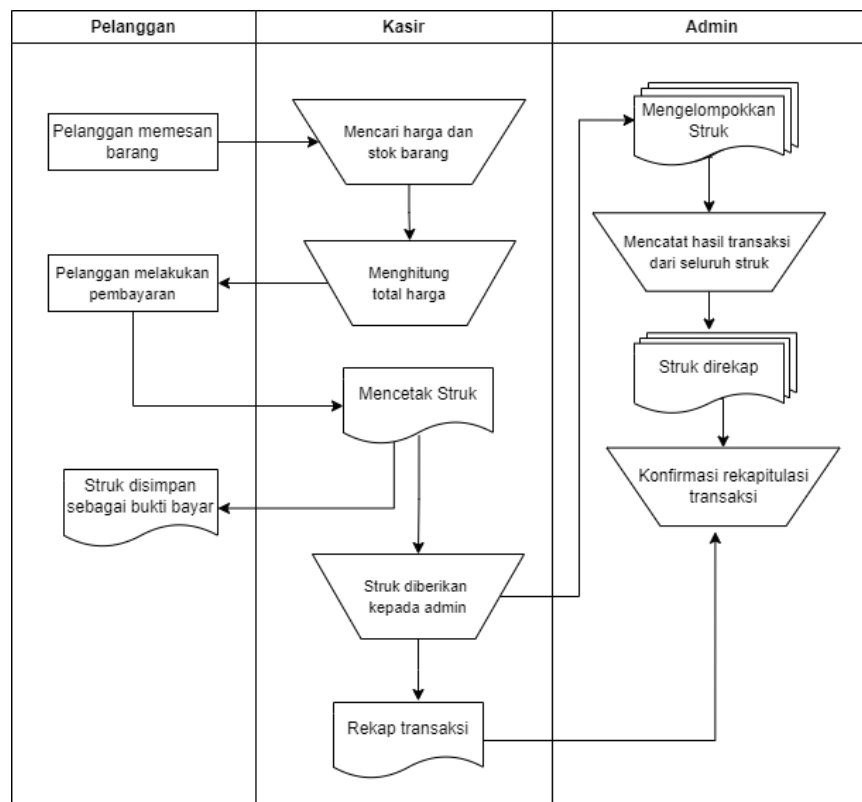


BAB IV

ANALISA DAN PERANCANGAN

4.1. Analis Sistem

4.1.1. Sistem yang sedang berjalan



Gambar 4. 1 Alur sistem yang sedang berjalan

Pada sistem yang berjalan saat ini, alur pembayaran menggunakan *QRIS* konvensional di Toko KIDO, BW *Coffee*, dan Arisa *Pet Shop* memiliki beberapa langkah penting yang melibatkan interaksi antara pelanggan dan kasir serta pencatatan manual. Berikut adalah penjelasan rinci alur tersebut:

Pertama, pelanggan yang ingin melakukan pembayaran meminta kode *QRIS* kepada kasir. Kasir di Toko KIDO, BW *Coffee*, dan Arisa *Pet Shop*

menunjukkan kode *QRIS* yang telah dicetak pada kertas. Kode *QRIS* ini biasanya diletakkan di meja kasir atau tempat yang mudah dijangkau oleh pelanggan. Dalam wawancara dengan Toko KIDO, diketahui bahwa mereka mencatat transaksi secara manual menggunakan buku kas selain menggunakan *QRIS* konvensional. *BW Coffee* dan *Arisa Pet Shop* juga menggunakan metode serupa dengan mengandalkan pencatatan manual di buku kas.

Setelah pelanggan menerima kode *QRIS*, mereka membuka aplikasi *e-wallet* atau *mobile banking* yang mereka gunakan, kemudian memindai kode *QRIS* tersebut. Pada tahap ini, pelanggan memasukkan jumlah pembayaran sesuai dengan total transaksi yang harus dibayar. Selanjutnya, pelanggan mengonfirmasi transaksi di aplikasi mereka.

Setelah transaksi dikonfirmasi oleh pelanggan, kasir mencatat transaksi tersebut secara manual di buku kas atau *spreadsheet*. Proses pencatatan manual ini dilakukan untuk memastikan bahwa semua transaksi tercatat dengan benar. Namun, metode ini memiliki beberapa kendala, seperti yang diungkapkan oleh para pelaku usaha dalam wawancara. Kendala utama yang dihadapi adalah risiko kehilangan atau kerusakan dokumen kertas, yang bisa menyebabkan kesulitan dalam pelacakan transaksi dan klaim pembayaran. Selain itu, proses pencatatan manual memakan waktu dan rawan kesalahan, terutama ketika bukti pembayaran kertas hilang atau rusak.

Para pemilik usaha di Toko KIDO, *BW Coffee*, dan *Arisa Pet Shop* juga menyampaikan bahwa mereka sering menghadapi masalah keamanan terkait transaksi. Kehilangan atau kerusakan bukti pembayaran kertas dapat menyebabkan transaksi yang tidak sah atau kehilangan data, sehingga sulit untuk

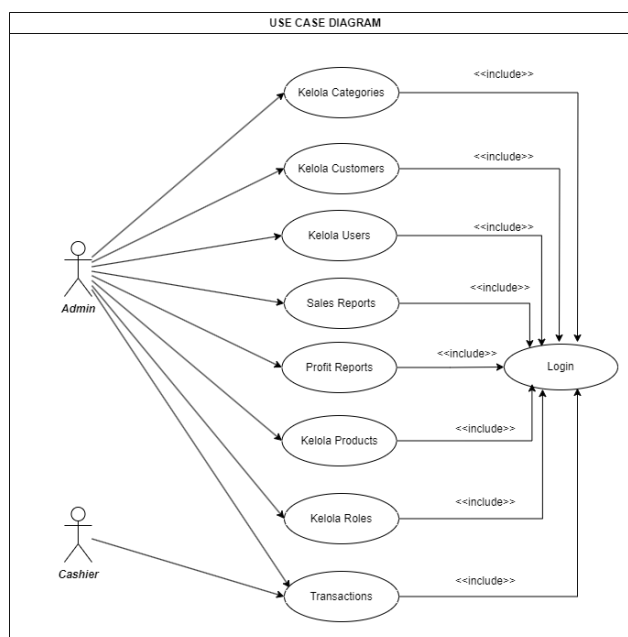
memverifikasi transaksi tersebut. Dalam wawancara, mereka mengungkapkan bahwa penggunaan *QRIS* konvensional yang dicetak kurang efisien dan praktis karena dapat rusak atau hilang. Mereka melihat potensi besar dalam penggunaan *Midtrans Payment Gateway* sebagai alternatif metode pembayaran yang lebih aman dan efisien.

Secara keseluruhan, alur pembayaran menggunakan *QRIS* konvensional di Toko KIDO, *BW Coffee*, dan *Arisa Pet Shop* melibatkan beberapa langkah penting yang membutuhkan interaksi manual dan pencatatan yang teliti. Namun, metode ini memiliki beberapa kelemahan yang dapat diatasi dengan implementasi sistem *POS* berbasis *website* yang terintegrasi dengan *Payment Gateway QRIS* dan sistem monitoring untuk meningkatkan efisiensi dan keamanan transaksi.

4.1.2. Sistem yang diusulkan

Sistem yang diusulkan adalah sistem *POS* berbasis *web* yang terintegrasi dengan *Midtrans Payment Gateway* dan dilengkapi dengan fitur monitoring transaksi secara *real-time* untuk meningkatkan efisiensi dan keamanan transaksi. Sistem ini dirancang untuk mengatasi kendala yang ada pada sistem manual, seperti yang dialami oleh Toko Kido, dengan memberikan solusi sebagai berikut: mencatat transaksi secara otomatis sehingga mengurangi risiko kehilangan atau kerusakan dokumen; mempercepat proses pencatatan dan pembayaran dengan mengintegrasikan *Midtrans Payment Gateway*; dan mengurangi kesalahan manusia dalam pencatatan data transaksi.

4.1.2.1. Use Case Diagram



Gambar 4. 2 Use Cas Diagram

Use Case Diagram diatas menunjukkan interaksi antara pengguna dan sistem yang diusulkan. Berikut adalah penjelasan dari diagram tersebut:

Use Case Diagram yang dibuat menggambarkan bagaimana pengguna dapat berinteraksi dengan sistem POS berbasis *web* yang diusulkan. Dalam diagram ini, ada tiga *use case* utama: cek transaksi, proses pembayaran, dan lihat laporan. Setiap *use case* ini merepresentasikan fungsi utama yang akan disediakan oleh sistem.

4.1.2.2. Activity Diagram

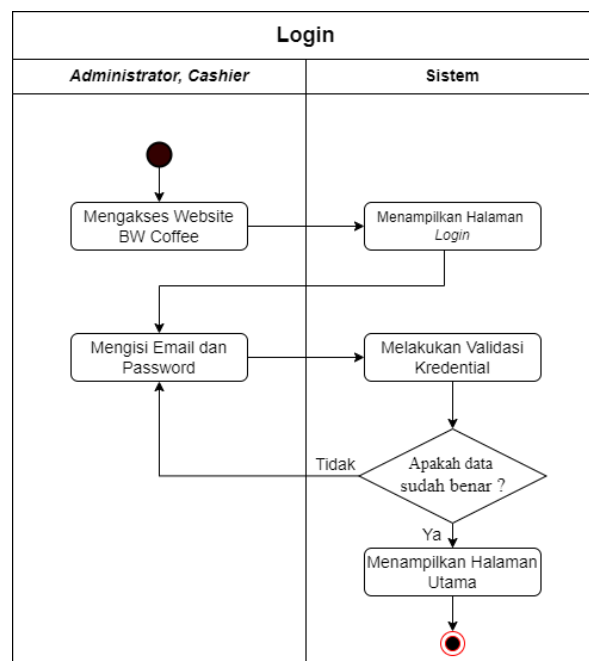
Activity Diagram menggambarkan alur aktivitas dalam sistem dari mulai proses pembayaran hingga pencatatan transaksi. Berikut adalah penjelasan dari diagram tersebut:

Activity Diagram ini menunjukkan langkah-langkah yang terjadi dalam sistem POS berbasis *web*. Alur dimulai dari pengguna membuka

aplikasi POS, memilih menu pembayaran, memasukkan detail pembayaran, dan sistem akan memvalidasi pembayaran tersebut. Jika pembayaran *valid*, sistem akan mencatat transaksi dan mengirim data ke *Midtrans Payment Gateway*. Jika tidak *valid*, sistem akan menampilkan pesan *error*.

Berikut diagram aktivitas dari setiap fitur yang ada pada *web* yang akan di rancang.

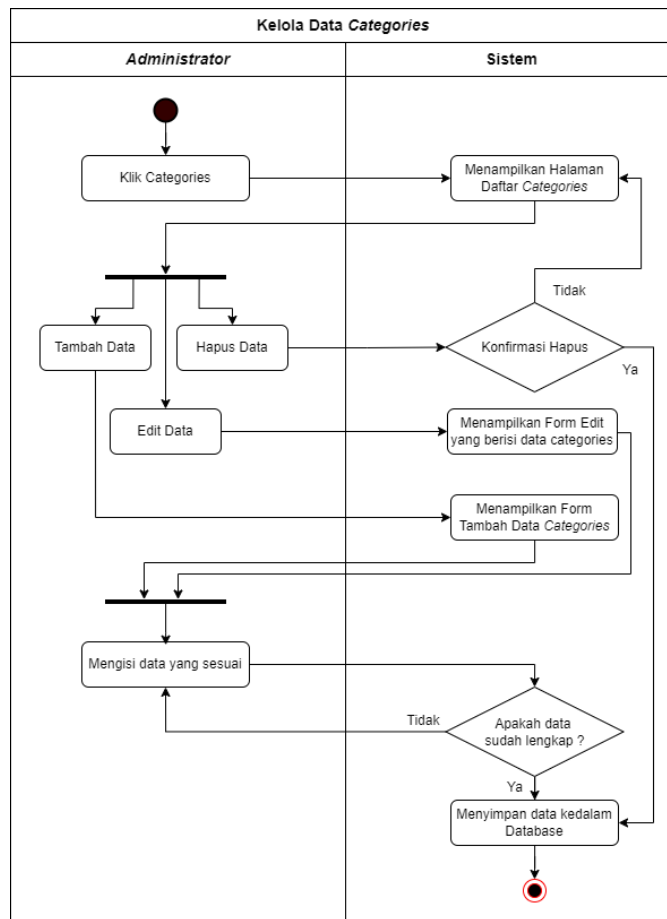
1. Activity Diagram Login



Gambar 4. 3 Activity Diagram Login

Activity diagram login menampilkan alur proses autentikasi pengguna dalam sebuah sistem. Diagram dimulai dari pengguna memasukkan *email* dan *password* ke dalam *form login*. Sistem kemudian melakukan validasi kredensial dengan mengecek kesesuaian *email* dan *password* dengan *database*. Saat data sudah benar, maka pengguna akan di alihkan ke halaman utama.

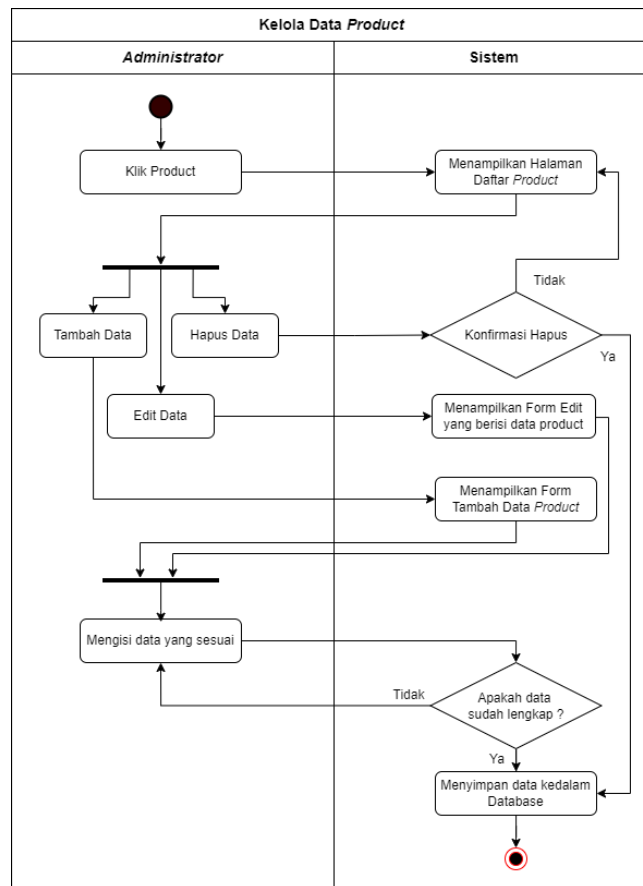
2. Activity Diagram Kelola Kategori



Gambar 4. 4 Activity Diagram Kelola Kategori

Activity diagram kelola kategori menggambarkan proses pengelolaan kategori dalam sistem. Proses dimulai ketika admin memilih opsi "Categories" dari menu. Admin kemudian dapat memilih untuk menambahkan, mengubah, atau menghapus kategori. Diagram ini menunjukkan alur interaksi mulai dari pengelolaan hingga pembaruan data kategori di sistem.

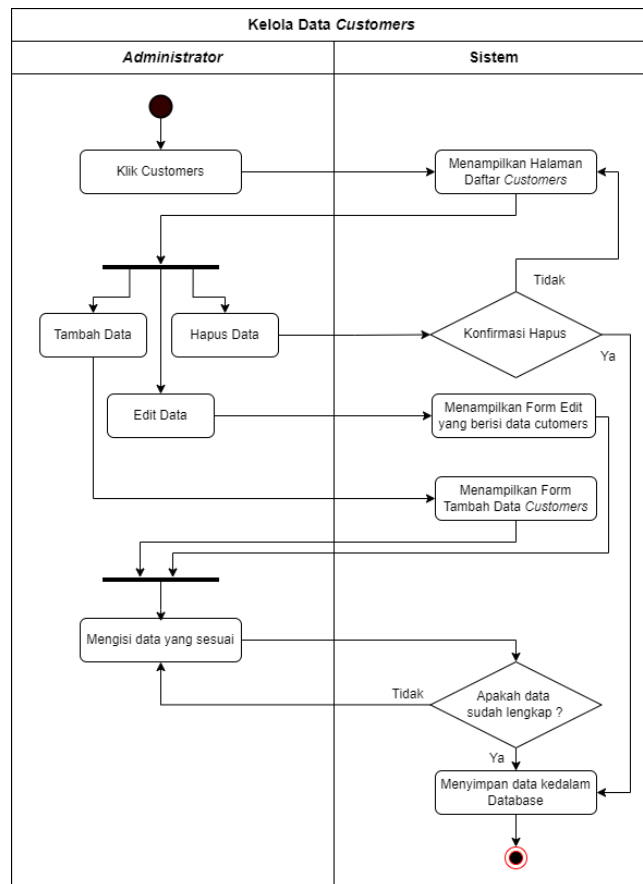
3. Activity Diagram Kelola Products



Gambar 4. 5 Activity Diagram Kelola Produk

Activity diagram kelola products menggambarkan alur pengelolaan produk dalam sebuah sistem. Proses dimulai ketika admin memilih menu "Products". Admin kemudian dapat memilih untuk menambahkan, mengubah, atau menghapus produk dari daftar. Diagram ini memetakan langkah-langkah pengelolaan produk dari awal hingga perubahan disimpan atau produk dihapus.

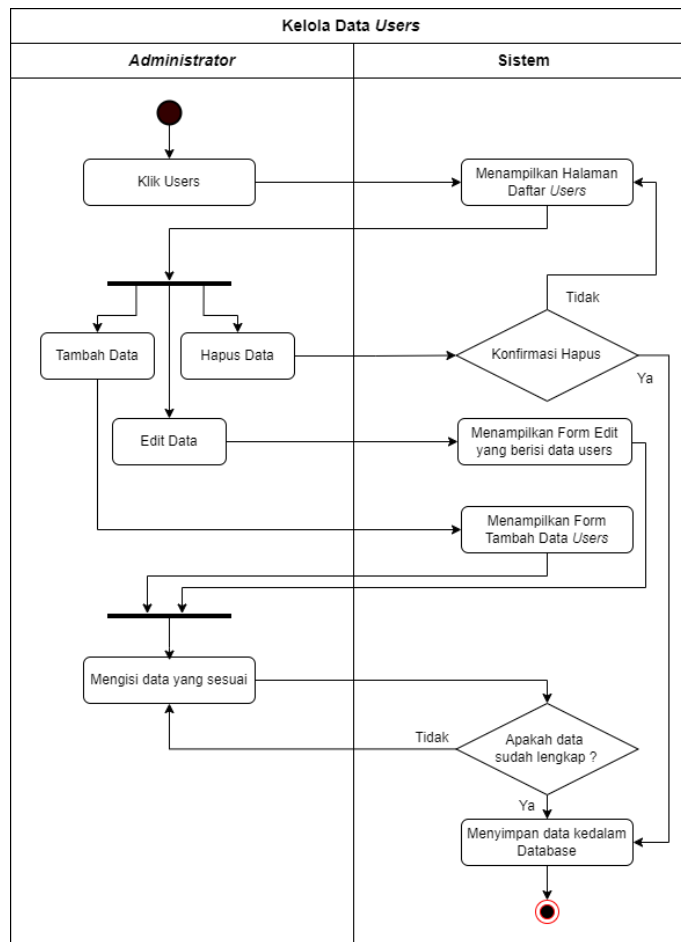
4. Activity Diagram Kelola Customers



Gambar 4. 6 Activity Diagram Kelola Customers

Activity diagram kelola customers menggambarkan alur proses pengelolaan data pelanggan dalam sebuah sistem. Proses dimulai ketika *admin* memilih opsi "*Customers*" dari menu. *Admin* dapat memilih untuk menambah, mengubah, atau menghapus data pelanggan. Diagram ini menunjukkan keseluruhan proses pengelolaan pelanggan, mulai dari penambahan hingga penghapusan data.

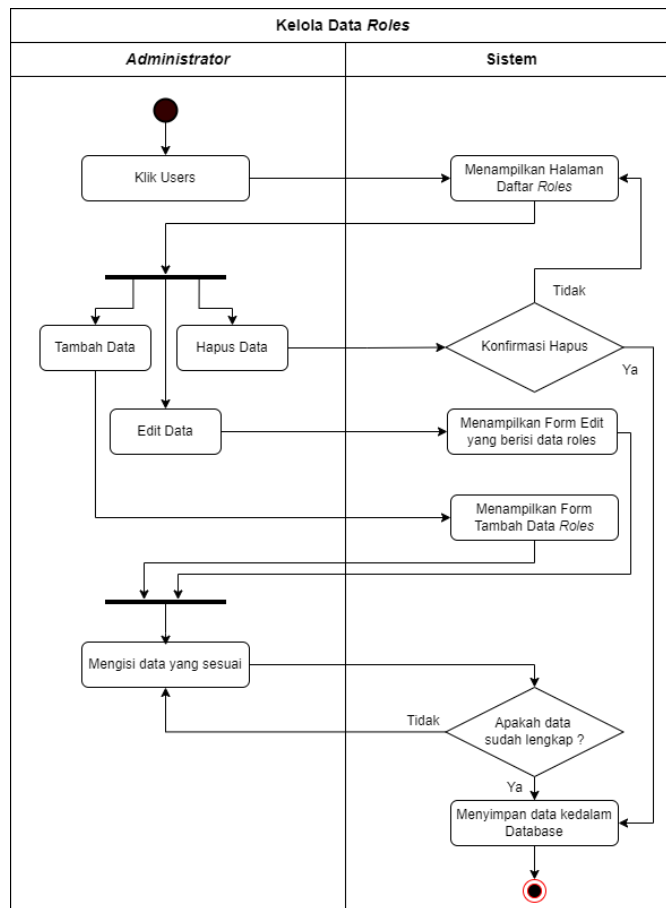
5. Activity Diagram Kelola Users



Gambar 4. 7 Activity Diagram Kelola Users

Activity diagram kelola users menggambarkan alur pengelolaan pengguna dalam sebuah sistem. Proses dimulai ketika *admin* memilih opsi "*Users*" dari menu. *Admin* kemudian dapat memilih untuk menambah, mengubah, atau menghapus pengguna. Diagram ini memetakan seluruh alur pengelolaan pengguna, termasuk penambahan, pengubahan, dan penghapusan data.

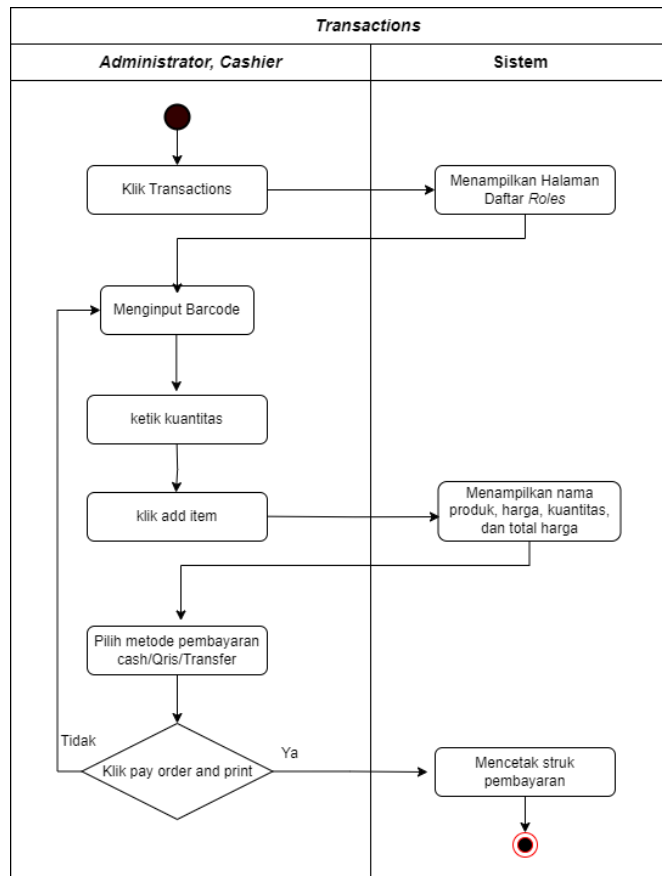
6. Activity Diagram Kelola Roles



Gambar 4. 8 Activity Diagram Kelola Roles

Activity diagram kelola roles menggambarkan alur pengelolaan peran (*roles*) dalam sebuah sistem. Proses dimulai ketika admin memilih menu "*Roles*". Admin kemudian dapat memilih untuk menambahkan, mengubah, atau menghapus peran. Diagram ini menggambarkan langkah-langkah dalam pengelolaan peran mulai dari penambahan, pengubahan, hingga penghapusan peran dalam sistem.

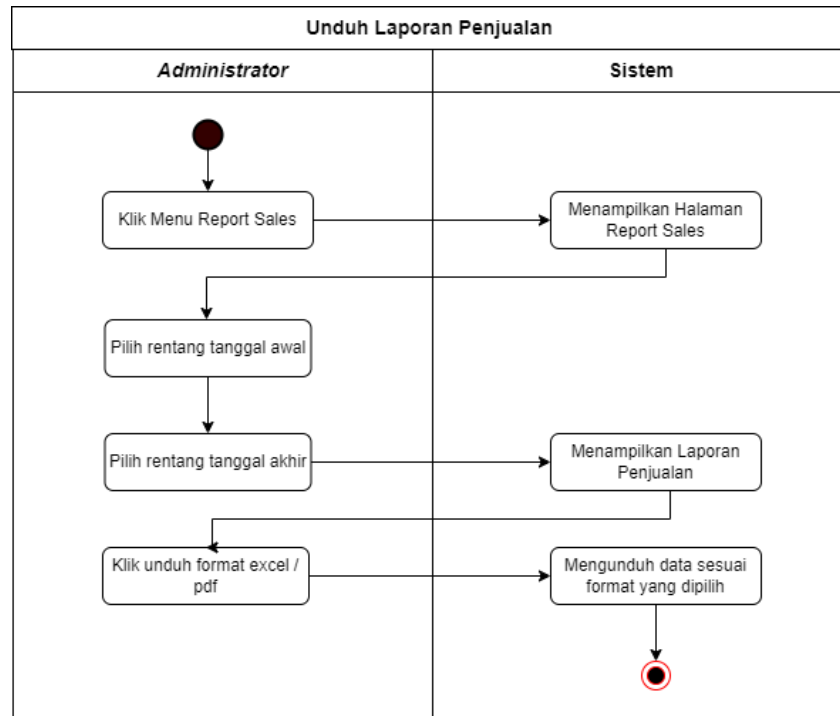
7. Activity Diagram Transaksi



Gambar 4. 9 Activity Diagram Transaksi

Activity diagram halaman transaksi menggambarkan alur proses transaksi dalam sebuah sistem. Proses dimulai ketika pengguna membuka halaman transaksi, di mana pengguna dapat memulai transaksi baru. pengguna harus mengisi detail transaksi seperti produk, jumlah, metode pembayaran, dan informasi pelanggan. Setelah diisi, sistem memvalidasi data, seperti ketersediaan produk dan kecukupan stok. Jika *valid*, transaksi diproses dan disimpan ke dalam *database*, lalu sistem akan mencetak struk.

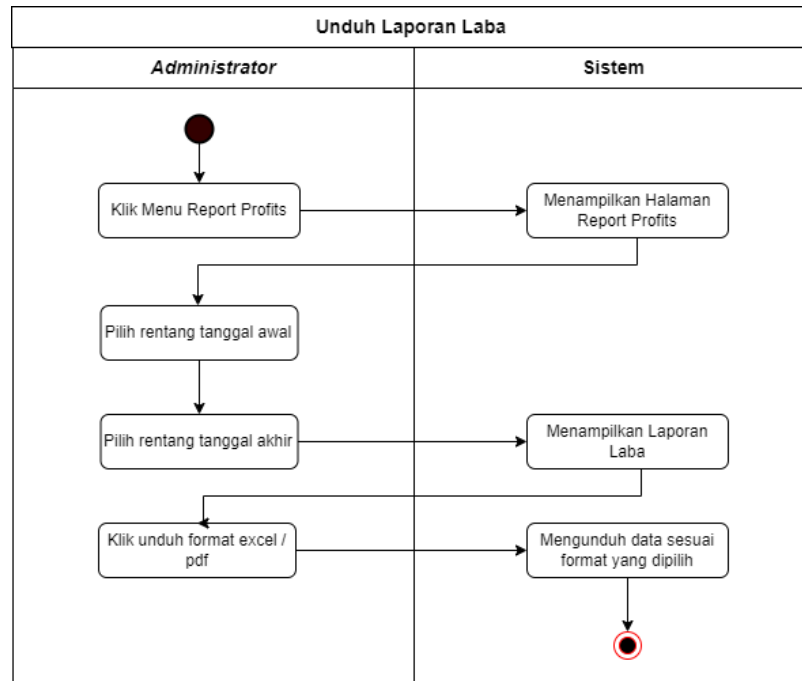
8. Activity Diagram Laporan Penjualan



Gambar 4. 10 Activity Diagram Laporan Penjualan

Activity diagram laporan penjualan menggambarkan alur menampilkan laporan penjualan dalam sebuah sistem. Proses dimulai ketika *admin* membuka halaman "*Report Sales*". *Admin* dapat memilih rentang waktu untuk laporan yang ingin ditampilkan. Sistem kemudian mengambil data penjualan dari *database* sesuai dengan periode yang dipilih dan menghasilkan laporan. Setelah laporan dihasilkan, *admin* dapat memilih mencetak laporan, atau mengunduhnya dalam format tertentu seperti PDF atau *Excel*.

9. Activity Diagram Laporan Laba



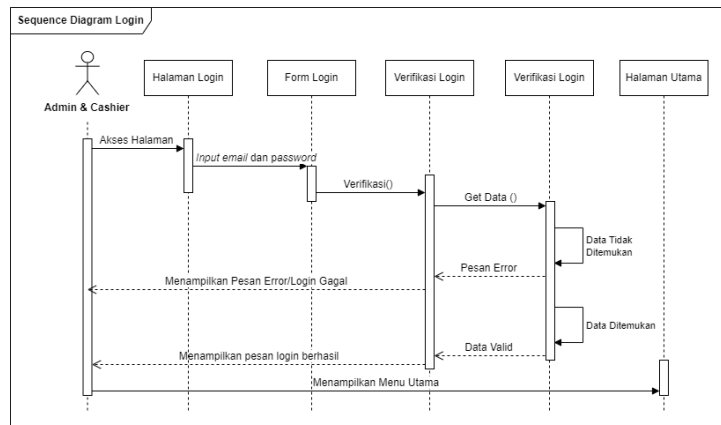
Gambar 4. 11 Activity Diagram Unduh Laporan Laba

Activity diagram laporan laba menggambarkan alur menampilkan laporan laba dalam sebuah sistem. Proses dimulai ketika *admin* membuka halaman "*Report Profits*". *Admin* dapat memilih rentang waktu untuk laporan yang ingin ditampilkan. Sistem kemudian mengambil data laba dari *database* sesuai dengan periode yang dipilih dan menghasilkan laporan. Setelah laporan dihasilkan, *admin* dapat memilih mencetak laporan, atau mengunduhnya dalam format tertentu seperti PDF atau *Excel*.

4.1.2.3. Sequence Diagram

Sequence Diagram menggambarkan urutan interaksi antara pengguna dan sistem dalam proses pembayaran. Berikut adalah penjelasan dari diagram tersebut:

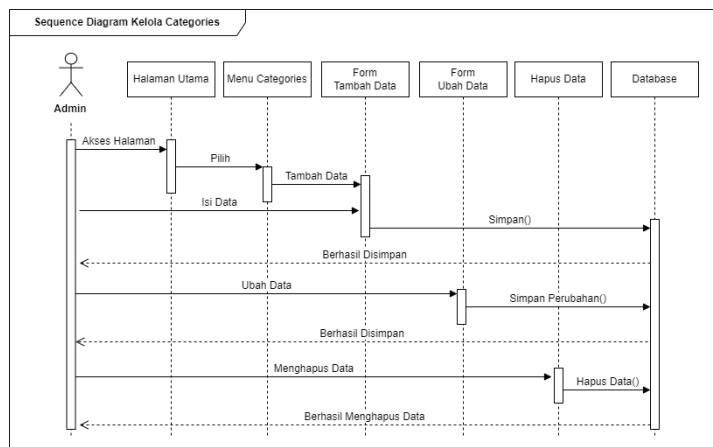
1. Sequence Diagram Login



Gambar 4. 12 Sequence Diagram Login

Sequence diagram login menggambarkan interaksi antara pengguna, antarmuka sistem, dan *server* selama proses autentikasi. Proses dimulai ketika pengguna memasukkan *email* dan *password* pada *form login* di halaman *login*. Antarmuka sistem kemudian mengirimkan kredensial tersebut ke *server* untuk validasi. Jika data sudah benar, maka pengguna akan dialihkan ke halaman utama.

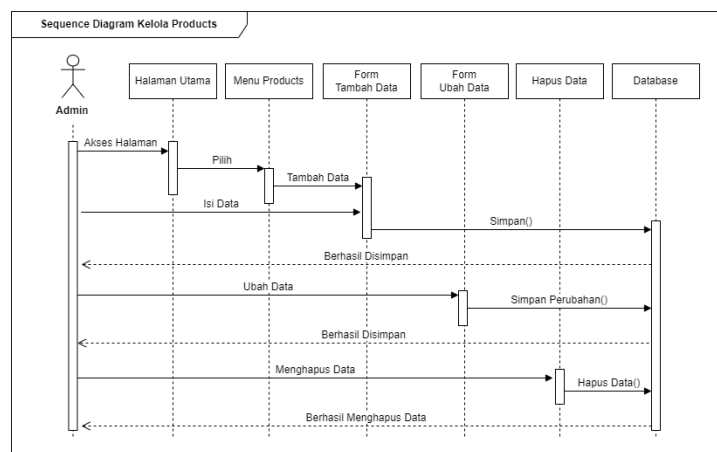
2. Sequence Diagram Kelola Categories



Gambar 4. 13 Sequence Diagram Kelola Categories

Sequence diagram kelola *categories* menggambarkan interaksi antara *admin*, antarmuka sistem, dan *server* saat mengelola kategori. Proses dimulai ketika *admin* memilih menu "*Categories*" pada antarmuka sistem. *Admin* kemudian dapat memilih untuk menambah, mengubah, atau menghapus kategori. Diagram ini menunjukkan alur komunikasi dari tindakan *admin* hingga pembaruan atau penghapusan kategori di sistem.

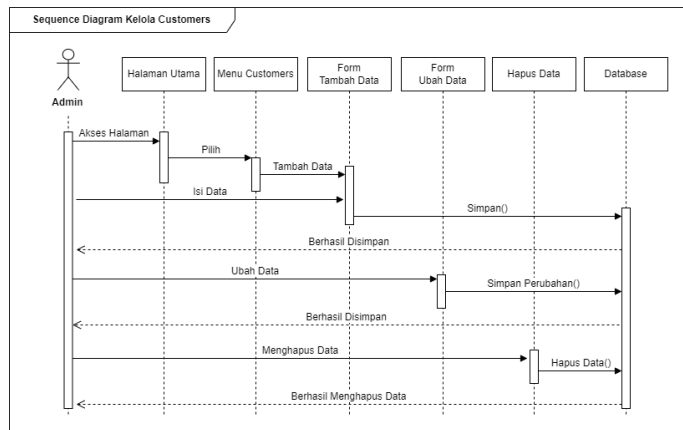
3. *Sequence Diagram Kelola Products*



Gambar 4. 14 *Sequence Diagram Kelola Products*

Sequence diagram kelola *products* menggambarkan interaksi antara *admin*, antarmuka sistem, dan *server* dalam proses pengelolaan produk. Proses dimulai ketika *admin* memilih menu "*Products*" pada antarmuka sistem. *Admin* kemudian dapat memilih untuk menambah, mengubah, atau menghapus produk. Diagram ini menunjukkan alur komunikasi dari tindakan *admin* hingga pembaruan atau penghapusan produk di sistem.

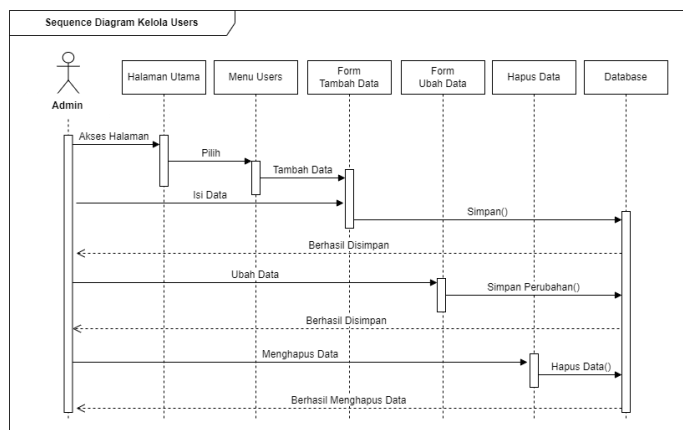
4. Sequence Diagram Kelola Customers



Gambar 4. 15 Sequence Diagram Kelola Customers

Sequence diagram kelola customers menggambarkan alur interaksi antara *admin*, antarmuka sistem, dan *server* dalam pengelolaan data pelanggan. Proses dimulai ketika *admin* mengakses menu "Customers" di antarmuka sistem. *Admin* dapat memilih untuk menambah, mengubah, atau menghapus data pelanggan. Diagram ini menggambarkan alur komunikasi dari tindakan *admin* hingga pembaruan atau penghapusan data pelanggan dalam sistem.

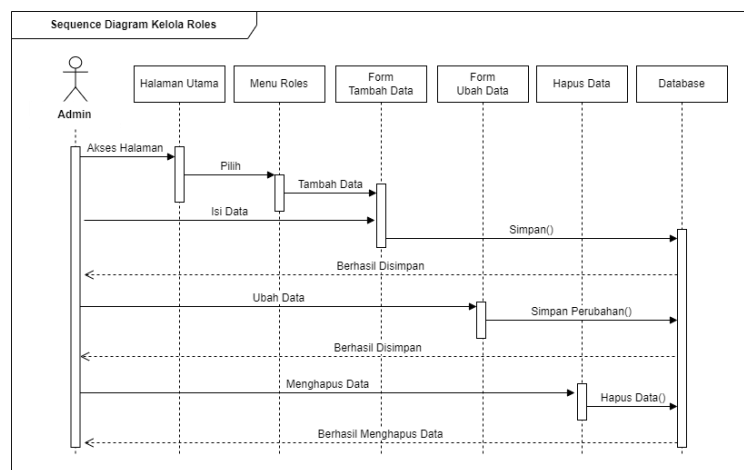
5. Sequence Diagram Kelola Users



Gambar 4. 16 Sequence Diagram Kelola Users

Sequence diagram kelola *users* menggambarkan proses pengelolaan data pengguna antara *admin*, antarmuka sistem, dan *server*. Proses dimulai saat *admin* mengakses menu "*Users*" di antarmuka sistem. *Admin* dapat memilih untuk menambah, mengubah, atau menghapus pengguna. Diagram ini menunjukkan alur komunikasi dari tindakan *admin* hingga perubahan data pengguna dalam sistem.

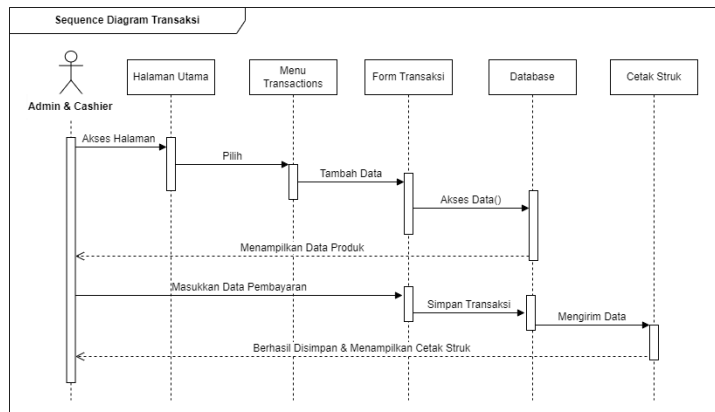
6. *Sequence Diagram Kelola Roles*



Gambar 4. 17 *Sequence Diagram Kelola Roles*

Sequence diagram kelola *roles* menggambarkan interaksi antara *admin*, antarmuka sistem, dan *server* dalam pengelolaan peran. Proses dimulai ketika *admin* membuka menu "*Roles*" di antarmuka sistem. *Admin* dapat memilih untuk menambah, mengubah, atau menghapus peran. Diagram ini menggambarkan alur komunikasi dari tindakan *admin* hingga perubahan atau penghapusan peran dalam sistem.

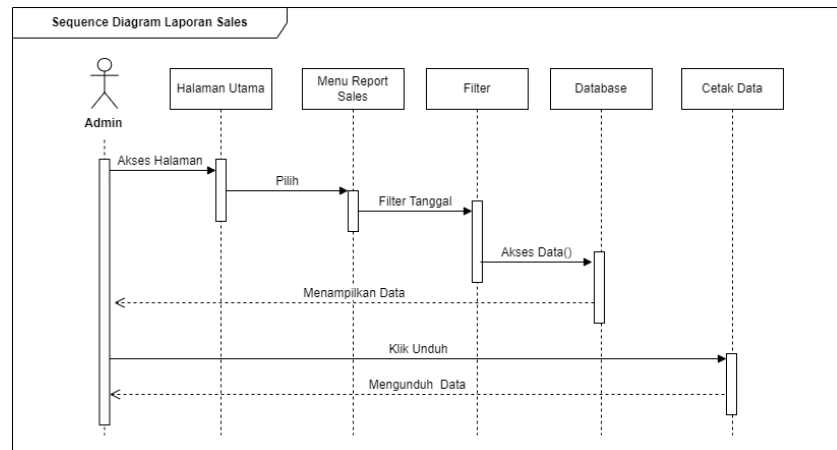
7. Sequence Diagram Transaksi



Gambar 4. 18 *Sequence Diagram* Transaksi

Sequence diagram transaksi menggambarkan alur interaksi antara pengguna, antarmuka sistem, dan *server* dalam proses transaksi. Proses dimulai ketika pengguna membuka halaman transaksi dan memulai transaksi baru. *Admin* Pengguna harus mengisi detail transaksi seperti produk, jumlah, dan metode pembayaran. Setelah pengguna mengisi *form* dan mengirimkan data, antarmuka sistem mengirimkan permintaan transaksi ke *server*. *Server* memproses data, memvalidasi informasi seperti ketersediaan produk dan metode pembayaran, lalu menyimpan transaksi ke *database*. Setelah proses selesai, *server* mengirimkan konfirmasi transaksi dan status pembayaran kembali ke antarmuka sistem.

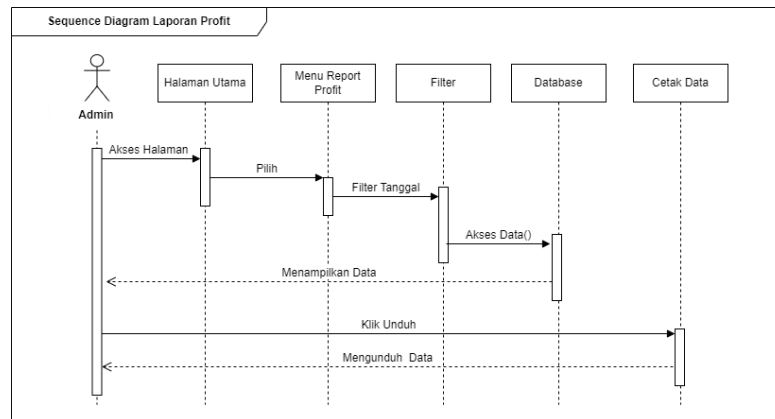
8. Sequence Diagram Laporan Sales



Gambar 4. 19 Sequence Diagram Laporan Sales

Sequence diagram laporan sales menggambarkan alur menampilkan laporan penjualan. Proses dimulai ketika *admin* membuka halaman "*Report Sales*" di antarmuka sistem dan memilih rentang waktu laporan yang diinginkan. Antarmuka sistem mengirimkan permintaan data berdasarkan rentang waktu tersebut ke *server*. *Server* memproses permintaan dengan mengambil data penjualan dari *database*. Pengguna dapat memilih untuk mengunduhnya dalam format seperti PDF atau *Excel*. Diagram ini menunjukkan alur komunikasi dari menampilkan laporan hingga *output* laporan yang dapat diakses atau diunduh oleh *admin*.

9. Sequence Diagram Laporan Profits

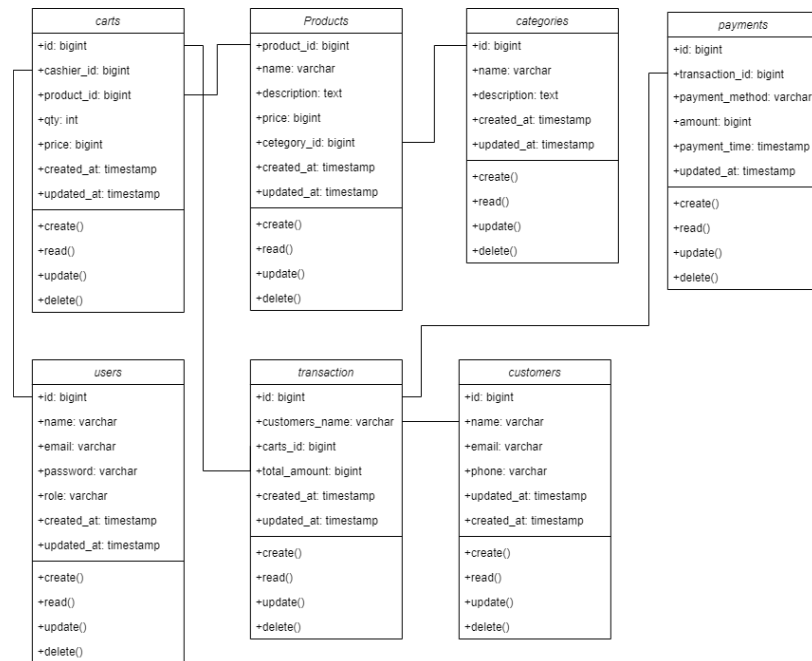


Gambar 4. 20 *Sequence Diagram Laporan Profits*

Sequence diagram laporan *profits* menggambarkan alur menampilkan laporan laba. Proses dimulai ketika *admin* membuka halaman "*Report Profits*" di antarmuka sistem dan memilih rentang waktu yang diinginkan. Antarmuka sistem kemudian mengirimkan permintaan ke *server* dengan periode waktu yang dipilih. *Server* memproses permintaan dan menampilkan laporan laba. Pengguna dapat memilih untuk mengunduhnya dalam format seperti PDF atau *Excel*. Diagram ini menunjukkan alur komunikasi dari pembuatan hingga penyajian laporan laba yang dapat diakses atau diunduh oleh *admin*.

4.1.2.4. Class Diagram

Class Diagram menunjukkan struktur kelas dari sistem POS dan bagaimana kelas-kelas tersebut berinteraksi. Berikut adalah penjelasan dari diagram tersebut:



Gambar 4. 21 Class Diagram

Class Diagram ini terdiri dari tiga kelas utama: *User*, *Transaction*, dan *PaymentGateway*. Kelas *User* berinteraksi dengan kelas *Transaction* untuk membuat transaksi. Kelas *Transaction* menggunakan kelas *PaymentGateway* untuk memproses pembayaran.

4.2. Analisis Kebutuhan Sistem

4.2.1. Kebutuhan Fungsional

Sistem *POS* yang diusulkan harus mampu mencatat transaksi secara otomatis untuk menghindari kesalahan manual dan kehilangan data. Selain itu, sistem harus terintegrasi dengan *Midtrans Payment Gateway* untuk memproses

pembayaran, yang akan mempercepat proses transaksi dan meningkatkan efisiensi. Sistem juga harus memiliki fitur monitoring transaksi secara *real-time*, yang memungkinkan UMKM untuk memantau transaksi yang sedang berlangsung dan mengidentifikasi masalah segera. Terakhir, sistem harus menyediakan laporan transaksi yang dapat diakses kapan saja, sehingga memudahkan UMKM dalam melacak dan menganalisis data transaksi mereka.

Perbandingan QRIS Konvensional dan *Midtrans Payment Gateway* dapat dilihat pada tabel dibawah ini:

Tabel 4. 1 QRIS Konvensional *vs Midtrans Payment Gateway*

QRIS Konvensional	<i>Midtrans Payment Gateway</i>
Pengguna memindai kode QR yang dicetak atau ditampilkan di merchant untuk melakukan pembayaran.	Pengguna melakukan pembayaran melalui aplikasi atau website yang terintegrasi dengan Midtrans, dan kode QR atau metode pembayaran lain disediakan secara dinamis.
Biasanya tidak memerlukan integrasi teknis yang kompleks, cukup menampilkan QR secara statis.	Memerlukan integrasi API ke dalam sistem POS atau platform, dengan fitur seperti <i>Snap</i> untuk pembayaran instan.
Proses sederhana, pengguna hanya memindai kode QR fisik atau statis di merchant.	Lebih fleksibel, pengguna dapat memilih berbagai metode pembayaran (QRIS, transfer, kartu kredit, dll.) yang disediakan oleh Midtrans.
Tingkat keamanan standar, dengan otentikasi melalui aplikasi mobile banking atau e-wallet pengguna.	Tingkat keamanan tinggi, menggunakan fitur seperti <i>3D Secure</i> , <i>fraud detection</i> , dan notifikasi pembayaran otomatis.
Tidak ada notifikasi otomatis, status pembayaran harus diperiksa secara manual oleh merchant.	Notifikasi pembayaran secara real-time melalui webhook Midtrans, status pembayaran dapat di-update secara otomatis di sistem.

4.2.2. Kebutuhan Nonfungsional

Selain kebutuhan fungsional, sistem *POS* yang diusulkan juga harus

memenuhi beberapa kebutuhan nonfungsional. Pertama, sistem harus aman dan melindungi data transaksi dari ancaman siber. Keamanan data sangat penting untuk memastikan bahwa informasi sensitif tidak jatuh ke tangan yang salah. Kedua, sistem harus mudah digunakan dan diakses oleh pengguna. Antarmuka pengguna yang intuitif dan *user-friendly* akan memudahkan UMKM dalam mengadopsi dan menggunakan sistem baru. Ketiga, sistem harus memiliki performa yang cepat dan responsif, sehingga transaksi dapat diproses dengan efisien tanpa mengalami keterlambatan.

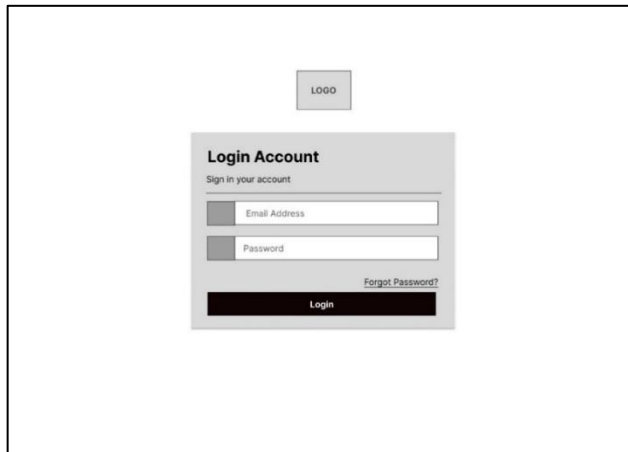
4.3. Perancangan Sistem

Desain sistem meliputi arsitektur aplikasi, antarmuka pengguna, dan komponen utama lainnya. Penjelasan rinci akan mencakup bagaimana setiap komponen berinteraksi dan bagaimana data mengalir melalui sistem. Sistem ini dirancang untuk memastikan bahwa semua transaksi tercatat dengan akurat dan dapat diakses kapan saja oleh pengguna.

4.3.1. Desain Sistem

4.3.1.1. Desain Sistem *Admin*

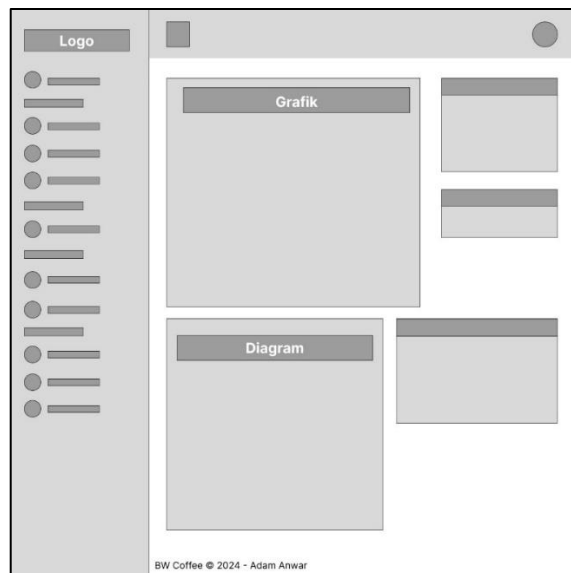
1. Desain Halaman *Login*



Gambar 4. 22 Desain sistem Halaman *Login*

Gambar diatas merupakan desain tampilan halaman *login*. *user* memasukkan data *email* dan *password* yang telah terdaftar untuk bisa mengakses halaman.

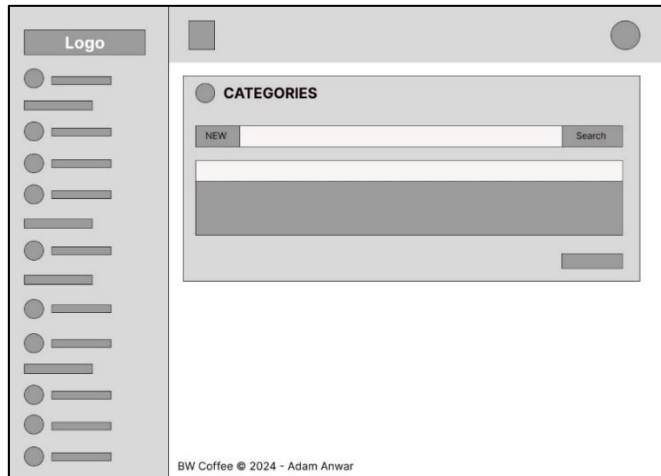
2. Desain Halaman Beranda



Gambar 4. 23 Desain sistem Halaman Beranda

Gambar diatas merupakan desain tampilan halaman beranda yang akan ditampilkan setelah berhasil melakukan proses *login*.

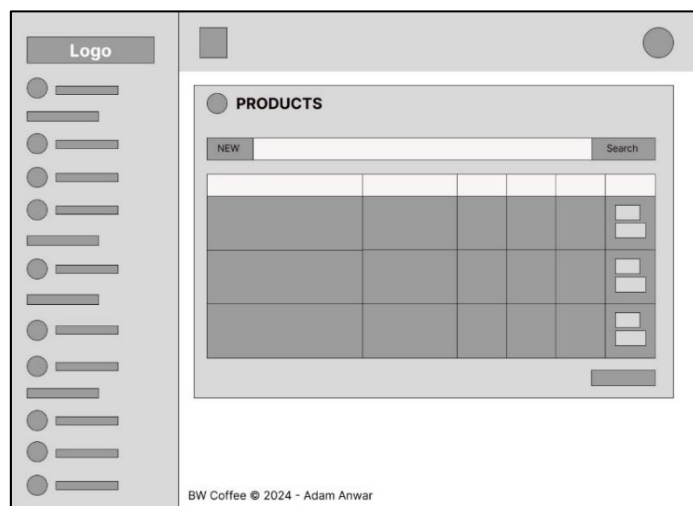
3. Desain Halaman Kelola Kategori



Gambar 4. 24 Desain sistem Halaman Kelola Kategori

Gambar diatas merupakan desain tampilan halaman kelola kategori, *admin* dapat menambah, mengubah dan menghapus data kategori.

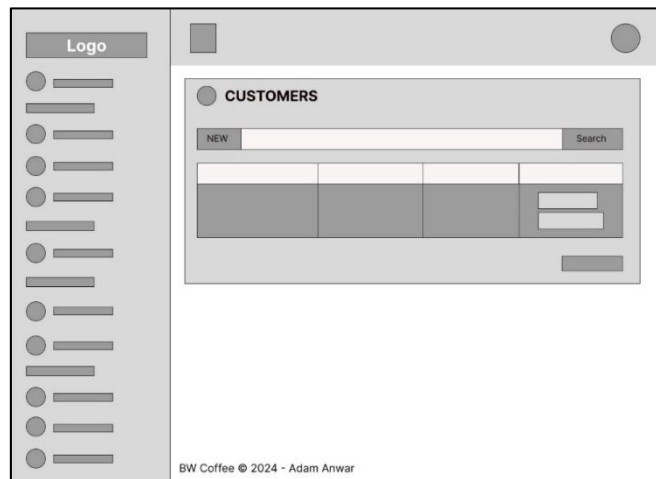
4. Desain Halaman Kelola Produk



Gambar 4. 25 Desain sistem Halaman Kelola Produk

Gambar diatas merupakan desain tampilan halaman kelola produk, *admin* dapat menambah, mengubah dan menghapus data produk.

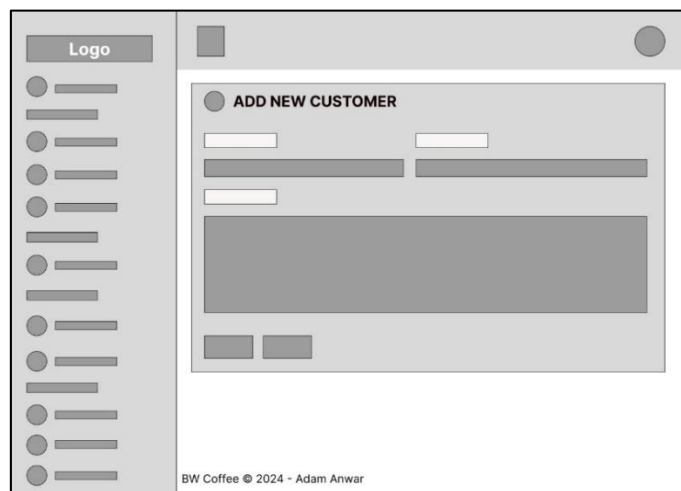
5. Desain Halaman Kelola Pelanggan



Gambar 4. 26 Desain sistem Halaman Kelola Pelanggan

Gambar diatas merupakan desain tampilan halaman kelola pelanggan, *admin* dapat menambah, mengubah dan menghapus data pelanggan.

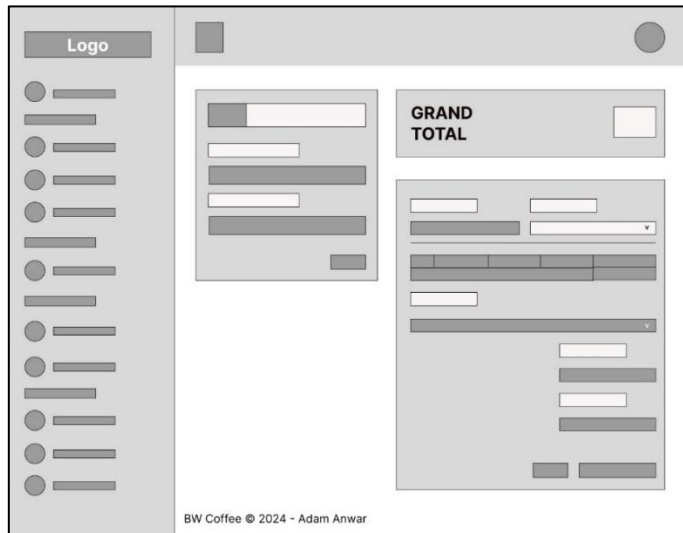
6. Desain Tambah Pelanggan



Gambar 4. 27 Desain sistem Tambah Pelanggan

Gambar diatas merupakan desain tampilan halaman kelola pelanggan, *admin* dapat menambah, mengubah dan menghapus data pelanggan.

7. Desain Halaman Transaksi



Gambar 4. 28 Desain sistem Transaksi

Gambar diatas merupakan desain tampilan halaman transaksi, admin dapat menambah, mengubah dan menghapus data transaksi yang dilakukan.

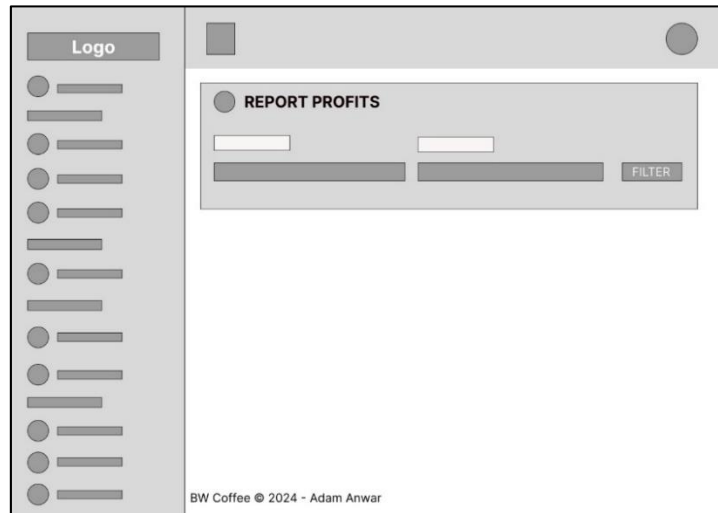
8. Desain Halaman Laporan Penjualan



Gambar 4. 29 Desain sistem Halaman Laporan Penjualan

Gambar diatas merupakan desain tampilan halaman laporan penjualan yang dapat diunduh.

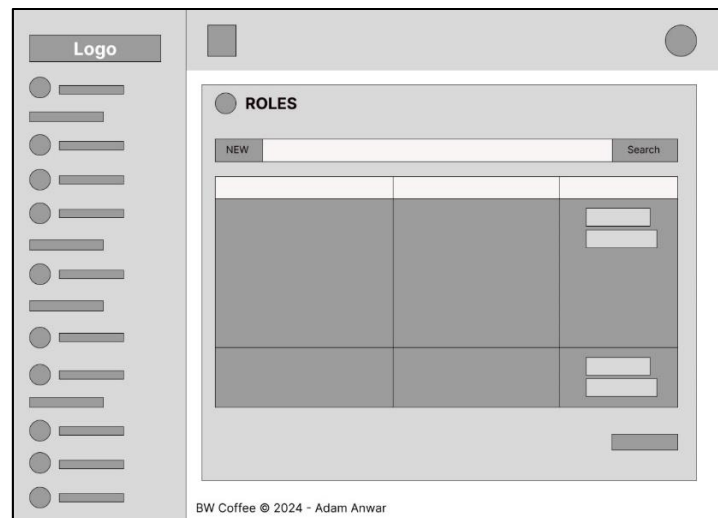
9. Desain Halaman Laporan Keuntungan



Gambar 4. 30 Desain sistem Halaman Laporan Keuntungan

Gambar diatas merupakan desain tampilan halaman laporan keuntungan yang dapat diunduh.

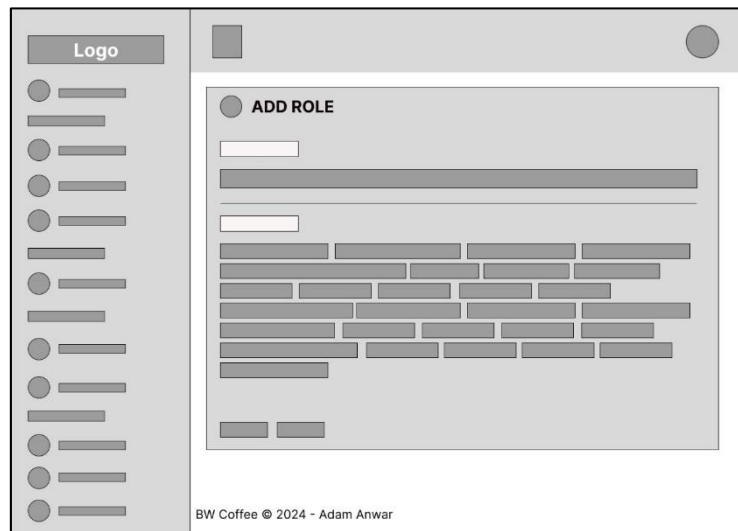
10. Desain Halaman Kelola *Role*



Gambar 4. 31 Desain sistem Halaman Kelola *Role*

Gambar diatas merupakan desain tampilan halaman kelola *role*, *admin* dapat menambah, mengubah dan menghapus data *role*.

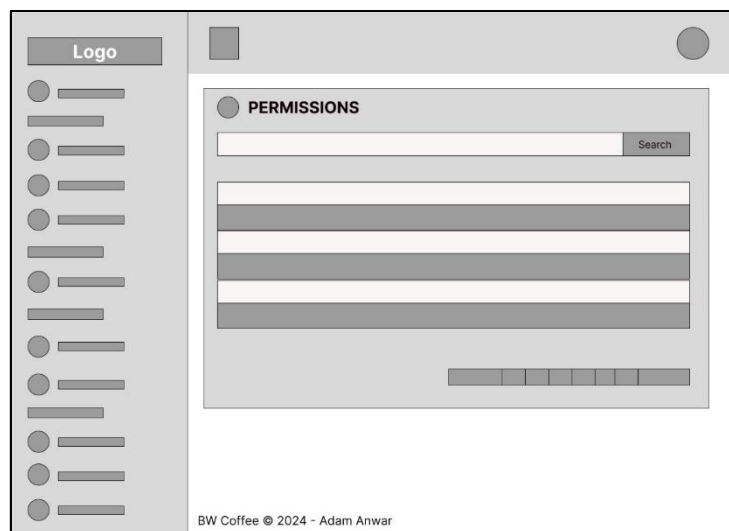
11. Desain Tambah *Role*



Gambar 4. 32 Desain sistem Tambah *Role*

Gambar diatas merupakan desain tampilan halaman *role*, *admin* dapat menambah, mengubah dan menghapus data *role*.

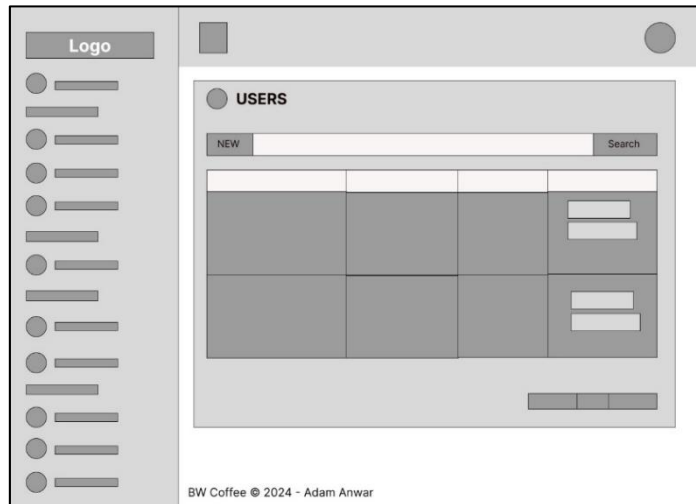
12. Desain Halaman Kelola Akses (*Permissions*)



Gambar 4. 33 Desain sistem Halaman Kelola Akses

Gambar diatas merupakan desain tampilan halaman kelola akses, *admin* dapat menambah, mengubah dan menghapus data akses.

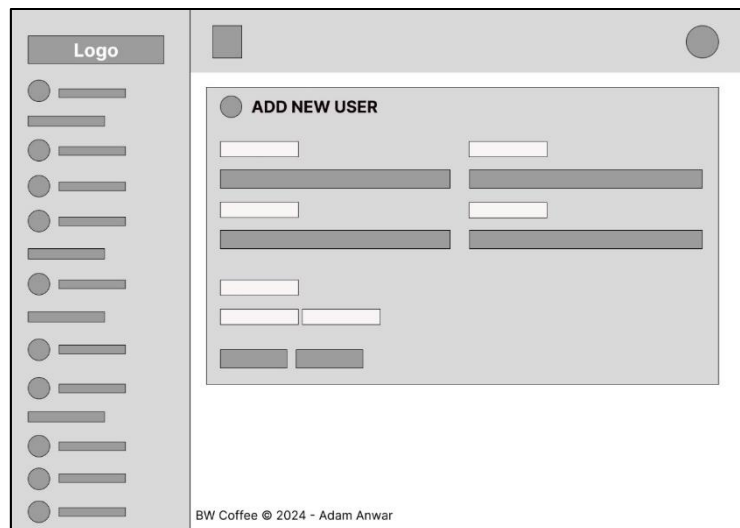
13. Desain Halaman Kelola Pengguna (*Users*)



Gambar 4. 34 Desain sistem Halaman Kelola Pengguna (*Users*)

Gambar diatas merupakan desain tampilan halaman kelola pengguna, *admin* dapat menambah, mengubah dan menghapus data pengguna.

14. Desain Tambah Pengguna

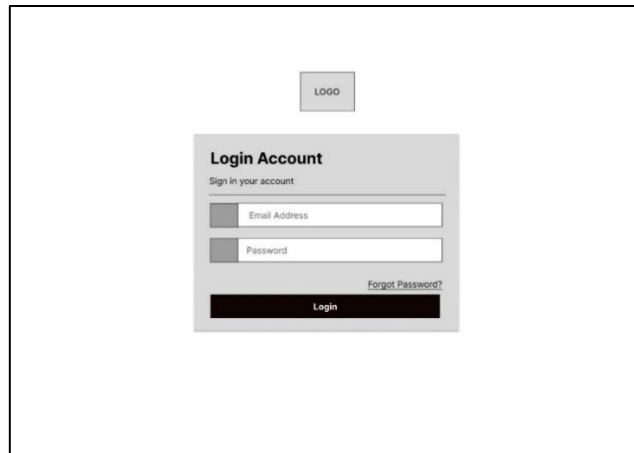


Gambar 4. 35 Desain sistem Tambah Pengguna

Gambar diatas merupakan desain tampilan tambah pengguna, *admin* harus memasukkan data sesuai kebutuhan untuk proses pendaftaran akun baru.

4.3.1.2. Desain Sistem *Cashier*

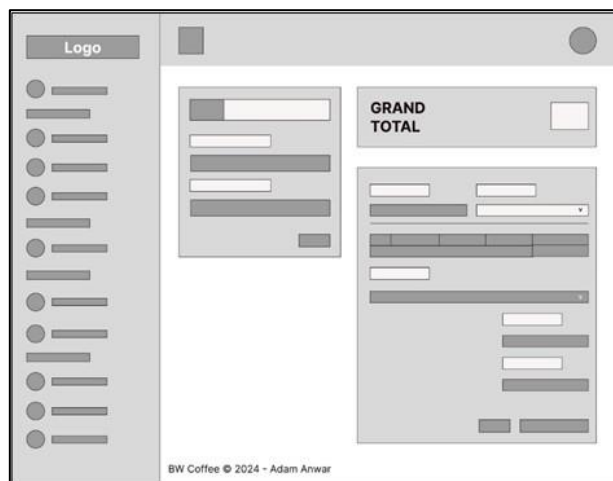
1. Desain Halaman *Login*



Gambar 4. 36 Desain sistem Halaman *Login*

Gambar diatas merupakan desain tampilan halaman *login*. *user* memasukkan data *email* dan *password* yang telah terdaftar untuk bisa mengakses halaman.

2. Desain Halaman Transaksi



Gambar 4. 37 Desain sistem Halaman Transaksi

Gambar diatas merupakan desain tampilan halaman transaksi, kasir dapat menambah data transaksi yang sedang berjalan.

4.3.2. Database

Desain *database* sistem POS berbasis *web* ini menggunakan *MySQL* dan terdiri dari beberapa tabel utama yang saling berhubungan. Tabel-tabel utama dalam *database* ini meliputi tabel *User*, *Transaction*, dan *Payment*.

1. Tabel *Carts*

Nama *Database* : inertia-pos

Nama Tabel : *carts*

Primary Key : id

Tabel 4. 2 Tabel *Carts*

Nama <i>Field</i>	Tipe Data	Ukuran	Deskripsi
<i>id</i>	<i>bigint</i>	20	Id unik, <i>Primary Key</i>
<i>cashier_id</i>	<i>bigint</i>	20	Id kasir
<i>product_id</i>	<i>bigint</i>	20	Id produk
<i>qty</i>	<i>int</i>	11	Kuantitas produk
<i>price</i>	<i>bigint</i>	20	<i>Harga barang</i>
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		

2. Tabel *Categories*

Nama *Database* : inertia-pos

Nama Tabel : *categories*

Primary Key : id

Tabel 4. 3 Tabel *Categories*

Nama Field	Tipe Data	Ukuran	Deskripsi
<i>id</i>	<i>bigint</i>	20	Id unik, <i>Primary Key</i>
<i>image</i>	<i>varchar</i>	255	Link gambar
<i>name</i>	<i>varchar</i>	255	Nama kategori
<i>description</i>	<i>text</i>		Deskripsi kategori
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		

3. Tabel *Customers*

Nama *Database* : inertia-pos

Nama Tabel : *customers*

Primary Key : id

Tabel 4. 4 Tabel *Customers*

Nama Field	Tipe Data	Ukuran	Deskripsi
<i>id</i>	<i>bigint</i>	20	Id unik, <i>Primary Key</i>
<i>name</i>	<i>varchar</i>	255	Nama pelanggan
<i>no_telp</i>	<i>int</i>		Nomor telp pelanggan
<i>address</i>	<i>text</i>		Alamat pelanggan
<i>created_at</i>	<i>timestamp</i>		

<i>updated_at</i>	<i>timestamp</i>		
-------------------	------------------	--	--

4. Tabel *Products*

Nama Database : inertia-pos

Nama Tabel : *products*

Primary Key : id

Tabel 4. 5 Tabel *Products*

Nama Field	Tipe Data	Ukuran	Deskripsi
<i>id</i>	<i>bigint</i>	20	Id unik, <i>Primary Key</i>
<i>category_id</i>	<i>bigint</i>	20	Id kategori
<i>image</i>	<i>varchar</i>	255	Link gambar
<i>barcode</i>	<i>varchar</i>	255	Barcode produk
<i>titled</i>	<i>varchar</i>	255	Judul produk
<i>description</i>	<i>text</i>		Deskripsi produk
<i>buy_price</i>	<i>bigint</i>	20	Harga beli
<i>sell_prices</i>	<i>bigint</i>	20	Harga jual
<i>stock</i>	<i>int</i>	11	Stok produk
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		

5. Tabel *Transaction*

Nama *Database* : inertia-pos

Nama Tabel : *transaction*

Primary Key : id

Tabel 4. 6 Tabel *Transaction*

Nama Field	Tipe Data	Ukuran	Deskripsi
<i>id</i>	<i>bigint</i>	20	Id unik, <i>Primary Key</i>
<i>is_paid</i>	<i>tinyint</i>	1	
<i>cashier_id</i>	<i>bigint</i>	20	Id kasir
<i>customer_id</i>	<i>bigint</i>	20	Id <i>customer</i>
<i>invoice</i>	<i>varchar</i>	255	invoice
<i>cash</i>	<i>bigint</i>	20	Total bayar
<i>change</i>	<i>bigint</i>	20	Kembalian
<i>discount</i>	<i>bigint</i>	20	Diskon jika ada
<i>grand_total</i>	<i>bigint</i>	20	Total keseluruhan
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		
<i>snap_token</i>	<i>varchar</i>	255	
<i>qris_link</i>	<i>varchar</i>	255	

6. Tabel *Users*

Nama *Database* : inertia-pos

Nama Tabel : *users*

Primary Key : id

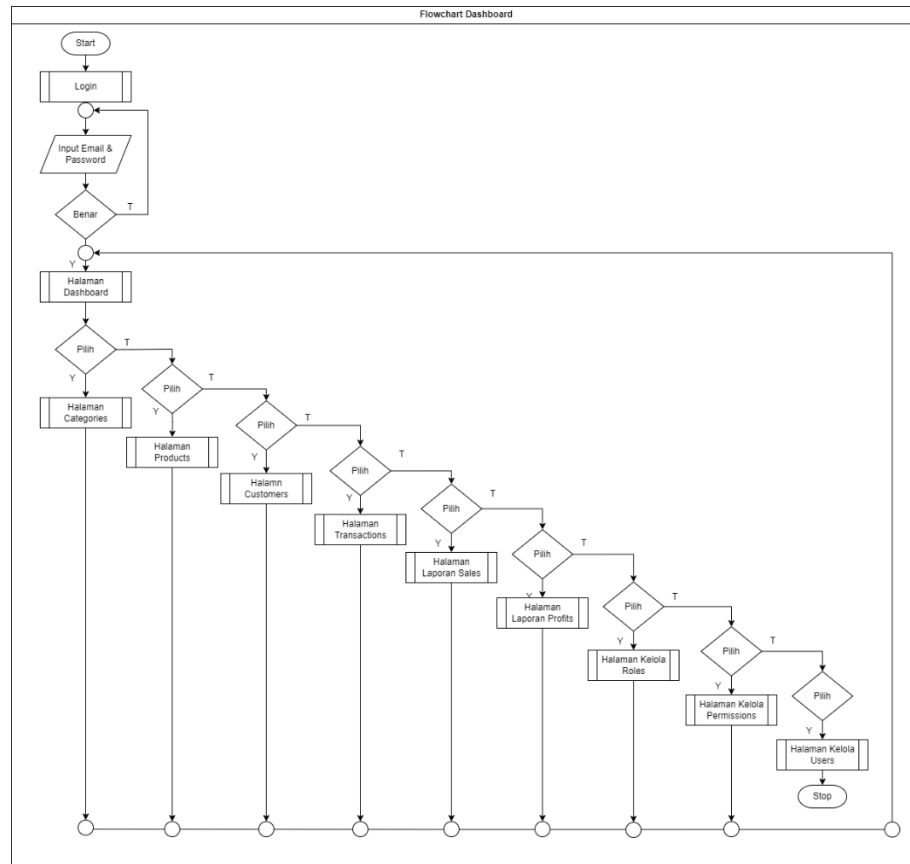
Tabel 4. 7 Tabel *Users*

Nama Field	Tipe Data	Ukuran	Deskripsi
<i>id</i>	<i>bigint</i>	20	Id unik, <i>Primary Key</i>
<i>Name</i>	<i>varchar</i>	255	
<i>email</i>	<i>varchar</i>	255	
<i>Email_verified_at</i>	<i>bigint</i>	20	
<i>password</i>	<i>varchar</i>	255	
<i>Two_factor_secret</i>	<i>Text</i>		
<i>Two_factor_recovery_codes</i>	<i>Text</i>		
<i>Remember_token</i>	<i>varchar</i>	100	
<i>grand_total</i>	<i>bigint</i>	20	
<i>created_at</i>	<i>timestamp</i>		
<i>updated_at</i>	<i>timestamp</i>		

4.3.3. Flowchart

4.3.3.1. Flowchart Admin

1. Flowchart Halaman Utama (Dashboard)



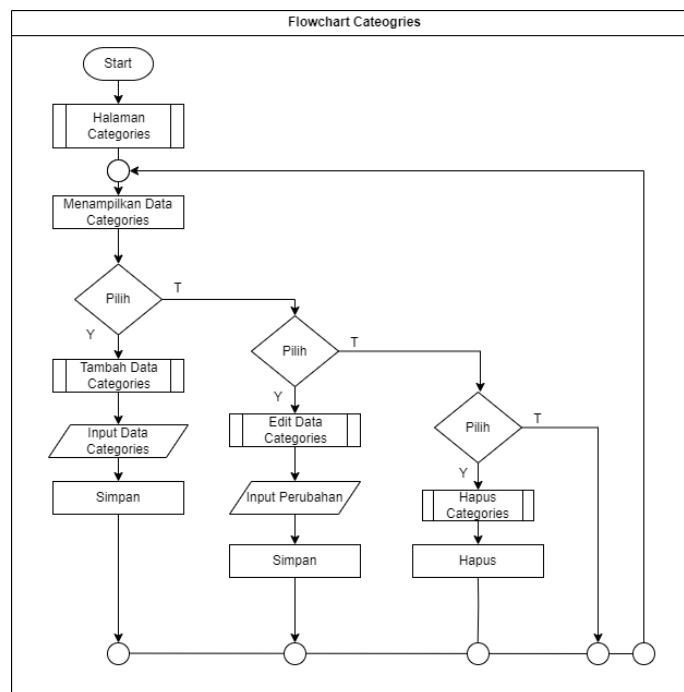
Gambar 4. 38 Flowchart Halaman Utama (Dashboard)

Gambar diatas menunjukkan proses yang terjadi ketika aktor mengakses sistem setelah *login*. Halaman *dashboard* hanya ditampilkan kepada aktor *admin*.

- 1) Proses dimulai dengan aktor melakukan *login* ke sistem dengan memasukkan *email* dan *password*. Setelah itu, sistem akan melakukan validasi *credential* yang dimasukkan.
- 2) Aktor akan diarahkan ke halaman *dashboard*, di mana aktor dapat memilih salah satu dari berbagai menu yang tersedia seperti kelola

categories, kelola *products*, kelola *customers*, halaman *transactions*, laporan penjualan, laporan keuntungan, kelola *roles*, kelola *permissions*, dan kelola *users*. Aktor juga dapat memilih untuk *logout*.

2. *Flowchart Kelola Categories*



Gambar 4. 39 *Flowchart Fitur Categories*

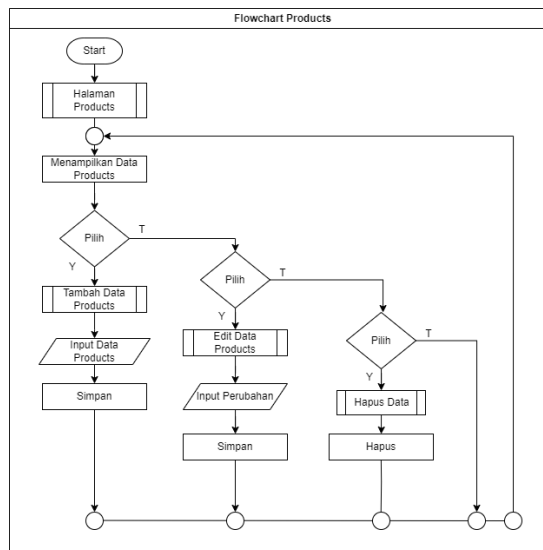
Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam kelola *categories*. Proses dimulai saat aktor membuka halaman *categories*. Setelah halaman terbuka, sistem akan menampilkan daftar kategori yang ada beserta fitur pencarian. Pada titik ini, aktor dapat memilih tindakan yang diinginkan seperti menambah kategori baru, mencari kategori yang sudah ada, mengubah kategori, atau menghapus kategori.

- 1) Menambah Kategori Baru: Sistem akan membuka formulir penambahan kategori. Aktor diharuskan mengisi rincian kategori seperti gambar, nama kategori, dan deskripsi. Setelah pengisian selesai, aktor dapat memilih untuk menyimpan kategori baru atau mereset formulir jika

terjadi kesalahan. Jika kategori disimpan, sistem akan memperbarui daftar kategori dengan menampilkan kategori yang baru ditambahkan.

- 2) Mengubah Kategori: Sistem akan membuka formulir perubahan data kategori yang sudah ada, aktor melakukan perubahan yang diperlukan. Setelah perubahan disimpan, sistem akan memperbarui daftar kategori.
- 3) Menghapus Kategori: Sistem akan menghapus kategori yang dipilih dari daftar. Semua tindakan diakhiri dengan menampilkan daftar kategori yang telah diperbarui, memastikan bahwa semua perubahan tercermin dalam tampilan.

3. *Flowchart Kelola Product*



Gambar 4. 40 *Flowchart Fitur Product*

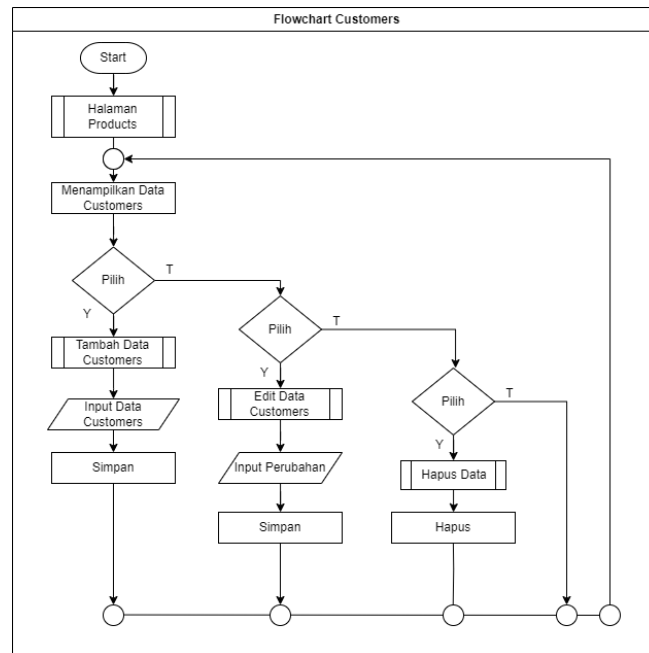
Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam kelola *products*. Proses dimulai ketika aktor membuka halaman *products*. Setelah halaman terbuka, sistem akan menampilkan daftar produk yang ada beserta fitur pencarian. Pada titik ini, aktor dapat memilih salah satu

tindakan yang diinginkan, seperti menambah produk baru, mencari produk yang sudah ada, mengubah produk, atau menghapus produk.

- 1) Menambah Produk Baru: Sistem akan membuka formulir untuk menambah produk baru. aktor diharuskan mengisi detail produk, termasuk *barcode*, *title*, *buy price*, *sell price*, dan *stock*. Setelah semua detail diisi, aktor dapat memilih untuk menyimpan produk baru atau mengatur ulang formulir jika terjadi kesalahan. Jika produk disimpan, sistem akan memperbarui daftar produk dengan menampilkan produk yang baru ditambahkan.
- 2) Mengubah Produk: Sistem akan membuka formulir ubah produk yang sudah ada, aktor mengubah data sesuai yang diperlukan. Setelah perubahan disimpan, sistem akan memperbarui daftar produk.
- 3) Menghapus Produk: Sistem akan menghapus produk yang dipilih dari daftar.
- 4) Mencari Produk: aktor dapat mencari produk tertentu menggunakan fitur pencarian, dan hasil pencarian akan ditampilkan.

Setelah melakukan salah satu tindakan di atas, sistem akan menampilkan daftar produk yang telah diperbarui, memastikan bahwa semua perubahan tercermin dalam tampilan.

4. *Flowchart Kelola Customers*



Gambar 4. 41 *Flowchart Fitur Customers*

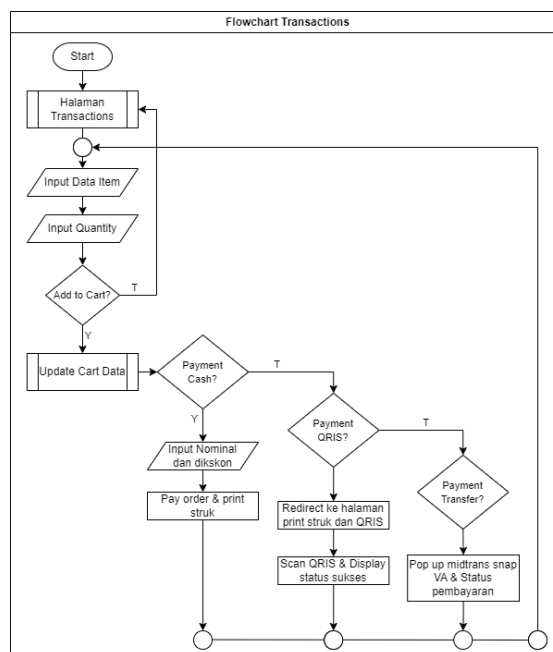
Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam kelola *customers*. Proses dimulai ketika aktor membuka halaman *customers*. Setelah halaman terbuka, sistem menampilkan daftar pelanggan yang ada beserta fitur pencarian. Aktor dapat memilih untuk melakukan beberapa tindakan seperti menambah pelanggan baru, mencari pelanggan yang sudah ada, mengubah informasi pelanggan, atau menghapus pelanggan dari daftar.

- 1) Menambah Pelanggan Baru: Sistem akan membuka formulir untuk menambahkan pelanggan baru. Aktor harus mengisi detail pelanggan seperti nama lengkap, nomor telepon, dan alamat. Setelah semua informasi diisi, aktor dapat memilih untuk menyimpan pelanggan baru atau mengatur ulang formulir jika ada kesalahan dalam pengisian. Jika pelanggan baru disimpan, sistem akan memperbarui daftar pelanggan

dengan menambahkan data yang baru ditambahkan.

- 2) Mengubah Pelanggan: Sistem membuka formulir perubahan untuk pelanggan yang sudah ada, aktor mengubah data sesuai yang diperlukan. Setelah perubahan disimpan, sistem akan memperbarui daftar pelanggan.
- 3) Menghapus Pelanggan: Sistem akan menghapus pelanggan yang dipilih dari daftar, dan daftar akan diperbarui untuk mencerminkan penghapusan tersebut.
- 4) Mencari Pelanggan: Aktor dapat memasukkan kata kunci untuk mencari pelanggan tertentu, dan hasil pencarian akan ditampilkan pada daftar pelanggan.

5. *Flowchart Halaman Transaction*

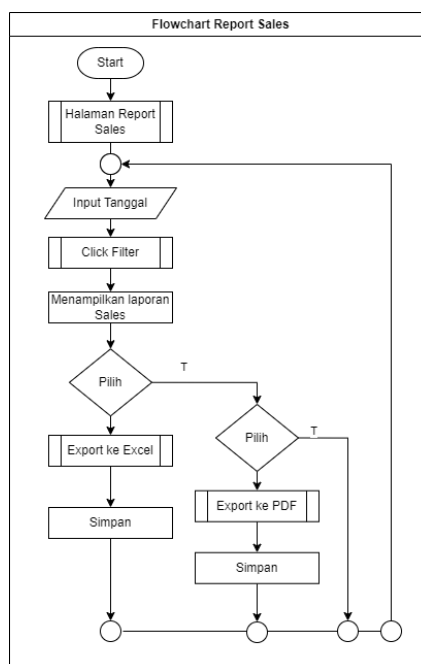


Gambar 4. 42 *Flowchart Transaction*

Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam proses *transaction*. Proses dimulai ketika aktor membuka halaman *transaction*.

- 1) Menambahkan Produk Transaksi: Aktor memasukkan *barcode* dan kuantitas sesuai pesanan pelanggan. Setelah itu, aktor menekan tombol “*Add Item*” untuk menambahkan data transaksi. Kemudian sistem akan menampilkan data produk yang ditambahkan pada daftar transaksi.
- 2) Memilih *Customers*: Aktor memilih *customer* melalui komponen “*select*” sesuai dengan nama *customers* yang sedang melakukan transaksi.
- 3) Pembayaran: Aktor memilih metode pembayaran yang dilakukan oleh *customers* kemudian memasukkan nominal pembayaran yang diberikan oleh *customers* dan memasukkian jumlah diskon jika ada.
- 4) Mencetak Struk: Jika tidak ada tambahin lain, maka aktor dapat memproses untuk mencetak struk belanja.

6. *Flowchart Report Sales*

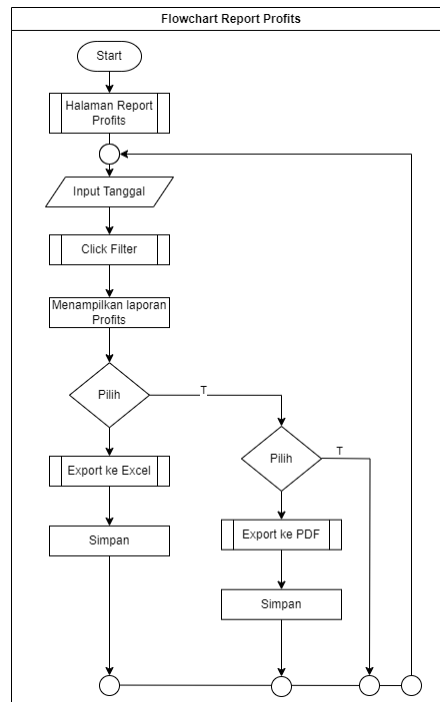


Gambar 4. 43 *Flowchart Report Sales*

Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam proses mengekspor laporan penjualan pada sistem POS. Proses dimulai ketika aktor membuka halaman laporan penjualan.

- 1) Memilih Periode Waktu: aktor memilih rentang periode awal dan akhir laporan penjualan yang diperlukan.
- 2) *Filter* Laporan: Setelah menentukan periode waktu, aktor menekan tombol “*filter*” untuk menampilkan data penjualan yang sesuai dengan periode yang telah dipilih.
- 3) Menampilkan Data Laporan: Sistem kemudian menampilkan daftar laporan penjualan berdasarkan periode waktu yang telah diatur.
- 4) Ekspor Laporan Penjualan: Aktor memiliki opsi untuk mengekspor laporan penjualan tersebut ke dalam format *excel* atau pdf. Jika aktor memilih format *excel*, maka laporan penjualan akan diekspor kedalam bentuk format tersebut. Jika memilih pdf, maka laporan penjualan akan diekspor kedalam bentuk format tersebut sehingga aktor dapat menyimpan *file* tersebut ke perangkat mereka sesuai format yang dipilih.

7. *Flowchart Report Profits*



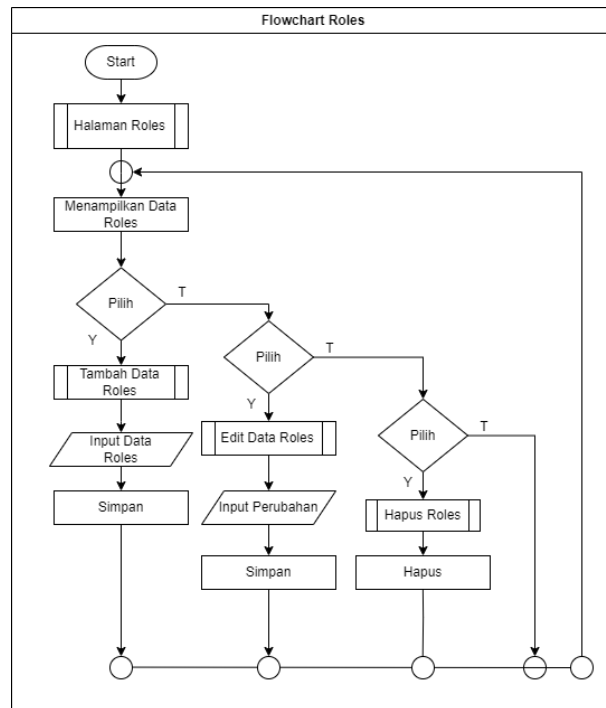
Gambar 4. 44 *Flowchart Report Profits*

Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam proses mengekspor laporan keuntungan pada sistem POS. Proses dimulai ketika aktor membuka halaman laporan keuntungan

- 1) Memilih Periode Waktu: aktor memilih rentang periode awal dan akhir laporan keuntungan yang diperlukan.
- 2) *Filter* Laporan: Setelah menentukan periode waktu, aktor menekan tombol “*filter*” untuk menampilkan data yang sesuai dengan periode yang telah dipilih.
- 3) Menampilkan Data Laporan: Sistem kemudian menampilkan daftar laporan berdasarkan periode waktu yang telah diatur.
- 4) Ekspor Laporan Keuntungan: Aktor memiliki opsi untuk mengekspor laporan tersebut ke dalam format excel atau pdf. Jika aktor memilih

format *excel*, maka laporan keuntungan akan diekspor kedalam bentuk format tersebut. Jika memilih pdf, maka laporan keuntungan akan diekspor kedalam bentuk format tersebut sehingga aktor dapat menyimpan *file* tersebut ke perangkat mereka sesuai format yang dipilih.

8. Flowchart Roles



Gambar 4. 45 Flowchart Roles

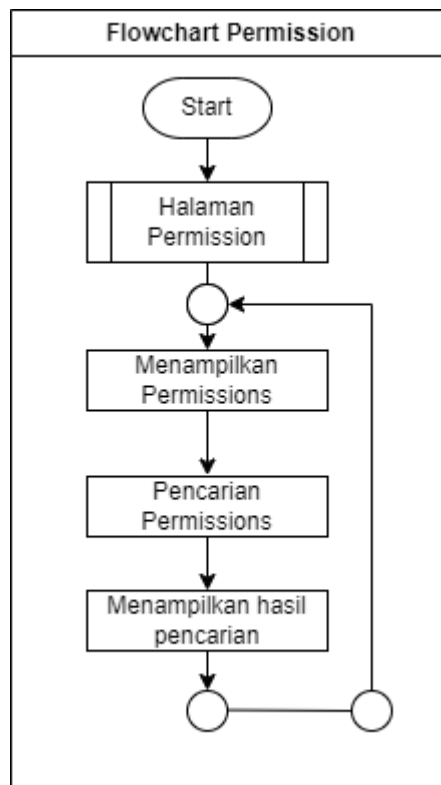
Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam proses kelola *roles*. Proses dimulai ketika aktor membuka halaman *roles* kemudian sistem akan menampilkan daftar peran yang ada beserta fitur pencarian untuk memudahkan aktor dalam menemukan peran tertentu. Aktor dapat memilih tindakan yang ingin dilakukan seperti menambah peran baru, mengubah peran yang sudah ada, mencari peran, atau menghapus peran.

- 1) Menambahkan *Role*: Sistem akan membuka formulir penambahan peran dan mengharuskan aktor mengisi formulir dengan nama peran baru.

Aktor juga harus mengatur izin-izin yang dimiliki oleh peran tersebut kemudian menyimpan peran baru atau mengatur ulang formulir. Jika aktor memilih untuk menyimpan, peran baru disimpan ke dalam sistem, dan akan sistem akan menampilkan peran baru yang telah ditambahkan.

- 2) Mengubah *Role*: Sistem membuka formulir perubahan untuk peran yang sudah ada, aktor mengubah data sesuai yang diperlukan. Setelah perubahan disimpan, sistem akan memperbarui daftar peran.
- 3) Menghapus *Role*: Sistem akan menghapus peran yang dipilih dari daftar, dan daftar akan diperbarui untuk mencerminkan penghapusan tersebut.
- 4) Mencari *Role*: Aktor dapat memasukkan kata kunci untuk mencari peran tertentu, dan hasil pencarian akan ditampilkan pada daftar peran.

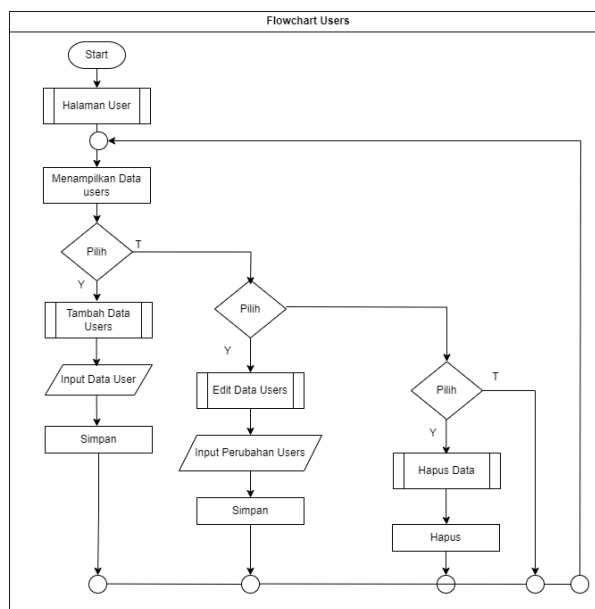
9. *Flowchart Permissions*



Gambar 4. 46 *Flowchart Kelola Permissions*

Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam proses kelola *permissions*. Proses dimulai ketika aktor membuka halaman *permissions* kemudian sistem akan menampilkan daftar *permissions* yang ada beserta fitur pencarian untuk memudahkan aktor dalam menemukan data tertentu. Aktor dapat memasukkan kata kunci untuk mencari *role* tertentu, dan hasil pencarian akan ditampilkan pada daftar *role*.

10. Flowchart Kelola Users



Gambar 4. 47 Flowchart Kelola Users

Gambar diatas menunjukkan proses dan langkah-langkah yang terjadi dalam kelola *users*. Proses dimulai ketika aktor membuka halaman *users*. Setelah halaman terbuka, sistem menampilkan daftar *users* yang ada beserta fitur pencarian. Aktor dapat memilih untuk melakukan beberapa tindakan seperti menambah *user* baru, mencari *user* yang sudah ada, mengubah informasi *user*, atau menghapus *user* dari daftar.

- 1) Menambah *User* Baru: Sistem akan membuka formulir untuk menambahkan *user* baru. Aktor harus mengisi detail pelanggan seperti