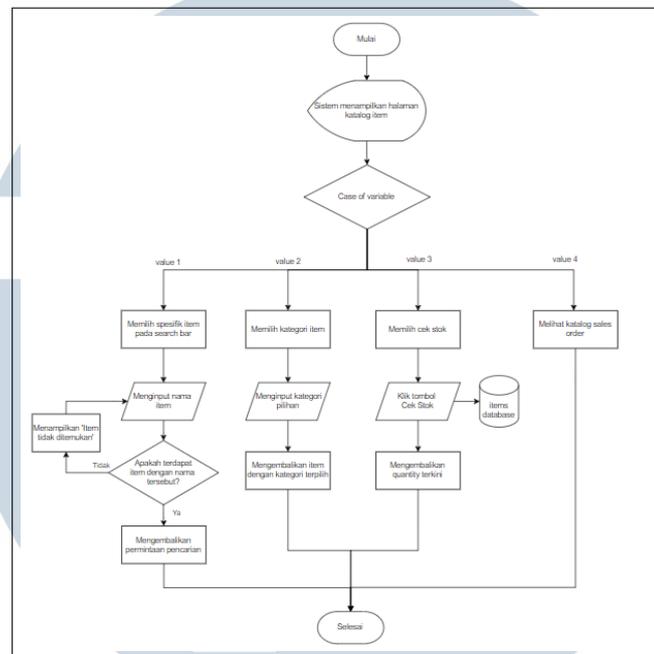
Gambar 3.8. *Flowchart* fitur Pelanggan Baru

Pada fitur Pelanggan Baru, menu tersebut mengakomodir *user* agar dapat melakukan daftar atau registrasi untuk toko baru yang ingin didaftarkan sebagai *customer* pada perusahaan. Setelah sistem berhasil membentuk beberapa dokumen *form* Pelanggan Baru, maka *backend* akan mengirimkan kembali kepada *frontend* sebuah notifikasi bahwa dokumen tersebut telah berhasil terbuat.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

(g) Perancangan fitur Katalog

Gambar 3.9. Menggambarkan *flowchart* terkait fitur Katalog.



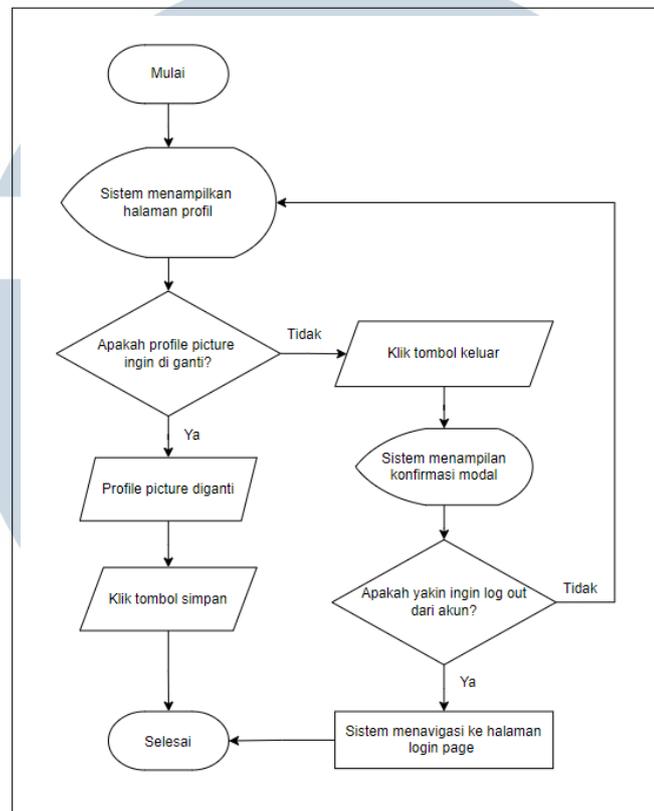
Gambar 3.9. *Flowchart* fitur Katalog

Ketika *user* melakukan akses pada fitur *Katalog*, sistem akan menampilkan produk-produk yang dijual serta terdapat pengelompokan berdasarkan merk dari masing-masing produk. Jika *user* memiliki kesulitan untuk mencari produk, *user* dapat langsung mencari melalui *search bar* yang ada. Melalui fitur *Katalog*, *user* dapat mengecek data stok secara *realtime* dengan menekan tombol "Cek Stok". Sistem akan mengembalikan permintaan *user* dengan menampilkan *quantity* produk berdasarkan data yang ada pada *backend*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

(h) Perancangan fitur Profil

Gambar 3.10. Menggambarkan *flowchart* terkait fitur Profil.



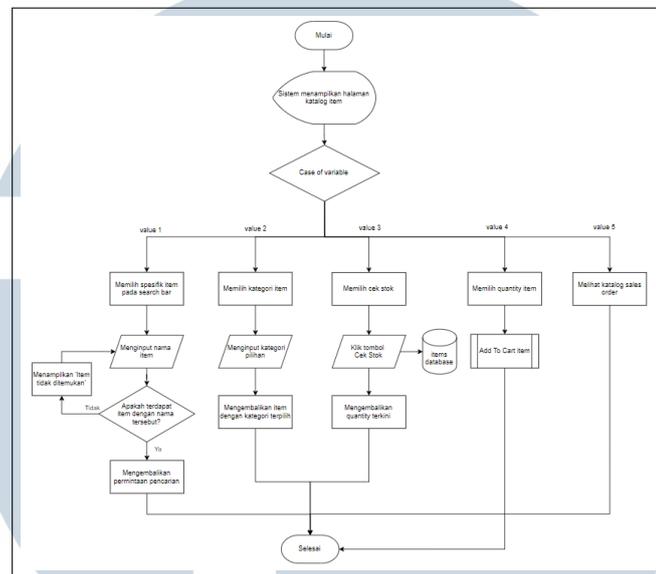
Gambar 3.10. *Flowchart* fitur Profil

Fitur Profil menampilkan informasi pribadi *user* yang sedang aktif di dalam sistem. Informasi yang ditampilkan meliputi nama *user*, alamat *email*, serta terdapat tombol *Log Out* untuk keluar dari akun yang sedang digunakan. Fitur ini membantu *user* untuk mengenali identitas akun yang digunakan dan memastikan keamanan sesi *login*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

(i) Perancangan fitur Detail Kunjungan

Gambar 3.11. Menggambarkan *flowchart* terkait fitur Detail Kunjungan.

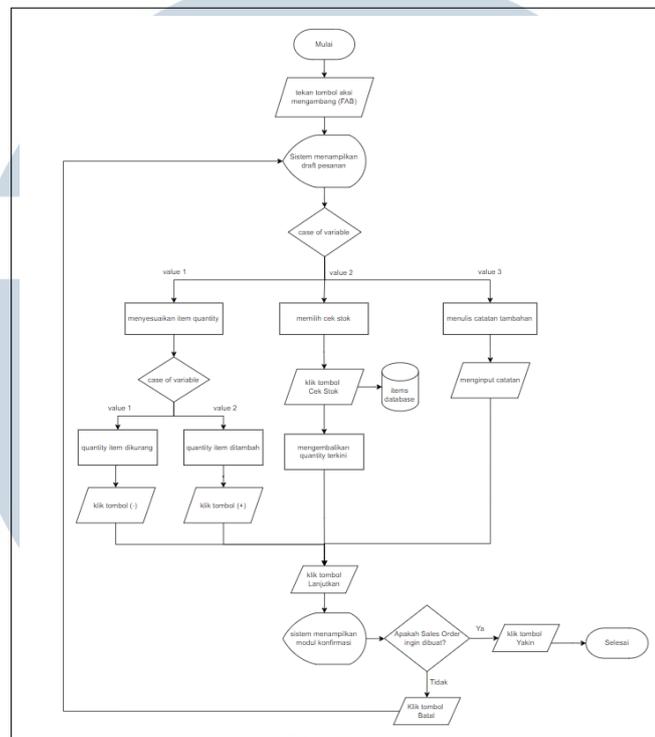


Gambar 3.11. *Flowchart* fitur *Detail Kunjungan*

Fitur ini digunakan setelah *user* sedang melakukan kunjungan ke toko tertentu dan telah melakukan pemesanan. Pada halaman *Detail Kunjungan*, informasi terkait toko yang dikunjungi, waktu kunjungan, serta detail pesanan ditampilkan secara lengkap. Hal ini membantu user dan manajemen untuk memantau aktivitas penjualan secara transparan dan terstruktur.

(j) Perancangan fitur *Add to Cart Item*

Gambar 3.12. Menggambarkan *flowchart* terkait fitur *Add to Cart Item*.



Gambar 3.12. *Flowchart* fitur *Add to Cart Item*

Fitur ini memungkinkan *user* untuk menambahkan produk ke dalam keranjang pembelian sebelum melakukan konfirmasi akhir. Dengan memilih produk dan menentukan jumlahnya, *user* secara langsung membentuk draft *sales order*. Aksi ini menandakan bahwa *user* telah memverifikasi pilihan produknya dan siap melanjutkan ke proses penentuan pengajuan pesanan.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.3.2 Rancangan Tampilan Antarmuka Aplikasi

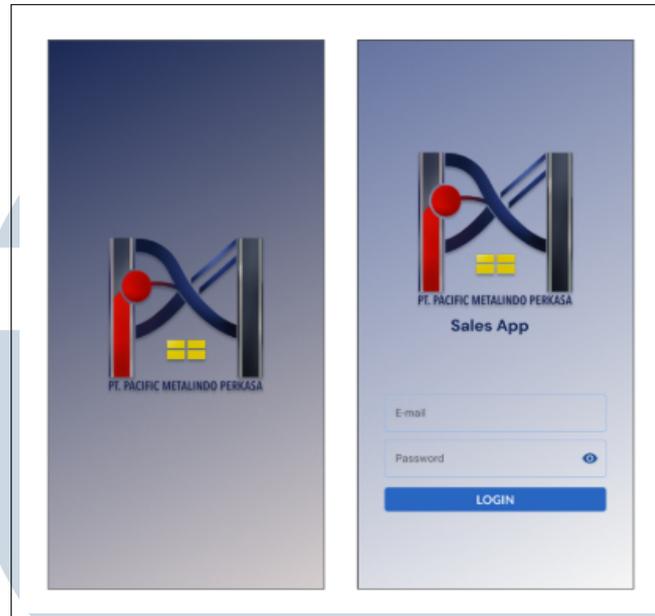
Setelah tahap perancangan sistem dan penyusunan prototipe dilakukan, langkah selanjutnya adalah implementasi desain antarmuka pengguna (User Interface) ke dalam bentuk visual yang representatif. Desain ini dibuat menggunakan *Figma*, sebuah *platform* desain berbasis *web* yang memungkinkan pengembangan dapat dilakukan secara efisien [20]. Implementasi ini bertujuan untuk menggambarkan tampilan awal sistem sebagaimana yang akan digunakan oleh user, sebelum diubah ke dalam bentuk kode program.

Sebagaimana dijelaskan dalam penelitian yang dilakukan oleh Warasi et al. [6], perancangan antarmuka pengguna yang intuitif dan efisien sangat penting untuk mendukung keberhasilan sistem informasi berbasis digital, terutama dalam konteks aplikasi penjualan. Oleh karena itu, penelitian berfokus untuk mengimplementasikan desain yang tidak hanya menarik secara visual, tetapi juga memperhatikan kemudahan navigasi dan pengalaman pengguna secara menyeluruh.

Berikut ditampilkan beberapa potongan desain UI yang telah dirancang, mencakup berbagai fitur inti dalam aplikasi, seperti halaman *login*, *homescreen*, *katalog*, profil, notifikasi, kunjungan *sales*, *detail* kunjungan *sales* dan *detail* pelanggan baru.



Pada Gambar 3.13 ditampilkan *design* untuk halaman *Login Page*.

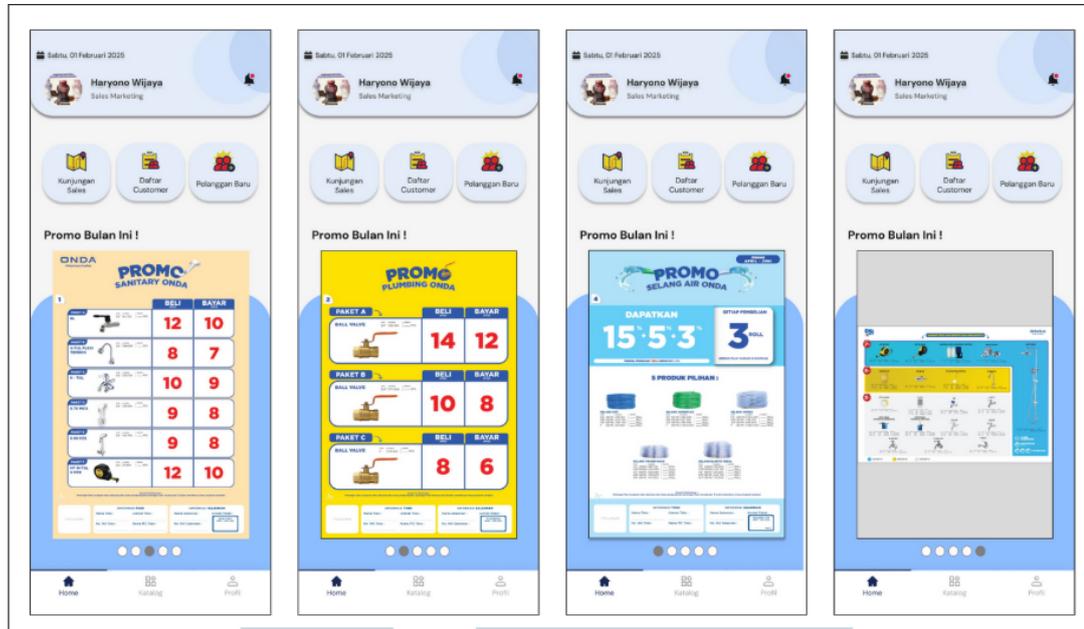


Gambar 3.13. *Design Login page*

Saat pertama kali, sistem akan menampilkan *splash screen* lalu *user* akan langsung ternavigasi ke halaman *Login*. Pada halaman ini, *user* perlu melakukan *input* pada bagian *email* dan *password* yang sudah di registrasikan melalui *backend*. Ketika *user credential valid*, maka *user* tersebut akan dinavigasi ke halaman selanjutnya yaitu Beranda.

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.14 ditampilkan *design* untuk halaman *Homescreen*.

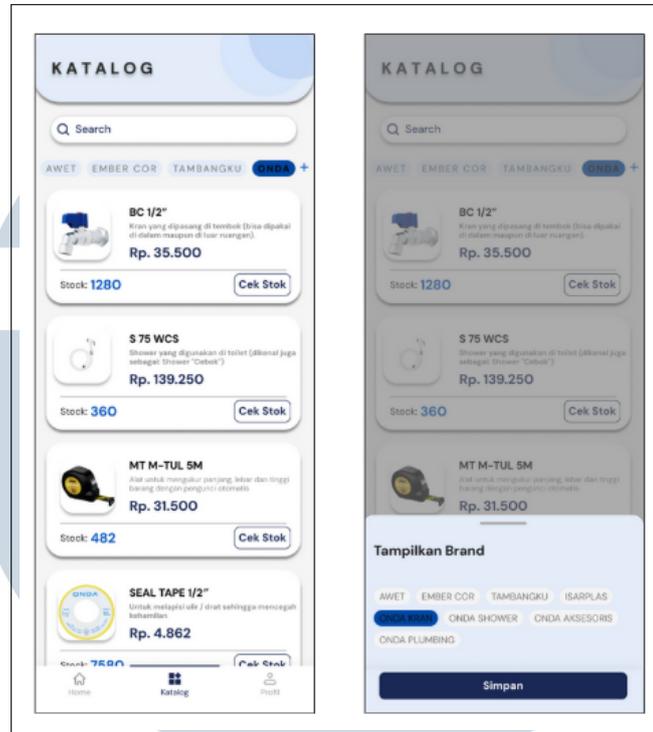


Gambar 3.14. *Design Homescreen page*

Pada halaman Beranda, terdapat beberapa menu yang bisa diakses melalui halaman berikut, yaitu Kunjungan *Sales*, Daftar *Customer*, dan Pelanggan Baru. Pada bagian *bottom navigation bar*, terdapat halaman Beranda itu sendiri, Katalog dan Profil. Terdapat *Carousel* sebagai komponen navigasi yang menampilkan konten promo yang sedang berjalan.



Pada Gambar 3.15 ditampilkan *design* untuk halaman *Katalog*.

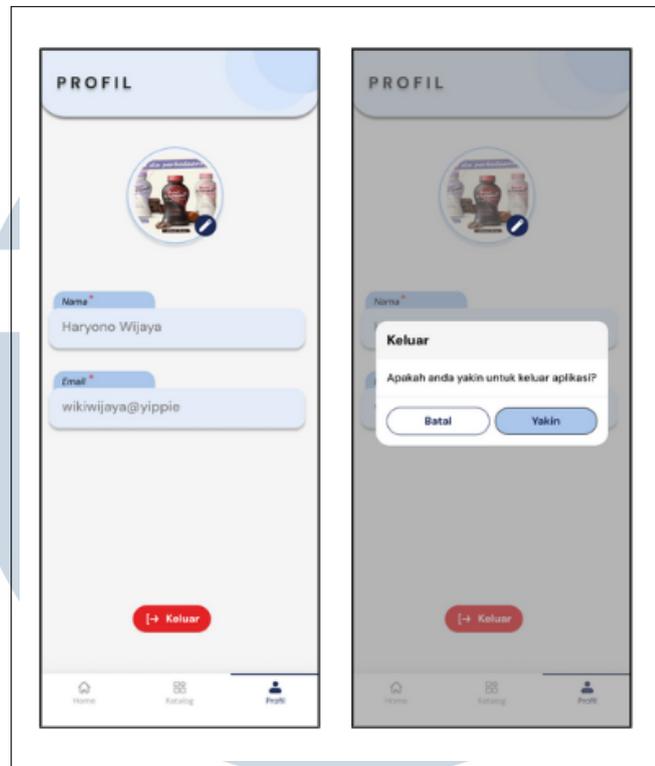


Gambar 3.15. *Design Katalog page*

Ketika *user* melakukan akses pada fitur *Katalog*, sistem akan menampilkan produk-produk yang dijual serta terdapat pengelompokan berdasarkan merk dari masing-masing produk. Jika *user* memiliki kesulitan untuk mencari produk, *user* dapat langsung mencari melalui *search bar* yang ada. Melalui fitur *Katalog*, *user* dapat mengecek data stok secara *realtime* dengan menekan tombol "Cek Stok". Sistem akan mengembalikan permintaan *user* dengan menampilkan *quantity* produk berdasarkan data yang ada pada *backend*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.16 ditampilkan *design* untuk halaman *Profile*.

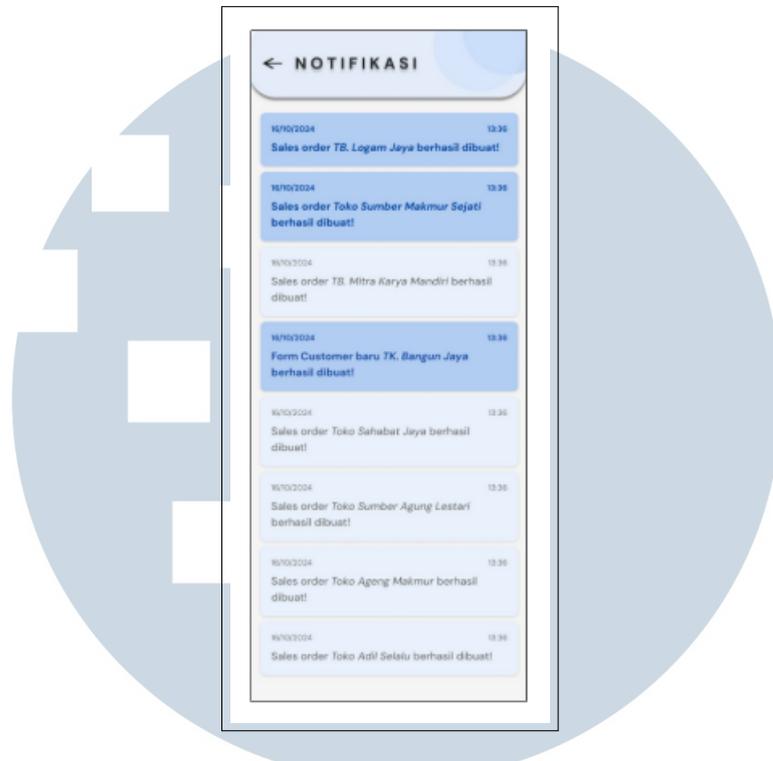


Gambar 3.16. *Design Profile page*

Fitur Profil menampilkan informasi pribadi *user* yang sedang aktif di dalam sistem. Informasi yang ditampilkan meliputi nama *user*, alamat *email*, serta terdapat tombol *Log Out* untuk keluar dari akun yang sedang digunakan. Fitur ini membantu *user* untuk mengenali identitas akun yang digunakan dan memastikan keamanan sesi *login*.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.17 ditampilkan *design* untuk halaman Notifikasi.

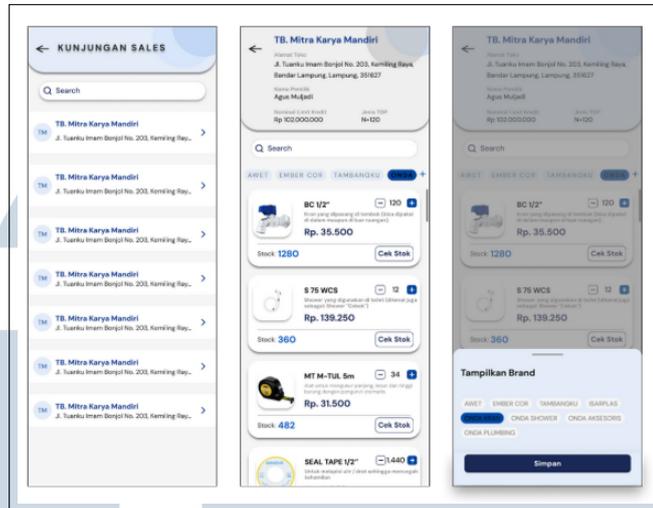


Gambar 3.17. *Design* Notifikasi page

Pada halaman fitur notifikasi, sistem menampilkan notifikasi atas dokumen *Sales Order* atau dokumen *form* Pelanggan Baru yang telah berhasil dibuat pada hari tersebut. *Font* dan *card* pada halaman notifikasi dapat berubah warna menjadi lebih gelap jika notifikasi tersebut telah terbuka atau terbaca oleh *user*.

U M M N
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada Gambar 3.18 ditampilkan *design* untuk halaman Kunjungan Sales.

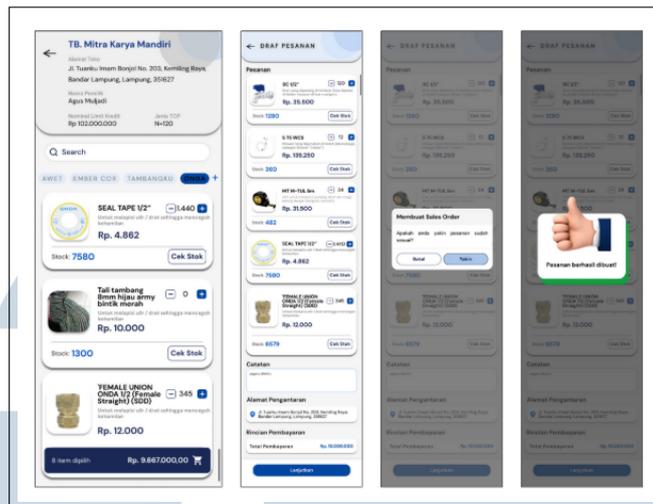


Gambar 3.18. *Design* Kunjungan Sales page

Ketika *user* akses pada fitur Kunjungan Sales, sistem list toko yang akan dikunjungi oleh user pada hari tersebut. Setelah *user* memilih toko yang sedang dikunjungi, sistem akan menampilkan produk-produk yang dijual serta terdapat pengelompokan berdasarkan merk dari masing-masing produk. Jika user memiliki kesulitan untuk mencari produk, user dapat langsung mencari melalui *search bar* yang ada. Melalui fitur *Katalog*, user dapat mengecek data stok secara *realtime* dengan menekan tombol "Cek Stok". Sistem akan mengembalikan permintaan user dengan menampilkan *quantity* produk berdasarkan data yang ada pada *backend*. Pada halama ini, *user* dapat menambahkan dan mengurangi *quantity* yang akan diinput kedalam *sales order*.

Pada Gambar 3.19 ditampilkan *design* untuk halaman *Detail* Kunjungan Sales.

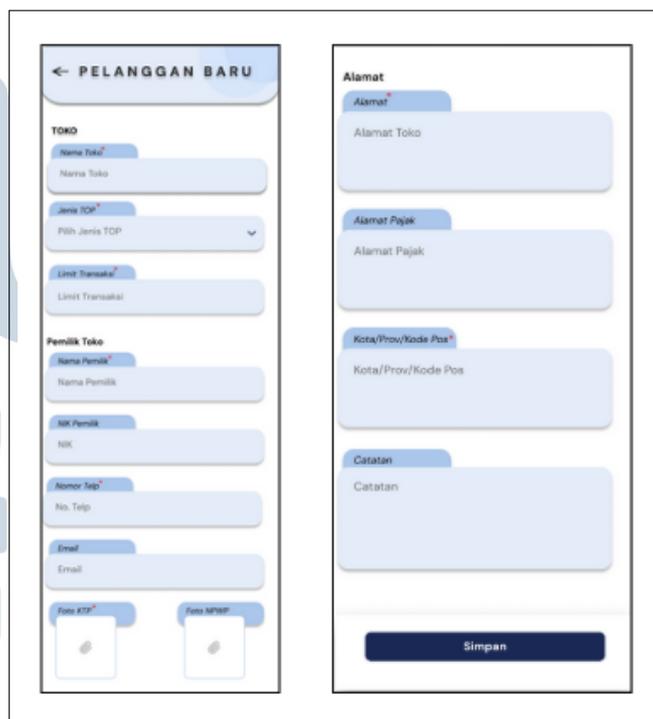
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.19. *Design Detail Kunjungan Sales page*

Fitur menampilkan *draft* pesanan setelah *user* menginput produk. Pada halaman *Detail Kunjungan*, draft pesanan menampilkan berupa detail produk yang telah diinput lalu terdapat tombol sebagai konfirmasi pembuatan *sales order* tersebut.

Pada Gambar 3.20 ditampilkan *design* untuk halaman Pelanggan Baru.



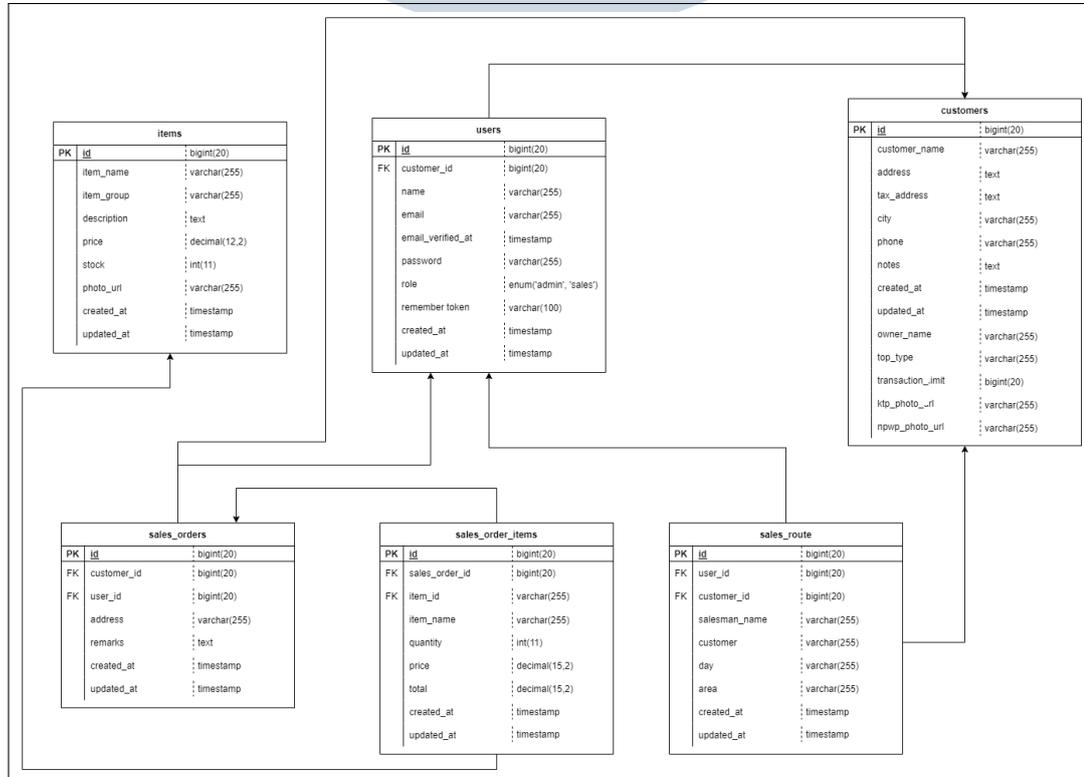
Gambar 3.20. *Design Pelanggan Baru page*

Pada fitur Pelanggan Baru, *user* dapat melakukan daftar atau registrasi untuk toko baru yang ingin didaftarkan sebagai *customer* pada perusahaan. Setelah *user* telah berhasil menginput dokumen *form* Pelanggan Baru, maka *backend* akan mengirimkan kembali kepada *frontend* sebuah notifikasi bahwa dokumen tersebut telah berhasil terbuat.

3.3.3 Rancangan Database

Database ini dirancang untuk menyimpan data utama seperti informasi pengguna (tabel *users*), pelanggan (*customers*), barang atau produk (*items*), serta data transaksi penjualan yang tercatat dalam tabel *sales_orders* dan *sales_order_items*. Selain itu, untuk mendukung pengelolaan aktivitas lapangan, terdapat tabel *sales_route* yang merekam rute kunjungan sales ke outlet. Relasi antar tabel menggunakan pendekatan *relational database* dengan pemanfaatan *foreign key*.

Berikut ditampilkan rancangan *database* yang digunakan pada aplikasi:



Gambar 3.21. Rancangan *database*

Tabel *users* menyimpan informasi pengguna sistem, termasuk *sales* dan *admin*, dengan atribut seperti *name*, *email*, dan *role* yang dibatasi oleh *enum* (*admin*, *sales*). Tabel ini memiliki hubungan *one-to-many* dengan tabel *sales_route* karena setiap *user* (*sales*) dapat memiliki lebih dari satu rute kunjungan.

Tabel *items* merepresentasikan data produk yang tersedia, dengan atribut seperti *item_name*, *item_group*, *description*, *price*, *stock*, dan *photo_url*. Data pada tabel ini digunakan dalam transaksi penjualan melalui relasi *one-to-many* ke tabel *sales_order_items*.

Tabel *sales_orders* berfungsi mencatat transaksi pesanan penjualan yang dilakukan oleh *sales* kepada pelanggan. Setiap entri pada tabel ini berelasi dengan satu pengguna (*salesman_id*) dan satu pelanggan (*customer_id*), serta memiliki atribut tambahan seperti *address* dan *remarks/ catatan*.

Untuk detail barang dalam setiap transaksi, digunakan tabel *sales_order_items*, yang merekam item apa saja yang dipesan pada suatu *sales_order*, lengkap dengan atribut *quantity*, *price*, dan *total*. Tabel ini memiliki *foreign key sales_order_id* yang menunjuk ke tabel *sales_orders*, dan *item_code* yang mengacu pada item yang dipesan.

Terakhir, tabel *sales_route* digunakan untuk mencatat rute kunjungan *sales* ke *customer* berdasarkan kombinasi *user_id*, *customer_id*, dan informasi pendukung seperti hari kunjungan, area, dan nama *salesman*. Hal ini memungkinkan perusahaan memantau aktivitas lapangan *sales* serta mengevaluasi efektivitas rute kunjungan.

Seluruh entitas dalam rancangan ini menggunakan atribut *created_at* dan *updated_at* untuk mencatat waktu penciptaan dan pembaruan data, guna mendukung fitur *log* perubahan dan *audit trail*. Relasi antar tabel menggunakan *foreign key* untuk menjaga integritas data, serta mendukung kebutuhan integrasi modul yang efisien dalam aplikasi *mobile*.