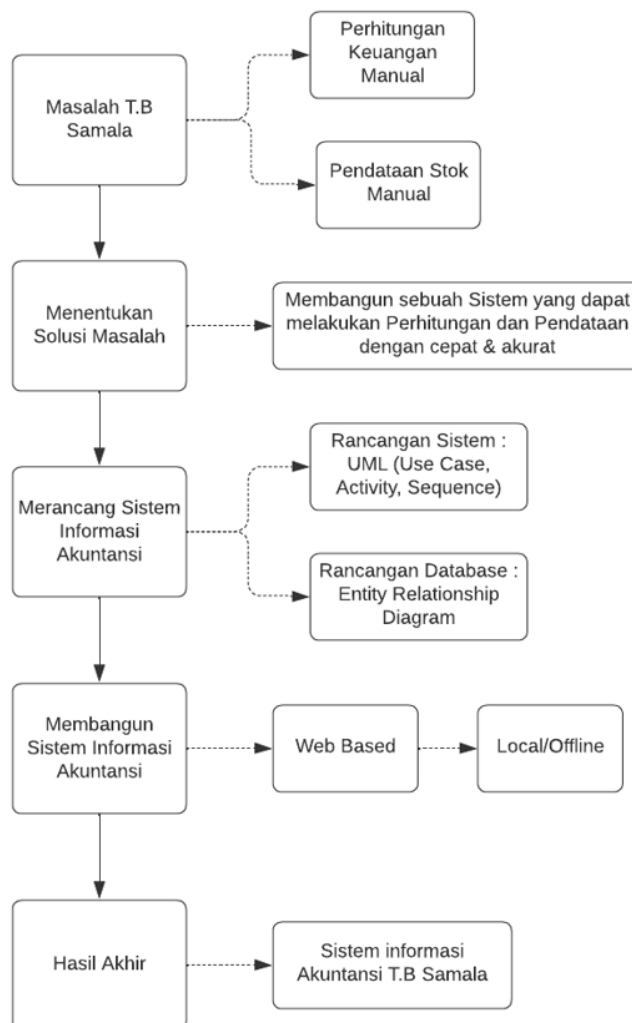


BAB III METODE PENELITIAN

3.1 Kerangka Berpikir

Alur berpikir dimulai dari menganalisis masalah apa saja yang terjadi di T.B Samala, menentukan solusi dari masalah-masalah yang ada, melakukan perancangan arsitektur sistem, melakukan pembangunan sistem dari arsitektur yang telah dirancang dan didapatkanlah hasil akhir yaitu sebuah sistem informasi akuntansi yang membantu menyelesaikan masalah-masalah yang dihadapi T.B Samala.



Gambar 3.1.1 Kerangka Berpikir

3.2 Jenis Penelitian

Penelitian ini menggunakan jenis penelitian *Research and Development* (Penelitian dan Pengembangan) menggunakan metode *waterfall*. Jenis penelitian ini digunakan untuk menghasilkan dan menguji efektifitas sebuah produk tertentu. (Fransisca & Putri, 2019)

3.3 Waktu dan Tempat Penelitian

3.3.1 Waktu Penelitian

Penelitian dilakukan pada tahun 2023 selama kurang lebih 6 bulan mulai dari bulan Februari sampai Juli dengan pembagian sebagai berikut:

Tabel 3.3.1 Tabel Waktu Kegiatan

Kegiatan	Bulan					
	Feb	Mar	Apr	Mei	Jun	Jul
Pengumpulan Data						
Analisis Kebutuhan Sistem						
Perancangan Sistem						
Pembuatan Sistem						
Pengujian dan <i>Maintenance</i>						

3.3.2 Tempat Penelitian

Toko Bangunan Samala adalah sebuah perusahaan yang sekarang beroperasi di Semarang tepatnya di Jl. Erlangga Tengah No.24B, Kec. Semarang Selatan dan bergerak di bidang jual/beli barang bangunan seperti semen, pipa, mur dan baut, dan lain-lain.

Penelitian ini sendiri berlokasi di rumah pemilik Toko yang Bernama Ibu Ria Yunike dan berlokasi di Jalan Karangrejo Tengah No.88, Kec. Gajahmungkur, Semarang.

3.4 Teknik Pengumpulan Data

3.4.1 Wawancara

Menurut (Makbul, 2021) Wawancara adalah sebuah proses komunikasi antara pewawancara dan sumber informasi melalui interaksi langsung. Bisa juga diartikan sebagai percakapan atau pertemuan pribadi antara pewawancara dan sumber informasi.

Pada penelitian ini wawancara dimaksudkan untuk memperoleh informasi tentang masalah yang dihadapi dan kebutuhan apa saja yang diperlukan oleh pemilik perusahaan.

3.4.2 Studi Dokumen

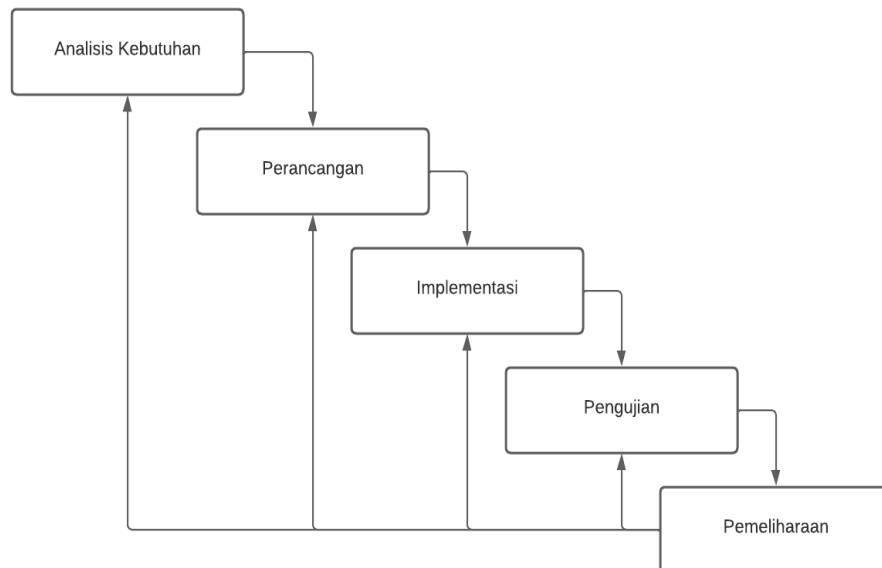
Dalam teknik ini, peneliti mengumpulkan data dengan memeriksa dan menganalisis dokumen yang relevan, seperti arsip, laporan, jurnal, surat, foto, atau rekaman *audio/video*. Dokumentasi dapat memberikan wawasan tentang perkembangan suatu topik yang sedang diteliti.

Pada penelitian ini peneliti diberikan beberapa dokumen berupa jurnal dan catatan stok yang dapat peneliti gunakan sebagai referensi dalam pembuatan sistem.

3.5 Metode Penelitian

Metode *Waterfall* adalah salah satu pendekatan yang sering digunakan dalam pengembangan perangkat lunak. Metode ini mengikuti pendekatan yang

sistematis dan berurutan, di mana setiap tahap bergantung pada keberhasilan tahap sebelumnya.



Gambar 3.5.1 Tahapan Metode Waterfall

Berikut adalah tahap-tahap penelitian menggunakan metode *waterfall* :

3.5.1 Analisis Kebutuhan

Tahap pertama dalam metode *Waterfall* adalah analisis kebutuhan. Tahap ini adalah bagian penting metode *Waterfall* yang bertujuan untuk memahami kebutuhan dan kegunaan software yang diinginkan oleh pengguna. Analisis kebutuhan dapat dibagi menjadi 2 yaitu analisis kebutuhan fungsional dan non-fungsional. Kebutuhan fungsional meliputi proses-proses apa saja yang harus dimiliki oleh sistem, sedangkan kebutuhan non-fungsional lebih menitikberatkan kepada kriteria sistem yang dibutuhkan.

Analisis kebutuhan dilakukan dengan cara wawancara dan studi dokumen. Pertama, peneliti melakukan wawancara dengan pemilik mengenai kendala yang dialami oleh T.B Samala saat ini dan solusi apa yang diinginkan pemilik. Peneliti juga meminta dokumentasi berupa contoh jurnal dan neraca milik T.B Samala yang menjadi referensi peneliti dalam proses pembuatan sistem.

3.5.2 Perancangan

Pada tahap ini, dilakukan perancangan arsitektur sistem secara keseluruhan dan desain modul-modul atau komponen-komponen yang akan digunakan. Tahapan ini bertujuan untuk memberikan gambaran lengkap tentang apa yang perlu dilakukan dan seperti apa sistem yang diinginkan.

Perancangan sistem akan dibuat menggunakan metode perancangan *UML (Unified Modelling Language)* dan *ERD (Entity Relationship Diagram)*. Berikut adalah perincian langkah-langkah yang dilakukan saat perancangan sistem :

- a. Mengumpulkan data kebutuhan sistem yang didapatkan dari tahap analisis kebutuhan.
- b. Dari data yang sudah dikumpulkan dibuatlah *use case diagram* untuk menggambarkan apa saja interaksi yang dapat dilakukan aktor (pengguna) didalam sistem.
- c. Langkah selanjutnya adalah membuat *activity diagram* untuk menggambarkan alur kerja atau aktivitas dari sistem. *Activity diagram* dibuat berdasarkan jumlah *use case* yang ada didalam *use case diagram*.

- d. Membuat *sequence diagram* untuk menggambarkan interaksi antara objek-objek dari sistem secara runtut. *Sequence diagram* dibuat berdasarkan jumlah *use case* yang ada didalam *use case diagram*.
- e. Membuat *ERD (Entity Relationship Diagram)* untuk menggambarkan struktur basis data dari sistem dan relasi setiap tabelnya.
- f. Terakhir, dilakukanlah validasi dan *review* atas seluruh diagram *UML* dan *ERD* yang telah dirancang untuk memastikan desain sistem telah mencakup semua kebutuhan dan tujuan.

3.5.3 Implementasi

Tahap implementasi melibatkan penulisan kode program berdasarkan desain yang telah dibuat pada tahap sebelumnya. Peneliti akan mengimplementasikan logika dan fungsi yang diperlukan sesuai dengan kebutuhan sistem.

Sistem akan dibuat berbasis web dan bersifat *local/offline* berdasarkan permintaan pemilik perusahaan. Pembuatan sistem dilakukan menggunakan Bahasa pemrograman *PHP* dengan *framework Laravel*. *Database* yang digunakan adalah *MySQL*. Peneliti juga menggunakan Bahasa pemrograman *JavaScript* dalam pembuatan fitur-fitur yang tidak dapat dibuat dengan hanya menggunakan *PHP*. *HTML* dan *CSS* digunakan untuk membuat tampilan dari sistem informasi akuntansi.

3.5.4 Pengujian

Pada tahap ini sistem akan diuji untuk memastikan semua persyaratan telah terpenuhi dan sistem berjalan dengan baik dan tidak terjadi kerusakan atau kesalahan.

Penelitian ini akan menggunakan metode pengujian *Blackbox*. *Blackbox* merupakan metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas dari perangkat lunak. Tujuan dari pengujian *blackbox* adalah memastikan bahwa perangkat lunak menghasilkan output yang sesuai dengan kebutuhan fungsionalnya (Vetdri, Mulyono, & Junaidi, 2023), tanpa memeriksa atau memahami bagaimana perangkat lunak mencapai hasil tersebut.

Penelitian dilakukan menggunakan alat bantu *iMacros*. *iMacros* adalah sebuah ekstensi dari *google chrome* yang dapat merekam aktivitas yang dilakukan didalam *web browser* dan mengulang aktivitas tersebut secara otomatis. Maka dari itu aplikasi ini dapat digunakan untuk melakukan pengujian aplikasi berbasis *web*.

Tabel 3.5.1 Contoh Pengujian Black Box

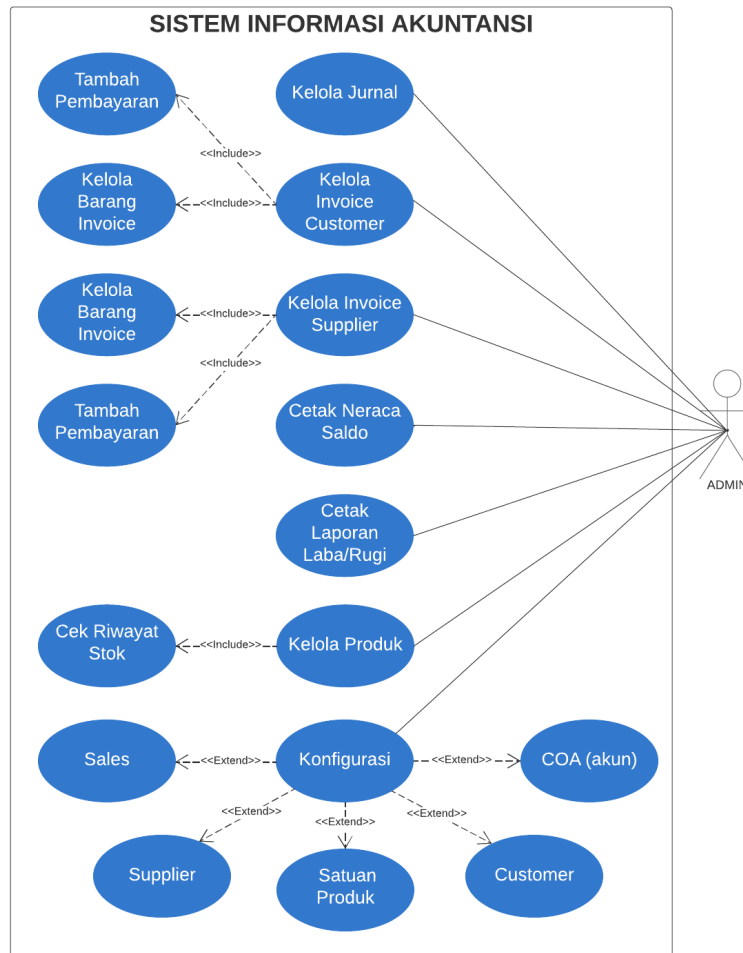
Halaman	Prosedur	Output yang diharapkan	Hasil
<i>Login</i>	<i>User</i> menginput <i>username</i> dan <i>password</i> dengan benar	<i>User</i> berhasil <i>login</i>	Berhasil
	<i>User</i> menginput <i>username</i> dan <i>password</i> yang salah	<i>User</i> tidak dapat <i>login</i>	Berhasil

3.5.5 Pemeliharaan

Tahap ini berlangsung setelah sistem mulai digunakan. Pada tahap ini, peneliti bertanggung jawab untuk melakukan pemeliharaan, pembaruan, dan perbaikan atas masalah yang mungkin muncul. Pengembangan dan penambahan fitur juga terjadi di tahapan ini.

3.6 Perancangan Sistem

3.6.1 Use Case Diagram



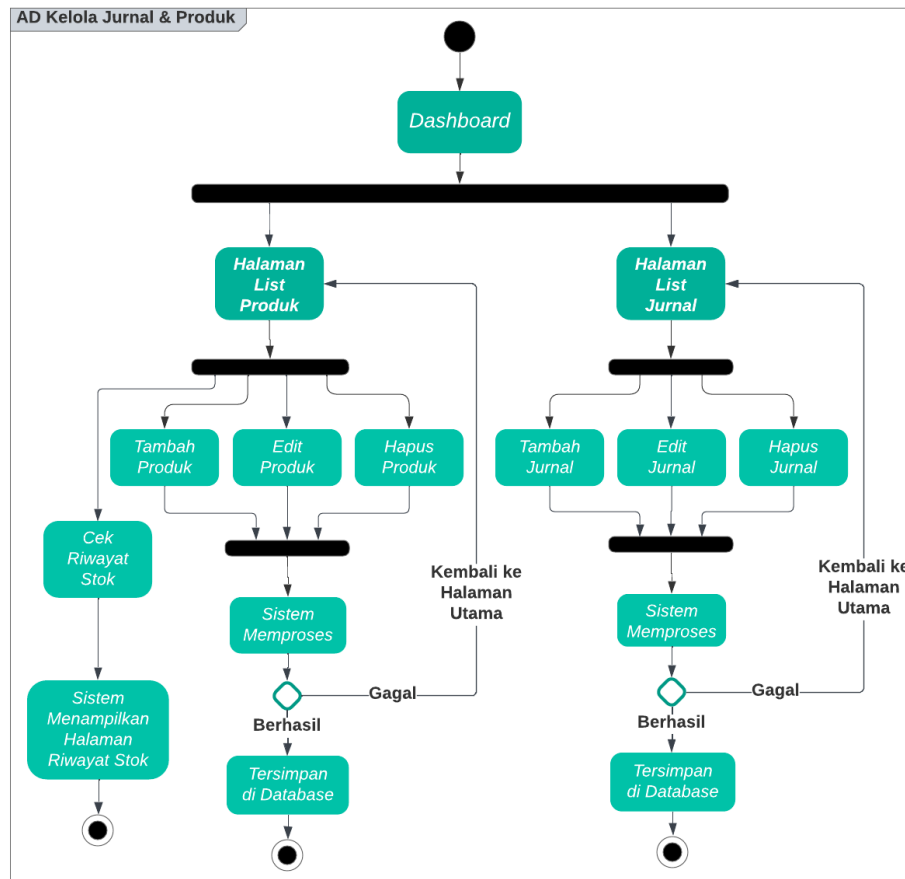
Gambar 3.6.1 Use Case Diagram T.B Samala

Use Case Diagram di atas menggambarkan user sebagai Admin. Admin dapat melakukan beberapa fitur yaitu Kelola Jurnal, Kelola *Invoice Customer*, Kelola *Invoice Supplier*, Cetak Neraca Saldo, Cetak Laporan Laba/Rugi, Kelola Produk dan Konfigurasi.

Fitur Kelola Barang *Invoice* dan Tambah Pembayaran termasuk kedalam fitur Kelola *Invoice Customer* dan *Supplier*. Fitur Konfigurasi terbagi menjadi 5 yaitu Konfigurasi *Supplier*, *Customer*, Satuan, Akun dan Sales.

3.6.2 Activity Diagram

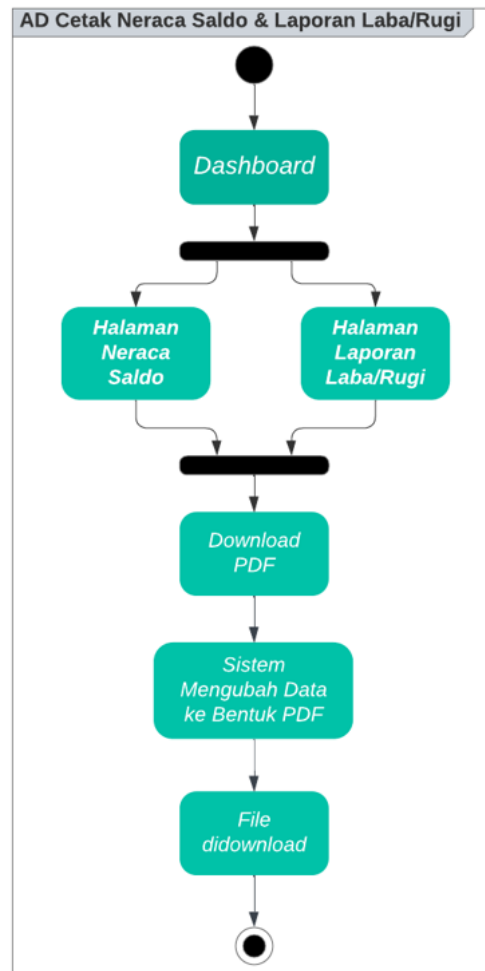
3.6.2.1 Kelola Jurnal & Produk



Gambar 3.6.2 Activity Diagram Jurnal & Produk

Activity Diagram diatas menunjukkan alur aktivitas yang dapat dilakukan dalam fitur Kelola Jurnal dan Kelola Produk. Keduanya dapat diakses melalui halaman *dashboard* dan memiliki alur yang sama yaitu dapat melakukan tambah, *edit* dan hapus yang kemudian akan diproses oleh sistem. Jika berhasil, data akan tersimpan di database, jika gagal maka akan dialihkan ke halaman utama agar admin dapat mencoba lagi. Halaman *List Produk* memiliki fitur cek Riwayat stok untuk melihat Riwayat stok dari produk yang dipilih.

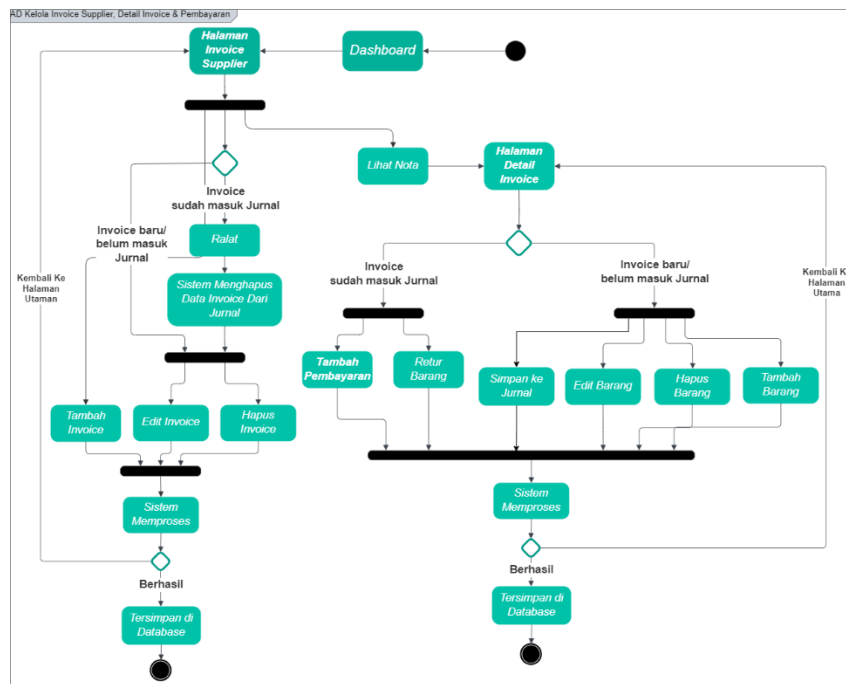
3.6.2.2 Cetak Neraca Saldo & Laporan Laba/Rugi



Gambar 3.6.3 Activity Diagram Neraca Saldo & Laporan Laba/Rugi

Activity Diagram diatas menunjukkan alur aktivitas yang dapat dilakukan dalam fitur Cetak Neraca Saldo dan Cetak Laporan Laba/Rugi. Keduanya dapat diakses melalui halaman *dashboard* dan memiliki alur yang sama yaitu dapat mencetak data dalam bentuk *PDF*. Saat aktivitas dijalankan maka sistem akan memproses dengan mengubah data menjadi *PDF* kemudian *file PDF* akan didownload ke komputer Admin.

3.6.2.3 Kelola *Invoice Supplier*, Detail *Invoice* & Pembayaran



Gambar 3.6.4 Activity Diagram Invoice Supplier & Detail Invoice

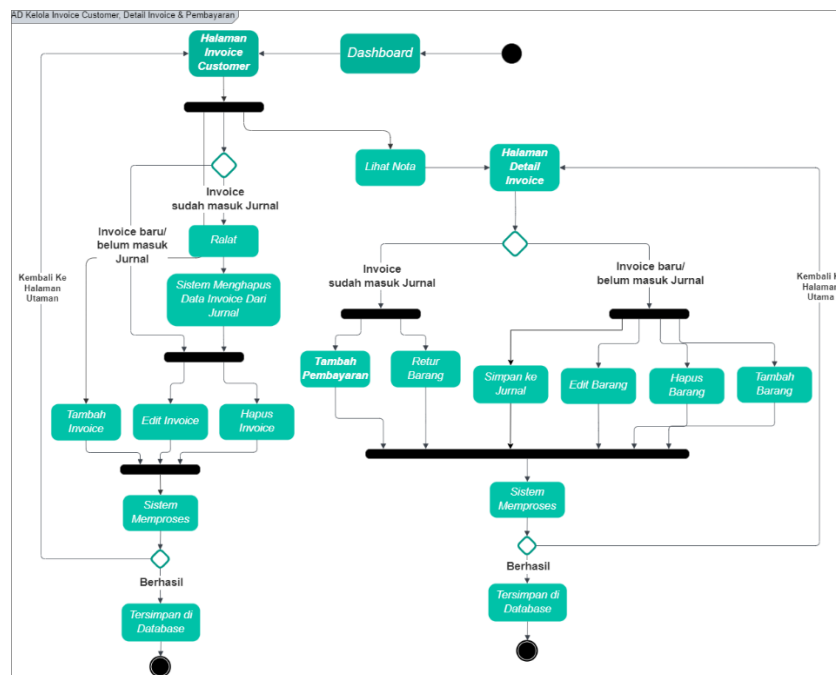
Activity Diagram diatas menunjukkan alur aktivitas yang dapat dilakukan dalam fitur Kelola *Invoice Supplier*, Detail *Invoice* dan Tambah Pembayaran. Admin dapat mengakses fitur Kelola *Invoice Supplier* untuk menambahkan *invoice* baru. Setelah dibuat, *invoice* dapat diubah atau dihapus oleh admin, kecuali jika sudah dimasukkan ke jurnal; dalam hal ini, diperlukan ralat sebelum dapat mengedit atau menghapus. Sistem akan memproses ketiga aktivitas tersebut (tambah, *edit*, hapus). Jika berhasil, data akan disimpan ke *database*; jika gagal maka akan dialihkan ke halaman utama agar admin dapat mencoba lagi.

Admin dapat mengakses halaman detail *invoice* dan tambah pembayaran melalui aktivitas “Lihat Nota”. Pada halaman ini admin dapat

melakukan tambah, *edit* dan hapus barang yang kemudian akan diproses oleh sistem. Jika berhasil, data akan tersimpan di database, jika gagal maka akan dialihkan ke halaman detail *invoice* agar admin dapat mencoba lagi.

Setelah semua data barang selesai dimasukkan admin dapat menyimpan data ke Jurnal dan mengakses fungsi retur dan tambah pembayaran.

3.6.2.4 Kelola *Invoice Customer*, Detail *Invoice* dan Pembayaran



Gambar 3.6.5 Activity Diagram Invoice Supplier & Detail Invoice

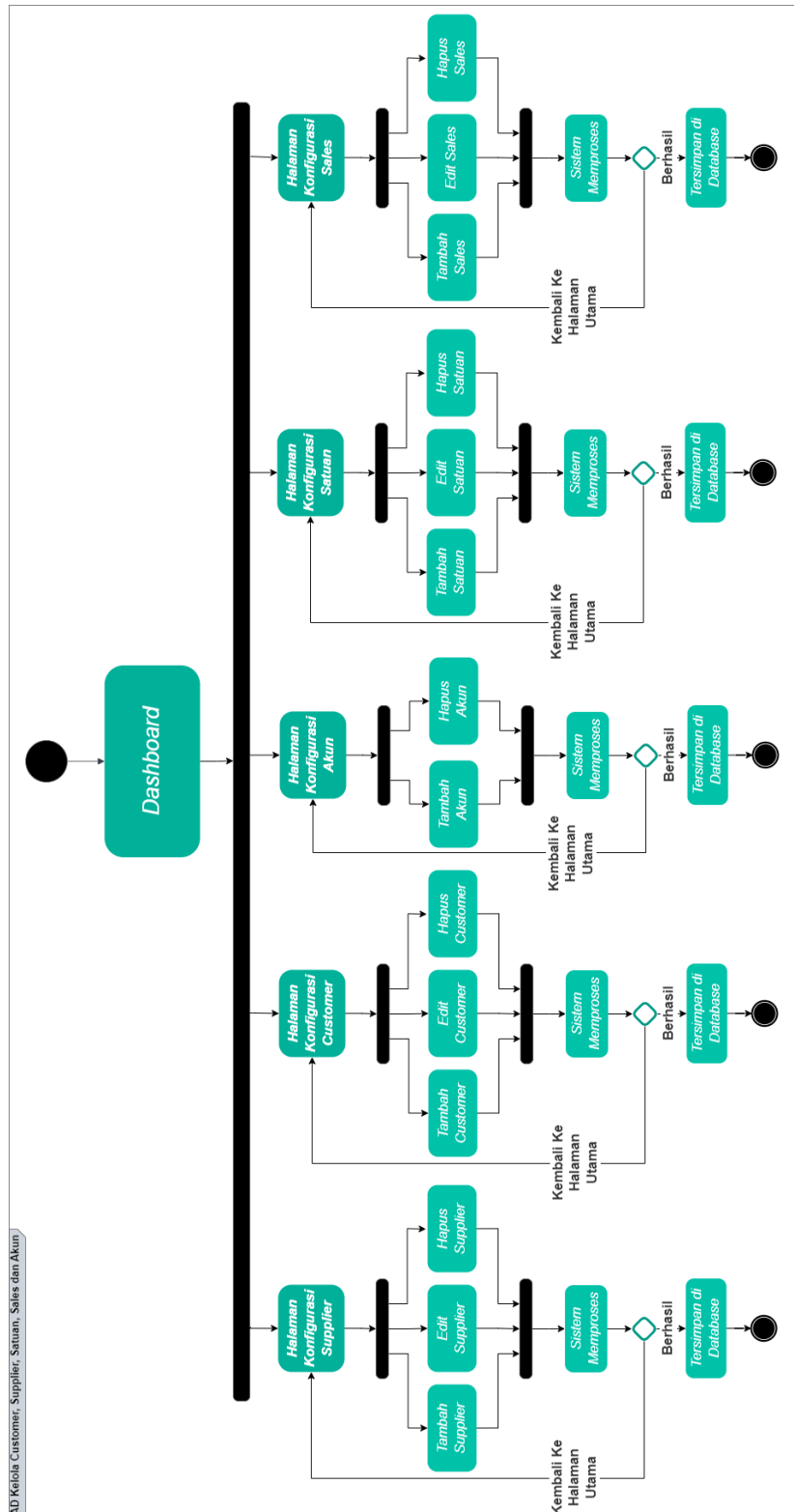
Activity Diagram diatas menunjukkan alur aktivitas yang dapat dilakukan dalam fitur Kelola *Invoice Customer*, Detail *Invoice* dan Tambah Pembayaran. Admin dapat mengakses fitur Kelola *Invoice Customer* untuk menambahkan *invoice* baru. Setelah dibuat, *invoice* dapat diubah atau dihapus oleh admin, kecuali jika sudah dimasukkan ke jurnal; dalam hal ini,

diperlukan ralat sebelum dapat mengedit atau menghapus. Sistem akan memproses ketiga aktivitas tersebut (tambah, *edit*, hapus). Jika berhasil, data akan disimpan ke *database*; jika gagal maka akan dialihkan ke halaman utama agar admin dapat mencoba lagi.

Admin dapat mengakses halaman detail *invoice* dan tambah pembayaran melalui aktivitas “Lihat Nota”. Pada halaman ini admin dapat melakukan tambah, *edit* dan hapus barang yang kemudian akan diproses oleh sistem. Jika berhasil, data akan tersimpan di *database*, jika gagal maka akan dialihkan ke halaman detail *invoice* agar admin dapat mencoba lagi.

Setelah semua data barang selesai dimasukkan admin dapat menyimpan data ke Jurnal dan mengakses fungsi retur dan tambah pembayaran.

3.6.2.5 Konfigurasi (Customer, Supplier, Satuan, Akun dan Sales)

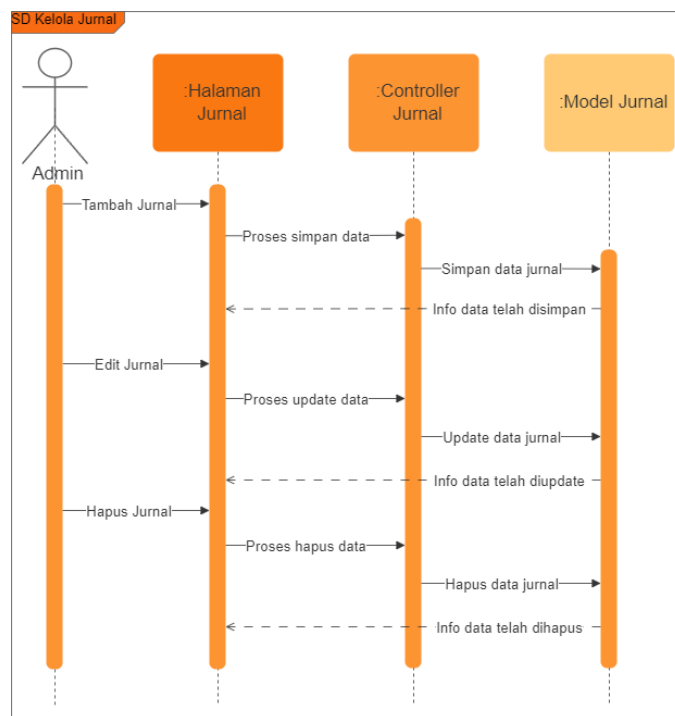


Gambar 3.6.6 Activity Diagram Konfigurasi

Activity Diagram diatas menunjukkan alur aktivitas yang dapat dilakukan dalam fitur Konfigurasi. Fitur Konfigurasi terbagi menjadi 5 yaitu Konfigurasi *Customer*, *Supplier*, Satuan, Akun dan Sales. Fitur-fitur ini dapat diakses melalui halaman *dashboard* dan memiliki alur yang sama yaitu dapat melakukan tambah, *edit* dan hapus (kecuali Akun, hanya memiliki tambah dan hapus) yang kemudian akan diproses oleh sistem. Jika berhasil, data akan tersimpan di *database*, jika gagal maka akan dialihkan ke halaman utama agar admin dapat mencoba lagi.

3.6.3 Sequence Diagram

3.6.3.1 Kelola Jurnal

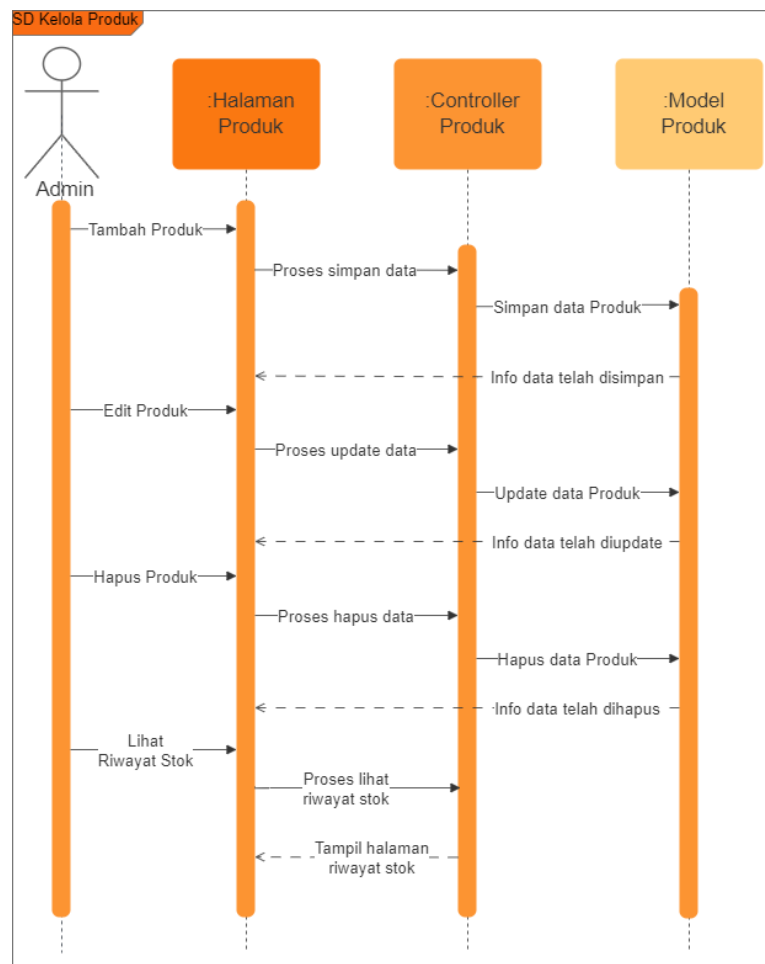


Gambar 3.6.7 Sequence Diagram Kelola Jurnal

Sequence Diagram diatas menggambarkan alur proses fitur Kelola Jurnal. Saat admin mengirim permintaan Tambah Jurnal maka data akan

diproses oleh *Controller* Jurnal yang kemudian akan disimpan ke *database* oleh *Modal* Jurnal, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan *hapus* memiliki alur yang sama hanya saja Saat mengedit, *modal* akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari *database* dan memberikan informasi tentang keberhasilan penghapusan.

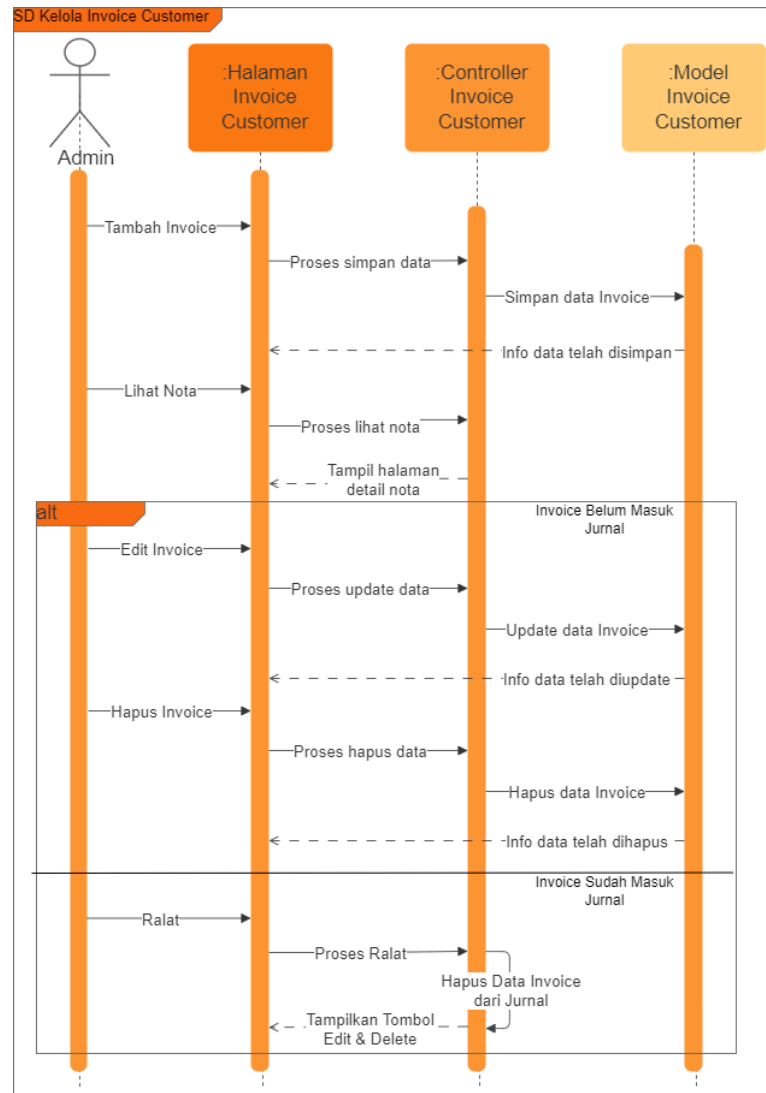
3.6.3.2 Kelola Produk



Gambar 3.6.8 Sequence Diagram Kelola Produk

Sequence Diagram diatas menggambarkan alur proses fitur Kelola Produk. Saat admin mengirim permintaan Tambah Produk data akan diproses oleh *Controller* Produk yang kemudian akan disimpan ke *database* oleh *Modal* Produk, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan hapus memiliki alur yang sama hanya saja Saat mengedit, *modal* akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, modal akan menghapus data dari database dan memberikan informasi tentang keberhasilan penghapusan. Terakhir, saat admin mengirim permintaan untuk melihat riwayat stok maka *Controller* Produk akan memproses dan menampilkan halaman Riwayat stok.

3.6.3.3 Kelola Invoice Customer



Gambar 3.6.9 Sequence Diagram Kelola Invoice Customer

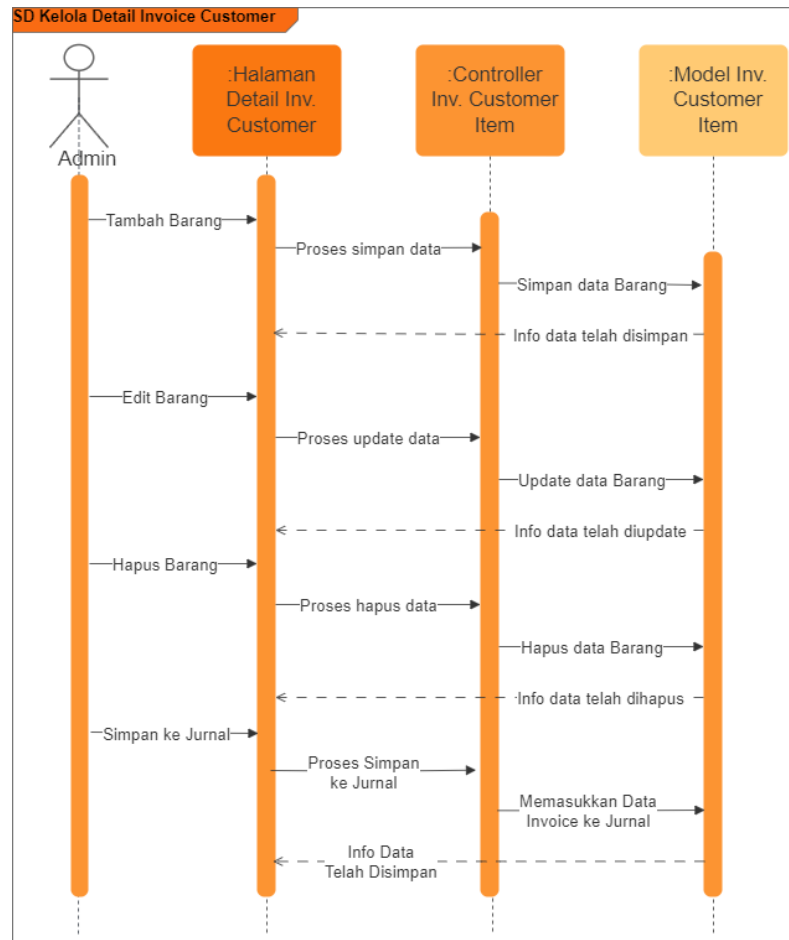
Sequence Diagram diatas menggambarkan alur proses fitur Kelola Invoice Customer. Saat admin mengirim permintaan Tambah Invoice maka data akan diproses oleh Controller Invoice Customer yang kemudian akan disimpan ke database oleh Modal Invoice Customer, kemudian Modal akan mengirimkan informasi bahwa data berhasil disimpan. saat admin mengirim

permintaan untuk melihat nota maka *Controller Invoice Customer* akan memproses dan menampilkan halaman *Detail Invoice Customer*.

Saat *invoice* belum masuk Jurnal admin dapat melakukan proses *edit* dan hapus. Proses *edit* dan hapus memiliki alur yang sama hanya saja Saat mengedit, modal akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari database dan memberikan informasi tentang keberhasilan penghapusan.

Saat *invoice* sudah masuk Jurnal admin dapat melakukan proses ralat. Saat mengirim permintaan ralat maka sistem akan memproses dan menghapus data *invoice* dari Jurnal dan menampilkan Kembali tombol *edit* & hapus.

3.6.3.4 Kelola Detail Invoice Customer

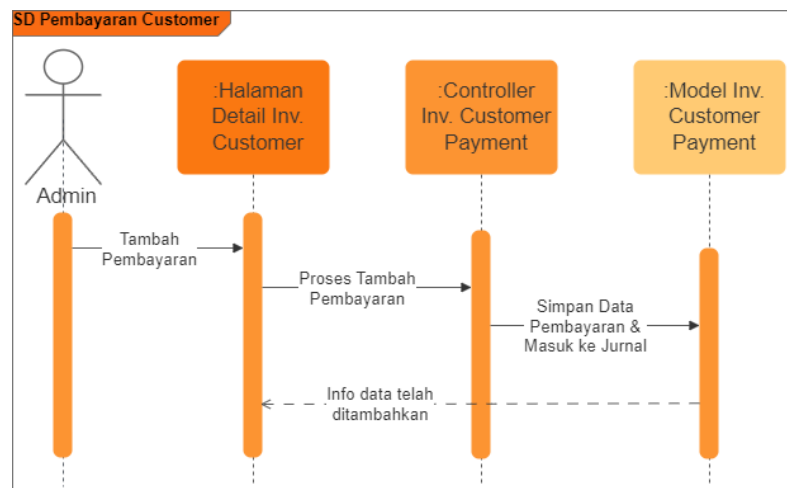


Gambar 3.6.10 Sequence Diagram Kelola Detail Invoice Customer

Sequence Diagram diatas menggambarkan alur proses fitur Kelola Detail Invoice Customer. Admin dapat melakukan proses tambah, edit, hapus dan memasukkan invoice ke jurnal. Saat admin mengirim permintaan Tambah Barang maka data akan diproses oleh Controller Invoice Customer Item yang kemudian akan disimpan ke database oleh Modal Invoice Customer Item, kemudian Modal akan mengirimkan informasi bahwa data berhasil disimpan. Proses edit dan hapus memiliki alur yang sama hanya saja Saat mengedit, modal akan memperbarui data dan memberikan

informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari database dan memberikan informasi tentang keberhasilan penghapusan. Terakhir, saat mengirim permintaan “simpan ke jurnal” maka data akan diproses oleh *Controller Invoice Customer Item* yang kemudian akan disimpan ke *database* oleh *Modal Invoice Customer Item*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan.

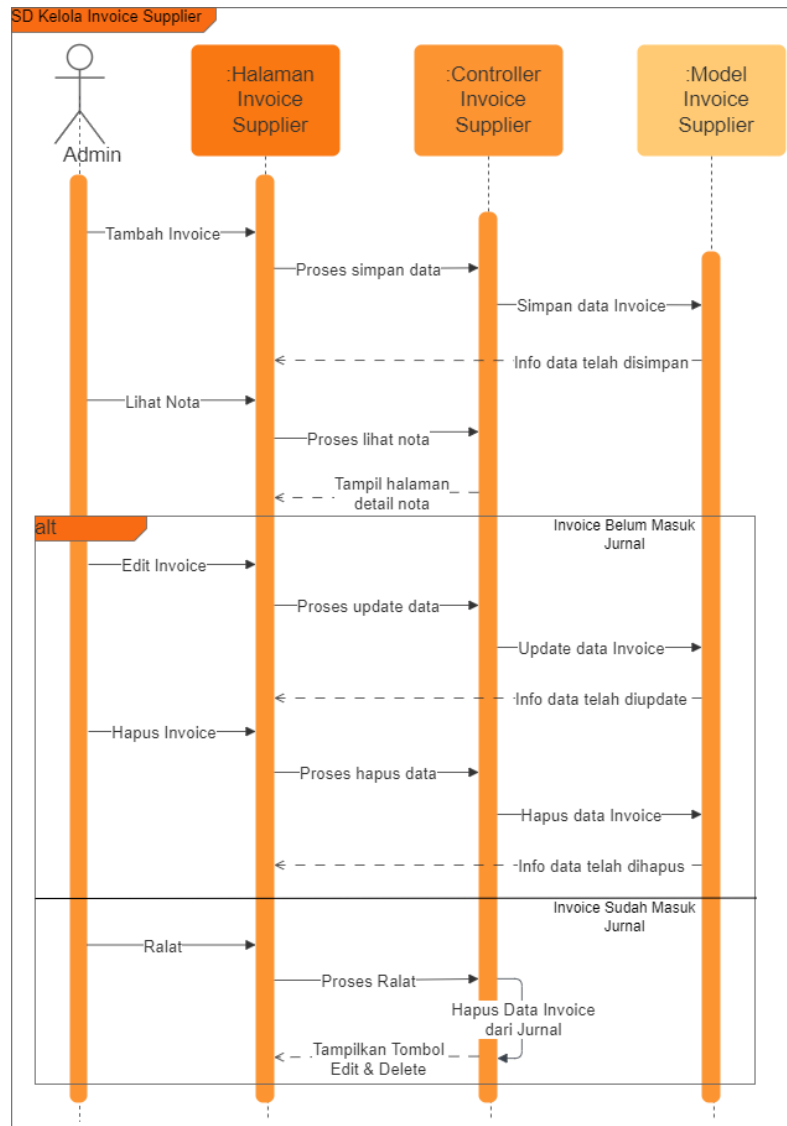
3.6.3.5 Pembayaran Customer



Gambar 3.6.11 Sequence Diagram Pembayaran Customer

Sequence Diagram diatas menggambarkan alur proses pembayaran customer. Saat admin mengirim permintaan pembayaran maka data akan diproses oleh *Controller Invoice Customer Payment* yang kemudian akan disimpan ke *database* oleh *Modal Invoice Customer Payment*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil ditambahkan.

3.6.3.6 Kelola Invoice Supplier



Gambar 3.6.12 Sequence Diagram Kelola Invoice Supplier

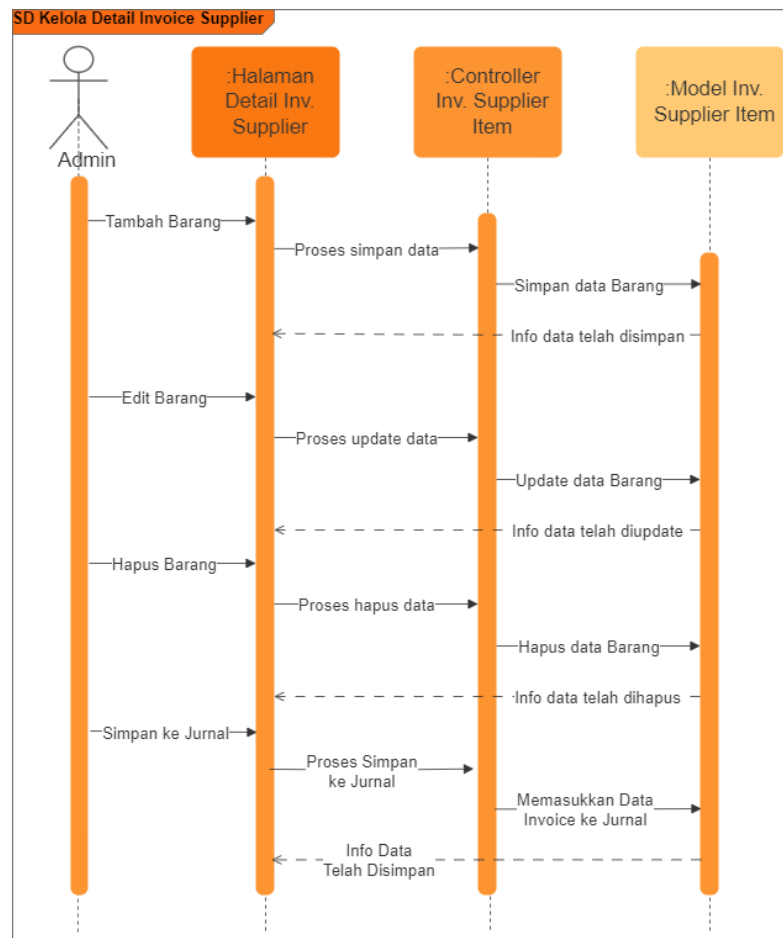
Sequence Diagram diatas menggambarkan alur proses fitur Kelola Invoice Supplier. Saat admin mengirim permintaan Tambah Invoice maka data akan diproses oleh Controller Invoice Supplier yang kemudian akan disimpan ke database oleh Modal Invoice Supplier, kemudian Modal akan mengirimkan informasi bahwa data berhasil disimpan. saat admin mengirim

permintaan untuk melihat nota maka *Controller Invoice Supplier* akan memproses dan menampilkan halaman *Detail Invoice Customer*.

Saat *invoice* belum masuk Jurnal admin dapat melakukan proses *edit* dan hapus. Proses edit dan hapus memiliki alur yang sama hanya saja Saat mengedit, modal akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari database dan memberikan informasi tentang keberhasilan penghapusan.

Saat *invoice* sudah masuk Jurnal admin dapat melakukan proses ralat. Saat mengirim permintaan ralat maka sistem akan memproses dan menghapus data *invoice* dari Jurnal dan menampilkan Kembali tombol *edit* & hapus.

3.6.3.7 Kelola Detail Invoice Supplier

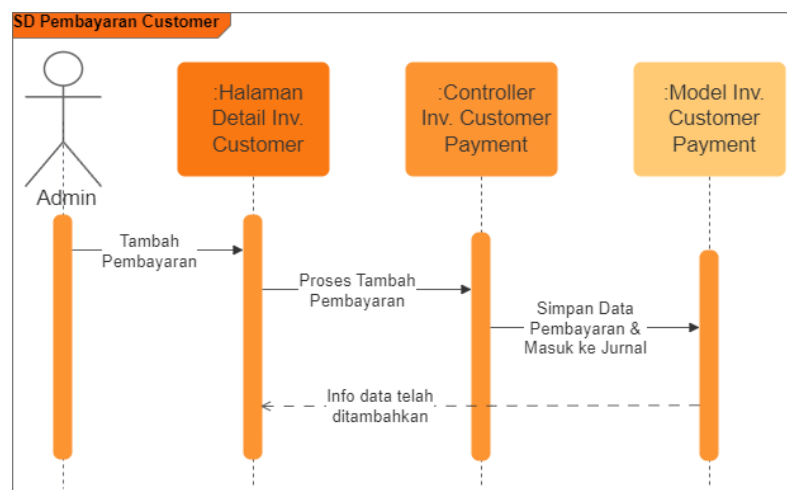


Gambar 3.6.13 *Sequence Diagram* Kelola Detail Invoice Supplier

Sequence Diagram diatas menggambarkan alur proses fitur Kelola Detail Invoice Supplier. Admin dapat melakukan proses tambah, edit, hapus dan memasukkan invoice ke jurnal. Saat admin mengirim permintaan Tambah Barang maka data akan diproses oleh *Controller Invoice Supplier Item* yang kemudian akan disimpan ke *database* oleh *Modal Invoice Supplier Item*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan hapus memiliki alur yang sama hanya saja Saat mengedit, modal akan memperbarui data dan memberikan

informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari database dan memberikan informasi tentang keberhasilan penghapusan. Terakhir, saat mengirim permintaan “simpan ke jurnal” maka data akan diproses oleh *Controller Invoice Supplier Item* yang kemudian akan disimpan ke *database* oleh *Modal Invoice Supplier Item*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan.

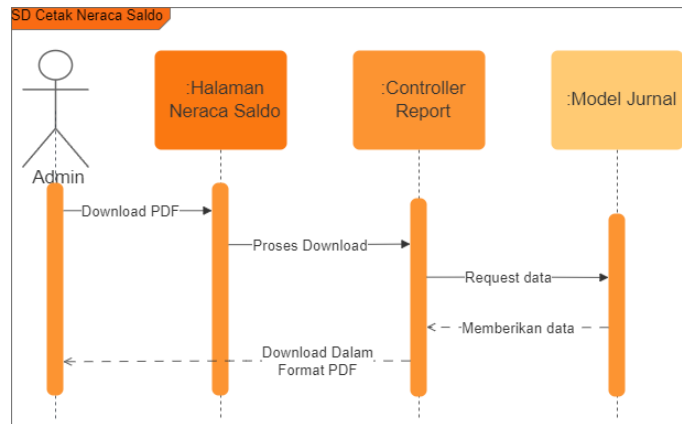
3.6.3.8 Pembayaran *Supplier*



Gambar 3.6.14 *Sequence Diagram* Pembayaran *Supplier*

Sequence Diagram diatas menggambarkan alur proses pembayaran *supplier*. Saat admin mengirim permintaan pembayaran maka data akan diproses oleh *Controller Invoice Supplier Payment* yang kemudian akan disimpan ke *database* oleh *Modal Invoice Supplier Payment*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil ditambahkan.

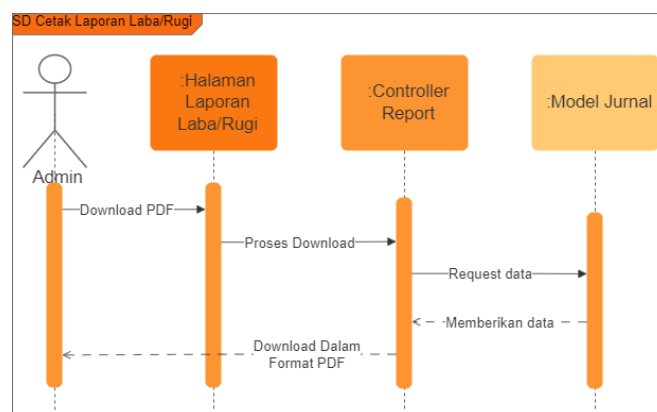
3.6.3.9 Cetak Neraca Saldo



Gambar 3.6.15 *Sequence Diagram* Cetak Neraca Saldo

Sequence Diagram diatas menggambarkan alur proses fitur Cetak Neraca Saldo. Saat admin mengirim permintaan *Download PDF* maka *Controller Report* akan memproses dan meminta data yang diperlukan dari Modal Jurnal. Kemudian *Controller* akan mengunduh data yang diminta ke dalam komputer admin dalam bentuk *PDF*.

3.6.3.10 Cetak Laporan Laba/Rugi

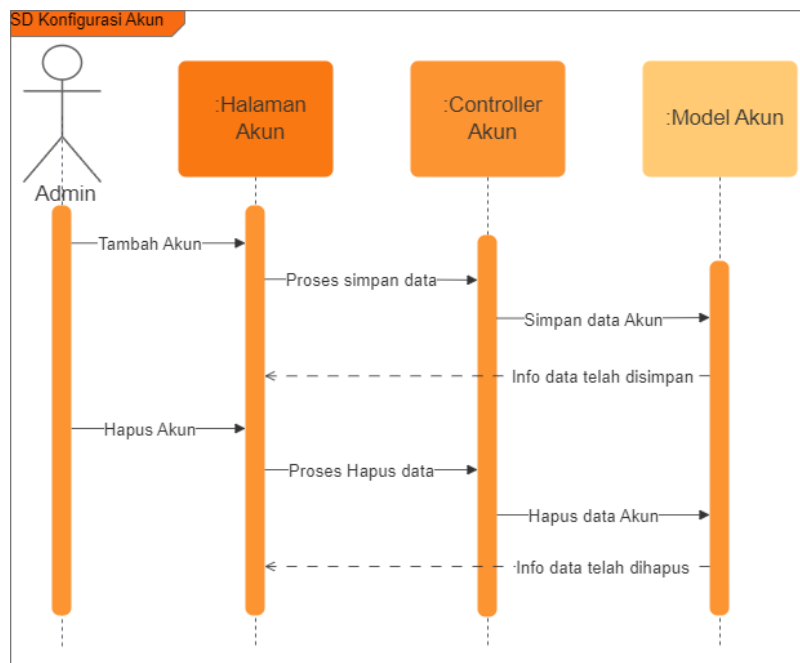


Gambar 3.6.16 *Sequence Diagram* Cetak Laporan Laba/Rugi

Sequence Diagram diatas menggambarkan alur proses fitur Cetak Laporan Laba/Rugi. Saat admin mengirim permintaan *Download PDF*

maka *Controller Report* akan memproses dan meminta data yang diperlukan dari Modal Jurnal. Kemudian *Controller* akan mengunduh data yang diminta ke dalam komputer admin dalam bentuk *PDF*.

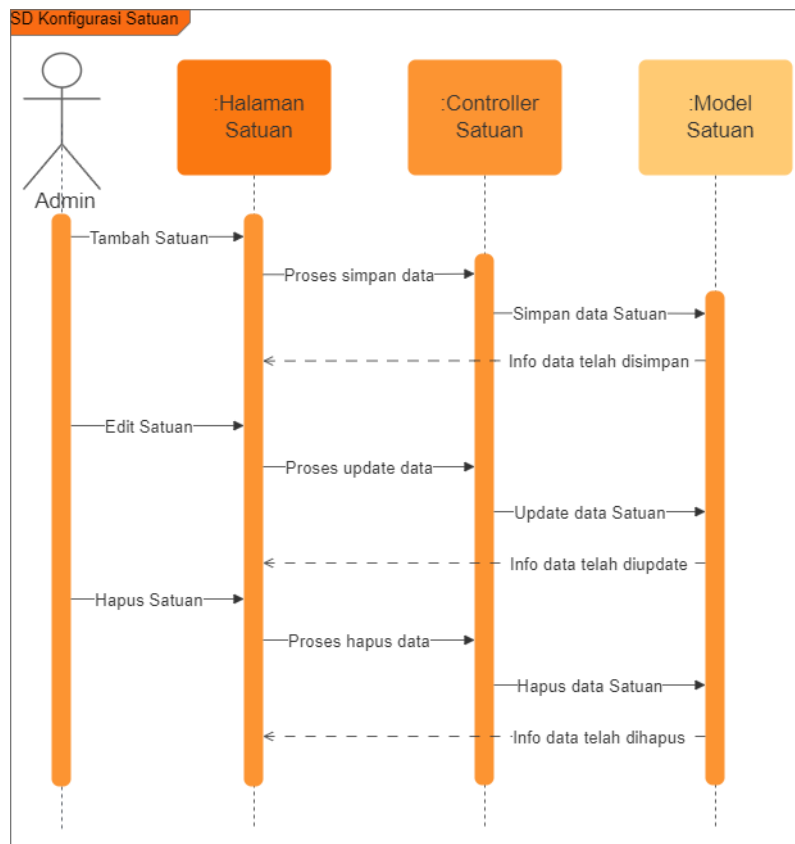
3.6.3.11 Konfigurasi Akun



Gambar 3.6.17 *Sequence Diagram* Konfigurasi Akun

Sequence Diagram diatas menggambarkan alur proses fitur Konfigurasi Akun. Saat admin mengirim permintaan Tambah Akun maka data akan diproses oleh *Controller* Akun yang kemudian akan disimpan ke *database* oleh *Modal* Akun, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses hapus memiliki alur yang sama hanya saja ketika menghapus, *modal* akan menghapus data dari *database* dan memberikan informasi tentang keberhasilan penghapusan.

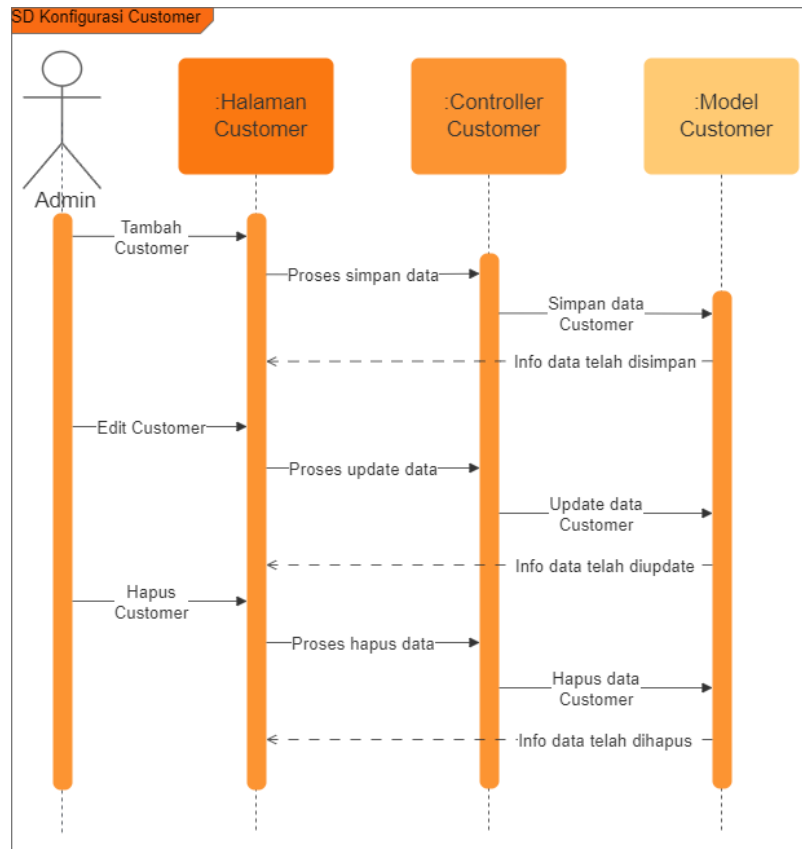
3.6.3.12 Konfigurasi Satuan



Gambar 3.6.18 *Sequence Diagram* Konfigurasi Satuan

Sequence Diagram diatas menggambarkan alur proses fitur Konfigurasi Satuan. Saat admin mengirim permintaan Tambah Satuan maka data akan diproses oleh *Controller* Satuan yang kemudian akan disimpan ke *database* oleh *Modal* Satuan, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan hapus memiliki alur yang sama hanya saja Saat mengedit, *modal* akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari *database* dan memberikan informasi tentang keberhasilan penghapusan.

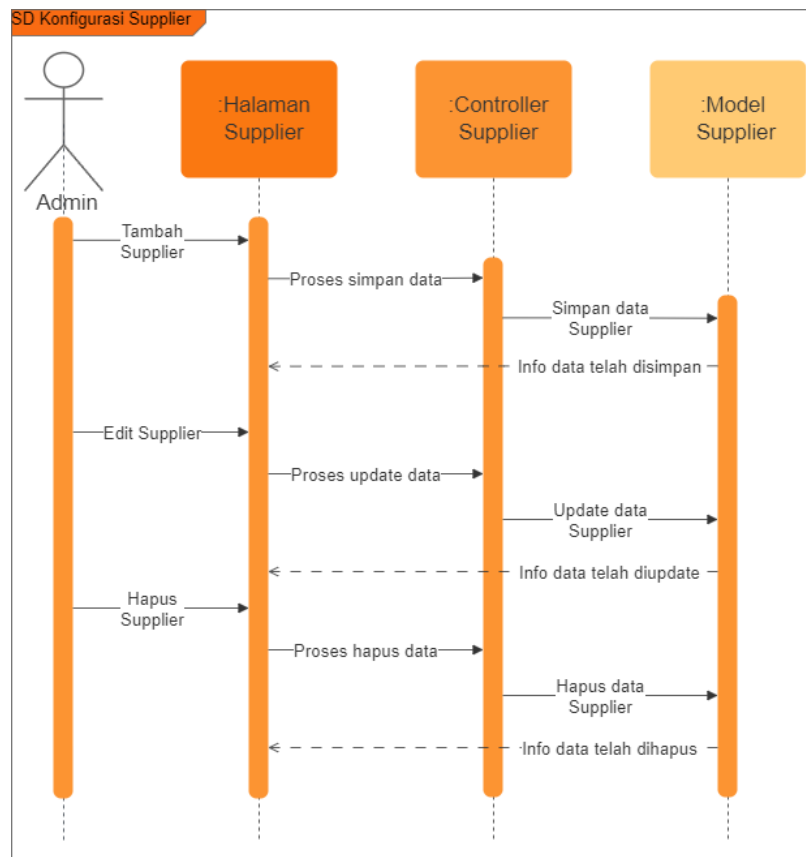
3.6.3.13 Konfigurasi *Customer*



Gambar 3.6.19 *Sequence Diagram* Konfigurasi *Customer*

Sequence Diagram diatas menggambarkan alur proses fitur Konfigurasi *Customer*. Saat admin mengirim permintaan *Tambah Customer* maka data akan diproses oleh *Controller Customer* yang kemudian akan disimpan ke *database* oleh *Modal Customer*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan *hapus* memiliki alur yang sama hanya saja Saat mengedit, *modal* akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari *database* dan memberikan informasi tentang keberhasilan penghapusan.

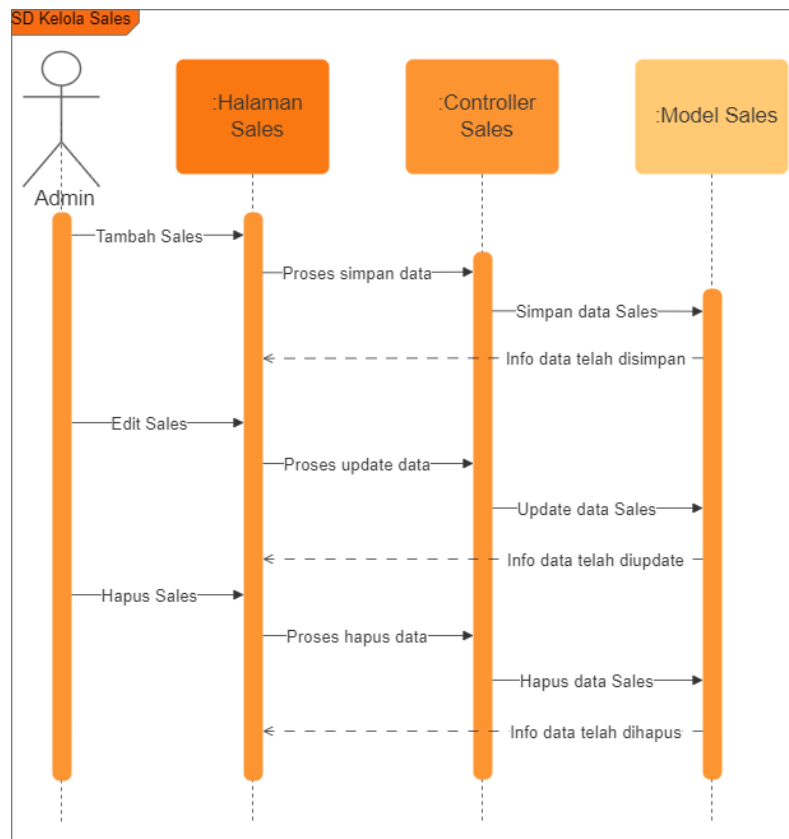
3.6.3.14 Konfigurasi *Supplier*



Gambar 3.6.20 *Sequence Diagram* Konfigurasi *Supplier*

Sequence Diagram diatas menggambarkan alur proses fitur Konfigurasi *Supplier*. Saat admin mengirim permintaan Tambah *Supplier* maka data akan diproses oleh *Controller Supplier* yang kemudian akan disimpan ke *database* oleh *Modal Supplier*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan hapus memiliki alur yang sama hanya saja Saat mengedit, *modal* akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari *database* dan memberikan informasi tentang keberhasilan penghapusan.

3.6.3.15 Konfigurasi Sales



Gambar 3.6.21 *Sequence Diagram* Konfigurasi Sales

Sequence Diagram diatas menggambarkan alur proses fitur Konfigurasi Sales. Saat admin mengirim permintaan Tambah Sales maka data akan diproses oleh *Controller Sales* yang kemudian akan disimpan ke *database* oleh *Modal Sales*, kemudian *Modal* akan mengirimkan informasi bahwa data berhasil disimpan. Proses *edit* dan hapus memiliki alur yang sama hanya saja Saat mengedit, *modal* akan memperbarui data dan memberikan informasi tentang keberhasilan perubahan. Ketika menghapus, *modal* akan menghapus data dari *database* dan memberikan informasi tentang keberhasilan penghapusan.

3.6.4 Entity Relationship Diagram



Gambar 3.6.22 Entity Relationship Diagram T.B Samala

ERD Sistem Informasi Akuntansi ini akan dibagi menjadi 4 bagian yaitu bagian Tengah yang berisi tabel *chart_of_account*, *journal* dan *sales*. Bagian kiri terdiri dari tabel *customer*, *customer_invoice*, *customer_invoice_items* dan *customer_invoice_payment*. Bagian Kanan terdiri dari tabel *supplier*, *supplier_invoice*, *supplier_invoice_items* dan *supplier_invoice_payment*. Terakhir adalah bagian bawah yang terdiri dari tabel *product*, *unit* dan *stock*.

3.6.4.1 Tabel *chart_of_account*

Tabel pertama bagian Tengah adalah *chart_of_account* memiliki 4 *field* yaitu *id*, *name*, *type* dan *relation*. Tabel ini berfungsi untuk menyimpan data dari akun-akun akuntansi yang dipakai oleh jurnal.

3.6.4.2 Tabel *journal*

Tabel kedua bagian Tengah adalah tabel *journal* memiliki 10 *field* yaitu *id*, *date*, *description*, *field_sales*, *store*, *amount*, *debit_credit*, *coa_id*, *supplier_invoice_id*, *customer_invoice_id*. Tabel ini berfungsi untuk menyimpan data jurnal akuntansi. Tabel ini memiliki relasi dengan tabel *chart_of_account*, *supplier_invoice* dan *customer_invoice*.

3.6.4.3 Tabel *sales*

Tabel ketiga bagian Tengah adalah tabel *sales* memiliki 2 *field* yaitu *id* dan *name*. Tabel ini berfungsi untuk menyimpan data nama dari para sales T.B Samala.

3.6.4.4 Tabel *customer*

Tabel pertama bagian kiri adalah *customer* memiliki 7 *field* yaitu *id*, *name*, *address*, *phone_number*, *phone_number_2*, *phone_number_3* dan *description*. Tabel ini berfungsi untuk menyimpan data para *customer* dari T.B Samala.

3.6.4.5 Tabel *customer_invoice*

Tabel kedua adalah *customer_invoice* memiliki 8 *field* yaitu *id*, *invoice_number*, *purchase_date*, *tax*, *sales_id*, *customer_id*, *status*, *saved*. Tabel ini berfungsi untuk menyimpan *invoice*/nota penjualan T.B Samala. Tabel ini berelasi dengan tabel *sales* dan *customer*.

3.6.4.6 Tabel *customer_invoice_items*

Tabel ketiga adalah *customer_invoice_items* memiliki 9 *field* yaitu *id*, *qty*, *discount_1*, *discount_2*, *discount_3*, *price*, *is_return*, *product_id*, *customer_invoice_id*. Tabel ini berfungsi untuk menyimpan data barang dari *invoice*/nota customer. Tabel ini memiliki relasi dengan tabel *customer_invoice* dan tabel *product*.

3.6.4.7 Tabel *customer_invoice_payment*

Tabel keempat adalah *customer_invoice_payment* memiliki 5 *field* yaitu *id*, *payment_date*, *amount*, *payment_method*, *customer_invoice_id*. Tabel ini berfungsi untuk menyimpan data pembayaran *invoice*/nota customer. Tabel ini berelasi dengan tabel *customer_invoice*.

3.6.4.8 Tabel *supplier*

Tabel pertama bagian kanan adalah *supplier* memiliki 7 *field* yaitu *id*, *company*, *address*, *phone_number*, *phone_number_2*, *phone_number_3* dan *description*. Tabel ini berfungsi untuk menyimpan data para *supplier* dari T.B Samala.

3.6.4.9 Tabel *supplier_invoice*

Tabel kedua adalah *supplier_invoice* memiliki 7 *field* yaitu *id*, *purchase_date*, *tax*, *sales_id*, *supplier_id*, *status*, *saved*. Tabel ini berfungsi untuk menyimpan *invoice*/nota pembelian T.B Samala. Tabel ini berelasi dengan tabel sales dan *supplier*.

3.6.4.10 Tabel *supplier_invoice_items*

Tabel ketiga adalah *supplier_invoice_items* memiliki 9 *field* yaitu *id*, *qty*, *discount_1*, *discount_2*, *discount_3*, *price*, *is_return*, *product_id*, *supplier_invoice_id*. Tabel ini berfungsi untuk menyimpan data barang dari *invoice*/nota *supplier*. Tabel ini memiliki relasi dengan tabel *supplier_invoice* dan tabel *product*.

3.6.4.11 Tabel *supplier_invoice_payment*

Tabel keempat adalah *supplier_invoice_payment* memiliki 5 *field* yaitu *id*, *payment_date*, *amount*, *payment_method*, *supplier_invoice_id*. Tabel ini berfungsi untuk menyimpan data pembayaran *invoice*/nota *supplier*. Tabel ini berelasi dengan tabel *supplier_invoice*.

3.6.4.12 Tabel *Unit*

Tabel pertama dari bagian bawah adalah tabel *unit* memiliki 3 *field* yaitu *id*, *name*, *symbol*. Tabel ini berfungsi untuk menyimpan jenis satuan dari barang seperti kg, pcs, dll.

3.6.4.13 Tabel *Product*

Tabel kedua adalah tabel *product* memiliki 5 *field* yaitu *id*, *product_name*, *unit_id*, *buy_price*, *sell_price*. Tabel ini berfungsi untuk menyimpan daftar barang atau produk yang dimiliki T.B Samala. Tabel ini berelasi dengan tabel *unit*.

3.6.4.14 Tabel *stock*

Tabel ketiga dan terakhir adalah tabel *stock* memiliki 10 *field* yaitu *id*, *product_id*, *date*, *description*, *price*, *qty*, *status*, *store*, *customer_invoice_item_id*, *supplier_invoice_item_id*. Tabel ini berfungsi untuk menyimpan data stok dari barang atau produk yang dimiliki T.B Samala. Tabel ini memiliki relasi dengan tabel *customer_invoice_item*, tabel *supplier_invoice_item* dan tabel *product*.