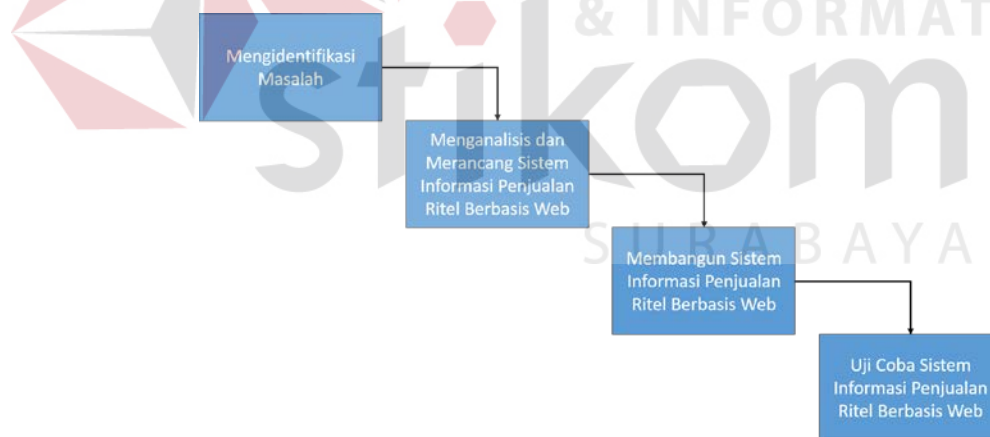


BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas tentang identifikasi masalah, analisis sistem, perancangan sistem, rancangan pengujian dan evaluasi sistem dalam Rancang Bangun Sistem Informasi Penjualan Berbasis Web Untuk Usaha Mikro dan Kecil. Model pengembangan yang digunakan pada penelitian ini adalah *waterfall* dengan tahapan-tahapan yang telah ditentukan, yaitu mengidentifikasi masalah, menganalisis dan merancang sistem informasi penjualan ritel berbasis web, membangun sistem dan uji coba sistem informasi penjualan ritel berbasis web. Secara garis besar digambarkan seperti di bawah ini.



Gambar 3.1 Tahapan-tahapan Pembuatan Sistem

3.1 Identifikasi Permasalahan

Untuk mengidentifikasi masalah yang ada pada ritel tradisional, maka dilakukan wawancara dan observasi pada beberapa pemilik toko ritel tradisional. Ritel tradisional terdapat dua bagian penting dalam proses bisnisnya yaitu bagian penjualan dan pemilik toko. Namun dalam beberapa toko yang mampu

memperkerjakan banyak pegawai, pada toko tersebut terdapat pengawas toko atau dapat dikatakan sebagai supervisor atau manajer yang berfungsi untuk mengatur dan mengawasi toko ketika pemilik toko tidak berada di tempat.

Kasir dan pramuniaga merupakan deskripsi pekerjaan yang ada pada bagian penjualan. Kasir memiliki tanggung jawab untuk mencatat semua transaksi penjualan, menerima retur penjualan, dan membuat laporan pendapatan. Sedangkan pramuniaga memiliki tanggung jawab untuk melayani pelanggan, memberikan informasi yang benar kepada pelanggan terkait produk yang dijual, menjaga dan mengawasi barang-barang yang dijual. Namun bagi beberapa toko yang belum mampu untuk memperkerjakan banyak pegawai, pramuniaga diwajibkan untuk merangkap sebagai kasir. Selain sebagai kasir, pramuniaga juga mempunyai pekerjaan lain yang sebenarnya bukan bagian dari pekerjaannya yaitu menghitung jumlah stok barang yang tersisa di toko. Bagian penjualan sering memiliki masalah hilang atau rusaknya nota penjualan yang akan direkap bahkan di beberapa toko transaksi penjualan tidak menggunakan nota penjualan melainkan hanya sekedar dicatat pada buku besar. Hilang dan rusaknya nota-nota tersebut menjadikan informasi penjualan yang dihasilkan menjadi kurang tepat dan tidak bisa dijadikan tolak ukur bagi pemilik toko untuk pengambilan keputusan dalam mengembangkan tokonya. Selain itu, proses penghitungan dan perekapan data yang dilakukan secara manual memerlukan waktu yang lama dan sering terjadi kesalahan perhitungan terutama ketika transaksi dengan jenis dan jumlah barang yang banyak.

Pemilik toko memiliki tanggung jawab untuk mengelola dan mengawasi semua kegiatan operasional toko berdasarkan informasi yang diberikan oleh

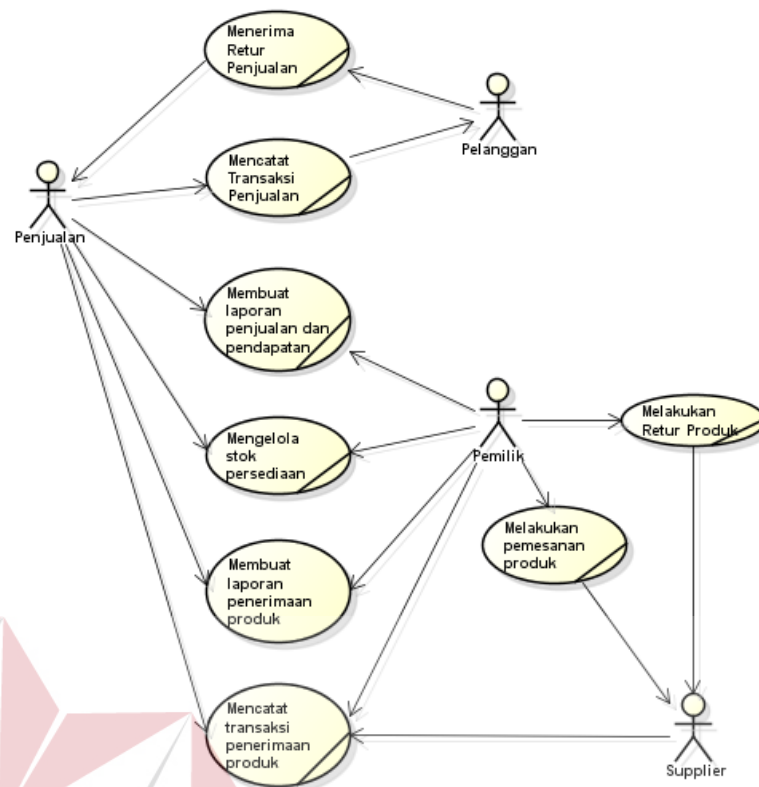
bagian penjualan dalam laporan mereka. Pemilik toko juga memiliki tanggung jawab untuk pengelolaan stok barang seperti mencatat transaksi penerimaan barang sekaligus memperbarui jumlah stok dan membuat laporan penerimaan barang, dalam melakukan tugas tersebut pemilik toko dibantu oleh pegawainya yang dalam hal ini kasir atau pramuniaga. Namun ada beberapa permasalahan yang dialami oleh pemilik toko yaitu pemilik tidak dapat mengetahui tren penjualan yang diperlukan untuk mengetahui lifetime sebuah produk dan menentukan jumlah pemesanan kepada supplier pada periode berikutnya. Akibatnya jenis barang dagangan yang menjadi tidak terarah dan menyebabkan kerugian karena membeli barang yang sudah tidak laku dalam jumlah banyak.

Pemilik toko juga mengalami kesulitan mengetahui stok yang tersedia karena harus menghitung terlebih dahulu dan hal tersebut membutuhkan waktu yang tidak sebentar. Selain itu pemilik juga sering mengalami kehabisan stok atau penumpukan stok karena kesalahan perkiraan waktu saat reoder ke supplier. Pemilik tidak memiliki data history transaksi penjualan per periode baik minggu, bulan atau tahun. Data history tersebut dapat dimanfaatkan untuk mengetahui barang yang paling sering dibeli, menentukan target penjualan di periode selanjutnya, dan mengevaluasi produk yang selama ini dijual. Selain hal di atas masalah lain yang dialami adalah terkendala biaya dalam pengadaan perangkat teknologi informasi yang dapat menunjang proses bisnis seperti, biaya maintenance yang cukup tinggi untuk perangkat teknologi informasi serta SDM (sumber daya manusia) yang melakukan maintenance. Informasi yang dihasilkan juga tidak dapat diakses kapanpun dan dimanapun.

Proses promosi barang yang selama ini dilakukan hanya mengandalkan sebuah papan yang berisi promo potongan harga yang dipajang di depan toko, mengandalkan promosi dari mulut ke mulut dan sebaran brosur yang tidak dapat diketahui secara pasti peningkatan penjualan yang dialami dari proses tersebut. Hal ini menyebabkan pelanggan dari toko tersebut terbatas pada lingkungan mereka saja. Selain itu pemilik toko juga kurang memanfaatkan internet untuk dapat memasarkan produknya secara luas sehingga produk yang dijual tidak diketahui oleh banyak orang yang bisa berpotensi menjadi pelanggan mereka.

3.2 Analisis Sistem

Tahap selanjutnya yang dilakukan adalah menganalisis sistem penjualan yang telah ada saat ini. Dengan adanya tahap ini maka akan dapat diketahui kelemahan-kelemahan yang ada pada sistem penjualan saat ini dan kebutuhan yang perlu dipenuhi. Berikut diagram *business use case* yang digunakan untuk menggambarkan model bisnis dari sistem penjualan yang ada pada saat ini:



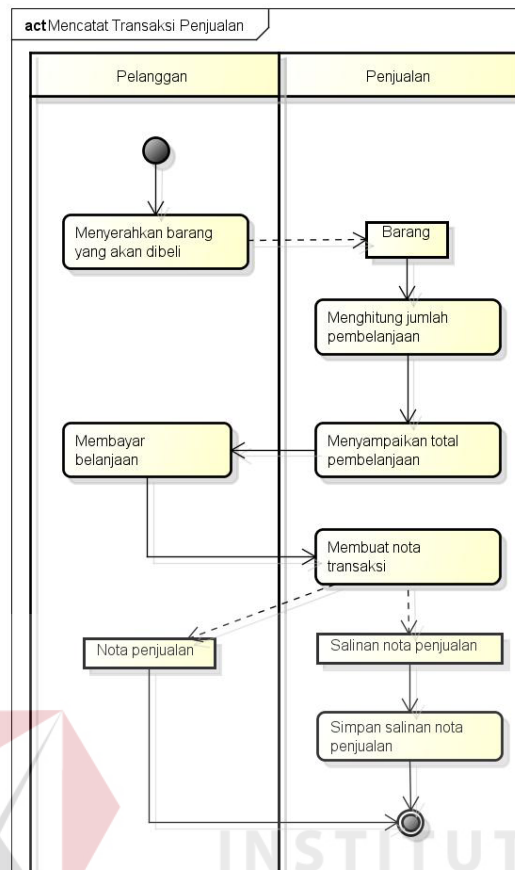
Gambar 3.2 Business Use Case Saat Ini

Activity Diagram di bawah ini akan menjelaskan secara detail masing-masing proses bisnis pada diagram bisnis *use case* di atas.

3.2.1 Activity Diagram

3.2.1.1 Mencatat Transaksi Penjualan

Proses mencatat transaksi penjualan dimulai dari pelanggan memilih dan menyerahkan produk yang akan dibeli kepada bagian penjualan. Bagian penjualan menghitung produk yang dibeli oleh pelanggan satu persatu menggunakan kalkulator. Setelah melakukan perhitungan, bagian penjualan menyampaikan harga yang harus dibayarkan oleh pelanggan. Setelah itu pelanggan melakukan pembayaran dan bagian penjualan menyerahkan produk yang dibeli dan nota berwarna putih kepada pelanggan dan nota berwarna merah disimpan sebagai rekam. Untuk lebih jelasnya dapat melihat Gambar 3.3.

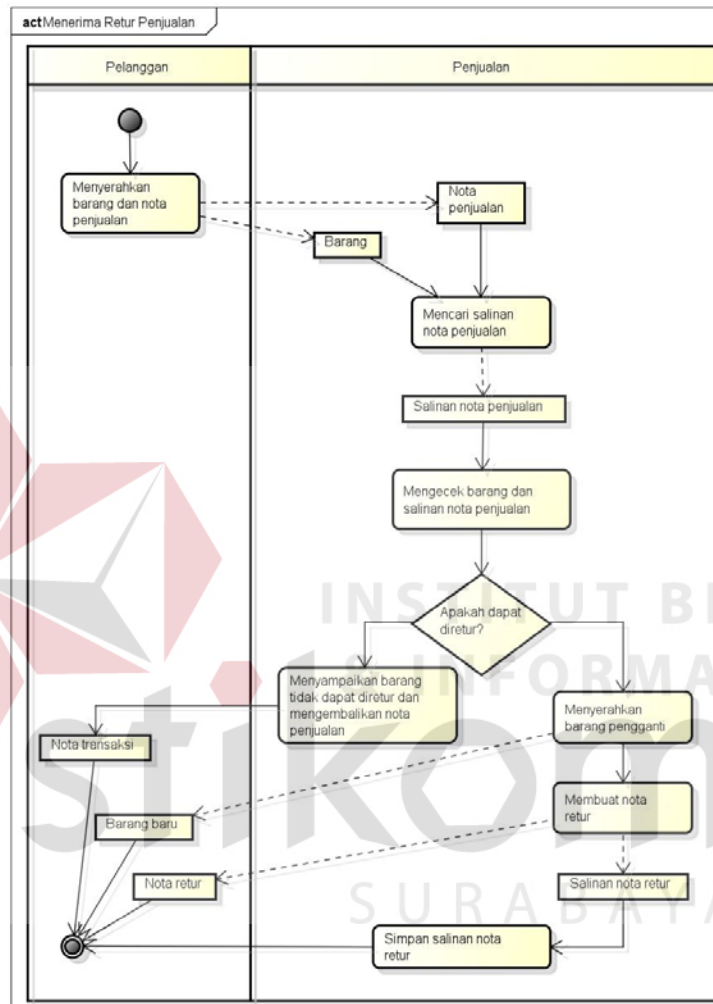


Gambar 3.3 Activity Diagram Mencatat Transaksi Penjualan

3.2.1.2 Menerima Retur Penjualan

Proses menerima retur penjualan dimulai dari pelanggan menyerahkan nota penjualan produk yang akan diretur. Namun apabila pelanggan tidak dapat menunjukkan nota penjualan maka secara otomatis bagian penjualan tidak dapat mengganti produk tersebut dengan yang baru. Setelah menerima nota penjualan, bagian penjualan akan mencari salinan nota penjualan tersebut lalu pelanggan menyerahkan produk yang akan diretur. Sebelum mengganti produk dengan yang baru, bagian penjualan mengecek terlebih dahulu produk dan nota penjualan yang dikembalikan oleh pelanggan. Hal tersebut untuk mengetahui apakah produk yang diretur dapat diganti dengan yang baru. Jika tidak, bagian penjualan menyampaikan ke pelanggan dan mengembalikan nota penjualan. Jika ya, bagian

penjualan menyerahkan barang pengganti lalu membuat nota retur. Bagian penjualan menyerahkan nota berwarna putih ke pelanggan dan nota berwarna merah disimpan sebagai rekap. Untuk lebih jelasnya dapat melihat Gambar 3.4.

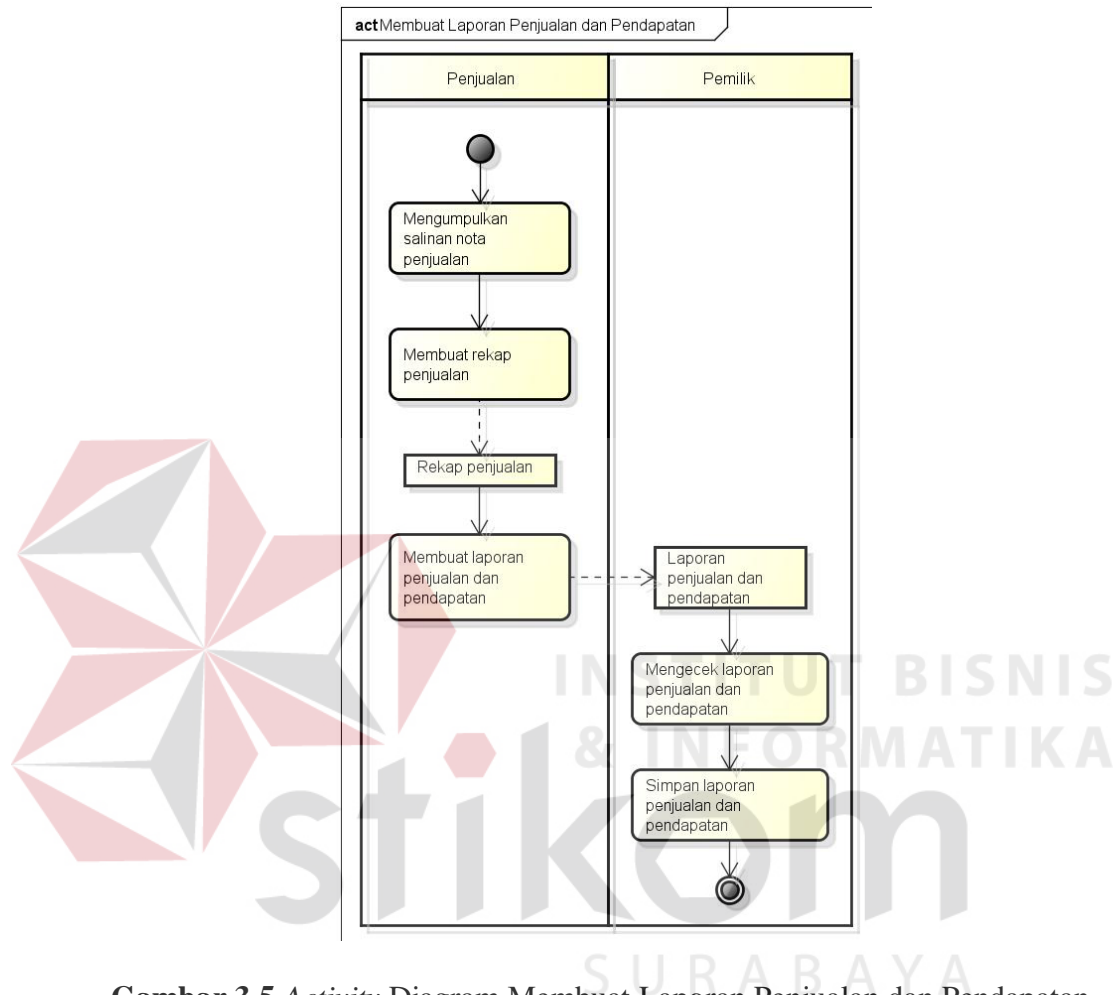


Gambar 3.4 Activity Diagram Menerima Retur Penjualan

3.2.1.3 Membuat Laporan Penjualan dan Pendapatan

Untuk proses membuat laporan penjualan dan pendapatan bagian penjualan mengumpulkan salinan nota penjualan untuk kemudian dihitung dan direkap. Setelah itu bagian penjualan membuat laporan penjualan dan pendapat yang kemudian dilaporkan kepada pemilik. Setelah mendapat laporan penjualan dan pendapatan dari bagian penjualan, pemilik toko mengecek kembali kebenaran

dari laporan yang dibuat dan kemudian menyimpan laporan tersebut. Untuk lebih jelasnya dapat melihat Gambar 3.5.



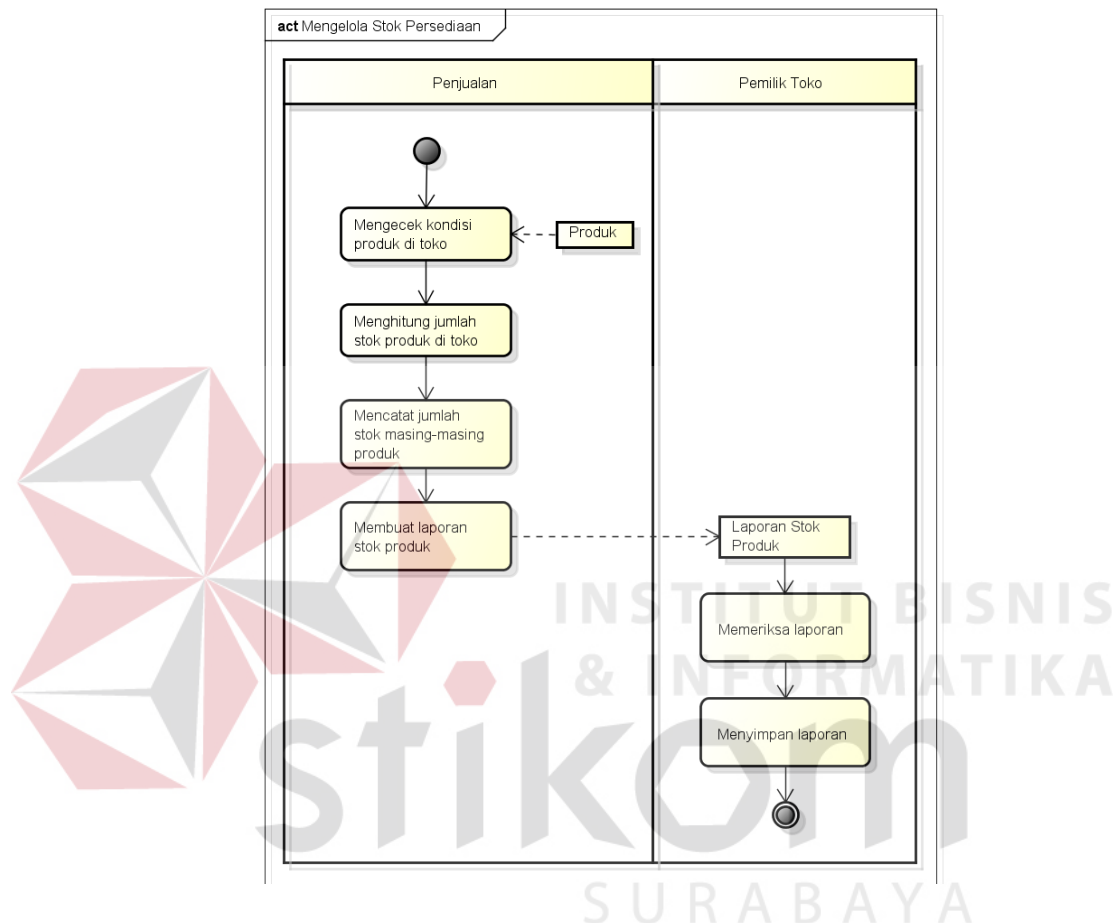
Gambar 3.5 Activity Diagram Membuat Laporan Penjualan dan Pendapatan

3.2.1.4 Mengelola Stok Persediaan

Proses mengelola stok persediaan dimulai dengan bagian penjualan melakukan perhitungan fisik produk yang ada pada toko ataupun gudang untuk kemudian dicatat jumlah total masing-masing produk yang tersedia. Dalam proses pengelolaan ini bagian penjualan juga mengecek kondisi produk yang berada ditoko ataupun digudang. Setelah selesai bagian penjualan melakukan *update* data stok untuk kemudian diberikan kepada pemilik toko. Sebelum menerima dan

menyetujui data tersebut pemilik toko terlebih dahulu memeriksa laporan dan mencocokkan dengan kondisi sebenarnya yang terdapat ditoko ataupun digudang.

Untuk lebih jelasnya dapat melihat Gambar 3.6.

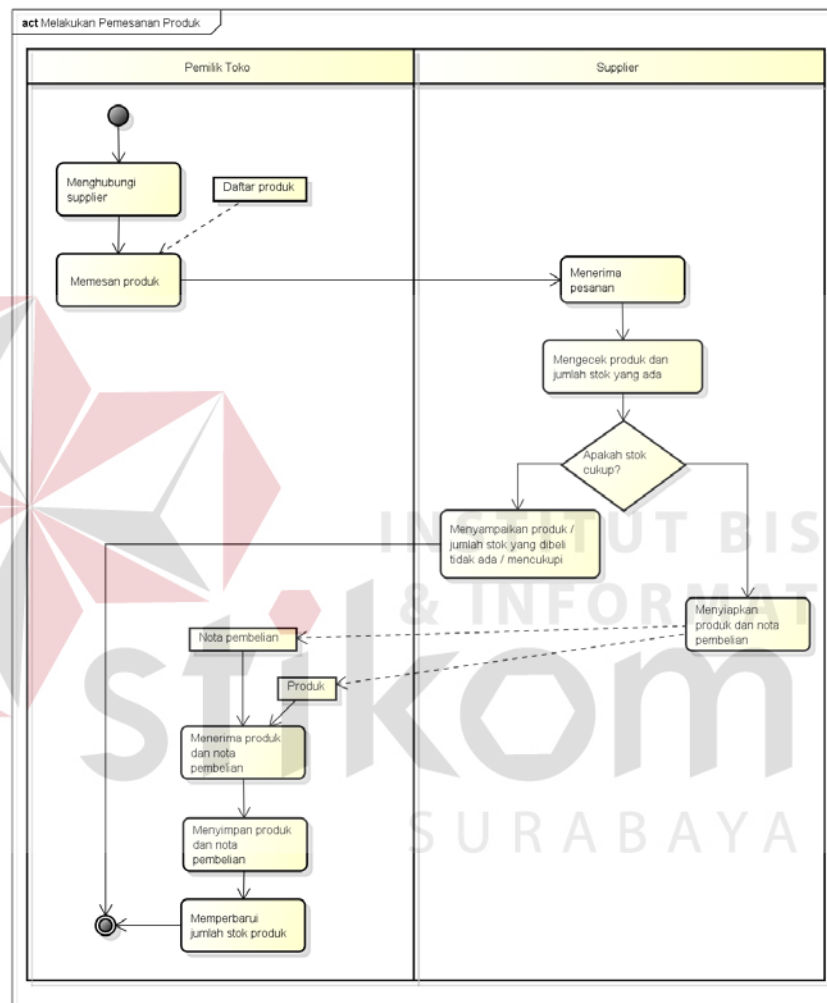


Gambar 3.6 Activity Diagram Mengelola Stok Persediaan

3.2.1.5 Melakukan Pemesanan Produk

Proses melakukan pemesanan produk dimulai ketika setelah proses mengelola stok persediaan selesai dilakukan karena setelah proses tersebut selesai dilakukan dapat diketahui mana saja produk yang harus dilakukan pemesanan ulang kepada supplier. Pemilik toko mencatat produk mana saja yang harus dilakukan pemesanan ulang kepada supplier. Setelah itu pemilik toko melakukan pemesanan produk kepada supplier baik melalui secara langsung ataupun melalui

panggilan telepon. Supplier mengirim pesanan kepada pemilik toko, kemudian pemilik toko mencocokkan produk yang dikirim sesuai dengan pesanan. Setelah selesai pemilik toko melakukan *update* data stok. Untuk lebih jelasnya dapat melihat Gambar 3.7.

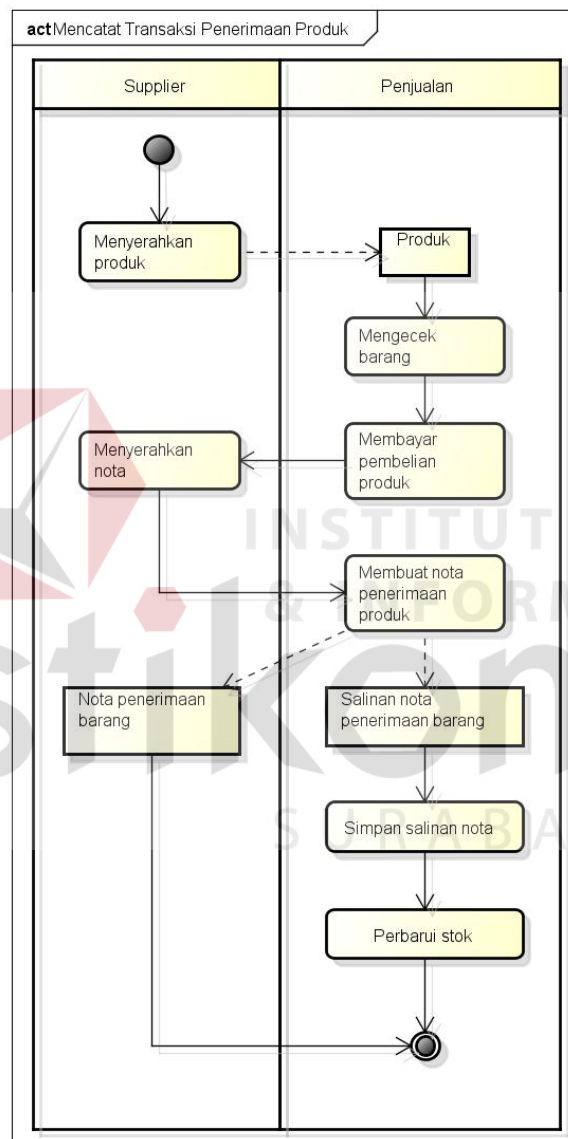


Gambar 3.7 Activity Diagram Melakukan Pemesanan Produk

3.2.1.6 Mencatat Transaksi Penerimaan Produk

Proses pencatatan penerimaan produk dimulai dari supplier menyerahkan produk ke bagian penjualan. Bagian penjualan mengecek kondisi produk yang diterima dan mencocokkan dengan pemesan yang dilakukan. Setelah itu penjualan akan membayar dan menerima nota transaksi dari supplier. Bagian penjualan akan

membuat nota penerimaan produk dua rangkap lalu menyerahkan nota berwarna putih kepada supplier dan nota berwarna merah akan disimpan sebagai rekap. Bagian penjualan juga akan melakukan update data stok. Untuk lebih jelasnya dapat melihat Gambar 3.8.

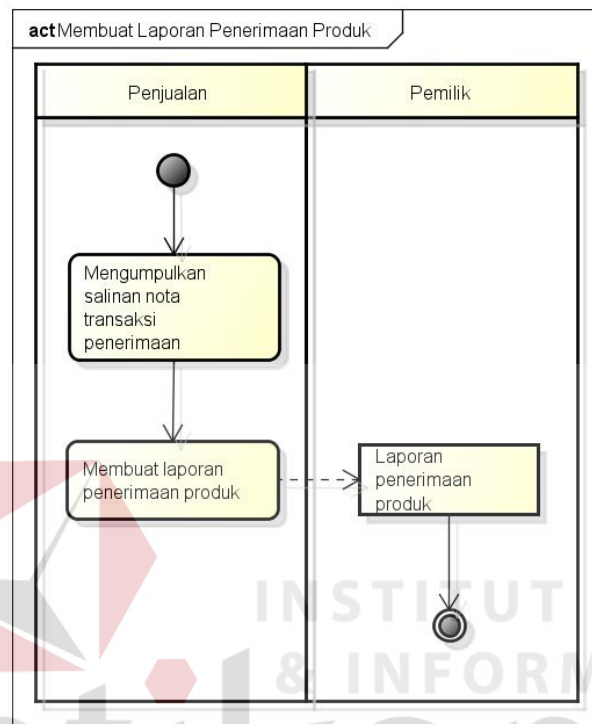


Gambar 3.8 Activity Diagram Mencatat Transaksi Penerimaan

3.2.1.7 Membuat Laporan Penerimaan Produk

Proses membuat laporan penerimaan produk dimulai dari bagian penjualan mengumpulkan salinan nota penerimaan. Setelah itu bagian penjualan

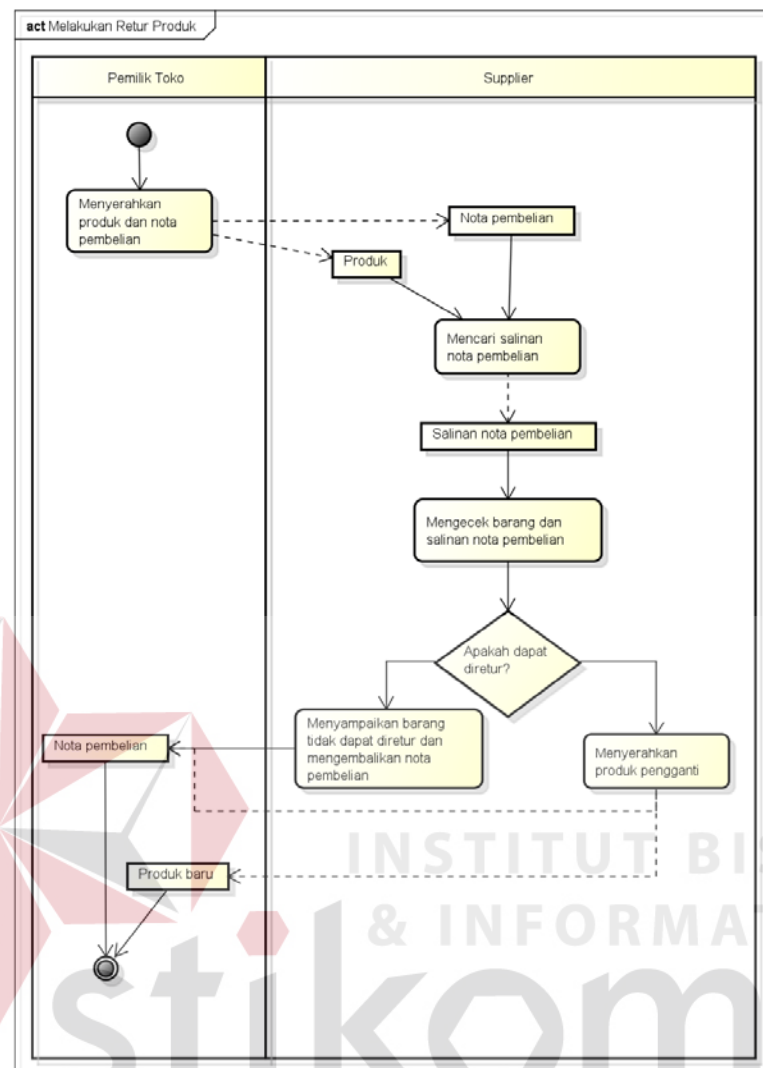
membuat laporan penerimaan produk untuk kemudian diserahkan kepada pemilik. Pemilik akan menerima dan mengecek laporan yang telah dibuat untuk kemudian disimpan sebagai arsip. Untuk lebih jelasnya dapat melihat Gambar 3.9.



Gambar 3.9 Activity Diagram Membuat Laporan Penerimaan Produk

3.2.1.8 Melakukan Retur Produk

Proses melakukan retur produk dimulai ketika pemilik toko menemukan produk cacat atau rusak yang dikirim oleh supplier. Setelah itu pemilik toko merekap produk yang rusak tersebut, kemudian dikembalikan kepada supplier untuk ditukarkan dengan produk yang baru. Supplier menerima produk yang rusak dan mengganti produk tersebut. Setelah menerima produk yang baru saja ditukar, pemilik toko melakukan update data stok. Untuk lebih jelasnya dapat melihat Gambar 3.10.



Gambar 3.10 Activity Diagram Melakukan Retur Produk

3.2.2 Hasil Analisis

Dari proses analisis, didapatkan kelemahan-kelemahan sistem yang ada pada saat ini. Adapun kelemahan-kelemahan sistem pada saat ini adalah sebagai berikut :

1. Pencatatan penjualan yang masih manual dengan menggunakan buku.
 Penggunaan buku sebagai media pencatatan sangat rawan untuk terjadi kehilangan atau kerusakan akibat ketelodoran pemilik dalam merawat. Selain itu bagian penjualan sering melakukan kesalahan perhitungan jumlah barang

yang dibeli beserta harga yang diberikan karena proses transaksi yang masih konvensional dengan mengandalkan bantuan nota dan kalkulator.

2. Pemilik toko tidak dapat mengetahui tren penjualan dan *history* transaksi dari pelanggan dan supplier. Pemilik toko tidak memiliki data *history* transaksi penjualan per periode baik minggu, bulan atau tahun. Data *history* tersebut dapat dimanfaatkan untuk mengetahui barang yang paling sering dibeli, menentukan target penjualan di periode selanjutnya, dan mengevaluasi produk yang selama ini dijual.
3. Pemilik toko kesulitan untuk mengetahui stok yang tersedia karena harus menghitung terlebih dahulu. Hal ini mengakibatkan pemilik toko sering mengalami kehabisan stok atau penumpukan stok. Pemilik toko sering kali melewatkan order penjualan dengan jumlah besar dari pelanggan dikarenakan tidak dapat mengetahui jumlah stok produk yang dijual.
4. Laporan yang dibutuhkan pemilik toko tidak dapat diakses kapanpun dan dimanapun, karena diharuskan untuk datang langsung ke toko atau harus menunggu laporan yang dibutuhkan selesai dibuat.
5. Permasalahan biaya dalam pengadaan perangkat teknologi informasi serta SDM (sumber daya manusia) yang melakukan *maintenance*.

Untuk mengatasi kelemahan-kelemahan yang ada dalam sistem penjualan saat ini, perlu dibuat sebuah sistem informasi penjualan yang dapat memenuhi kebutuhan pemilik toko ritel tradisional dalam manajemen barang dagangan. Adapun kebutuhan sistem informasi penjualan yang akan dibangun adalah sebagai berikut :

1. Sistem informasi penjualan ritel yang dapat membantu proses pencatatan transaksi penjualan dan meminimalkan kesalahan perhitungan jumlah barang yang dibeli beserta harga yang diberikan. Sistem informasi penjualan yang dibangun juga dapat menangani retur penjualan maupun pembelian.
2. Sistem informasi penjualan ritel yang dapat menampilkan *history* transaksi dari pelanggan maupun supplier. *History* transaksi penjualan dari pelanggan ditampilkan sesuai periode mingguan, bulanan, dan tahunan.
3. Sistem informasi penjualan ritel dapat menampilkan jumlah stok produk yang tersedia sehingga tidak perlu menghitung terlebih dahulu.
4. Menghasilkan informasi-informasi yang dibutuhkan pemilik secara *up to date* seperti omset penjualan, barang yang paling sering dibeli oleh pelanggan, barang kurang laku yang harus dipromosikan, keuntungan yang diperoleh dalam sehari dan dapat diakses kapanpun dan dimanapun melalui jaringan internet.
5. Pemilik toko dapat memakai layanan sistem informasi penjualan ritel tanpa dikenakan biaya berlangganan, sehingga dapat membantu menekan ongkos biaya operasional toko.

Selain untuk mengatasi kelemahan sistem penjualan saat ini dan memenuhi kebutuhan pemilik toko, terdapat fungsi lain dari sistem yang dibangun, yaitu :

1. Sistem informasi penjualan ritel juga dapat mempromosikan produk yang dijual kedalam bentuk web katalog produk yang dapat diakses oleh banyak orang melalui internet.

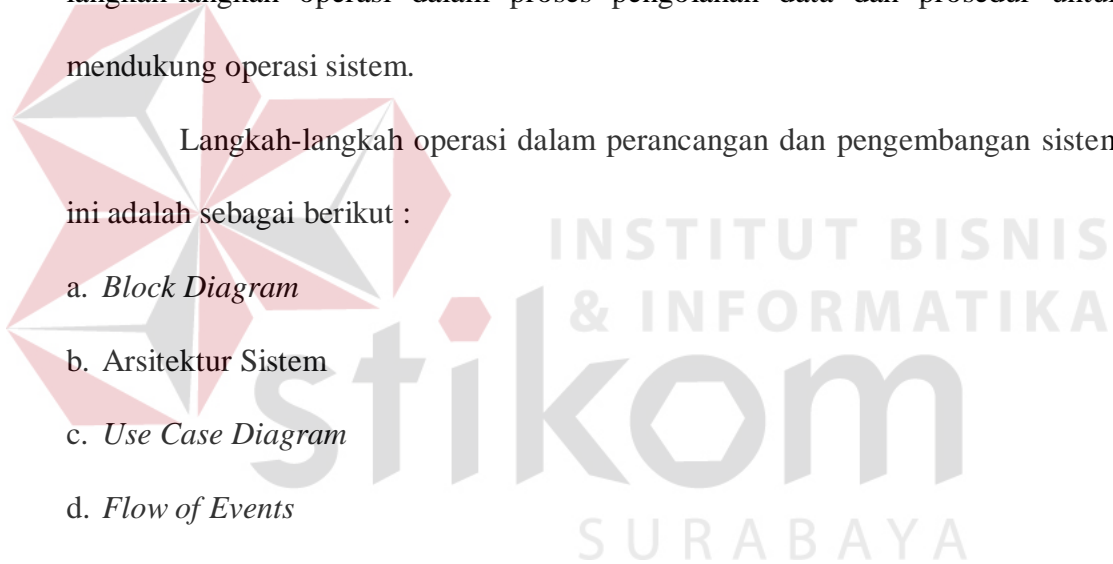
2. Tampilan sistem informasi penjualan ritel yang dibangun dapat menyesuaikan secara otomatis dengan ukuran layar perangkat keras yang digunakan seperti laptop, *smartphone*, komputer tablet, dll.

3.3 Perancangan Sistem

Perancangan sistem ini bertujuan untuk mendefinisikan kebutuhan-kebutuhan fungsional, menggambarkan aliran data dan alur sistem, dan sebagai tahap persiapan sebelum implementasi sistem. Perancangan sistem ini diharapkan dapat merancang dan mendesain sistem dengan baik, yang isinya meliputi langkah-langkah operasi dalam proses pengolahan data dan prosedur untuk mendukung operasi sistem.

Langkah-langkah operasi dalam perancangan dan pengembangan sistem ini adalah sebagai berikut :

- a. *Block Diagram*
- b. *Arsitektur Sistem*
- c. *Use Case Diagram*
- d. *Flow of Events*
- e. *Diagram Sekuensial*
- f. *Class Diagram*
- g. *Component Diagram*
- h. *Deployment Diagram*
- i. *Physical Data Model (PDM)*
- j. *Sitemaps*
- k. *Desain User Interface*



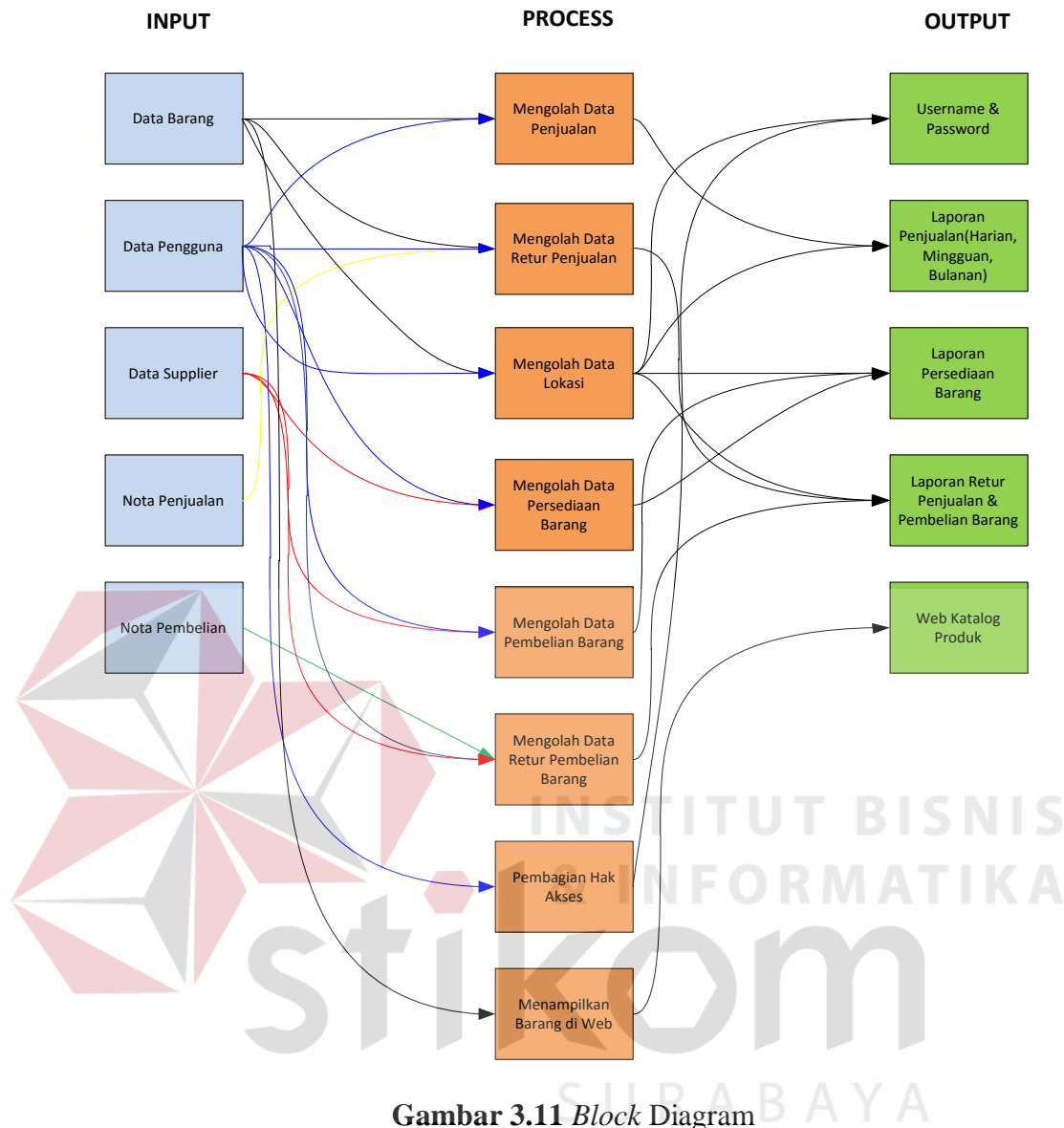
l. Desain *Input/Output*

m. Rancangan Pengujian dan Evaluasi Sistem

3.3.1 *Block Diagram*

Sistem informasi penjualan ritel yang dibangun terdiri dari point of sales untuk mencatat kegiatan transaksi penjualan dan laporan penjualan. Sistem juga mampu untuk mengelola stok produk dengan multi gudang sehingga memudahkan pemilik toko apabila memiliki lokasi gudang lebih dari satu. Selain itu sistem juga menampilkan produk pilihan ke dalam bentuk web katalog, sehingga memudahkan untuk pemilik toko dalam memasarkan produknya melalui internet. Berikut merupakan bentuk *Block diagram* sistem informasi penjualan ritel yang akan dibuat:

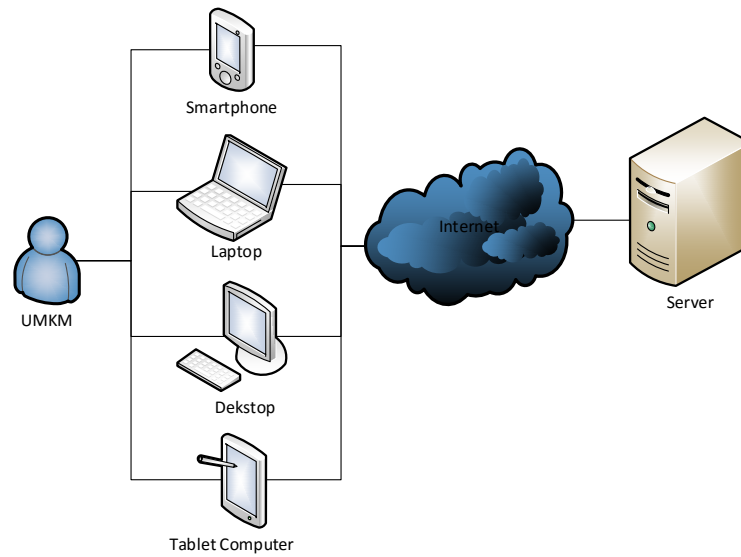




Gambar 3.11 Block Diagram

3.3.2 Arsitektur Sistem

Sistem informasi penjualan yang dibangun merupakan sistem berbasis web sehingga sistem dan data disimpan di data *center* milik penyedia layanan. Arsitektur sistem yang akan dibangun adalah seperti pada Gambar 3.12.



Gambar 3.12 Arsitektur Sistem Informasi Penjualan

Dengan arsitektur sistem berbasis web, pengguna tidak perlu melakukan pengelolaan dan perawatan infrastruktur sistem karena hal tersebut merupakan tanggung jawab dari penyedia layanan. Pengguna cukup mendaftarkan diri melalui halaman web penyedia layanan maka sistem informasi penjualan ritel bisa langsung digunakan. Untuk mengakses sistem, pengguna perlu menyediakan perangkat keras seperti komputer, laptop, *tablet* atau *smartphone* yang tersambung dengan koneksi internet.

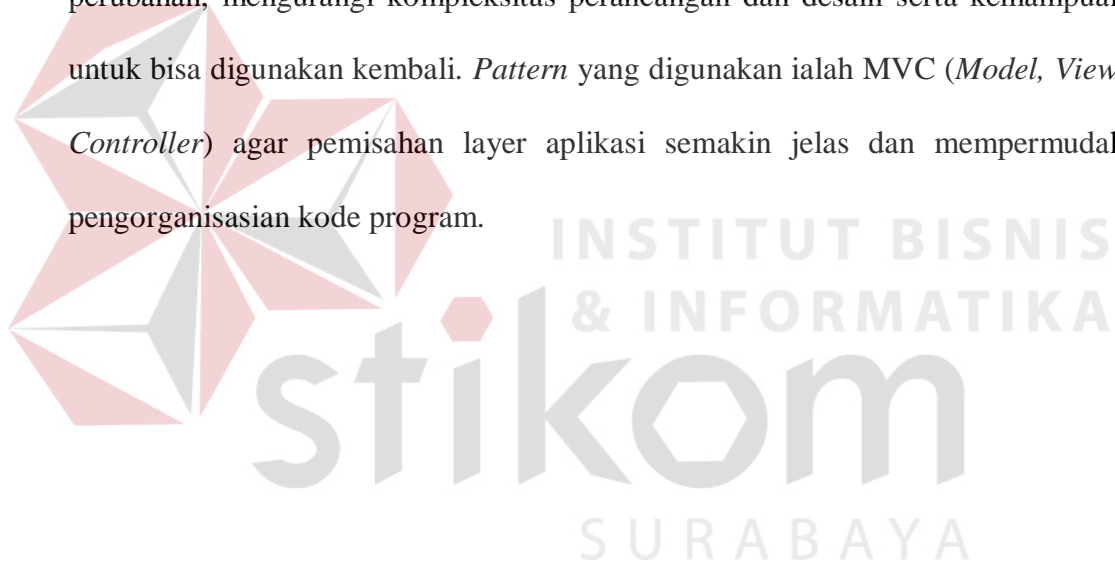
Model pengguna utama dari sistem didefinisikan sesuai fungsi dan tanggung jawab seperti di bawah ini.

Tabel 3.1 Model Pengguna

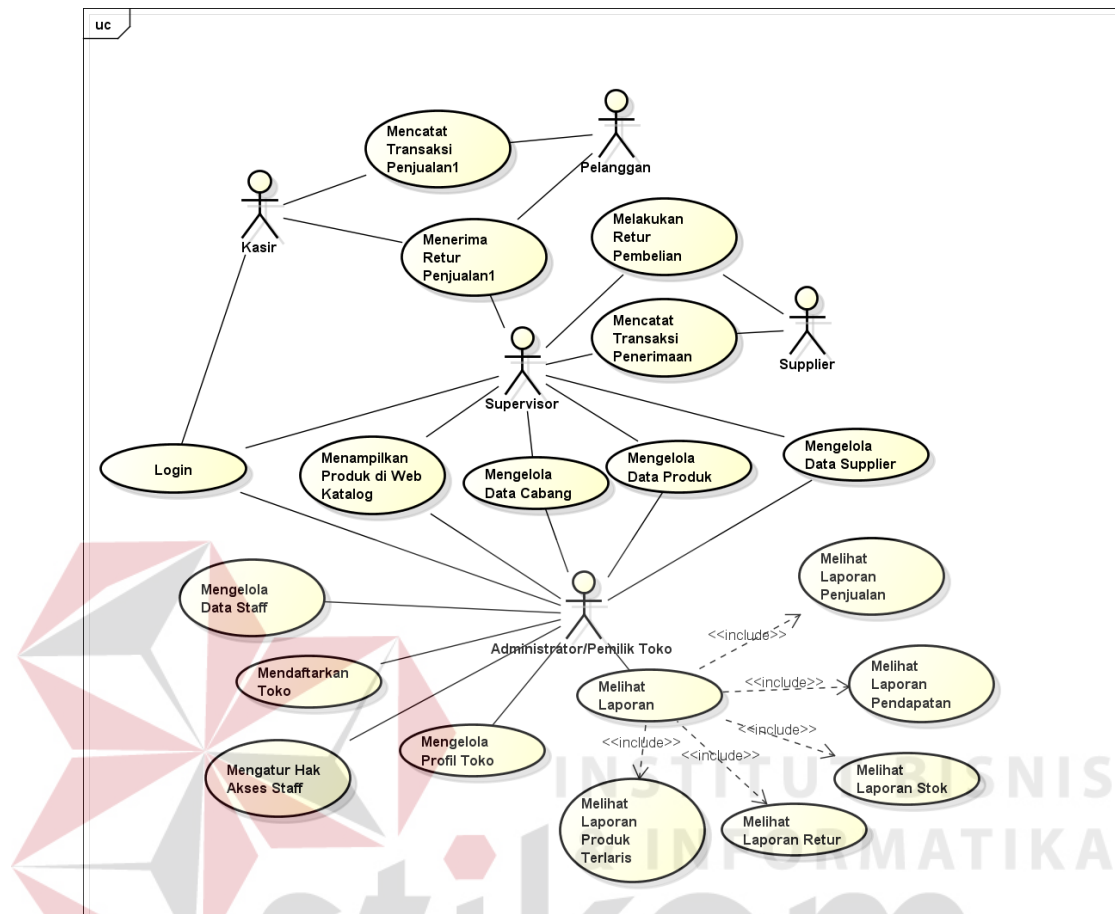
Pengguna	Fungsi
Administrator/Pemilik Toko	Memantau penjualan melalui laporan.
	Mengelola data karyawan beserta hak aksesnya
	Mengelola data master produk
	Mengelola data supplier
	Menampilkan produk di web katalog
Supervisor	Mengelola data master produk

Pengguna	Fungsi
	Mengelola persediaan produk
	Mencatat transaksi penerimaan produk
	Menampilkan produk di web katalog
	Melayani dan mencatat retur penjualan
Kasir	Melayani penjualan dan mencatat semua transaksi penjualan
	Melayani dan mencatat retur penjualan

Model perancangan dan pembangunan sistem akan menggunakan konsep *object oriented*. Konsep ini dipilih karena lebih mudah dalam mengakomodasi perubahan, mengurangi kompleksitas perancangan dan desain serta kemampuan untuk bisa digunakan kembali. *Pattern* yang digunakan ialah MVC (*Model, View, Controller*) agar pemisahan layer aplikasi semakin jelas dan mempermudah pengorganisasian kode program.



3.3.3 Use Case Diagram Sistem Informasi Penjualan



Gambar 3.13 Use Case Diagram Sistem Informasi Penjualan

Sistem informasi penjualan yang akan dibangun memiliki 5 aktor, aktor dalam *use case* merupakan semua yang ada diluar lingkup sistem perangkat lunak dan berinteraksi dengan sistem perangkat lunak tersebut. Aktor administrator/pemilik toko, supervisor dan kasir merupakan aktor internal dari toko. Ketiga aktor tersebut memberikan *input* dan menerima *output* dari sistem sesuai dengan peran dan tanggung jawab mereka. Sedangkan aktor pelanggan dan *supplier* merupakan aktor eksternal dari toko. Kedua aktor menerima *output* dari sistem berupa nota transaksi yang telah mereka lakukan.

3.3.4 *Flow of Events*

Proses selanjutnya yang dilakukan adalah dengan membuat *flow of events* dari masing-masing *use case*. Pembuatan *flow of events* bertujuan mendokumentasikan alur logika dalam *use case* yang menjelaskan secara rinci apa yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan.

3.3.4.1 *Flow of Events Mendaftarkan Toko*

Proses mendaftar toko dilakukan sebelum sebuah toko dapat menggunakan sistem informasi penjualan ritel berbasis web yang akan dibangun. Pendaftaran dilakukan dengan mengisi form pendaftaran yang disediakan dan melakukan konfirmasi email yang dikirim oleh sistem. Pengguna bisa mengakses halaman login jika proses pendaftaran dilalui. *Flow of events* proses mendaftar toko dapat dilihat pada Tabel 3.2.

Tabel 3.2 *Flow of Events Mendaftarkan Toko*

Nama Use Case	Mendaftarkan Toko
Kebutuhan terkait	Sebelum dapat menggunakan sistem informasi penjualan ritel berbasis web, toko harus melakukan registrasi dengan mengisi data toko. Setelah semua data diisi, sistem akan mengirimkan email ke pendaftar untuk melakukan konfirmasi pendaftaran.
Tujuan	Menangani proses registrasi toko
Prasyarat	Tidak ada
Kondisi akhir sukses	Toko berhasil terdaftar
Kondisi akhir gagal	Pengguna mendapatkan pesan registrasi toko gagal
Aktor utama	Pemilik Toko
Aktor sekunder	Tidak ada
Pemicu	Pengguna membuka halaman registrasi pada website sistem informasi penjualan

Nama Use Case	Mendaftarkan Toko	
<i>Primary Flow</i>	Langkah	Aksi
	1	Pengguna membuka halaman pendaftaran pada website sistem informasi penjualan
	2	Pengguna memasukkan nama lengkap, jenis kelamin, nomor telepon, alamat, kota, <i>email</i> , <i>password</i> dan nama domain.
	3	Sistem memeriksa <i>format</i> masukan dari pengguna. Jika ada <i>format</i> yang salah, masuk ke langkah 3.1
	4	Sistem melakukan koneksi ke <i>database</i> untuk memeriksa <i>email</i> yang didaftarkan telah digunakan. Jika <i>email</i> telah digunakan, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem melakukan koneksi <i>database</i> untuk menyimpan data toko baru. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 5.1 .
	6	Sistem mengirim <i>email</i> ke alamat <i>email</i> yang didaftarkan pengguna untuk melakukan konfirmasi pendaftaran. Jika pengiriman <i>email</i> gagal, masuk ke langkah 6.1
	7	Pengguna mengaktifkan akun melalui email konfirmasi yang dikirimkan sistem.
	8	Sistem melakukan koneksi <i>database</i> untuk mengaktifkan akun dan menampilkan halaman login ke pengguna sebagai tanda akun telah diaktifkan. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 8.1
	9	<i>Use case</i> berakhir
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan <i>format</i> data pendaftaran toko salah dan meminta pengguna mengulangi langkah ke-2
	4.1	Sistem menampilkan pesan <i>email</i> telah digunakan dan meminta pengguna mengulangi langkah ke-2

Nama Use Case	Mendaftarkan Toko	
<i>Error Flow</i>	Langkah	Aksi Percabangan
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk memeriksa <i>email</i> gagal. Pengguna kembali ke langkah 2.
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data pengguna baru gagal. Pengguna kembali ke langkah 2.
	6.1	Sistem menampilkan pesan pengiriman <i>email</i> gagal dan meminta pengguna mengulang langkah ke-2.
	8.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengaktifkan akun gagal. Pengguna kembali ke langkah 7.

3.3.4.2 Flow of Events Login

Proses login dilakukan sebelum pengguna masuk ke dalam sistem dan mengakses menu-menu yang ada. Proses login dilakukan untuk memastikan apakah orang yang mengakses sistem merupakan orang yang memiliki hak akses untuk membuka sistem tersebut. Sistem memberikan hak akses kepada pengguna berdasarkan login yang dilakukan. *Flow of events* proses *login* dapat dilihat pada Tabel 3.3.

Tabel 3.3 *Flow of Events Login*

Nama Use Case	Login
Kebutuhan terkait	Salah satu proses untuk masuk ke dalam sistem adalah login. Pengguna memiliki data berupa <i>username</i> dan <i>password</i> yang harus diverifikasi oleh sistem agar pengguna dapat masuk ke dalam sistem sesuai dengan hak akses yang dimiliki
Tujuan	Menangani proses login pengguna
Prasyarat	Tidak ada
Kondisi akhir sukses	Pengguna berhasil <i>login</i> dan dapat mengakses menu sesuai dengan hak akses yang dimiliki

Nama Use Case		Login
Kondisi akhir gagal	Pengguna mendapatkan pesan gagal dan mengulang proses login dengan mengisi <i>form login</i>	
Aktor utama	Pengguna	
Aktor sekunder	Tidak ada	
Pemicu	Pengguna membuka halaman login web sistem informasi penjualan	
<i>Primary Flow</i>	Langkah	Aksi
	1	Pengguna membuka halaman <i>login</i> melalui web <i>browser</i> .
	2	Pengguna mengisi <i>email</i> dan <i>password</i>
	3	Sistem melakukan koneksi ke <i>database</i> untuk melakukan verifikasi <i>email</i> dan <i>password</i> . Jika kombinasi <i>email</i> dan <i>password</i> salah, masuk ke langkah 3.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 3.2 .
	4	Sistem menampilkan halaman utama
	5	<i>Use case</i> berakhir
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan <i>email</i> dan <i>password</i> salah dan meminta pengguna mengulangi langkah ke- 3
<i>Error Flow</i>	Langkah	Aksi Percabangan
	3.2	Sistem menampilkan pesan koneksi <i>database</i> untuk melakukan verifikasi <i>email</i> dan <i>password</i> gagal. Pengguna kembali ke langkah 3 .

3.3.4.3 Flow of Events Mengelola Profil Toko

Pemilik toko diwajibkan untuk mengisi identitas toko yang dimiliki. Pemilik toko diharuskan untuk mengisi nama toko, deskripsi toko, alamat toko, telepon, kota, provinsi, email dan logo. Identitas ini digunakan untuk ditampilkan di halaman web katalog, sehingga calon pelanggan dapat dengan mudah untuk

berkomunikasi dengan pemilik toko ketika akan membeli produk yang dijual.

Flow of events proses mengelola profil toko dapat dilihat pada Tabel 3.4.

Tabel 3.4 *Flow of Events* Mengelola Profil Toko

Nama Use Case	Mengelola Profil Toko	
Kebutuhan terkait	Proses mengelola profil toko, pemilik toko diwajibkan untuk mengisi identitas toko yang dimiliki	
Tujuan	Memastikan bahwa identitas toko yang dimiliki merupakan data terkini yang dimiliki oleh pemilik toko.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengelola profil toko	
Kondisi akhir sukses	Data profil toko telah diperbaharui	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Pemilik Toko / Administrator	
Aktor sekunder	Tidak ada	
Pemicu	Pemilik Toko / Administrator memilih menu “Profil Toko”	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Profil Toko”.
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data profil toko dan menampilkannya. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	2	Pengguna melakukan operasi mengubah nama toko, deskripsi toko, alamat, nomor telp toko, <i>email</i> dan logo toko.
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data profil toko. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data

Nama Use Case	Mengelola Profil Toko	
		profil berhasil
	6	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data profil yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3 untuk mengisi ulang masukkan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data profil toko gagal. Pengguna kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data profil toko gagal. Pengguna kembali ke langkah 3 .

3.3.4.4 *Flow of Events* Mengelola Data Staff

Staff yang akan masuk ke dalam sistem wajib dimasukkan terlebih dahulu ke dalam data master staff. Pemilik toko bertanggung jawab untuk memasukkan data tersebut. Proses mengelola data staff meliputi menambah, mengubah, menghapus dan mencari data staff. *Flow of events* proses mengelola data staff disajikan pada Tabel 3.5.

Tabel 3.5 *Flow of Events* Mengelola Data Staff

Nama Use Case	Mengelola Data Staff
Kebutuhan terkait	Proses mengelola data staff meliputi proses menambah, mengubah, menghapus dan mencari data staff.
Tujuan	Memastikan data staff yang dimiliki merupakan data terkini.
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengelola data staff
Kondisi akhir sukses	Data staff telah diperbaharui

Nama Use Case		Mengelola Data Staff
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Pemilik Toko / Administrator	
Aktor sekunder	Tidak ada	
Pemicu	Pemilik Toko / Administrator memilih menu “Master Staff”	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Master Staff”.
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data staff dan menampilkannya. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	2	Pengguna melakukan operasi mencari, menambah, mengubah atau menghapus data staff.
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data staff. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data staff berhasil
	6	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data staff yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3 untuk mengisi ulang masukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data staff gagal. Pengguna kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data staff gagal. Pengguna kembali ke langkah 3 .

3.3.4.5 Flow of Events Mengatur Hak Akses Staff

Pemilik Toko/ Administrator dapat menentukan jabatan staff terhadap sistem. Hal ini memudahkan pembagian tanggung jawab dan mencari penyebab terjadinya kesalahan atau kekurangan dalam sistem. *Flow of events* proses mengatur jabatan staff disajikan pada Tabel 3.6.

Tabel 3.6 *Flow of Events* Mengatur Hak Akses Staff

Nama Use Case	Mengatur Hak Akses Staff	
Kebutuhan terkait	Proses mengatur jabatan staff akan membantu pemilik toko untuk melakukan pembagian tugas dalam sistem. Hak akses dibagi menjadi proses <i>insert</i> , <i>update</i> dan <i>delete</i> terhadap transaksi penjualan, penerimaan dan pengelolaan data <i>master staff</i> , <i>supplier</i> dan membaca laporan-laporan yang disediakan sistem.	
Tujuan	Membagi tugas dan tanggung jawab staff dengan pembagian hak akses terhadap sistem.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengatur hak akses staff	
Kondisi akhir sukses	Data staff dan hak aksesnya telah diperbaharui	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Pemilik Toko / Administrator	
Aktor sekunder	Tidak ada	
Pemicu	Pemilik Toko / Administrator memilih menu "Jabatan Staff"	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu "Jabatan Staff".
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data staff dan jabatan kemudian menampilkannya. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	2	Pengguna melakukan operasi mencari dan mengubah data jabatan staff.

Nama Use Case		Mengatur Hak Akses Staff
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data jabatan staff. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data jabatan staff berhasil
	6	<i>Use case</i> berakhir
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data jabatan staff yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3 untuk melakukan dengan <i>format</i> yang benar.
<i>Error Flow</i>	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data staff dan jabatan staff gagal. Pengguna kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data jabatan staff gagal. Pengguna kembali ke langkah 3 .

3.3.4.6 Flow of Events Mengelola Data Supplier

Proses mengelola data supplier dilakukan untuk memastikan data *supplier* yang dimiliki ialah data terkini yang berhasil didapatkan dari *supplier*. Proses ini meliputi menambah, mengubah, menghapus dan mencari data supplier. Proses ini dilakukan ketika toko mendapatkan informasi terbaru tentang *supplier* mereka. *Flow of events* proses mengelola data *supplier* disajikan pada Tabel 3.7.

Tabel 3.7 *Flow of Events* Mengelola Data *Supplier*

Nama Use Case		Mengelola Data <i>Supplier</i>	
Kebutuhan terkait	Proses mengelola data <i>supplier</i> meliputi proses menambah, mengubah, menghapus dan mencari data <i>supplier</i> .		
Tujuan	Memastikan bahwa data <i>supplier</i> yang dimiliki merupakan data terkini berdasarkan yang didapatkan dari <i>supplier</i> . Data ini dapat digunakan toko untuk berinteraksi dengan <i>supplier</i> .		
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengelola data <i>supplier</i>		
Kondisi akhir sukses	Data <i>supplier</i> telah diperbaharui		
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.		
Aktor utama	Administrator/Pemilik Toko dan Supervisor		
Aktor sekunder	Tidak ada		
Pemicu	Administrator/Pemilik Toko dan Supervisor memilih menu “Master <i>Supplier</i> ”		
Primary Flow	Langkah	Aksi	
	1	Pengguna memilih menu “Master <i>Supplier</i> ”.	
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data <i>supplier</i> dan menampilkannya. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1	
	2	Pengguna melakukan operasi mencari, menambah, mengubah atau menghapus data <i>supplier</i> .	
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data <i>supplier</i> . Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .	
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data <i>supplier</i> berhasil	
	6	<i>Use case</i> berakhir	
Alternate Flow	Langkah	Aksi Percabangan	

Nama Use Case		Mengelola Data <i>Supplier</i>
	4.1	Sistem menampilkan pesan <i>format</i> data <i>supplier</i> yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3 untuk melakukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data <i>supplier</i> gagal. Pengguna kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data <i>supplier</i> gagal. Pengguna kembali ke langkah 3 .

3.3.4.7 Flow of Events Mengelola Data Produk

Proses mengelola data produk merupakan proses ketika pemilik toko memasukkan data-data produk yang dijual. Proses ini meliputi menambah, mengubah, menghapus dan mencari data produk. *Flow of events* proses mengelola data produk disajikan pada Tabel 3.8.

Tabel 3.8 *Flow of Events* Mengelola Data Produk

Nama Use Case		Mengelola Data Produk
Kebutuhan terkait		Proses mengelola data produk meliputi proses menambah, mengubah, menghapus dan mencari data produk.
Tujuan		Memasukkan data produk yang dijual agar dapat digunakan untuk melakukan transaksi penjualan.
Prasyarat		<i>Login</i> dan memiliki hak akses untuk mengelola data produk
Kondisi akhir sukses		Data produk telah diperbahurui
Kondisi akhir gagal		Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.
Aktor utama		Administrator/Pemilik Toko dan Supervisor
Aktor sekunder		Tidak ada

Nama Use Case		Mengelola Data Produk
Pemicu	Administrator/Pemilik Toko dan Supervisor memilih menu “Master Produk”	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Master Produk”.
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data produk dan menampilkannya. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	2	Pengguna melakukan operasi mencari, menambah, mengubah atau menghapus data produk.
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data produk. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data produk berhasil
	6	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data produk yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3 untuk melakukan dengan <i>format</i> yang benar.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data produk gagal. Pengguna kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data produk gagal. Pengguna kembali ke langkah 3 .

3.3.4.8 Flow of Events Mengelola Data Cabang

Proses mengelola data cabang digunakan oleh pemilik toko memasukkan data lokasi cabang toko yang dimiliki. Proses ini meliputi menambah, mengubah, menghapus dan mencari data cabang. *Flow of events* proses mengelola data cabang disajikan pada Tabel 3.9.

Tabel 3.9 *Flow of Events* Mengelola Data Cabang

Nama Use Case	Mengelola Data Cabang	
Kebutuhan terkait	Proses mengelola data cabang meliputi proses menambah, mengubah, menghapus dan mencari data cabang.	
Tujuan	Memasukkan data cabang toko yang dimiliki.	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mengelola data cabang	
Kondisi akhir sukses	Data cabang telah diperbahurui	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberi tahu bahwa data gagal disimpan.	
Aktor utama	Administrator/Pemilik Toko dan Supervisor	
Aktor sekunder	Tidak ada	
Pemicu	Administrator/Pemilik Toko dan Supervisor memilih menu “Cabang Toko”	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Cabang Toko”.
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil data cabang dan menampilkannya. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	2	Pengguna melakukan operasi mencari, menambah, mengubah atau menghapus data cabang.
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data cabang. Jika <i>format</i> data masukan salah, masuk ke langkah 4.1 . Jika terjadi

Nama Use Case		Mengelola Data Cabang
		<i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2 .
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data cabang berhasil
	6	<i>Use case</i> berakhir
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data cabang yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3 untuk melakukan dengan <i>format</i> yang benar.
<i>Error Flow</i>	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data cabang gagal. Pengguna kembali ke langkah 1 .
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data cabang gagal. Pengguna kembali ke langkah 3 .

3.3.4.9 *Flow of Events* Mencatat Transaksi Penjualan

Proses mencatat transaksi penjualan dimulai dengan kasir memilih produk yang akan dibeli oleh pelanggan kemudian sistem akan menghitung jumlah yang harus dibayar oleh pelanggan. Pembayaran hanya dapat dilakukan secara tunai. Pelanggan memberikan uang pembayaran dan kasir mencetak nota penjualan untuk diserahkan kepada pelanggan dan menyimpan transaksi penjualan tersebut. *Flow of events* proses mencatat transaksi penjualan disajikan pada Tabel 3.10.

Tabel 3.10 *Flow of Events* Mencatat Transaksi Penjualan

Nama Use Case		Mencatat Transaksi Penjualan
Kebutuhan terkait		Kasir dapat memberikan diskon khusus pada masing-masing barang atau keseluruhan transaksi sesuai proses tawar menawar dengan pelanggan.

Nama Use Case	Mencatat Transaksi Penjualan	
Tujuan	Menangani proses penjualan secara tunai	
Prasyarat	<i>Login</i> dan memiliki hak akses sebagai kasir	
Kondisi akhir sukses	Transaksi penjualan berhasil dilakukan	
Kondisi akhir gagal	Jika stok produk habis maka transaksi gagal disimpan.	
Aktor utama	Kasir	
Aktor sekunder	Pelanggan	
Pemicu	Pengguna membuka halaman kasir	
Primary Flow	Langkah	Aksi
	1	Pengguna masuk ke halaman kasir
	2	Kasir memilih produk yang dibeli, jumlah yang dibeli dan diskon masing-masing barang (jika ada).
	3	Sistem melakukan koneksi <i>database</i> untuk memeriksa apakah jumlah yang dibeli tidak melebihi stok siap jual. Jika melebihi masuk ke langkah 3.1 . Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 3.2
	4	Kasir memasukkan diskon keseluruhan transaksi (jika ada).
	5	Sistem menampilkan total harga
	6	Kasir memasukkan jumlah uang yang dibayarkan oleh pelanggan sesuai dengan total harga. Jika jumlah uang yang dibayarkan pelanggan kurang, masuk ke langkah 6.1 .
	7	Sistem menampilkan jumlah kembalian.
	8	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan data transaksi dan <i>update</i> stok. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 8.2 .
	9	Sistem mencetak nota penjualan dan kasir menyerahkannya ke pelanggan.

Nama Use Case	Mencatat Transaksi Penjualan	
	10	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan “stok barang tidak mencukupi”. Kasir kembali ke langkah 2 untuk memilih produk.
	6.1	Sistem menampilkan “pesan uang yang dibayarkan kurang” dan meminta kasir mengulangi langkah ke- 8 .
Error Flow	Langkah	Aksi Percabangan
	3.2	Sistem menampilkan pesan koneksi <i>database</i> untuk memeriksa stok produk gagal. Pengguna kembali ke langkah 2 .
	8.2	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi penjualan gagal. Kasir kembali ke langkah- 7 .

3.3.4.10 *Flow of Events* Menerima Retur Penjualan

Proses menerima retur penjualan merupakan proses yang dilakukan ketika pelanggan mengembalikan barang yang dibeli karena kondisi rusak. Data retur penjualan perlu dicatat agar toko memiliki dokumentasi retur produk yang lengkap dan dapat dijadikan informasi bagi pemilik toko. *Flow of events* proses menerima retur penjualan disajikan pada Tabel 3.11.

Tabel 3.11 *Flow of Events* Menerima Retur Penjualan

Nama Use Case	Mencatat Retur Penjualan
Kebutuhan terkait	Retur penjualan dilakukan ketika pelanggan mengembalikan barang yang rusak. Retur harus berdasarkan data transaksi penjualan yang pernah dilakukan untuk memastikan kevalidan barang yang diretur
Tujuan	Menangani proses retur penjualan
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat retur penjualan

Nama Use Case		Mencatat Retur Penjualan
Kondisi akhir sukses	Retur penjualan sukses dilakukan	
Kondisi akhir gagal	Jika tidak ditemukan transaksi penjualan dari barang yang akan diretur maka retur tidak dapat dilakukan.	
Aktor utama	Supervisor / Kasir	
Aktor sekunder	Pelanggan	
Pemicu	Supervisor / Kasir memilih menu "Retur Penjualan"	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu "Retur Penjualan"
	2	Sistem melakukan koneksi <i>database</i> untuk mencari kode transaksi yang didapat dari nota penjualan. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	3	Sistem menampilkan <i>history</i> transaksi dan data produk
	4	Pengguna memilih produk yang akan diretur dan memasukkan jumlahnya. Jika jumlah yang diretur melebihi jumlah yang dibeli, masuk ke langkah 4.1
	5	Sistem melakukan koneksi <i>database</i> dan menyimpan transaksi retur. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 5.1
	6	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan jumlah yang diretur tidak dapat melebihi jumlah yang terjual dan meminta bagian penjualan mengulangi langkah ke-4.
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mencari kode transaksi gagal. Pengguna kembali ke langkah 2.
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi retur gagal. Kasir kembali ke langkah-4.

3.3.4.11 Flow of Events Mencatat Transaksi Penerimaan

Proses mencatat transaksi penerimaan dilakukan ketika pemilik menerima produk dari *supplier*. Transaksi penerimaan perlu dicatat agar pemilik toko dapat memantau transaksi penerimaan yang telah dilakukan beserta *supplier*-nya. *Flow of events* mencatat transaksi penerimaan disajikan pada Tabel 3.12.

Tabel 3.12 *Flow of Events* Mencatat Transaksi Penerimaan

Nama Use Case	Mencatat Transaksi Penerimaan	
Kebutuhan terkait	Transaksi penerimaan <i>supplier</i> dicatat dan update stok dilakukan sesuai yang ditentukan oleh pemilik toko.	
Tujuan	Menangani penerimaan produk	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat penerimaan produk	
Kondisi akhir sukses	Penerimaan produk sukses dilakukan	
Kondisi akhir gagal	Jumlah stok produk tidak bertambah dan jika kurang memenuhi stok minimal tidak dapat digunakan untuk transaksi penjualan.	
Aktor utama	Administrator / Pemilik Toko dan Supervisor	
Aktor sekunder	<i>Supplier</i>	
Pemicu	Administrator / Pemilik Toko dan Supervisor memilih menu “Penerimaan Stok Masuk”	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Penerimaan Stok Masuk”
	2	Pengguna memasukkan tanggal, <i>supplier</i> , lokasi dan memberikan catatan (jika ada). Sistem melakukan koneksi <i>database</i> dan menyimpan data. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.2
	3	Sistem menampilkan data produk
	4	Pengguna memilih produk yang akan ditambah dan memasukkan jumlahnya.
	5	Sistem melakukan koneksi <i>database</i> dan menyimpan transaksi penerimaan. Jika

Nama Use Case	Mencatat Transaksi Penerimaan	
		terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 5.1
	6	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	-	-
Error Flow	Langkah	Aksi Percabangan
	2.2	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data gagal. Pengguna kembali ke langkah 2 .
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi penerimaan produk gagal. Pengguna kembali ke langkah-4.

3.3.4.12 Flow of Events Melakukan Retur Pembelian

Proses melakukan retur pembelian merupakan proses yang dilakukan ketika produk yang diterima dari *supplier* dalam kondisi rusak. Data retur pembelian perlu dicatat agar toko memiliki dokumentasi retur produk yang lengkap dan dapat dijadikan informasi bagi pemilik toko mengenai kualitas produk dari *supplier*. *Flow of events* proses melakukan retur pembelian disajikan pada Tabel 3.13.

Tabel 3.13 *Flow of Events* Melakukan Retur Pembelian

Nama Use Case	Melakukan Retur Pembelian
Kebutuhan terkait	Retur pembelian dilakukan ketika produk yang diterima dari <i>supplier</i> dalam kondisi rusak. Retur harus berdasarkan data transaksi penerimaan stok masuk yang pernah dilakukan untuk memastikan kevalidan barang yang diretur
Tujuan	Menangani proses retur pembelian
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat retur pembelian

Nama Use Case		Melakukan Retur Pembelian
Kondisi akhir sukses	Retur pembelian sukses dilakukan	
Kondisi akhir gagal	Jika tidak ditemukan transaksi penerimaan stok masuk dari barang yang akan diretur maka retur tidak dapat dilakukan.	
Aktor utama	Administrator (Pemilik Toko) / Supervisor	
Aktor sekunder	<i>Supplier</i>	
Pemicu	Administrator (Pemilik Toko) / Supervisor memilih menu "Retur Pembelian"	
<i>Primary Flow</i>	Langkah	Aksi
	1	Pengguna memilih menu "Retur Pembelian"
	2	Sistem melakukan koneksi <i>database</i> untuk mencari kode transaksi penerimaan stok masuk. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 2.1
	3	Sistem menampilkan <i>history</i> transaksi dan data produk.
	4	Pengguna memilih produk yang akan diretur dan memasukkan jumlahnya.
	5	Sistem melakukan koneksi <i>database</i> dan menyimpan transaksi retur. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 5.1
	6	<i>Use case</i> berakhir
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	-	-
<i>Error Flow</i>	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mencari kode transaksi gagal. Pengguna kembali ke langkah 2 .
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan transaksi retur pembelian gagal. Pengguna kembali ke langkah- 4 .

3.3.4.13 Flow of Events Menampilkan Produk di Web Katalog

Proses menampilkan produk di web katalog digunakan untuk media promosi pemilik toko dalam memasarkan produk yang dijual. *Flow of events* menampilkan produk di web katalog disajikan pada Tabel 3.14.

Tabel 3.14 *Flow of Events* Menampilkan Produk di Web Katalog

Nama Use Case	Menampilkan Produk di Web Katalog	
Kebutuhan terkait	Menampilkan produk di web katalog digunakan untuk media promosi pemilik toko dalam memasarkan produk yang dijual.	
Tujuan	Menampilkan produk di web katalog	
Prasyarat	<i>Login</i> dan memiliki hak akses untuk menampilkan produk di web katalog	
Kondisi akhir sukses	Produk berhasil ditampilkan dalam web katalog	
Kondisi akhir gagal	Web katalog gagal menampilkan produk yang dipilih.	
Aktor utama	Administrator / Pemilik Toko dan Supervisor	
Aktor sekunder	Tidak ada	
Pemicu	Administrator / Pemilik Toko dan Supervisor memilih menu “Produk <i>Display</i> ”	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Produk <i>Display</i> ”
	2	Sistem melakukan koneksi <i>database</i> untuk mengambil dan menampilkan data produk. Jika terjadi <i>error</i> saat melakukan koneksi <i>database</i> , masuk ke langkah 2.1 .
	3	Sistem menampilkan data produk
	4	Pengguna memilih produk yang akan ditampilkan ke dalam web katalog.
	5	Sistem melakukan koneksi <i>database</i> dan menyimpan perubahan data. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah 5.1
	6	<i>Use case</i> berakhir

Nama Use Case		Menampilkan Produk di Web Katalog
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	-	-
<i>Error Flow</i>	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menampilkan data produk gagal. Pengguna kembali ke langkah 1.
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data produk <i>display</i> . Pengguna kembali ke langkah-4.

3.3.4.14 Flow of Events Melihat Laporan

Melihat laporan merupakan proses yang sangat penting bagi pemilik toko. Laporan-laporan yang dihasilkan akan membantu pemilik toko dalam mengambil keputusan. Laporan yang dihasilkan ialah *Flow of events* proses melihat laporan disajikan pada Tabel 3.15.

Tabel 3.15 *Flow of Events* Melihat Laporan

Nama Use Case	Melihat Laporan
Kebutuhan terkait	Transaksi penerimaan supplier dicatat dan update stok dilakukan sesuai yang ditentukan oleh pemilik toko.
Tujuan	Mendapatkan informasi secara <i>real time</i> yang dapat membantu pemilik toko dalam pengambilan keputusan.
Prasyarat	<i>Login</i> dan memiliki hak akses untuk mencatat melihat laporan
Kondisi akhir sukses	Mendapatkan informasi yang diperlukan
Kondisi akhir gagal	Jika <i>query</i> untuk laporan gagal dieksekusi, maka sistem akan menampilkan pesan <i>error</i> .
Aktor utama	Administrator / Pemilik Toko
Aktor sekunder	Tidak ada
Pemicu	Administrator / Pemilik Toko memilih menu "Laporan"

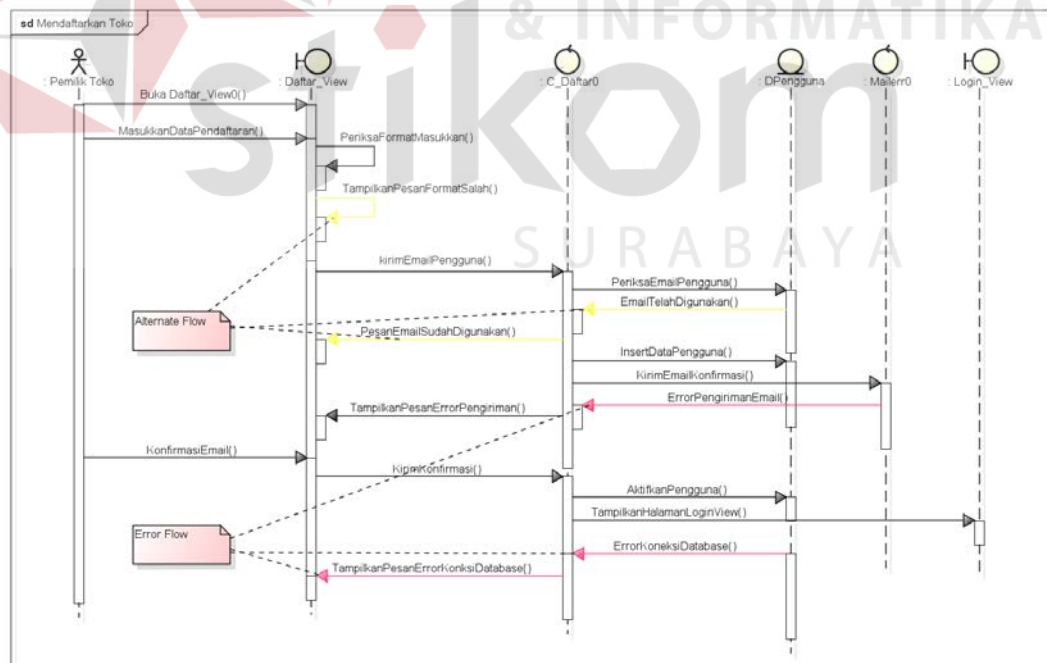
Nama Use Case	Melihat Laporan	
Primary Flow	Langkah	Aksi
	1	Pengguna memilih menu “Laporan”
	2	Pengguna memilih jenis laporan yang diinginkan
	3	Pengguna memasukkan parameter-parameter laporan sesuai informasi yang dibutuhkan. Jika <i>format</i> parameter masuk salah, masuk ke langkah 3.1
	4	Sistem melakukan koneksi <i>database</i> untuk mengambil data dan mengolahnya menjadi informasi sesuai dengan tampilan dan jenis laporan yang dipilih. Jika terjadi error saat melakukan koneksi ke <i>database</i> , masuk ke langkah 4.2
	5	Pengguna mencetak laporan
	6	<i>Use case</i> berakhir
Alternate Flow	Langkah	Aksi Percabangan
	3.1	Sistem menampilkan pesan format data yang dimasukkan salah dan meminta pengguna mengisi ulang parameter laporan sesuai dengan format yang benar. Kemudian mengulangi langkah ke-3.
Error Flow	Langkah	Aksi Percabangan
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk menampilkan laporan gagal. Pengguna kembali ke langkah 3 .

3.3.5 Diagram Sekuensial

Setelah menggambarkan *Flow of Events* langkah selanjutnya adalah membuat diagram sekuensial. Diagram sekuensial sendiri adalah diagram yang digunakan untuk menunjukkan aliran fungsionalitas dalam masing-masing use case yang disusun dalam urutan waktu.

3.3.5.1 Diagram Sekuensial Mendaftarkan Toko

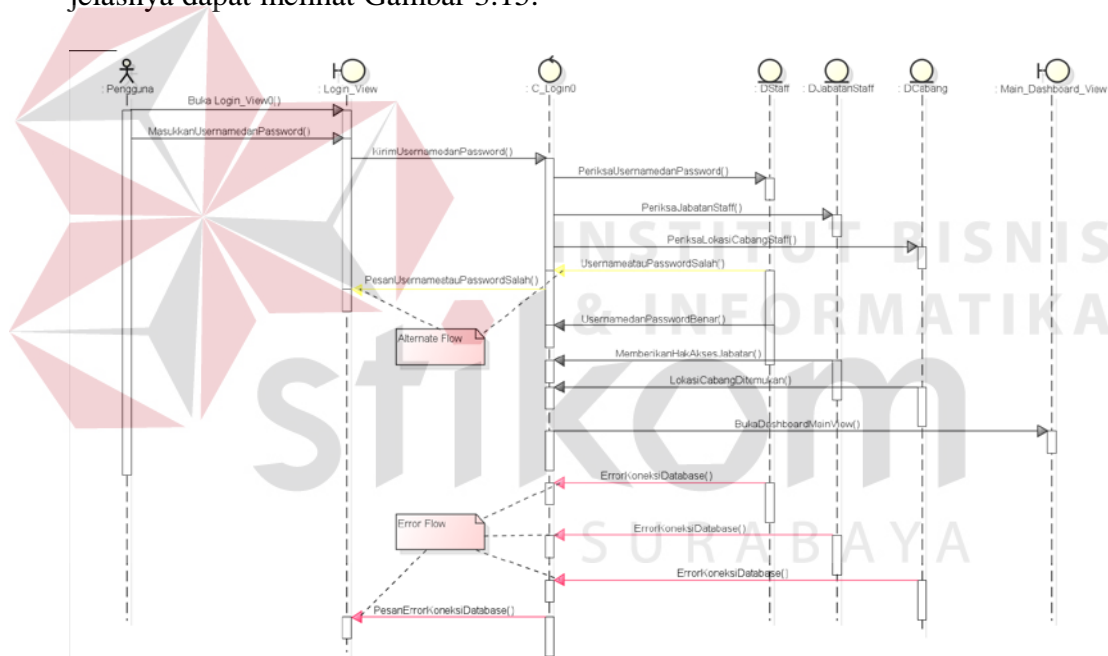
Proses mendaftarkan toko dimulai dari pengguna mengakses *boundary* Daftar_View dan mengisi *form* pendaftaran. Setelah itu, sistem memeriksa format data pendaftaran. Jika terdapat kesalahan, sistem akan menampilkan pesan ke pengguna. Apabila benar, sistem akan memeriksa apakah *email* yang didaftarkan telah digunakan melalui *control* C_Daftar. Jika telah digunakan, sistem meminta pengguna untuk memasukkan *email* yang lain. Jika belum, maka sistem akan menyimpan data pengguna dan mengirimkan *email* konfirmasi pada *email* yang didaftarkan. Pengguna diharuskan melakukan konfirmasi melalui *link* yang dikirimkan ke alamat *email* yang didaftarkan dan sistem akan mengaktifkan akun pengguna dan mengarahkan ke halaman *boundary* Login_View. Untuk lebih jelasnya dapat melihat Gambar 3.14.



Gambar 3.14 Diagram Sekuensial Mendaftarkan Toko

3.3.5.2 Diagram Sekuensial *Login*

Proses login dimulai dari pengguna mengakses *boundary Login_View* dan memasukkan *username* dan *password*. Setelah itu, sistem akan mengirim *username* dan *password* ke *control C_Login* untuk diperiksa. Setelah itu, pengguna melakukan operasi RU (*Read, Update*) pada *boundary Profil_View*. Jika *username* atau *password* salah sistem akan meminta pengguna untuk memasukkan *username* dan *password* yang benar. Apabila benar, sistem akan mengarahkan pengguna ke *boundary Main_Dashboard_View*. Untuk lebih jelasnya dapat melihat Gambar 3.15.



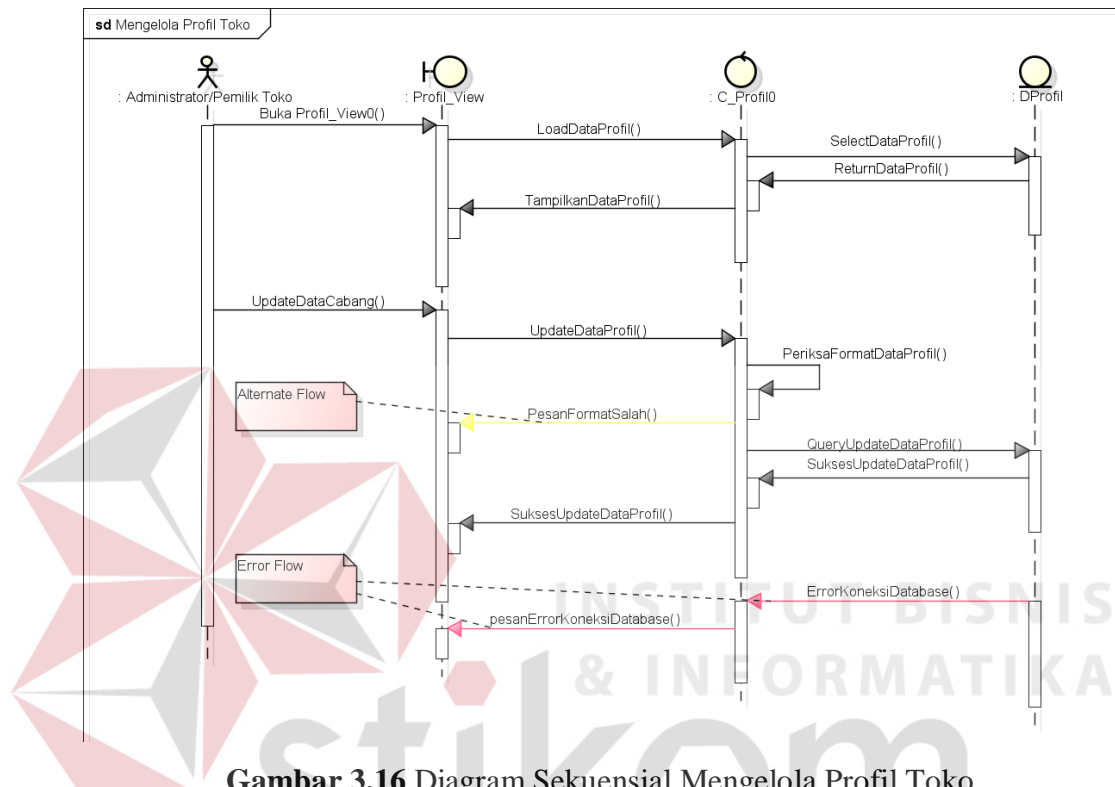
Gambar 3.15 Diagram Sekuensial *Login*

3.3.5.3 Diagram Sekuensial Mengelola Profil Toko

Proses mengelola profil toko dimulai dari pengguna mengakses *boundary Profil_View*. Lalu *boundary Profil_View* meminta *control C_Profil* mengambil data profil dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi RU (*Read, Update*) pada *boundary Profil_View*. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan, sistem meminta

pengguna untuk mengubahnya. Apabila benar, data profil disimpan oleh sistem di *database* melalui *control* C_Profil dan pesan sukses disampaikan oleh sistem.

Untuk lebih jelasnya dapat melihat Gambar 3.16.

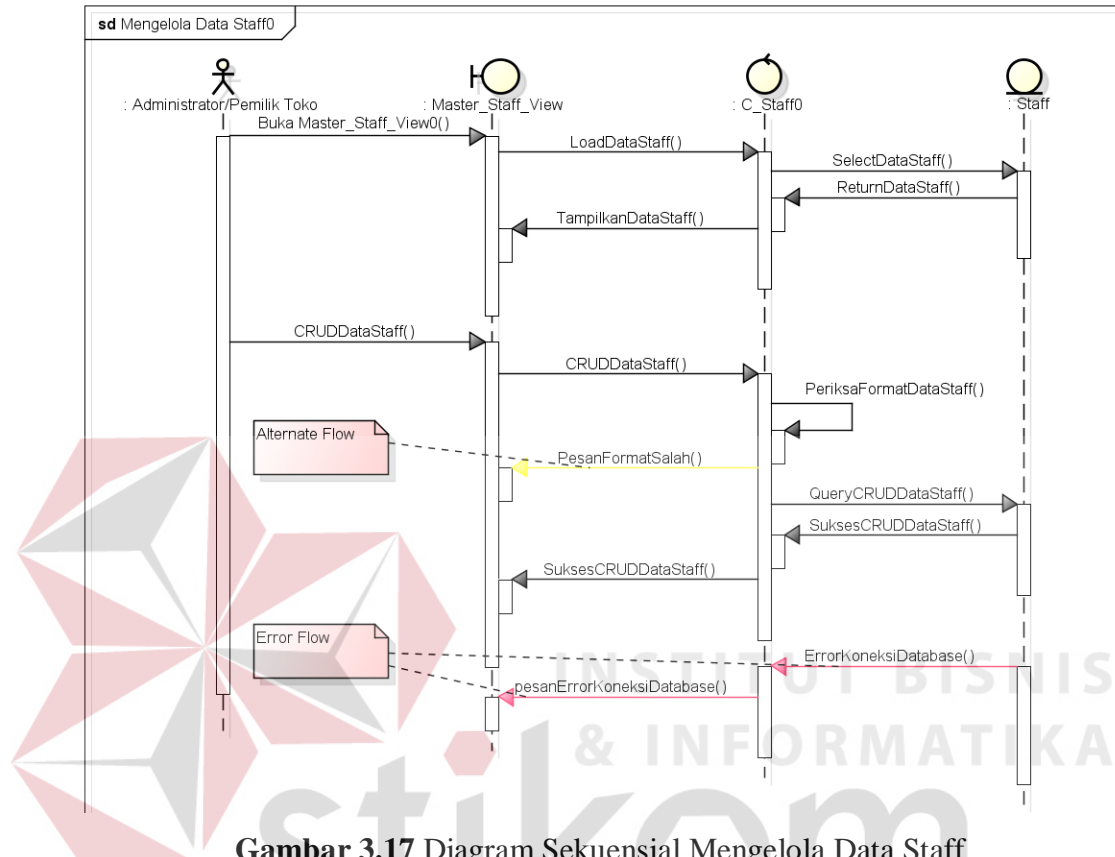


Gambar 3.16 Diagram Sekuensial Mengelola Profil Toko

3.3.5.4 Diagram Sekuensial Mengelola Data Staff

Proses mengelola data staff dimulai dari pengguna mengakses *boundary* Master_Staff_View. Lalu *boundary* Master_Staff_View meminta *control* C_Staff mengambil data staff dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* Master_Staff_View. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data staff disimpan oleh sistem di *database* melalui *control* C_Staff dan

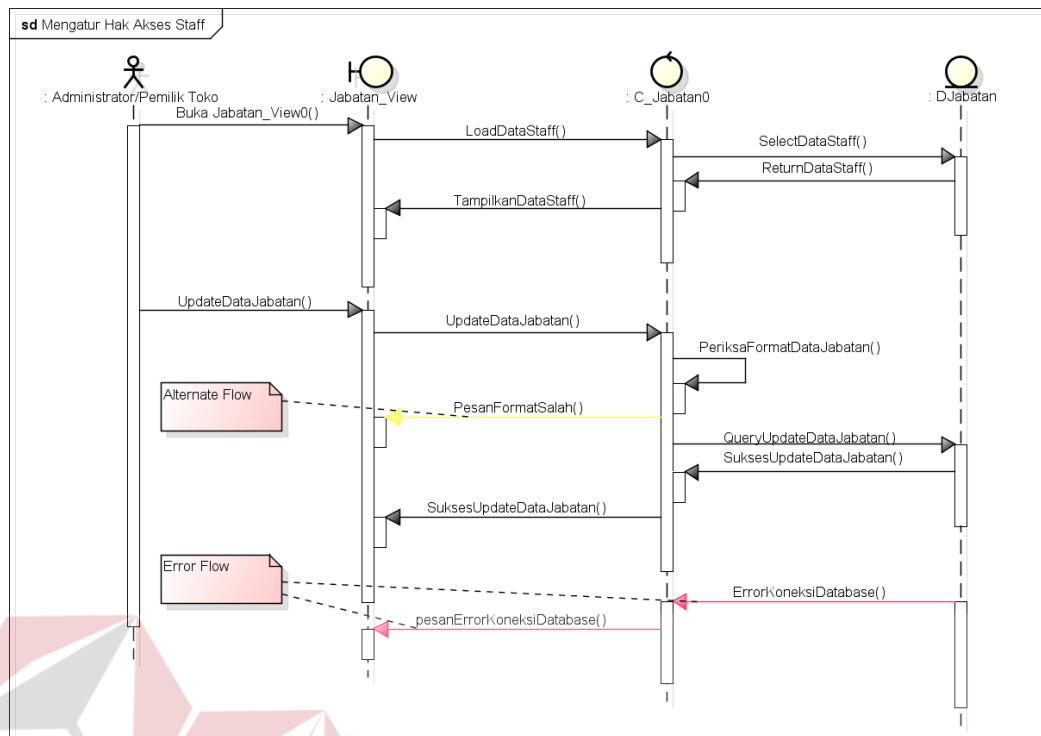
pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.17.



Gambar 3.17 Diagram Sekuensial Mengelola Data Staff

3.3.5.5 Diagram Sekuensial Mengatur Hak Akses Staff

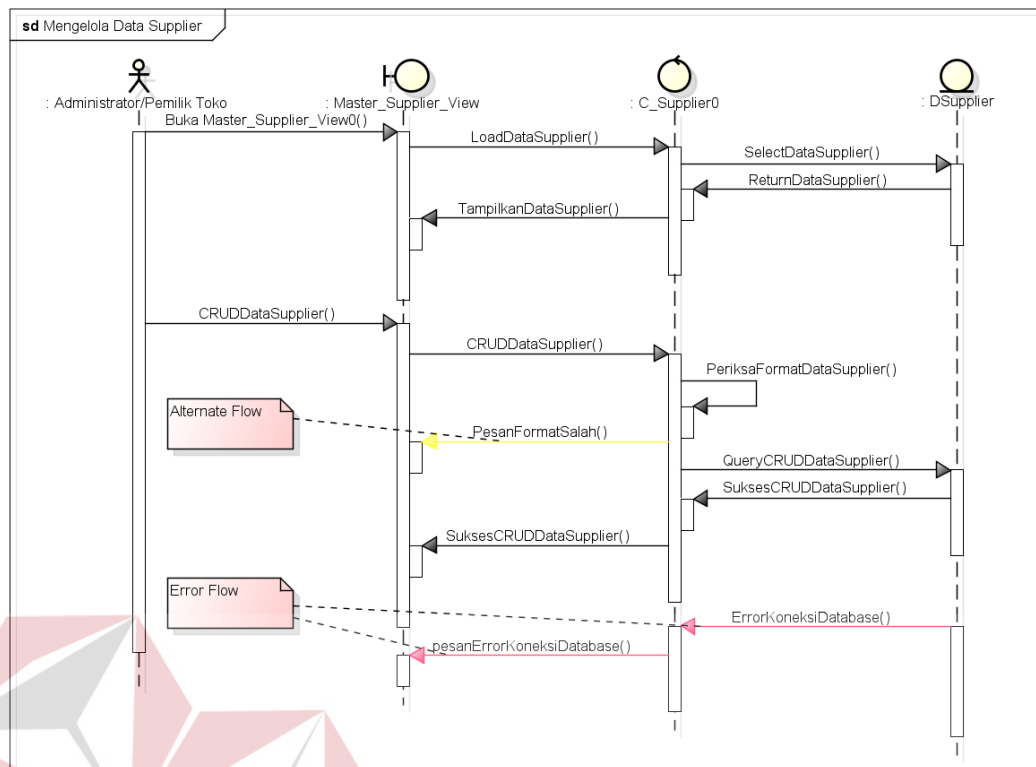
Proses mengelola hak akses staff dimulai dari pengguna mengakses *boundary* *Jabatan_View*. Lalu *boundary* *Jabatan_View* meminta *control* *C_Produk* mengambil data produk dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi RU (*Read, Update*) pada *boundary* *Jabatan_View*. Setelah selesai, sistem akan memeriksa format data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data jabatan staff disimpan oleh sistem di *database* melalui *control* *C_Jabatan* dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.18.



Gambar 3.18 Diagram Sekuensial Mengatur Hak Akses Staff

3.3.5.6 Diagram Sekuensial Mengelola Data *Supplier*

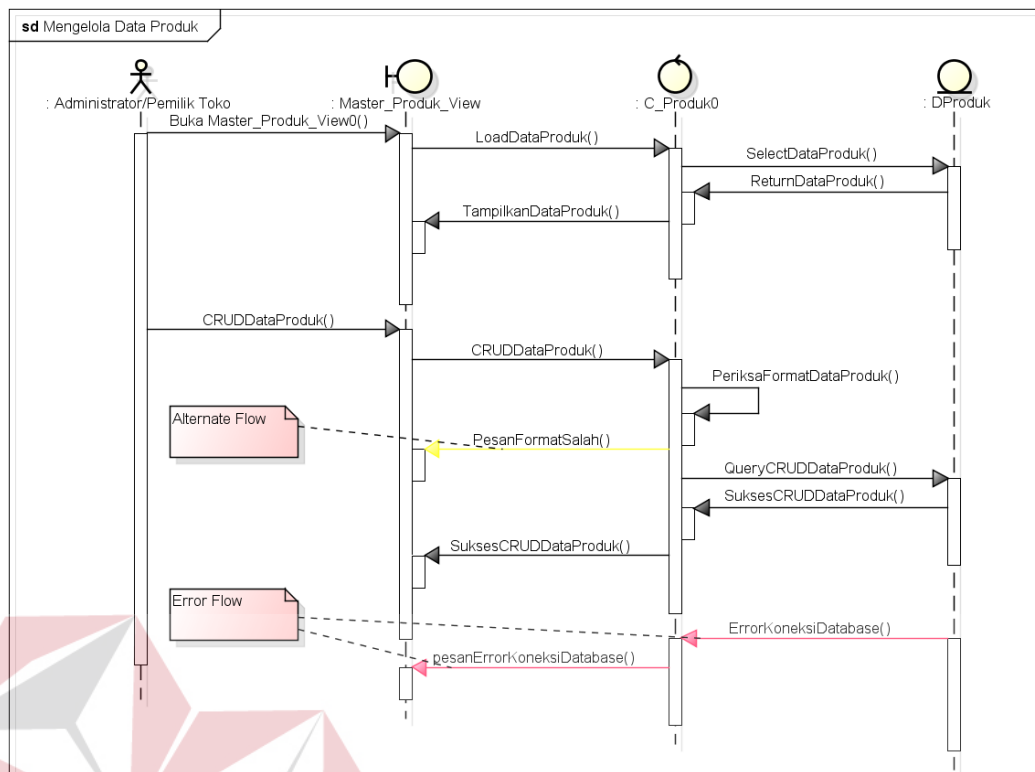
Proses mengelola data *supplier* dimulai dari pengguna mengakses *boundary* *Master_Supplier_View*. Lalu *boundary* *Master_Supplier_View* meminta *control* *C_Supplier* mengambil data *supplier* dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* *Master_Supplier_View*. Setelah selesai, sistem akan memeriksa format data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data *supplier* disimpan oleh sistem di *database* melalui *control* *C_Supplier* dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.19.



Gambar 3.19 Diagram Sekuensial Mengelola Data *Supplier*

3.3.5.7 Diagram Sekuensial Mengelola Data Produk

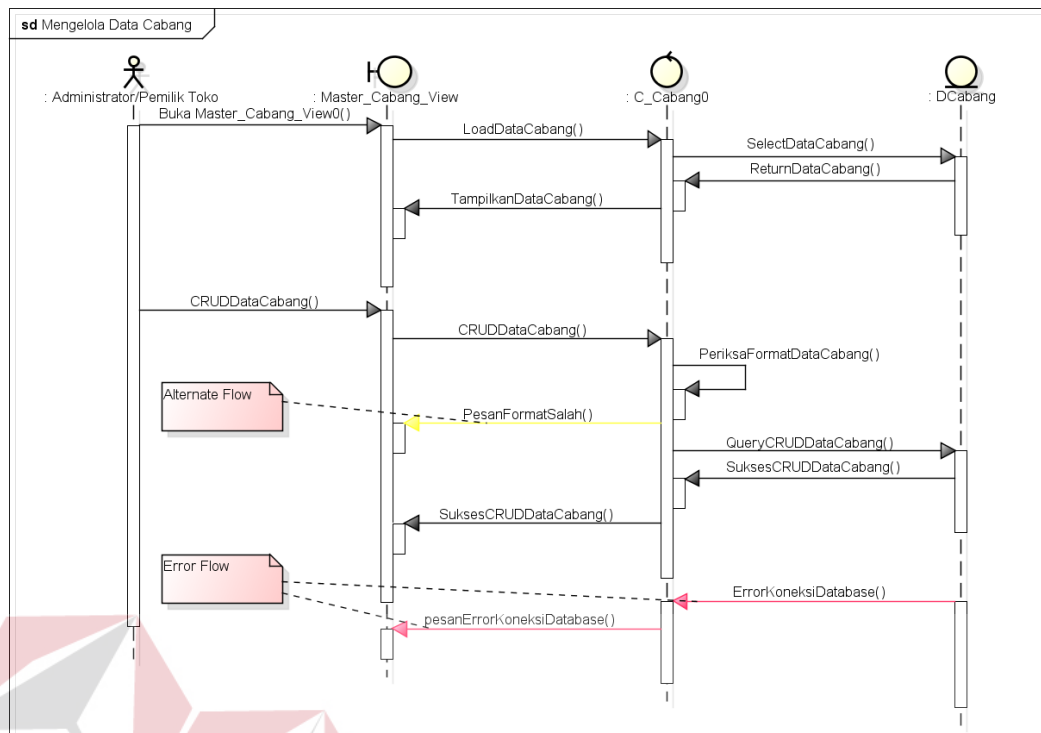
Proses mengelola data produk dimulai dari pengguna mengakses *boundary* *Master_Produk_View*. Lalu *boundary* *Master_Produk_View* meminta *control* *C_Produk* mengambil data produk dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* *Master_Produk_View*. Setelah selesai, sistem akan memeriksa format data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data produk disimpan oleh sistem di *database* melalui *control* *C_Produk* dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.20.



Gambar 3.20 Diagram Sekuensial Mengelola Data Produk

3.3.5.8 Diagram Sekuensial Mengelola Data Cabang

Proses mengelola data cabang dimulai dari pengguna mengakses *boundary* *Master_Cabang_View*. Lalu *boundary* *Master_Cabang_View* meminta *control* *C_Cabang* mengambil data cabang dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* *Master_Cabang_View*. Setelah selesai, sistem akan memeriksa format data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data cabang disimpan oleh sistem di *database* melalui *control* *C_Cabang* dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.21.

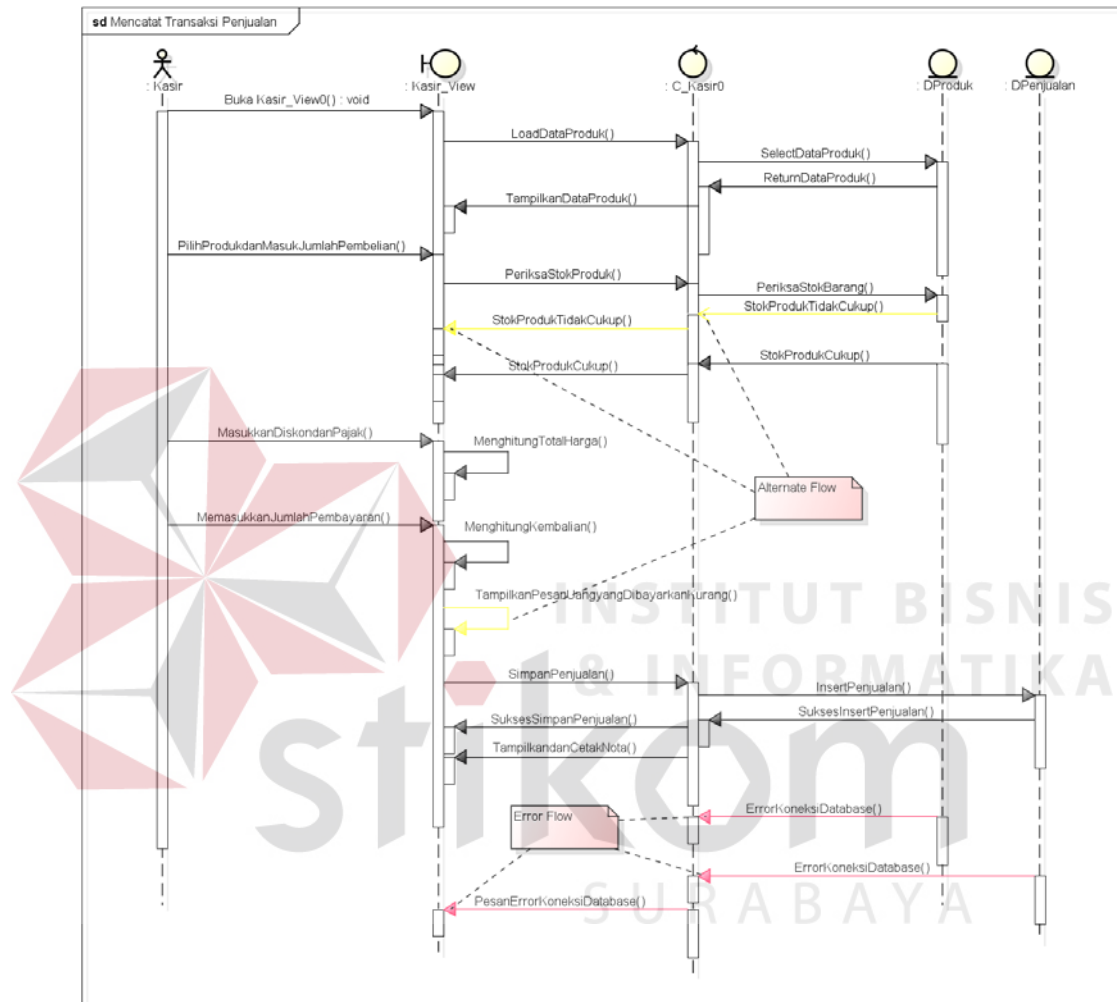


Gambar 3.21 Diagram Sekuensial Mengelola Data Cabang

3.3.5.9 Diagram Sekuensial Mencatat Transaksi Penjualan

Proses mencatat transaksi penjualan dimulai dari pengguna mengakses *boundary* Kasir_View. Pengguna memilih produk dan memasukkan jumlah yang dibeli oleh pelanggan, lalu *boundary* Kasir_View meminta *control* C_Kasir untuk menghitung apakah jumlah stok yang ada mencukupi dengan jumlah yang dibeli. Apabila stok tidak mencukupi, maka sistem meminta untuk mengurangi jumlah yang dibeli. Jika mencukupi, sistem akan menampilkan total harga sebelum pengguna memberikan diskon atau pajak. Jika pengguna memberikan diskon atau pajak, maka sistem akan menghitung ulang total harga. Setelah itu, pengguna memasukkan jumlah uang yang dibayar oleh pelanggan. Apabila uang yang dibayar kurang maka sistem akan memberi pesan bahwa uang yang dibayarkan kurang dan meminta pengguna untuk memasukkan jumlah yang sesuai. Apabila uang yang dibayarkan sesuai maka sistem akan menampilkan jumlah kembalian

yang harus diberikan kepada pelanggan. Setelah selesai maka *boundary* Kasir_View akan meminta *control* C_Kasir untuk menyimpan data penjualan dan mencetak nota penjualan. Untuk lebih jelasnya dapat melihat Gambar 3.22.

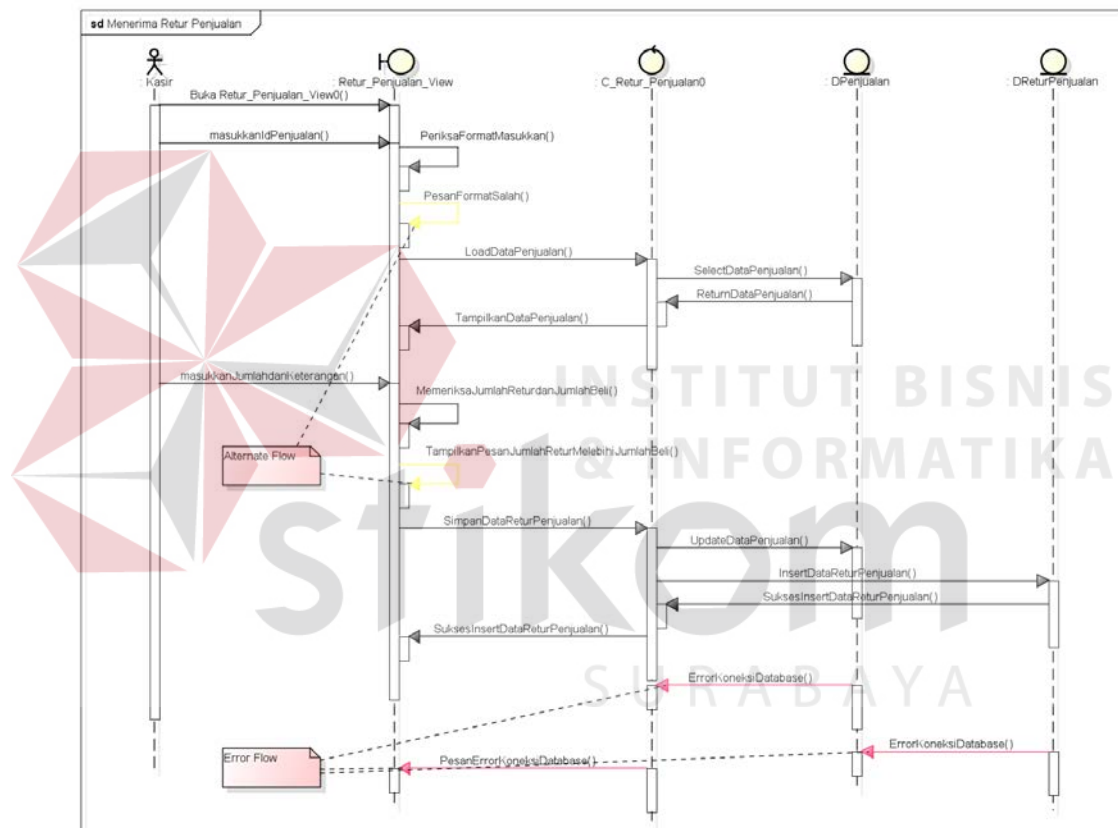


Gambar 3.22 Diagram Sekuensial Mencatat Transaksi Penjualan

3.3.5.10 Diagram Sekuensial Menerima Retur Penjualan

Proses menerima retur penjualan dimulai dari pengguna mengakses *boundary* Retur_Penjualan_View dan memasukkan Id penjualan. Lalu *boundary* Retur_Penjualan_View meminta *control* C_Retur_Penjualan mengambil data penjualan dari *database* untuk ditampilkan. Setelah itu, pengguna memasukkan jumlah dan keterangan dari produk yang akan diretur. Setelah selesai, *boundary*

Retur_Penjualan_View meminta *control* C_Retur_Penjualan untuk menyimpan data retur penjualan. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data retur penjualan disimpan oleh sistem di *database* melalui *control* C_Retur_Penjualan dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.23.

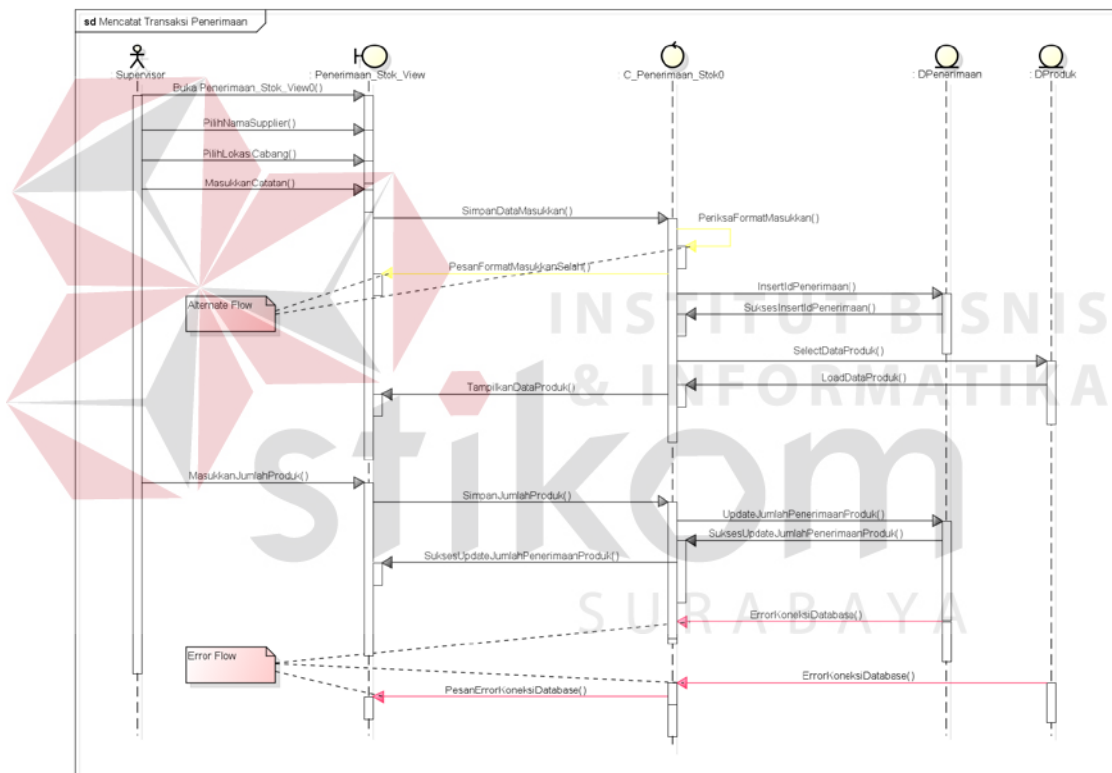


Gambar 3.23 Diagram Sekuensial Menerima Retur Penjualan

3.3.5.11 Diagram Sekuensial Mencatat Transaksi Penerimaan

Proses mencatat transaksi penerimaan dimulai dari pengguna mengakses *boundary* Penerimaan_Stok_View dan memasukkan data penerimaan. Sistem akan memeriksa format masukkan pengguna. Jika benar, sistem akan menyimpan untuk menghasilkan id penerimaan melalui *control* C_Penerimaan_Stok. Setelah

berhasil menyimpan *boundary* *Penerimaan_Stok_View* meminta *control* *C_Penerimaan_Stok* mengambil data produk dari *database* untuk ditampilkan. Setelah itu, pengguna memasukkan jumlah dan keterangan dari produk yang akan ditambah jumlahnya. Setelah selesai, *boundary* *Penerimaan_Stok_View* meminta *control* *C_Penerimaan_Stok* untuk menyimpan data penerimaan stok. Apabila data berhasil disimpan, pesan sukses akan disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.24.

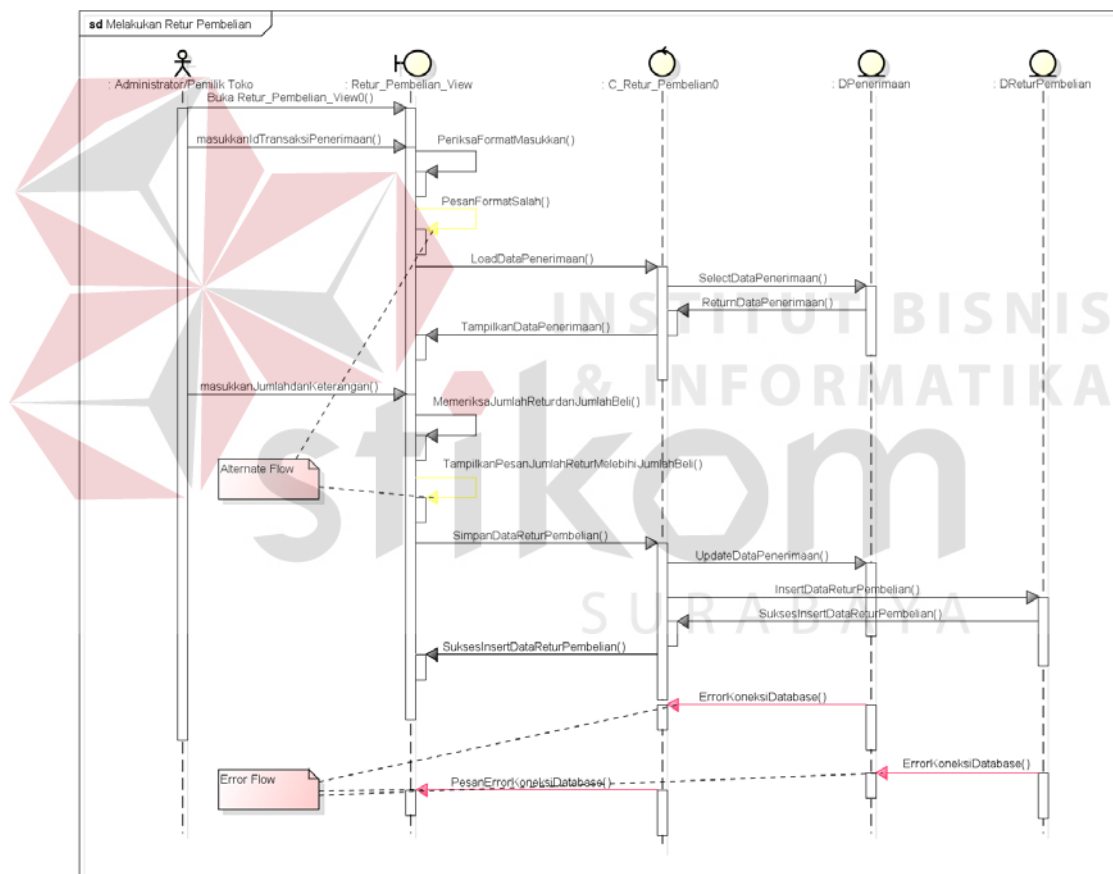


Gambar 3.24 Diagram Sekuensial Mencatat Transaksi Penerimaan

3.3.5.12 Diagram Sekuensial Melakukan Retur Pembelian

Proses menerima retur pembelian dimulai dari pengguna mengakses *boundary* *Retur_Pembelian_View* dan memasukkan Id penerimaan. Lalu *boundary* *Retur_Pembelian_View* meminta *control* *C_Retur_Pembelian* mengambil data penerimaan dari *database* untuk ditampilkan. Setelah itu,

pengguna memasukkan jumlah dan keterangan dari produk yang akan diretur. Setelah selesai, *boundary* *Retur_Pembelian_View* meminta *control* *C_Retur_Pembelian* untuk menyimpan data retur penjualan. Setelah selesai, sistem akan memeriksa *format* data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data retur penjualan disimpan oleh sistem di *database* melalui *control* *C_Retur_Pembelian* dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.25.

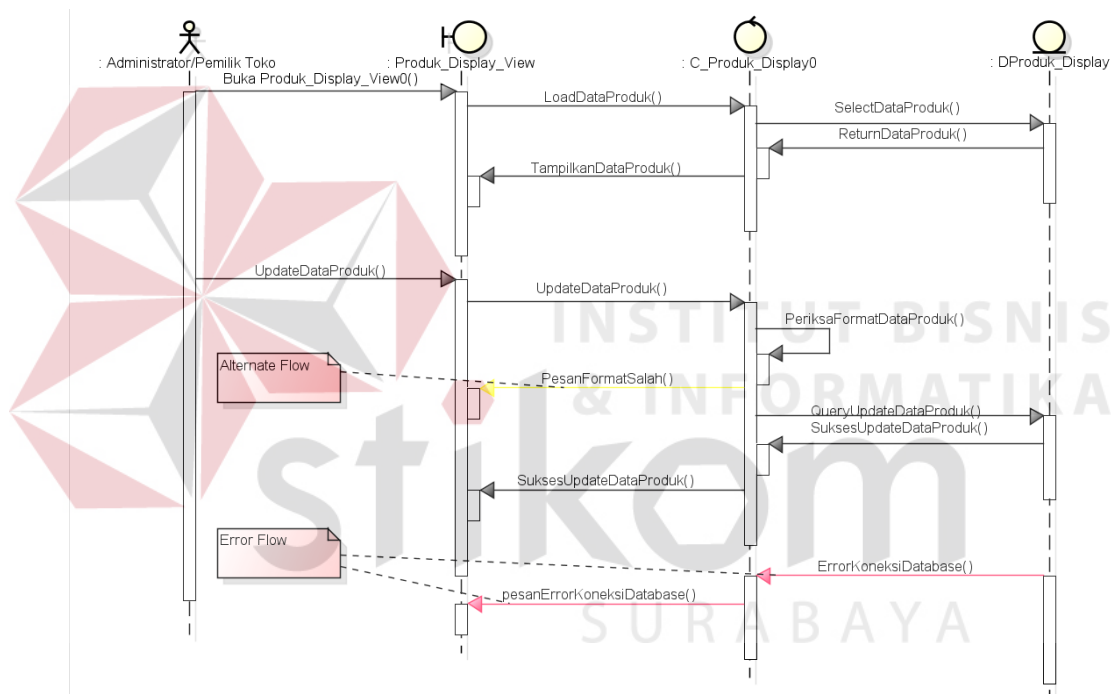


Gambar 3.25 Diagram Sekuensial Melakukan Retur Pembelian

3.3.5.13 Diagram Sekuensial Menampilkan Produk di Web Katalog

Proses menampilkan produk di web katalog dimulai dari pengguna mengakses *boundary* *Produk_Display_View*. Lalu *boundary* *Produk_Display_View* meminta *control* *C_Produk_Display* mengambil data

produk dari *database* untuk ditampilkan. Setelah itu, pengguna melakukan operasi *Update* pada *boundary* *Produk_Display_View* untuk memilih produk yang akan ditampilkan. Setelah selesai, sistem akan memeriksa format data yang baru. Jika ada kesalahan, sistem meminta pengguna untuk mengubahnya. Apabila benar, data produk *display* disimpan oleh sistem di *database* melalui *control* *C_Produk_Display* dan pesan sukses disampaikan oleh sistem. Untuk lebih jelasnya dapat melihat Gambar 3.26.

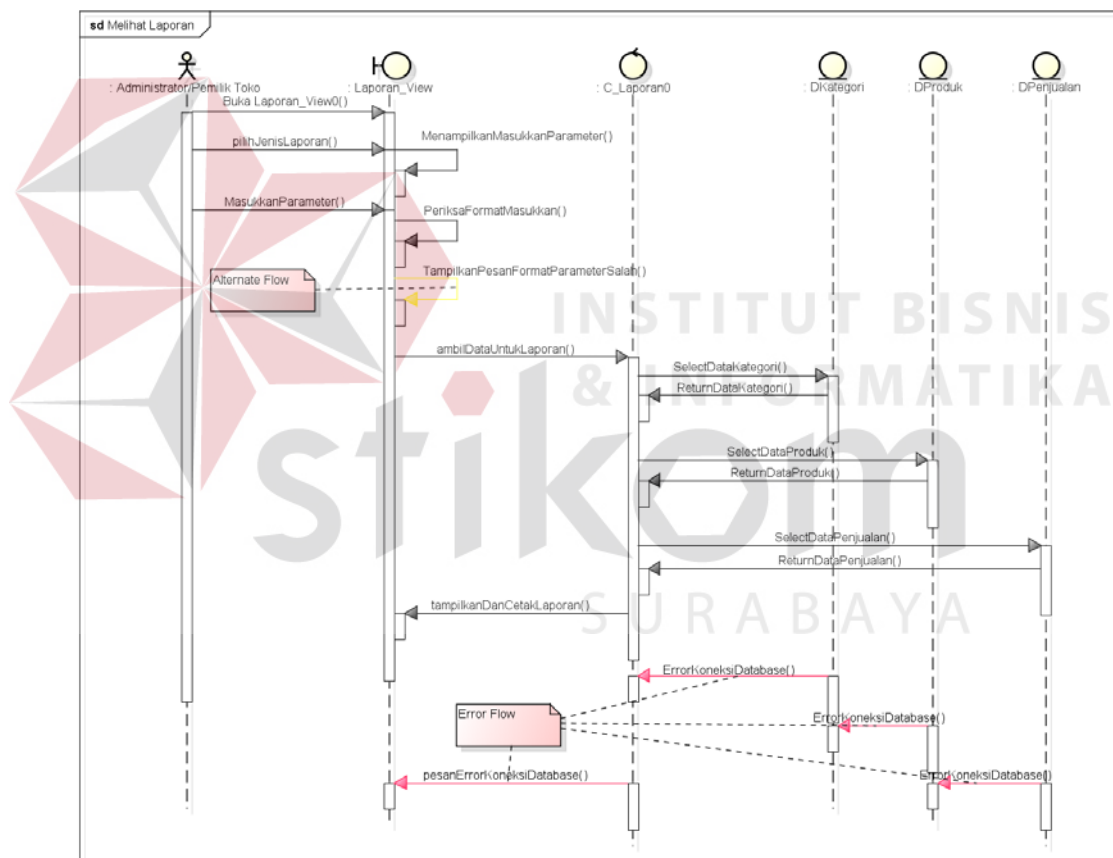


Gambar 3.26 Diagram Sekuensial Menampilkan Produk di Web Katalog

3.3.5.14 Diagram Sekuensial Melihat Laporan

Proses melihat laporan dimulai dari pengguna mengakses *boundary* *Laporan_View* dan memilih jenis laporan yang diinginkan. *Boundary* *Laporan_View* akan menampilkan masukan parameter yang harus diisi oleh pengguna untuk mendapatkan laporan yang diinginkan. Setelah pengguna memasukkan parameter, sistem akan memeriksa apakah ada kesalahan *format*

pada masukan parameter. Apabila ada, sistem akan menyampaikan pesan ke pengguna. Jika benar, *boundary* Laporan_View meminta control C_Laporan untuk mengambil data dari *database* sesuai dengan jenis laporan dan parameter yang telah dimasukkan oleh pengguna. Setelah itu, data akan diolah oleh sistem untuk menjadi informasi dan ditampilkan kepada pengguna untuk kemudian laporan tersebut dapat dicetak oleh pengguna. Untuk lebih jelasnya dapat melihat Gambar 3.27.



Gambar 3.27 Diagram Sekuensial Melihat Laporan

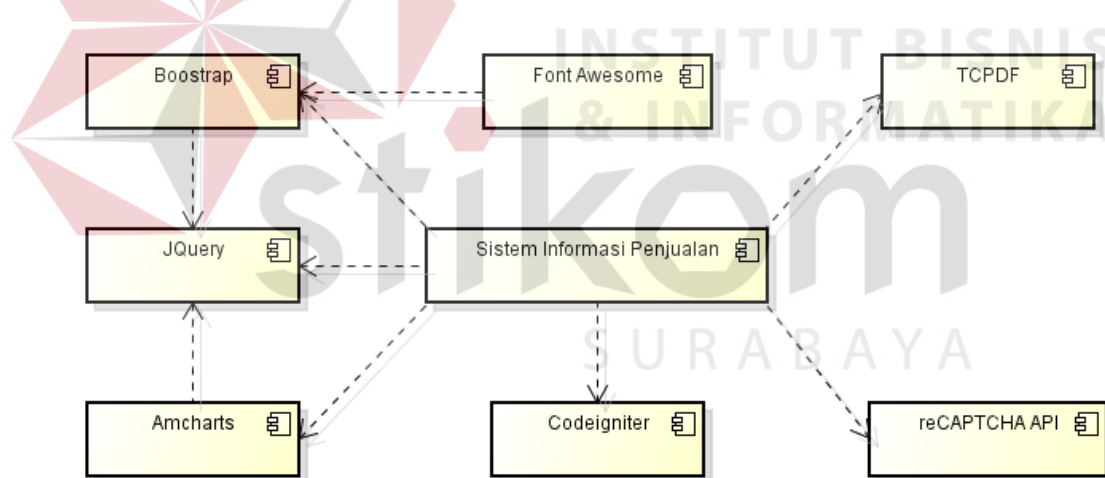
3.3.6 Class Diagram

Class yang dibuat dalam sistem ini menggunakan *pattern* MVC (*Model-View-Controller*). MVC *pattern* sendiri merupakan teknik pemrograman yang memisahkan ketiga bagian program menjadi tiga buah objek yang terpisah.

Pengertian *Class diagram* sendiri adalah diagram yang menunjukkan interaksi antar kelas dalam sistem. Diagram kelas adalah alat perancangan terbaik untuk membantu tim pengembang dalam mendapatkan pola kelas-kelas ke dalam sistem, struktur sistem sebelum menuliskan kode program dan membantu untuk memastikan bahwa sistem adalah rancangan terbaik dari beberapa alternatif. Desain *class* diagram dapat dilihat pada Lampiran 1.

3.3.7 Component Diagram

Gambar 3.28 dibawah menggambarkan *component* diagram dari sistem informasi penjualan berbasis web. Pengertian *component* diagram sendiri adalah diagram yang menunjukkan model secara fisik atau komponen perangkat lunak pada sistem dan hubungannya antar mereka (Sholiq, 2010).

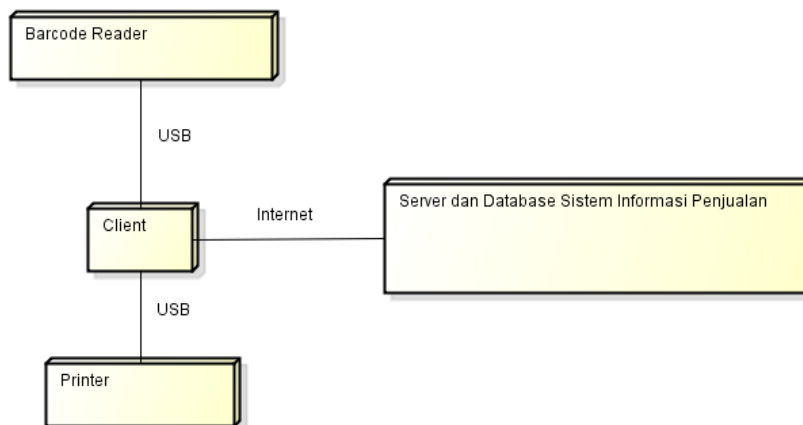


Gambar 3.28 Component Diagram Sistem Informasi Penjualan

3.3.8 Deployment Diagram

Deployment diagram adalah diagram yang menampilkan rancangan fisik jaringan tempat berbagai komponen akan diletakkan disana. Diagram ini menghubungkan titik-titik alat pemroses dan peralatan lainnya dalam sebuah

layout fisik (Sholiq, 2010). Gambar 3.31 menggambarkan *deployment* diagram dari sistem informasi penjualan berbasis web.



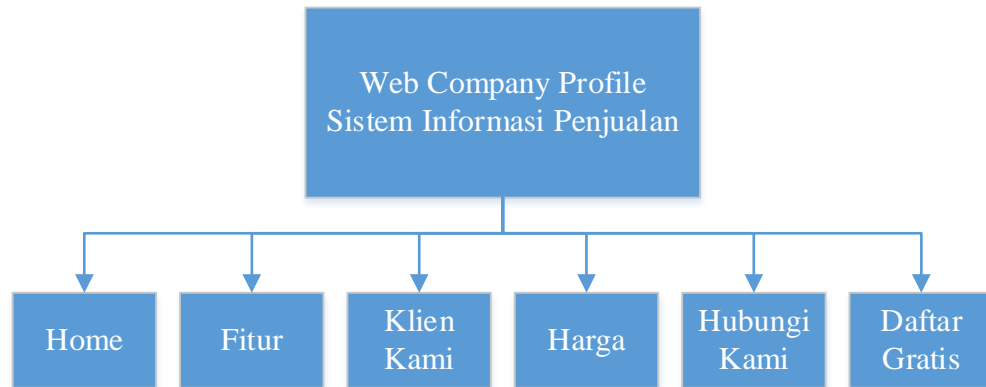
Gambar 3.29 *Deployment* Diagram Sistem Informasi Penjualan

3.3.9 *Physical Data Model (PDM)*

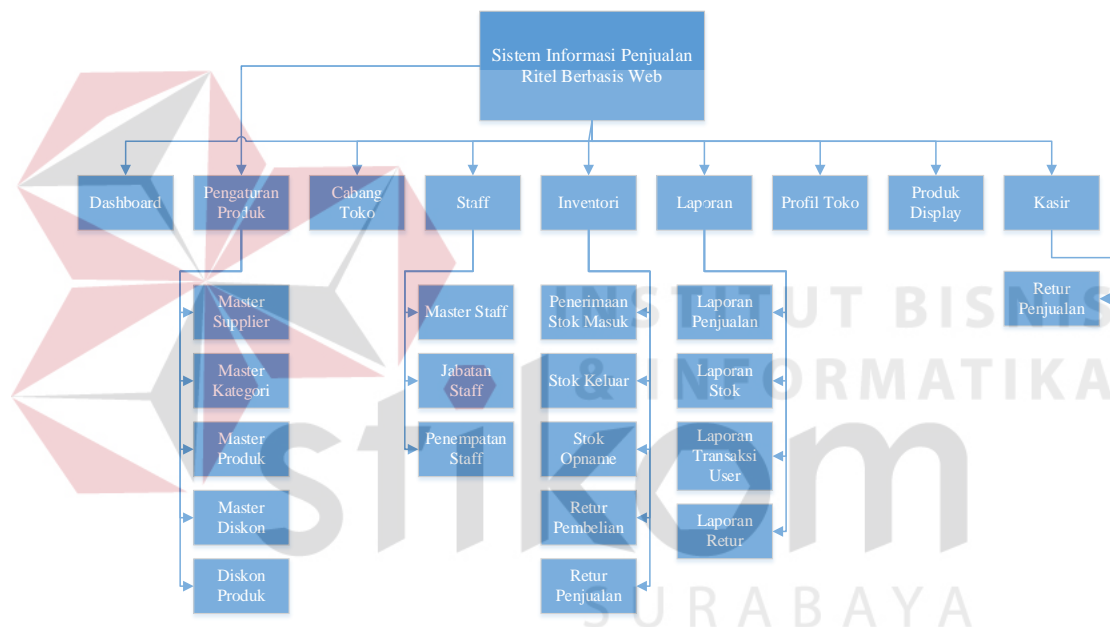
PDM menggambarkan secara detail tentang konsep struktur basis data yang dirancang untuk suatu sistem, yang mana hasil generate dari CDM. Pada PDM ini, juga sudah tergambar jelas relasi antar tabelnya, dengan ditunjukkan primary key dan foreign key dari masing-masing tabel. Semua field yang akan diimplementasikan dalam tabel-tabel pada database telah dipresentasikan secara lengkap, seperti yang terlihat pada Lampiran 2.

3.3.10 *Sitemap*

Sitemap dalam arti umum adalah sebuah file yang berisi daftar *link* yang terdapat pada sebuah web/blog. *Sitemap* adalah salah satu alat bantu yang dapat mempermudah dalam pengenalan peta situs dalam suatu website. Desain *sitemap* sistem informasi penjualan dibagi menjadi dua yaitu *sitemap web company profile* pendaftaran dan *sitemap* pengguna sistem. Desain *sitemap* dari sistem informasi penjualan ritel berbasis web dapat dilihat pada Gambar 3.30 dan Gambar 3.31.



Gambar 3.30 Sitemap Web Company Profile Pendaftaran



Gambar 3.31 Sitemap Sistem Informasi Penjualan Ritel Berbasis Web

3.3.11 Desain *User Interface*

Desain *user interface* adalah rancangan form-form yang akan diimplementasikan kedalam sistem dan berfungsi sebagai antar muka pengguna dengan sistem. Rancangan ini akan menerima *input* / masukan data dari pengguna dan memberikan hasilnya berupa *output* laporan. Selanjutnya masukan data dari pengguna tersebut akan dimasukkan dalam *database* dan dikeluarkan dalam bentuk laporan.

A. Desain *User Interface* Halaman Pendaftaran Toko

Halaman pendaftaran toko digunakan oleh pengguna yang akan mendaftarkan toko mereka untuk menggunakan sistem informasi penjualan berbasis web. Desain *user interface* halaman pendaftaran dapat dilihat pada Gambar 3.32.

Pendaftaran Toko

Nama Lengkap	<input type="text"/>
Jenis Kelamin	<input type="radio"/> Pria <input type="radio"/> Wanita
No Telp	<input type="text"/>
Alamat	<input type="text"/>
Kota	<input type="text" value="Pilih Kota"/>
Email	<input type="text"/>
Password	<input type="text"/>
Nama Domain	<input type="text"/> .ritelaa.com
<input type="checkbox"/> Setujui Persyaratan	<input type="button" value="Daftar"/> <input type="button" value="Batal"/>



Gambar 3.32 Desain Halaman Pendaftaran Toko

B. Desain *User Interface* Login

Halaman *Login* ini digunakan untuk keamanan sistem. Tujuannya adalah supaya sistem digunakan oleh *user* yang berhak memakai dan berjalan sesuai hak aksesnya masing-masing. Adapun *field* yang harus diisi dalam form ini adalah *username* dan *password*. Selanjutnya klik tombol login, secara otomatis sistem akan bekerja untuk validasi pengguna tersebut dan mencari hak aksesnya. *User* juga dapat memilih opsi untuk mengingat *username* dan *password*. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.33.

Gambar 3.33 Desain *Form* Login

C. Desain *User Interface* Halaman *Dashboard*

Desain *user interface* halaman ini dapat dilihat pada Gambar 3.34.

Halaman ini berfungsi untuk menampilkan laporan-laporan yang dibutuhkan Pemilik toko dan Supervisor. Laporan-laporan yang ditampilkan pada halaman *dashboard* antara lain :

- a. Laporan transaksi hari ini.
- b. Laporan pendapatan yang dibuat dalam tiga periode yaitu harian, mingguan dan bulanan.
- c. Grafik pendapatan baik pendapatan bersih maupun kotor yang ditampilkan

dalam rentang waktu enam bulan terakhir.

- d. Laporan kategori produk terlaris yang ditampilkan dalam periode tiap bulan.
- e. Laporan produk terlaris yang ditampilkan dalam periode tiap bulan.
- f. Laporan stok produk yang masuk dalam ambang batas minimum ketersediaan

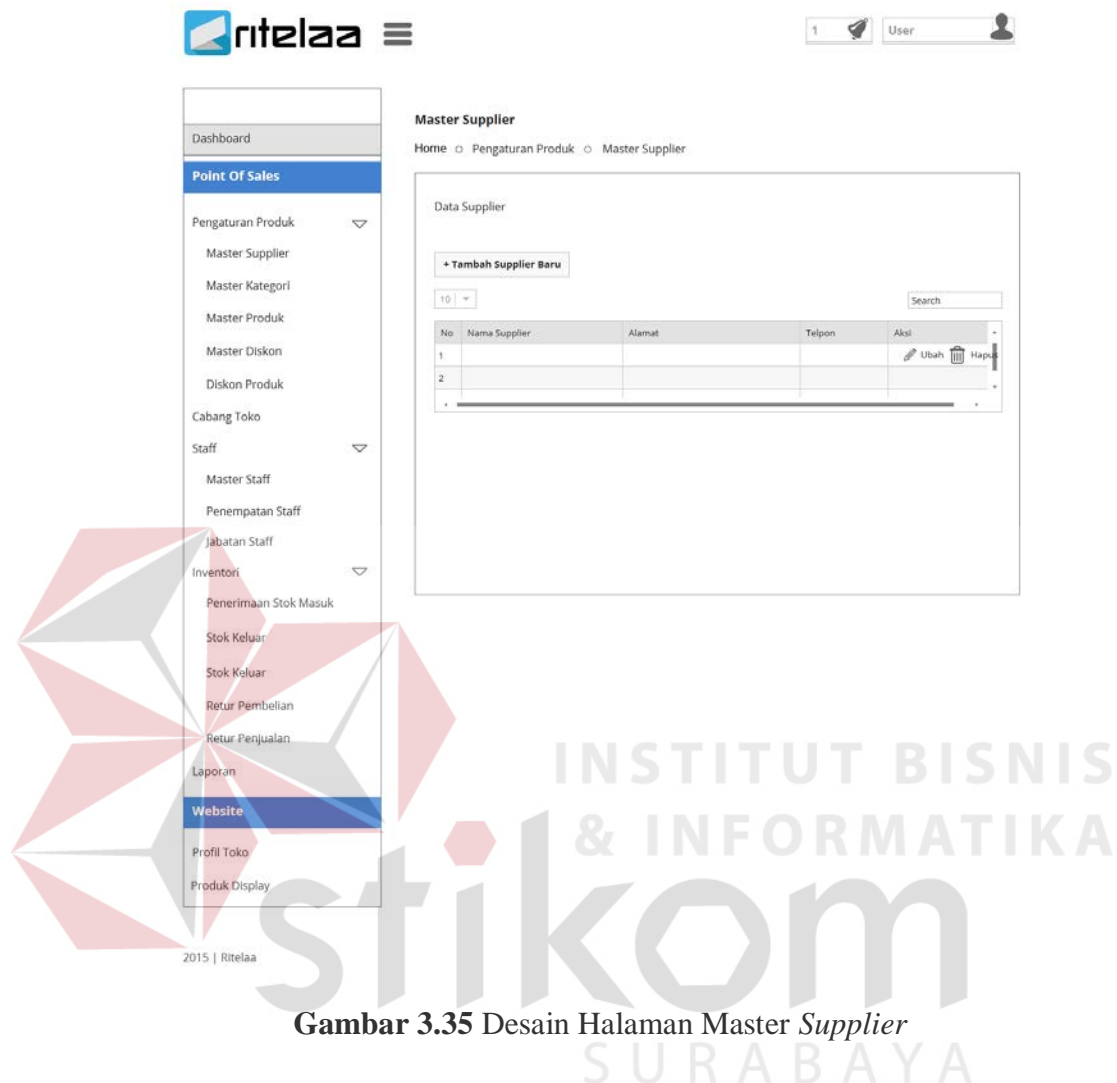


Gambar 3.34 Desain Halaman *Dashboard*

D. Desain *User Interface* Halaman *Master Supplier*

Halaman ini berfungsi untuk maintenance data supplier, yang bertugas untuk melakukan maintenance data supplier ini adalah Administrator / Pemilik Toko dan Supervisor. Administrator /Pemilik Toko dan Supervisor dapat melakukan *insert*, *update*, dan *delete* data supplier. Desain *user interface* halaman

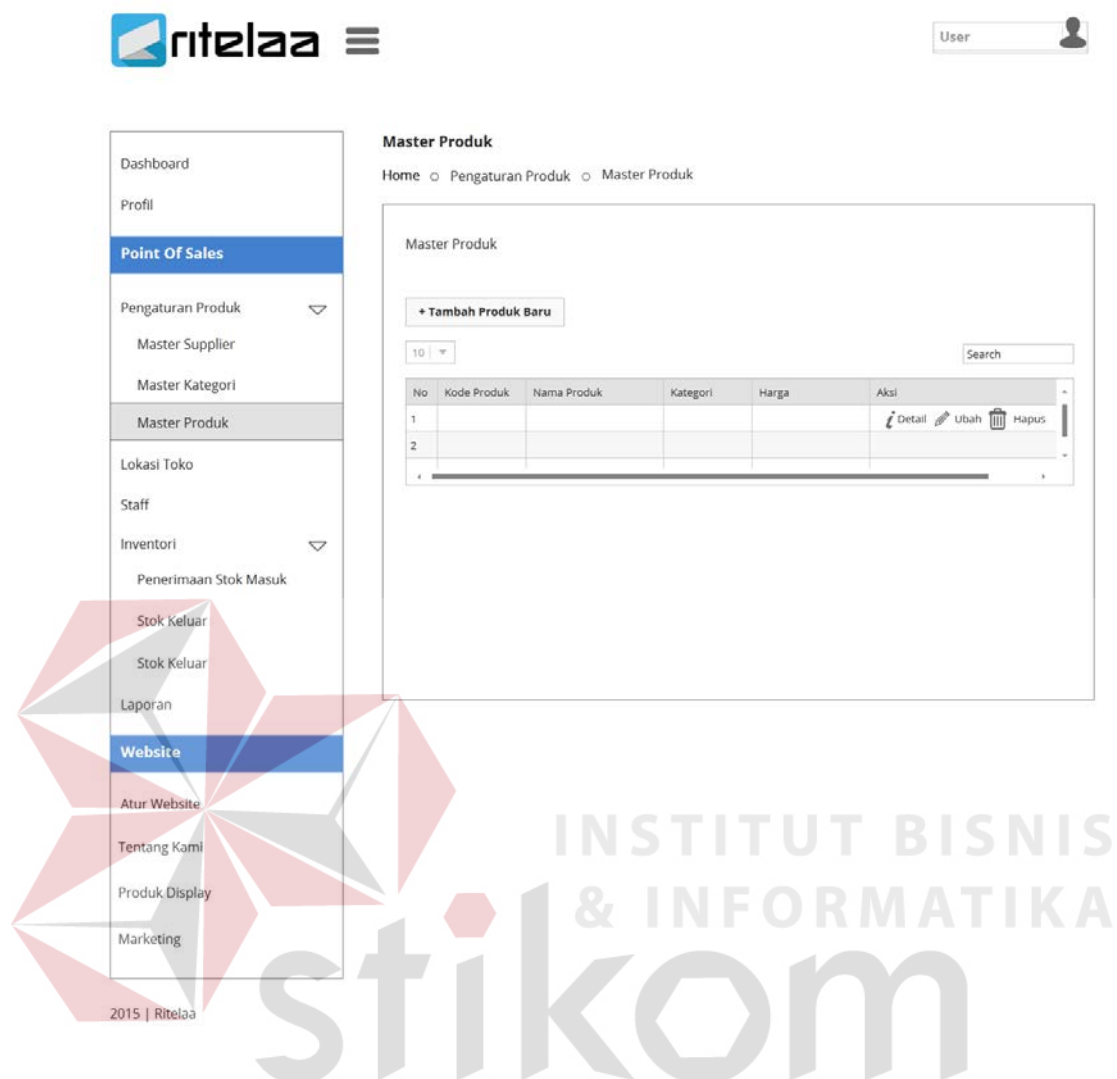
ini dapat dilihat pada Gambar 3.35.



Gambar 3.35 Desain Halaman Master Supplier

E. Desain User Interface Halaman Master Produk

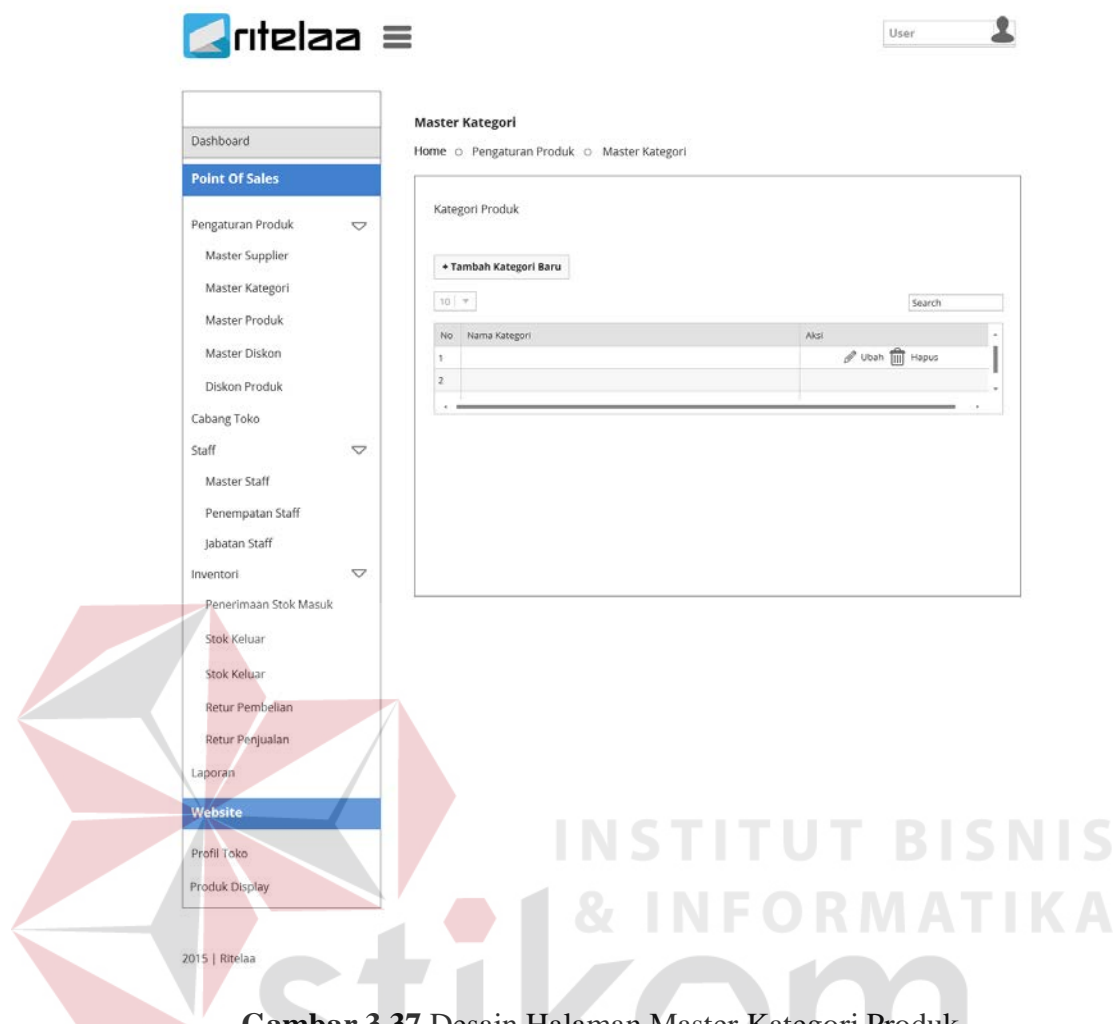
Halaman ini berfungsi untuk maintenance data produk, yang bertugas untuk melakukan maintenance data produk ini adalah Administrator / Pemilik Toko dan Supervisor. Administrator /Pemilik Toko dan Supervisor dapat melakukan *insert*, *update*, dan *delete* data produk. Data yang dimasukkan pada halaman ini yaitu nama, kategori, *supplier*, minimal stok, harga beli, harga jual, deskripsi, dan gambar produk. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.36.



Gambar 3.36 Desain Halaman Master Produk

F. Desain *User Interface* Halaman Master Kategori Produk

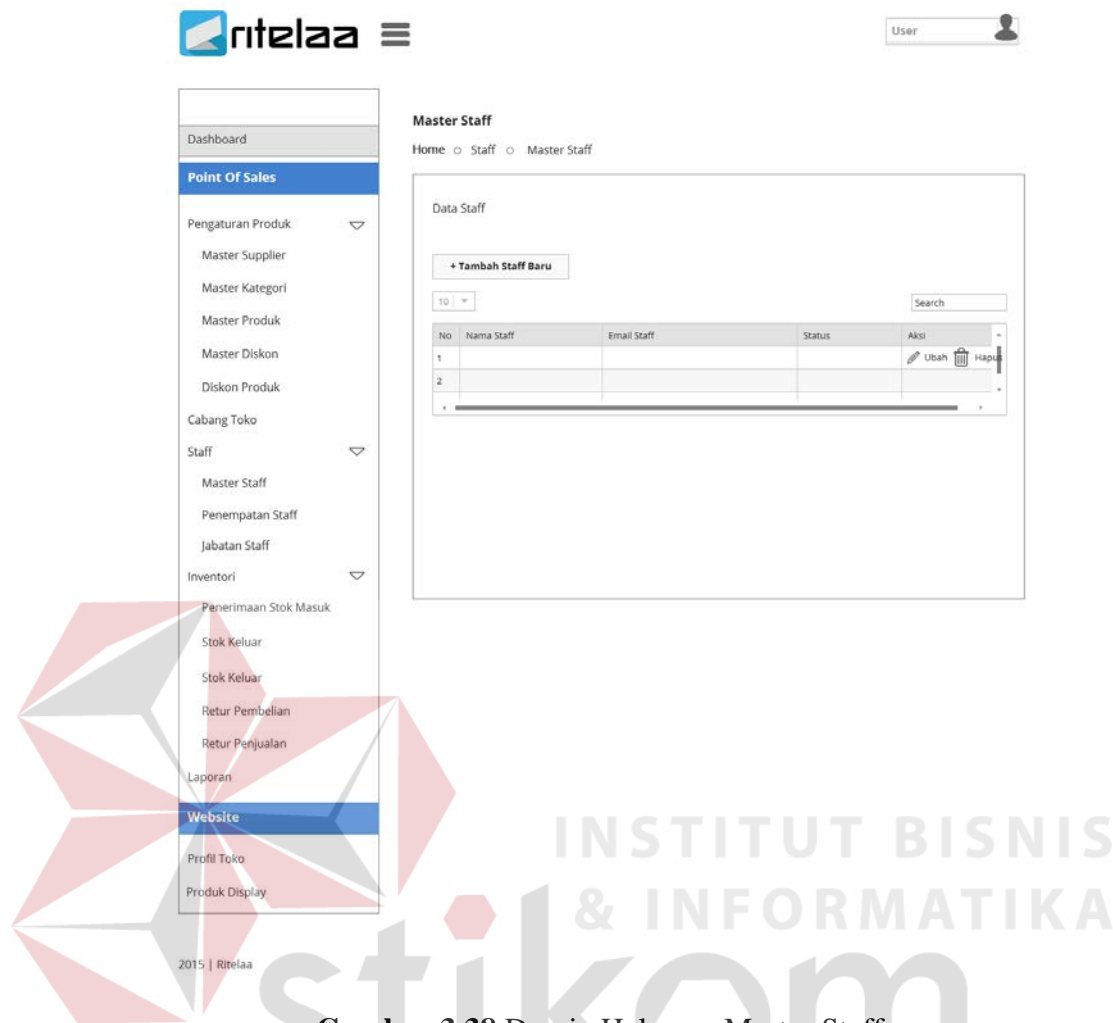
Halaman ini berfungsi untuk *maintenance* data kategori produk yang dijual, yang bertugas untuk melakukan *maintenance* data kategori produk ini adalah Administrator / Pemilik Toko dan Supervisor. Administrator /Pemilik Toko dan Supervisor dapat melakukan *insert*, *update*, dan *delete* data kategori produk. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.37.



Gambar 3.37 Desain Halaman Master Kategori Produk

G. Desain *User Interface* Halaman Master Staff

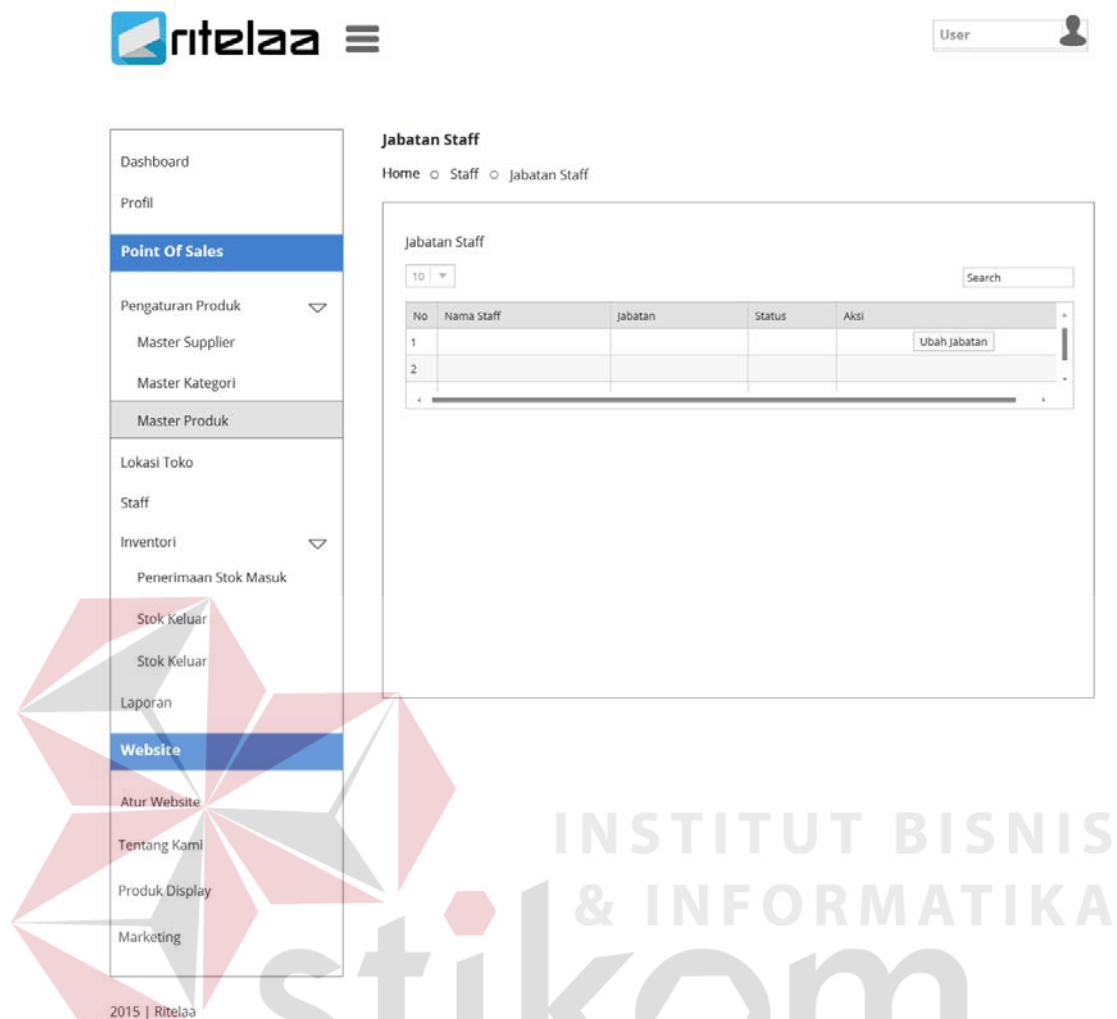
Halaman ini berfungsi untuk *maintenance* data staff, yang bertugas untuk melakukan *maintenance* data staff ini adalah Administrator / Pemilik Toko dan Supervisor. Administrator /Pemilik Toko dan Supervisor dapat melakukan *insert*, *update*, dan *delete* data staff. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.38.



Gambar 3.38 Desain Halaman Master Staff

H. Desain *User Interface* Halaman Jabatan Staff

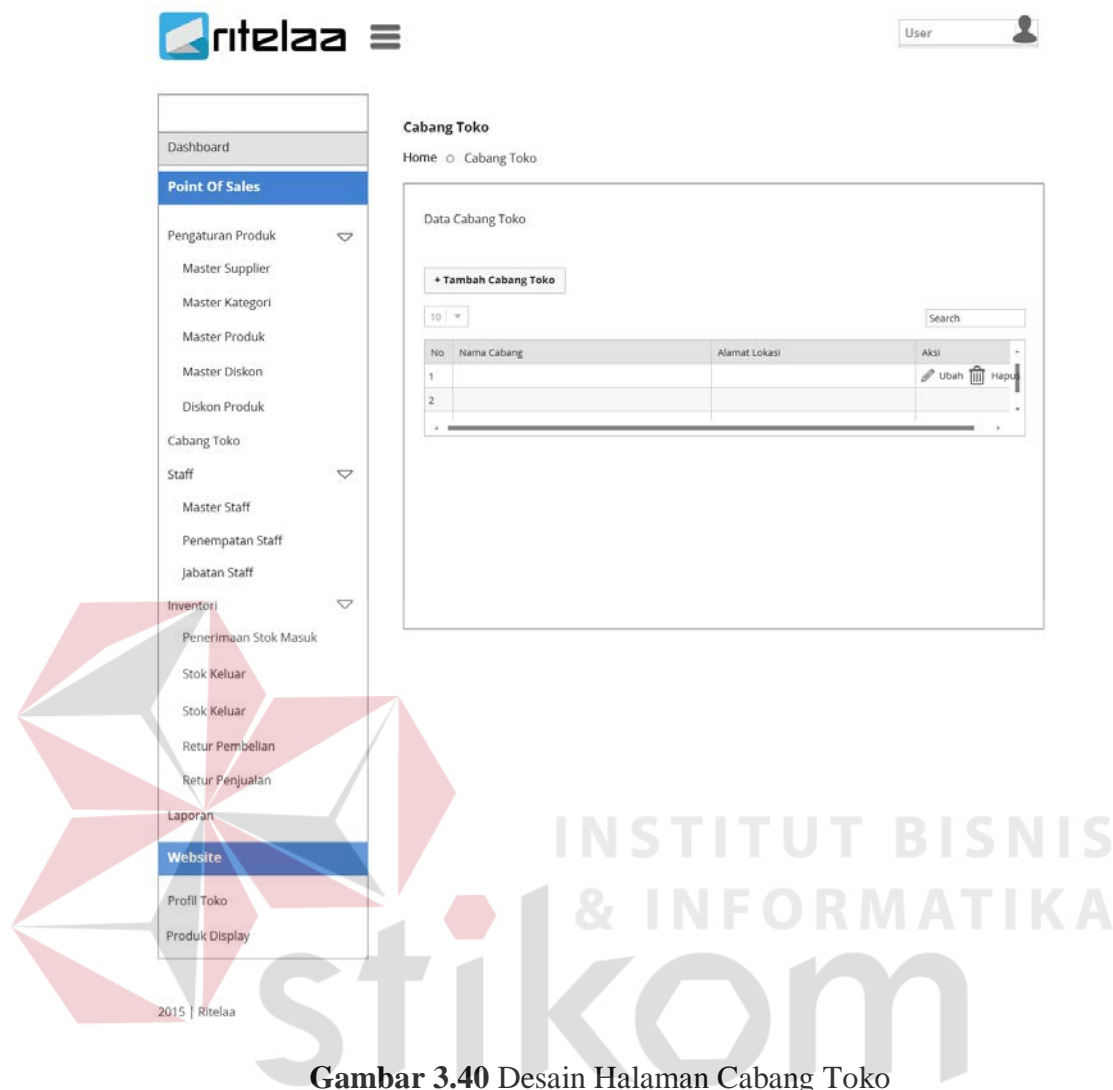
Halaman ini berfungsi untuk memberikan hak akses login ke dalam sistem, yang bertugas untuk memberikan hak akses ini adalah Administrator / Pemilik Toko. Administrator /Pemilik Toko dapat memilih staff yang akan diberikan hak akses sebagai Adminisrator, Supervisor atau Kasir. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.39.



Gambar 3.39 Desain Halaman Jabatan Staff

I. Desain *User Interface* Halaman Cabang Toko

Halaman ini berfungsi untuk maintenance data cabang toko apabila pengguna memiliki lebih dari satu lokasi usaha, sedangkan yang bertugas untuk melakukan maintenance data cabang toko ini adalah Administrator / Pemilik Toko dan Supervisor. Administrator /Pemilik Toko dan Supervisor dapat melakukan *insert*, *update*, dan *delete* data cabang toko. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.40.



Gambar 3.40 Desain Halaman Cabang Toko

J. Desain User Interface Halaman Produk Display

Halaman produk *display* digunakan untuk menampilkan semua daftar produk yang dimiliki toko untuk kemudian dipilih ditampilkan di web katalog. Dalam menampilkan produk di web katalog, pemilik dapat dengan bebas memilih produk apa saja yang ingin dipromosikan di web katalog. Selain itu pemilik dapat menambahkan keterangan tentang produk yang dijual. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.41.



Gambar 3.41 Desain Halaman Produk *Display*

K. Desain *User Interface* Halaman Profil Toko

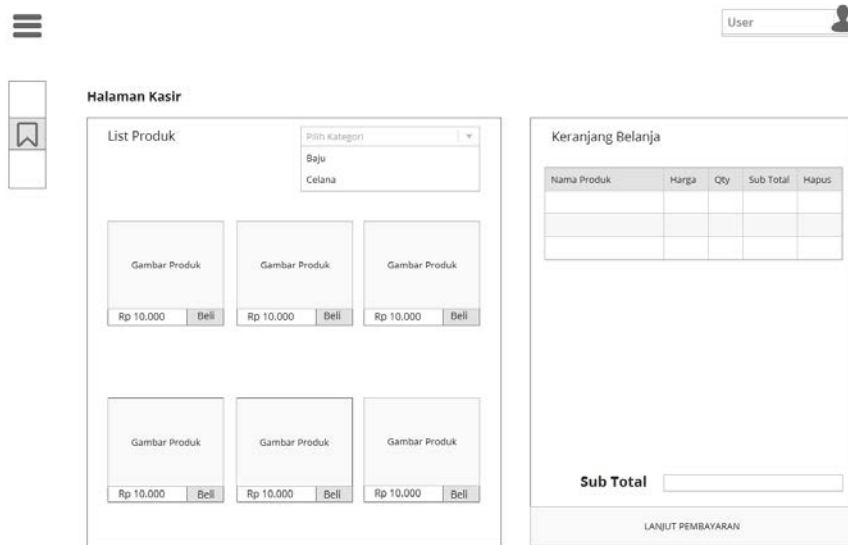
Pada halaman ini pemilik toko diwajibkan untuk mengisi identitas toko yang dimiliki. Pemilik toko diharuskan untuk mengisi nama toko, deskripsi toko, alamat toko, telepon, kota, provinsi, email dan logo. Identitas ini digunakan untuk ditampilkan di halaman web katalog, sehingga calon pelanggan dapat dengan mudah untuk berkomunikasi dengan pemilik toko ketika akan membeli produk yang dijual. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.42.

The image shows a web application interface for 'Ritelaa'. On the left is a sidebar menu with categories: 'Point Of Sales', 'Pengaturan Produk' (with sub-items: Master Supplier, Master Kategori, Master Produk, Master Diskon, Diskon Produk), 'Cabang Toko', 'Staff' (with sub-items: Master Staff, Penempatan Staff, Jabatan Staff), 'Inventori' (with sub-items: Penerimaan Stok Masuk, Stok Keluar, Stok Keluar, Retur Pembelian, Retur Penjualan), 'Laporan', 'Website' (with sub-items: Profil Toko, Produk Display), and 'Produk Display'. The main content area is titled 'Profil Toko' and has a breadcrumb 'Home > Profil Toko'. The form contains the following fields: 'Nama Toko' (text input), 'Deskripsi Toko' (text area), 'Alamat Toko' (text area), 'Provinsi' (dropdown menu with 'Pilih Provinsi'), 'Kota' (dropdown menu with 'Pilih Kota'), 'Kecamatan' (dropdown menu with 'Pilih Kecamatan'), 'Telpon 1' (text input), 'Telpon 2' (text input), 'Email' (text input), 'Logo' (file upload with 'Pilih File' button), 'Slogan' (text area), and 'Kata Kunci' (text input). A 'Simpan' button is located at the bottom of the form. The footer of the page reads '2015 | Ritelaa'.

Gambar 3.42 Desain Halaman Profil Toko

L. Desain *User Interface* Halaman Transaksi Penjualan

Halaman ini digunakan oleh staff dengan hak akses sebagai kasir untuk melakukan transaksi penjualan dengan pelanggan. Data transaksi penjualan akan disimpan secara otomatis ke dalam sistem informasi penjualan untuk kemudian diolah sistem untuk ditampilkan dalam bentuk informasi kepada Pemilik Toko / Administrator. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.43.

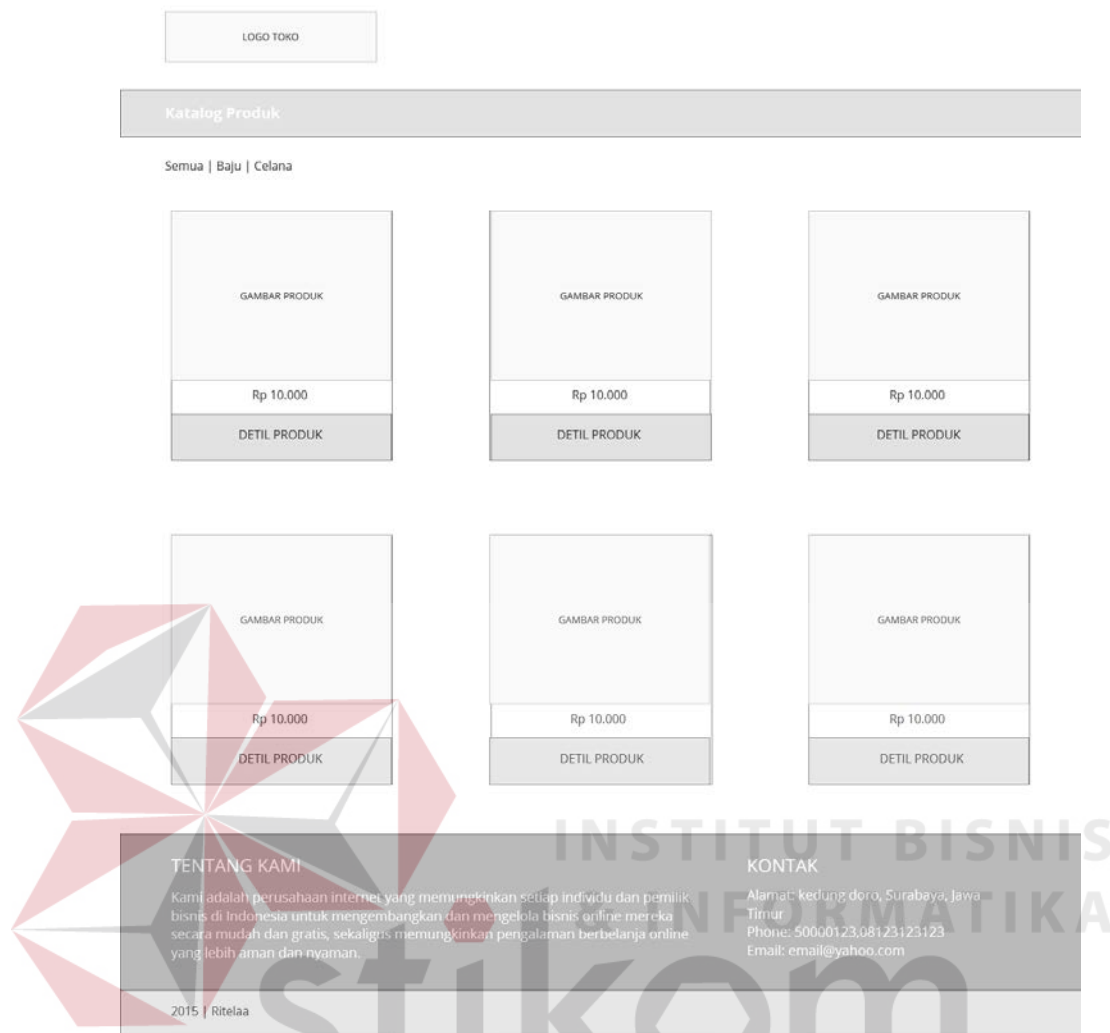


2015 | Ritelaa

Gambar 3.43 Desain Halaman Kasir

M. Desain *User Interface* Halaman Web Katalog

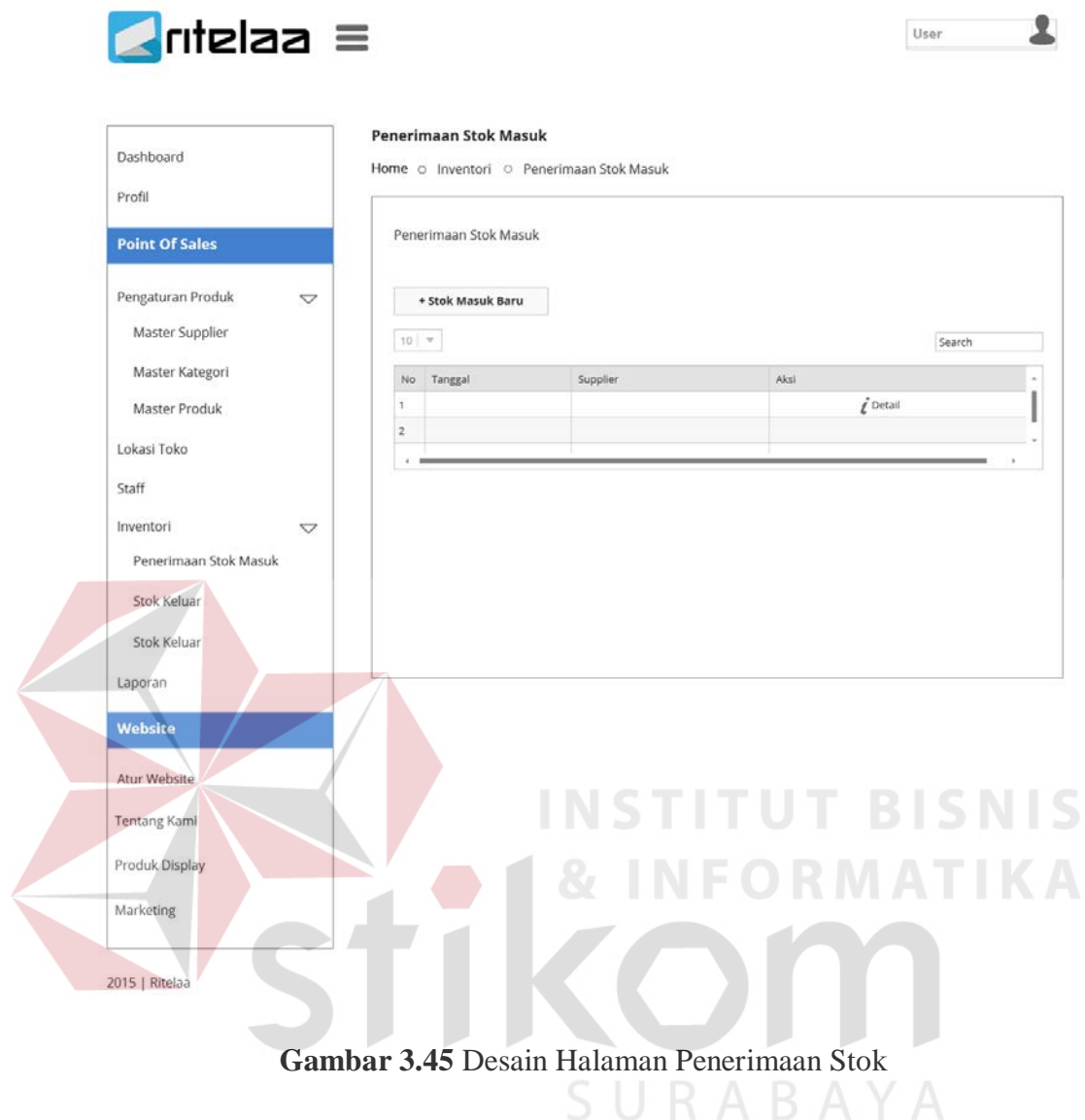
Halaman web katalog digunakan untuk menampilkan produk yang dipilih oleh pemilik toko. Dalam web katalog ini menampilkan detail produk seperti nama produk, gambar produk, harga produk, kategori produk dan deskripsi produk. Selain menampilkan detail produk, web katalog juga menampilkan informasi tentang toko, alamat toko, *email* toko, dan nomor telepon. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.44.



Gambar 3.44 Desain Halaman Web Katalog

N. Desain *User Interface* Halaman Penerimaan Stok Produk

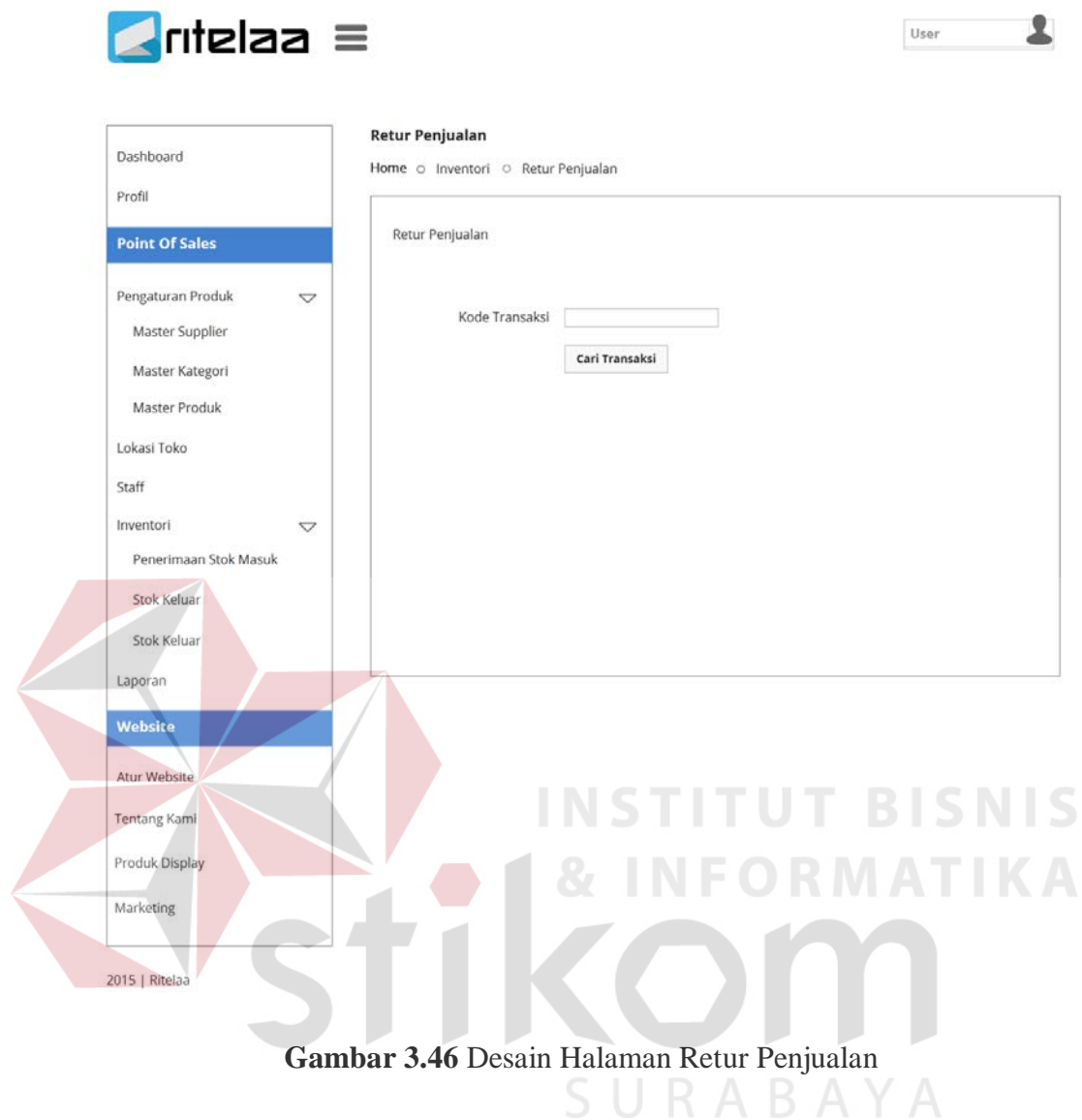
Halaman ini berfungsi untuk mencatat stok produk yang masuk dari supplier, yang bertugas untuk melakukan pencatatan ini adalah Administrator / Pemilik Toko dan Supervisor. Penerimaan stok masuk ini dijadikan acuan untuk perhitungan stok yang tersedia pada gudang. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.45.



Gambar 3.45 Desain Halaman Penerimaan Stok

O. Desain *User Interface* Halaman Retur Penjualan

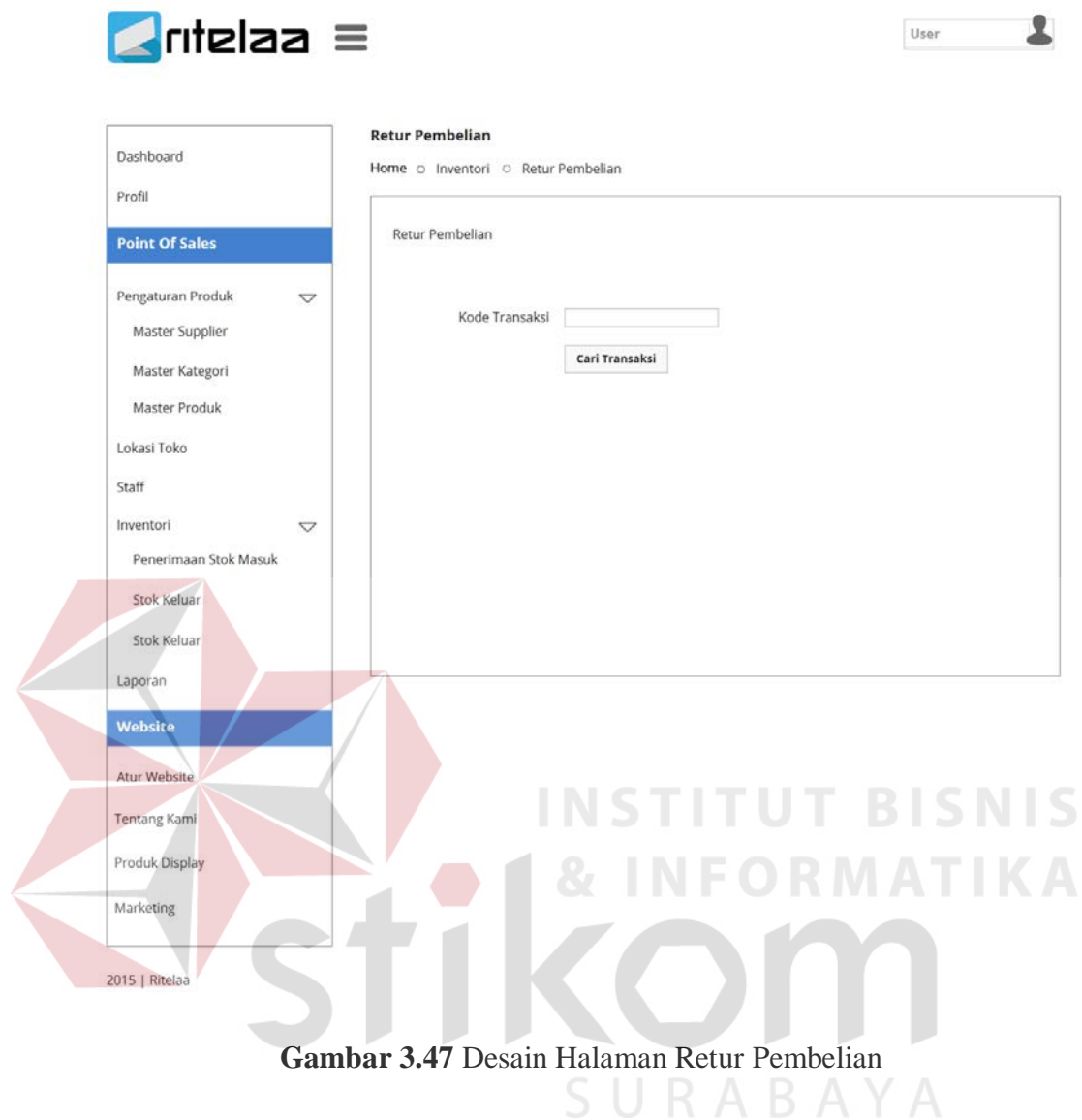
Halaman ini berfungsi untuk mencatat pengembalian barang dari pelanggan toko karena hal tertentu, seperti rusak dalam perjalanan atau pengiriman barang yang tidak memenuhi spesifikasi yang diinginkan pelanggan toko. Sedangkan yang bertugas untuk melakukan pencatatan ini adalah Administrator / Pemilik Toko dan Supervisor. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.46.



Gambar 3.46 Desain Halaman Retur Penjualan

P. Desain *User Interface* Halaman Retur Pembelian

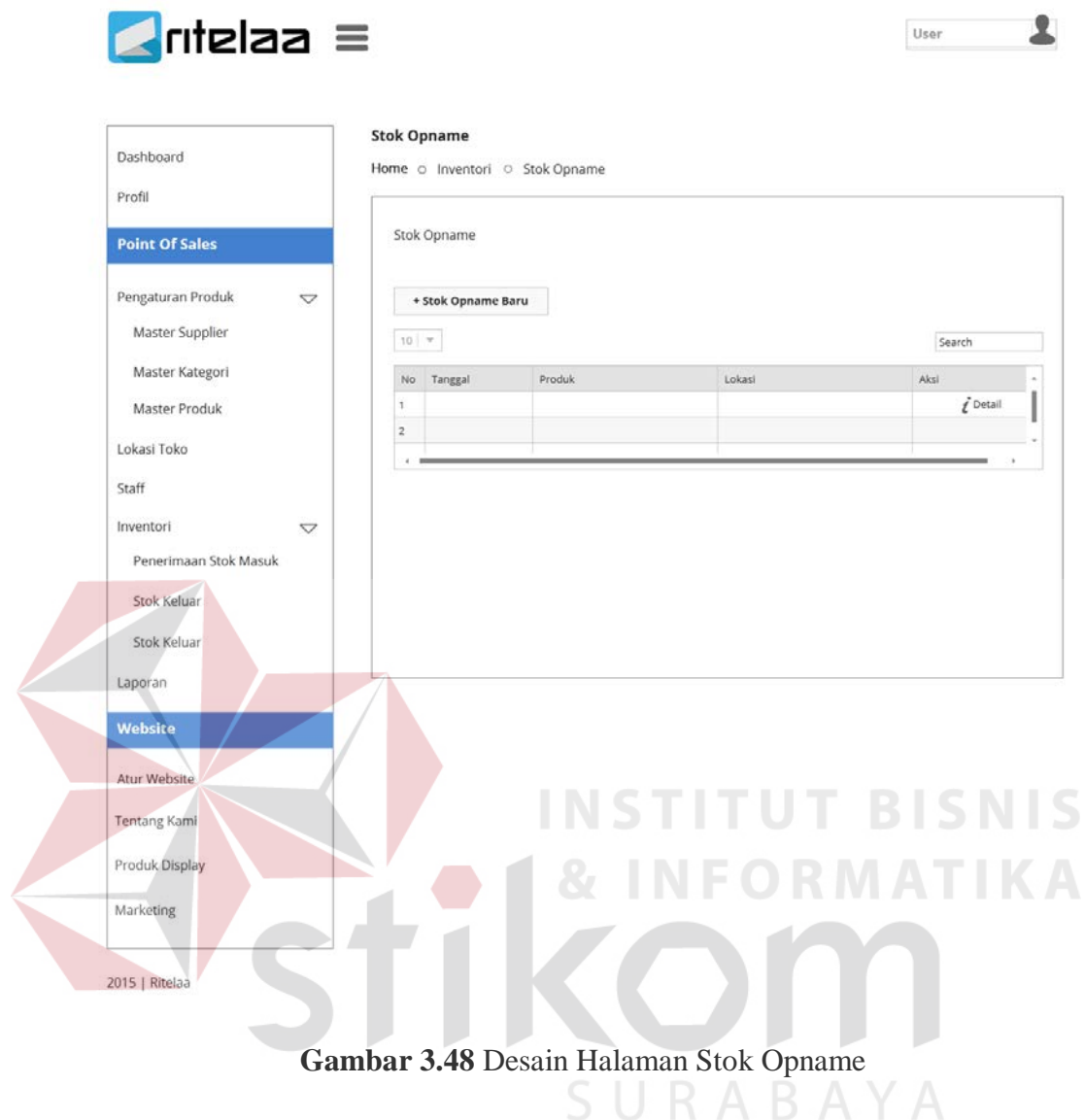
Halaman ini berfungsi untuk mencatat pengembalian barang yang sudah dibeli kepada *supplier*. Karena adanya ketidakcocokan dengan spesifikasi yang tercantum dalam surat order pembelian, barang mengalami kerusakan dalam pengiriman yang dijanjikan oleh *supplier*, yang bertugas untuk melakukan pencatatan ini adalah Administrator / Pemilik Toko dan Supervisor. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.47.



Gambar 3.47 Desain Halaman Retur Pembelian

Q. Desain *User Interface* Halaman Stok Opname

Halaman ini berfungsi untuk menyocokkan jumlah stok yang ada pada sistem dengan stok nyata yang ada pada toko, yang bertugas untuk melakukan stok opname ini adalah Administrator / Pemilik Toko dan Supervisor. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.48.



Gambar 3.48 Desain Halaman Stok Opname

R. Desain *User Interface* Halaman Laporan Penjualan

Halaman ini berfungsi untuk menampilkan laporan penjualan kepada Pemilik Toko / Administrator. Laporan yang ditampilkan dapat dipilih sesuai dengan periode waktu yang diinginkan. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.49.

The screenshot displays the 'Ritelaa' web application interface. On the left is a sidebar menu with options like 'Dashboard', 'Profil', 'Point Of Sales' (highlighted), 'Pengaturan Produk', 'Lokasi Toko', 'Staff', 'Inventori', and 'Website'. The main area is titled 'Laporan Penjualan' and includes a date range selector ('Tanggal' and 's/d'), 'Load' and 'Cetak' buttons, a search bar, and a table with the following structure:

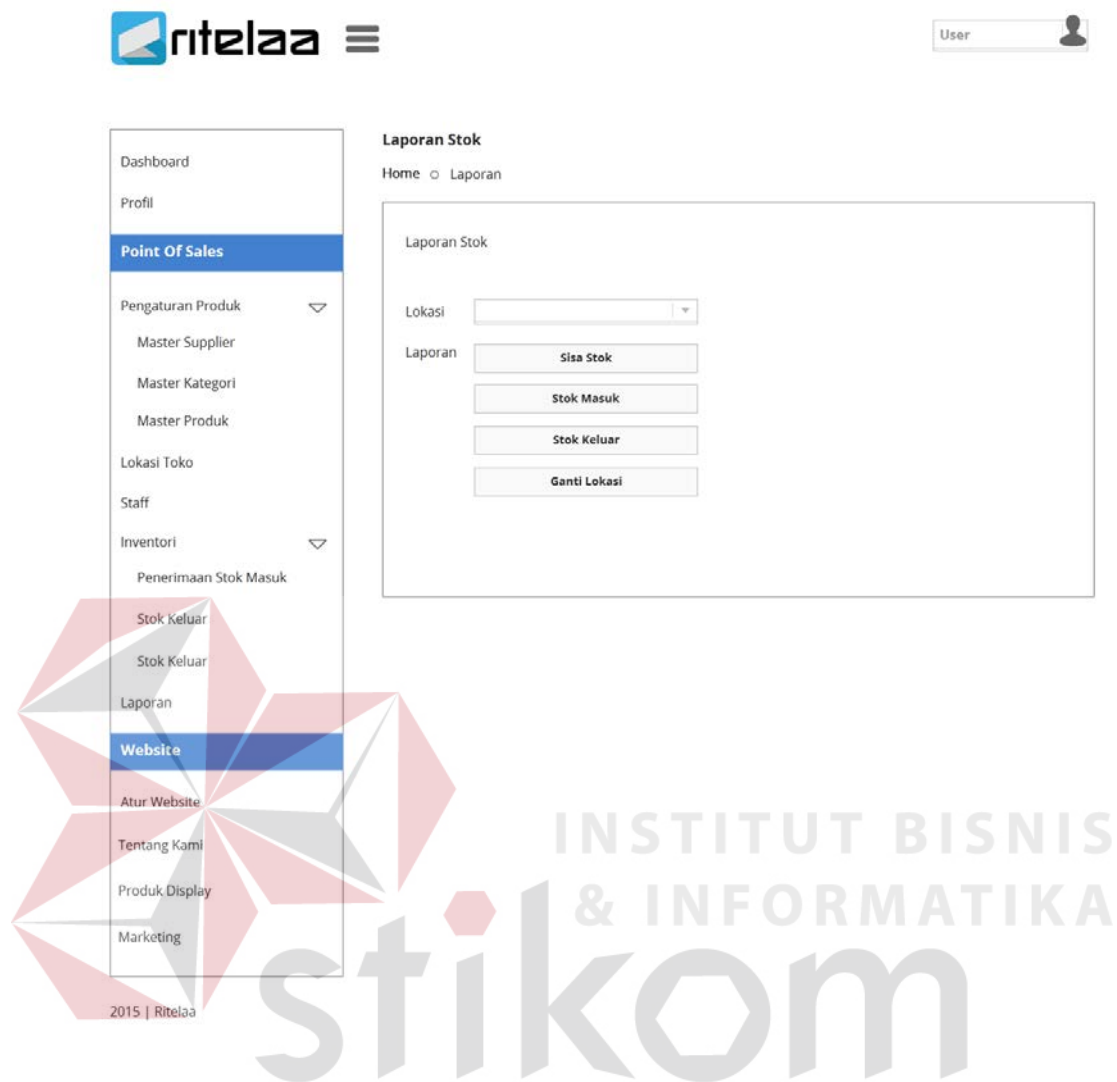
No	Kode Produk	Nama Produk	Jml	Sales	Diskon	Net
1						
2						

A large watermark for 'stikom SURABAYA' is overlaid on the bottom half of the image.

Gambar 3.49 Desain Halaman Laporan Penjualan

S. **Desain *User Interface* Halaman Laporan Stok**

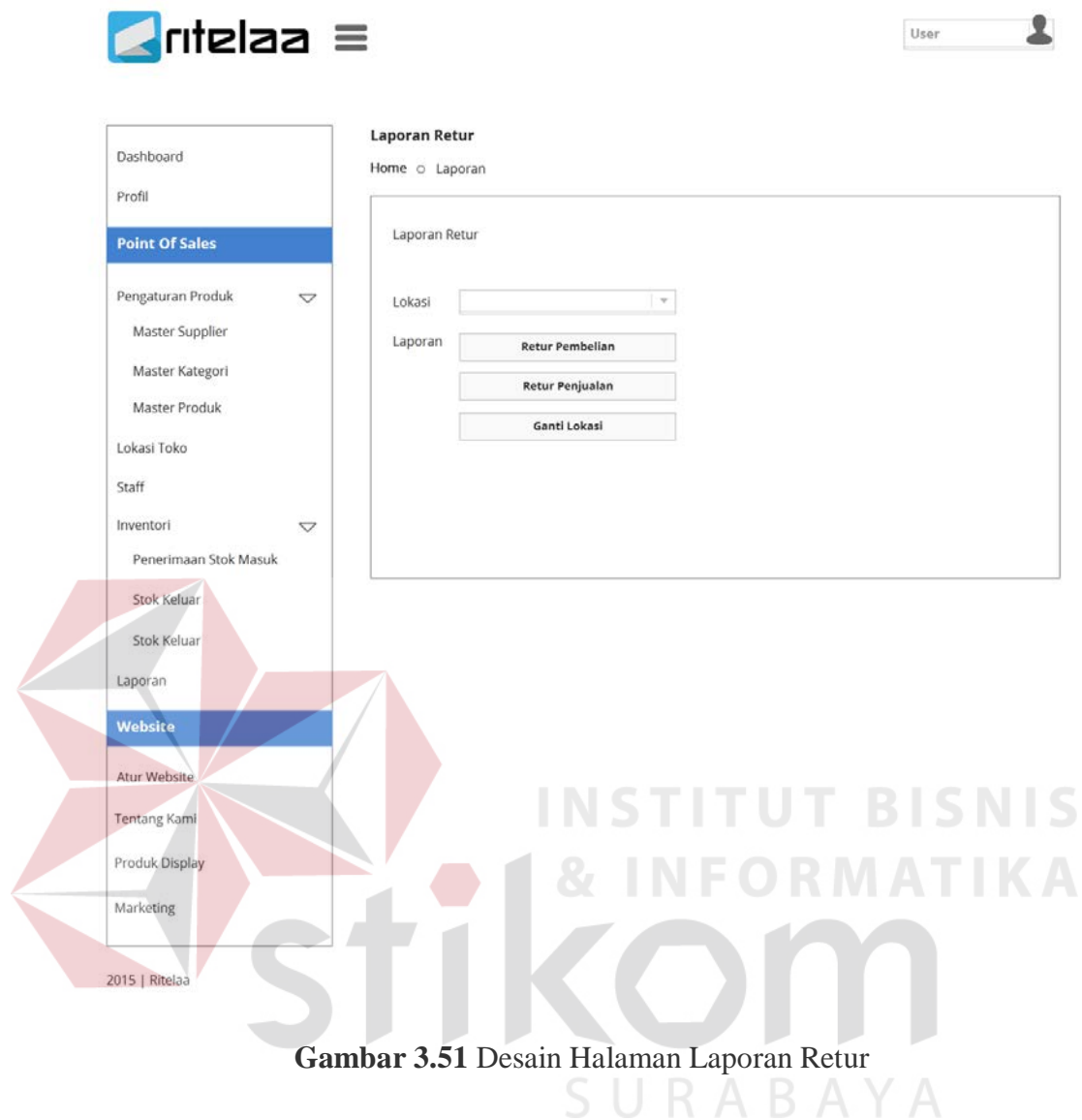
Halaman ini digunakan untuk menampilkan informasi stok produk yang tersisa pada toko kepada Pemilik Toko / Administrator. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.50.



Gambar 3.50 Desain Halaman Laporan Stok

T. Desain *User Interface* Halaman Laporan Retur

Halaman ini digunakan untuk menampilkan informasi laporan retur penjualan dan pembelian dalam periode waktu tertentu. Laporan ini dapat diakses oleh Pemilik Toko / Administrator. Desain *user interface* halaman ini dapat dilihat pada Gambar 3.51.



Gambar 3.51 Desain Halaman Laporan Retur

3.3.12 Desain *Input/Output*

Desain *input/output* digunakan untuk menggambarkan *input* dan *output* dari lingkungan eksternal sistem yang akan dimasukkan ke dalam sistem. Dalam sistem informasi penjualan yang akan dibangun, desain *input* tidak digambarkan karena semua *input* dimasukkan secara langsung ke dalam sistem. Desain *output* yang dihasilkan sistem adalah nota penjualan beserta laporan yang dibutuhkan oleh Pemilik Toko.

A. Desain *Ouput* Nota Transaksi Penjualan

Nota transaksi penjualan adalah *output* untuk setiap transaksi penjualan yang dimasukkan di halaman kasir. Nota penjualan berisi tanggal transaksi, nomor transaksi, nama produk yang dibeli beserta jumlah, harga satuan, diskon satuan dan total harga. Desain *output* nota penjualan dapat dilihat pada Gambar 3.52.

Nota Penjualan				
No. NOTA	: 1602130001	Lokasi	:	Pusat
Tanggal (Jam)	: 13-02-2016 (10-06)	Kasir	:	Ageng Permadi

Flatshoes Nadine LP	1	30.000	0	30.000
Sepatu Denim	3	35.000	0	105.000
Sub Total				: 135.000
Diskon				: - 0

Pajak 0%				: 0
Total				: 135.000
Pembayaran				: 140.000
Kembali				: 5.000

Gambar 3.52 Desain *Ouput* Nota Transaksi Penjualan

B. Desain *Ouput* Laporan Penjualan

Laporan penjualan adalah *output* dari sistem untuk membantu Pemilik Toko dalam mengetahui total penjualan yang diperoleh. Laporan penjualan ini dapat dilihat dalam periode harian, bulanan atau tahunan sesuai dengan kebutuhan. Laporan penjualan ini berisi tanggal cetak laporan, periode waktu laporan yang dihasilkan, kode produk, nama produk, jumlah produk yang terjual, diskon produk dan total penjualan. Desain *output* laporan penjualan dapat dilihat pada Gambar 3.53.

Dicetak pada: Jum at, 12 Februari 2016

Laporan Penjualan Produk

Periode : 01-12-2015 s/d 12-02-2016

No.	Kode Produk	Nama Produk	JML	Sales	Diskon	Net
1.	P00001	Flatsshoes Nadine LP	21	Rp.630.000	Rp.0	Rp.630.000
2.	P00002	Sepatu Denim	11	Rp.385.000	Rp.0	Rp.385.000
3.	P00003	Flatshoes Nadine Tali	5	Rp.175.000	Rp.0	Rp.175.000
4.	P00004	Sepatu Denim Tali	3	Rp.111.000	Rp.0	Rp.111.000
TOTAL						Rp.1.301.000

Gambar 3.53 Desain *Output* Laporan Penjualan

C. Desain *Output* Ranking Penjualan Berdasarkan Kategori

Laporan ranking penjualan berdasarkan kategori adalah *output* dari sistem untuk membantu Pemilik Toko dalam mengetahui kategori produk yang paling banyak dibeli oleh pelanggan. Laporan ranking penjualan berdasarkan kategori ini dapat dilihat dalam periode harian, bulanan atau tahunan sesuai dengan kebutuhan. Laporan ranking penjualan berdasarkan kategori ini berisi tanggal cetak laporan, periode waktu laporan yang dihasilkan, nama kategori dan jumlah produk yang terjual. Desain *output* laporan ranking penjualan berdasarkan kategori dapat dilihat pada Gambar 3.54.

Dicetak pada: Jum at, 12 Februari 2016

Rangking Penjualan Kategori

Periode : 01-12-2015 s/d 12-02-2016

No.	Nama Kategori	Jumlah Penjualan
1.	Sepatu	37 pcs
2.	Sepatu Sandal	3 pcs

Gambar 3.54 Desain *Output* Ranking Penjualan Berdasarkan Kategori

D. Desain *Output* Ranking Penjualan Berdasarkan Produk

Laporan ranking penjualan berdasarkan produk adalah *output* dari sistem untuk membantu Pemilik Toko dalam mengetahui produk yang paling banyak

dibeli oleh pelanggan. Laporan ini dapat membantu Pemilik Toko dalam mengambil keputusan untuk jumlah *reorder* produk ke *supplier*, pemberian diskon dan strategi promosi. Laporan ranking penjualan berdasarkan produk ini dapat dilihat dalam periode harian, bulanan atau tahunan sesuai dengan kebutuhan. Laporan ranking penjualan berdasarkan produk ini berisi tanggal cetak laporan, periode waktu laporan yang dihasilkan, kode produk, nama produk dan jumlah produk yang terjual. Desain *output* laporan ranking penjualan berdasarkan produk dapat dilihat pada Gambar 3.55.

Dicetak pada: Sabtu, 13 Februari 2016

Rangking Penjualan Produk

Periode : 01-12-2015 s/d 13-02-2016

No.	Kode Produk	Nama Produk	Jumlah Penjualan
1.	P00001	Flatshoes Nadine LP	22 pcs
2.	P00002	Sepatu Denim	14 pcs
3.	P00003	Flatshoes Nadine Tali	5 pcs
4.	P00004	Sepatu Denim Tali	3 pcs

Gambar 3.55 Desain *Output* Ranking Penjualan Berdasarkan Produk

E. Desain *Output* Laporan Sisa Stok Produk

Laporan sisa stok produk adalah *output* dari sistem untuk Pemilik Toko agar dapat memantau dan mengetahui stok terkini yang tersedia. Hal ini membantu Pemilik Toko untuk mengambil keputusan *order* ke *supplier* sehingga toko tidak kehabisan stok. Laporan sisa stok produk ini berisi tanggal cetak laporan, kode produk, nama produk, stok minimum dan sisa stok yang tersedia. Dalam *output* laporan sisa stok produk sistem akan secara otomatis memberikan peringatan dengan warna merah apabila terdapat produk yang memiliki jumlah stok dibawah ambang batas yang ditentukan. Desain *output* laporan sisa stok produk dapat dilihat pada Gambar 3.56.

dicetak pada: Jum at, 12 Februari 2016

Sisa Stok Produk

lokasi: Pusat

No.	Kode Produk	Nama Produk	Stok Minimum	Sisa Stok
1.	P00001	Flatshoes Nadine LP	30 pcs	23 pcs
2.	P00002	Sepatu Denim	20 pcs	23 pcs

keterangan

■ : Jumlah stok kurang atau sama dengan jumlah stok minimum produk.

Gambar 3.56 Desain *Output* Laporan Sisa Stok**F. Desain *Output* Laporan *History* Transaksi *User***

Laporan *history* transaksi *user* adalah output dari sistem untuk Pemilik Toko melihat *history* transaksi yang dilakukan oleh staff dengan hak akses sebagai kasir. Pada laporan ini Pemilik Toko dapat melihat total transaksi yang dilakukan oleh kasir. Laporan ini berisi berisi tanggal cetak laporan, nama staff, status staff, jumlah transaksi, nominal transaksi, dan jumlah diskon. Desain *output* laporan *history* transaksi *user* dapat dilihat pada Gambar 3.57.

dicetak pada: Sabtu, 13 Februari 2016

Laporan Transaksi Kasir

lokasi: Pusat

No.	Nama Staff	Status Aktif	Transaksi	Nominal Transaksi	Diskon
1.	Ageng Permadi	Aktif	11	Rp.1.436.000	Rp.10.000

Gambar 3.57 Desain *Output* Laporan *History* Transaksi *User***G. Desain *Output* Laporan Retur Penjualan**

Laporan retur penjualan adalah *output* dari sistem untuk membantu Pemilik Toko dalam mengetahui *history* produk yang di retur oleh pelanggan. Dengan adanya laporan ini Pemilik Toko dapat mengetahui kualitas produk yang dijual oleh *supplier* sehingga dapat meminimalisasi kerugian dalam membeli barang yang tidak berkualitas. Semakin tinggi intensitas retur penjualan maka

dapat merusak citra toko yang menjual. Laporan retur penjualan ini dapat dilihat dalam periode harian, bulanan atau tahunan sesuai dengan kebutuhan. Laporan retur penjualan ini berisi tanggal cetak laporan, periode waktu laporan yang dihasilkan, kode produk, nama produk, jumlah produk yang diretur, tanggal transaksi retur, staff yang melayani retur dan keterangan dari produk yang diretur. Desain *output* laporan retur penjualan dapat dilihat pada Gambar 3.58.

dicetak pada: Sabtu, 13 Februari 2016

Laporan Retur Penjualan

Lokasi: Pusat

Durasi Waktu : 01-12-2015 s/d 13-02-2016

No.	Kode Produk	Nama Produk	Jumlah Retur
1.	P00001	Flatshoes Nadine LP	1 pcs

No.	Produk	Jumlah	Tanggal	Staff	Keterangan
1.	P00001 - Flatshoes Nadine LP	1 pcs	03-02-2016	Diva Zhora	Tali lepas

Gambar 3.58 Desain *Output* Laporan Retur Penjualan

H. Desain *Output* Laporan Retur Pembelian

Laporan retur pembelian adalah output dari sistem untuk membantu Pemilik Toko dalam mengetahui *history* produk yang diretur kepada *supplier*. Dengan adanya laporan ini Pemilik Toko dapat mengetahui kualitas produk yang dijual oleh *supplier* sehingga dapat meminimalisasi kerugian dalam membeli barang yang tidak berkualitas. Laporan retur pembelian ini dapat dilihat dalam periode harian, bulanan atau tahunan sesuai dengan kebutuhan. Laporan retur penjualan ini berisi tanggal cetak laporan, periode waktu laporan yang dihasilkan, kode produk, nama produk, jumlah produk yang diretur, tanggal transaksi retur, nama *supplier* dan keterangan dari produk yang diretur. Desain *output* laporan retur pembelian dapat dilihat pada Gambar 3.59.

dicetak pada: Sabtu, 13 Februari 2016

Laporan Retur Pembelian

Lokasi: Pusat

Durasi Waktu : 01-12-2015 s/d 13-02-2016

No.	Kode Produk	Nama Produk	Jumlah Retur		
No.	Produk	Jumlah	Tanggal	Staff	Keterangan

Gambar 3.59 Desain *Ouput* Laporan Retur Pembelian

3.3.13 Rancangan Pengujian dan Evaluasi Sistem

Sistem informasi penjualan yang telah dirancang dan dibangun harus diuji untuk mengetahui apakah sistem berjalan sesuai dengan yang diharapkan. Secara garis besar, uji coba yang dilakukan adalah pendaftaran toko, proses login, pengelolaan data master, pencatatan transaksi, menampilkan laporan kepada pemilik toko serta menampilkan produk ke dalam web katalog. Selain yang telah disebutkan, pengujian akan dilakukan terhadap beberapa jenis toko yang memiliki jumlah jenis produk tidak lebih dari dua puluh *item* dan memiliki jumlah barang dengan bilangan bulat. Pengujian akan dilakukan dengan metode *Black Box Testing*. *Black Box Testing* fokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Cara pengujian dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit atau modul sesuai dengan proses bisnis yang diinginkan. Secara detil, hal-hal yang diujikan adalah sebagai berikut :

Tabel 3.16 Rencana Pengujian Sistem Informasi Penjualan Berbasis Web

Requirement yang diuji	Fungsi yang diuji
Registrasi Toko	Mendaftarkan toko
Login	Melakukan login
	Menjalankan fitur “Ingkat kode dan kata sandi”

Requirement yang diuji	Fungsi yang diuji
Data Profil Toko	Melakukan operasi RUD (<i>Read, Update, Delete</i>) pada data profil toko
Data Supplier	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data supplier
Data Kategori Produk	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data kategori produk
Data Produk	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data produk Menjalankan fitur “Menampilkan produk ke dalam web katalog”
Data Staff	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data staff Memberikan hak akses kepada staff
Data Penjualan	Mencatat transaksi penjualan Mencatat retur penjualan
Data Stok Produk	Mencatat transaksi penerimaan produk Mencatat transaksi retur pembelian
Laporan	Menampilkan laporan penjualan Menampilkan laporan pendapatan Menampilkan laporan stok Menampilkan laporan retur penjualan dan pembelian
Tampilan Responsif	Menampilkan sistem pada ukuran layar 360x640 Menampilkan sistem pada ukuran layar 1280x800 Menampilkan sistem pada ukuran layar 1366x768 Menampilkan sistem pada ukuran layar 1600x900 Menampilkan sistem pada ukuran layar 1920x1080 Menampilkan sistem pada ukuran layar 1920x1200
Implementasi Untuk Beberapa Jenis Toko	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data produk dengan berbagai jenis kategori Mencatat transaksi penjualan untuk berbagai jenis produk Mencatat transaksi penerimaan produk untuk berbagai jenis produk

A. Desain Uji Coba Registrasi Toko

Desain uji coba registrasi toko bertujuan untuk menguji apakah fungsi registrasi toko dan aktivasi toko dapat berjalan sesuai yang diharapkan. Desain uji coba registrasi toko dapat dilihat pada Tabel 3.17.

Tabel 3.17 Desain Uji Coba Registrasi Toko

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
1	Registrasi toko	Nama pelanggan, jenis kelamin, no telp, alamat, <i>email</i> , <i>password</i> dan nama domain toko	Registrasi Berhasil	

B. Desain Uji Coba Login

Desain uji coba *login* bertujuan untuk menguji apakah fungsi melakukan *login* dapat berjalan sesuai dengan yang diharapkan. Desain uji coba login dapat dilihat pada Tabel 3.18.

Tabel 3.18 Desain Uji Coba Login

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
2	Melakukan <i>login</i>	<i>Username</i> dan <i>password</i>	<i>Login</i> berhasil dan masuk ke halaman utama sistem	
3	Melakukan <i>login</i> dengan <i>format</i> salah	<i>Username</i> dan <i>password</i>	Muncul pesan “ <i>Username</i> atau <i>password</i> salah”	

C. Desain Uji Coba Data Profil Toko

Desain uji coba data profil toko bertujuan untuk menguji apakah fungsi RU (*Read*, *Update*) pada data profil toko dapat berjalan sesuai dengan yang

diharapkan. Desain uji coba data profil toko dapat dilihat pada Tabel 3.19.

Tabel 3.19 Desain Uji Coba Data Profil Toko

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
4	Menampilkan data profil toko	Nama toko, deskripsi toko, alamat toko, telepon, email, logo dan slogan	Data profil toko	
5	Mengubah data profil toko	Nama toko, deskripsi toko, alamat toko, telepon, email, logo dan slogan	Data profil toko berhasil diubah dan ditampilkan pada halaman profil toko	

D. Desain Uji Coba Data Supplier

Desain uji coba data *master supplier* dilakukan untuk menguji apakah fungsi CRUD (*Create, Read, Update, Delete*) pada data *master supplier* dapat berjalan sesuai dengan yang diharapkan. Desain uji coba *master supplier* dapat dilihat pada Tabel 3.20.

Tabel 3.20 Desain Uji Coba Data Supplier

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
6	Menampilkan data <i>supplier</i>	Memilih menu “Master Supplier”	Data <i>supplier</i>	
7	Memasukkan data <i>supplier</i>	Nama, alamat dan telepon	Data <i>supplier</i> berhasil disimpan dan ditampilkan pada tabel Data <i>supplier</i>	
8	Mengubah data <i>supplier</i>	Nama, alamat dan telepon	Data <i>supplier</i> berhasil diubah dan ditampilkan pada tabel <i>supplier</i>	

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
9	Menghapus data <i>supplier</i>	Nama, alamat dan telepon	Data <i>supplier</i> berhasil dihapus	

E. Desain Uji Coba Data Kategori Produk

Proses uji coba kategori produk dilakukan untuk menguji apakah fungsi CRUD (*Create, Read, Update, Delete*) pada data kategori produk dapat berjalan sesuai dengan yang diharapkan. Desain uji coba kategori produk dapat dilihat pada Tabel 3.21.

Tabel 3.21 Desain Uji Coba Data Kategori Produk

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
10	Menampilkan data kategori produk	Memilih menu “Master Kategori”	Data Kategori Produk	
11	Memasukkan data kategori produk	Nama kategori	Data kategori produk berhasil disimpan dan ditampilkan pada tabel data kategori produk	
12	Mengubah data kategori produk	Nama kategori	Data kategori produk berhasil disimpan dan ditampilkan pada tabel kategori produk	
13	Menghapus data kategori produk	Nama kategori	Data kategori produk berhasil dihapus	

F. Desain Uji Coba Data Produk

Proses uji coba master produk dilakukan untuk menguji apakah fungsi CRUD (*Create, Read, Update, Delete*) pada data produk dapat berjalan sesuai dengan yang diharapkan. Desain uji coba master produk dapat dilihat pada Tabel

3.22.

Tabel 3.22 Desain Uji Coba Data Produk

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
14	Menampilkan data produk	Memilih menu “Master Produk”	Data Produk	
15	Memasukkan data produk	Nama, kategori, <i>supplier</i> , minimal stok, harga beli, harga jual, deskripsi produk dan gambar produk	Data produk berhasil disimpan dan ditampilkan pada tabel data produk	
16	Mengubah data data produk	Nama, kategori, <i>supplier</i> , minimal stok, harga beli, harga jual, deskripsi produk dan gambar produk	Data produk berhasil disimpan dan ditampilkan pada tabel produk	
17	Menghapus data produk	Nama, kategori, <i>supplier</i> , minimal stok, harga beli, harga jual, deskripsi produk dan gambar produk	Data produk berhasil dihapus	

G. Desain Uji Coba Data Staff

Proses uji coba master staff dilakukan untuk menguji apakah fungsi CRUD (*Create, Read, Update, Delete*) pada master staff dapat berjalan sesuai dengan yang diharapkan. Desain uji coba master staff dapat dilihat pada Tabel

3.23.

Tabel 3.23 Desain Uji Coba Data Staff

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
18	Menampilkan data staff	Memilih menu "Master Staff"	Master Staff	
19	Memasukkan data staff	Nama staff, <i>email</i> dan password	Data staff berhasil disimpan dan ditampilkan pada tabel staff	
20	Mengubah data staff	Nama staff dan <i>email</i>	Data staff berhasil disimpan dan ditampilkan pada tabel staff	
21	Menghapus data staff	Nama staff dan <i>email</i>	Data staff berhasil dihapus	
22	Memberikan hak akses kepada staff	Jabatan staff	Staff berhasil diberikan hak akses	

H. Desain Uji Coba Data Penjualan

Desain uji coba data penjualan bertujuan untuk menguji apakah fungsi mencatat transaksi penjualan dan mencatat retur penjualan dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data penjualan dapat dilihat pada Tabel 3.24.

Tabel 3.24 Desain Uji Coba Data Penjualan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
23	Mencatat transaksi penjualan	Nama produk, jumlah, diskon satuan, pajak, diskon keseluruhan dan uang yang dibayarkan	Total harga, pesan transaksi penjualan berhasil disimpan dan nota penjualan	
24	Mencatat retur penjualan	<i>Id</i> nota penjualan,	Pesan retur penjualan	

		nama produk, jumlah retur, nama staff dan alasan	berhasil disimpan	
--	--	---	----------------------	--

I. Desain Uji Coba Data Stok Produk

Desain uji coba data stok produk bertujuan untuk menguji apakah fungsi mencatat transaksi penerimaan dan retur pembelian dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data stok produk dapat dilihat pada Tabel 3.25.

Tabel 3.25 Desain Uji Coba Data Stok Produk

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
25	Mencatat transaksi penerimaan produk	Id atau nama produk, jumlah dan nama <i>supplier</i>	Pesan transaksi penerimaan berhasil disimpan	
26	Mencatat retur pembelian	Id transaksi penerimaan, nama produk, jumlah retur, nama <i>supplier</i> dan alasan	Pesan retur pembelian berhasil disimpan	

J. Desain Uji Coba Laporan

Desain uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem, yaitu laporan penjualan, laporan pendapatan, laporan stok, laporan retur penjualan dan pembelian. Desain uji coba ini bertujuan apakah fungsi laporan yang dijalankan sesuai dengan yang diharapkan. Desain uji coba laporan dapat dilihat pada Tabel 3.26.

Tabel 3.26 Desain Uji Coba Laporan

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
27	Menampilkan menu-menu	Memilih menu "Laporan "	Tampilan laporan dalam	

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
	laporan		bentuk tombol untuk masuk ke dalam masing-masing laporan	
28	Menampilkan laporan penjualan	Tampilan laporan, nama produk dan periode laporan	Tampilan laporan penjualan	
29	Menampilkan laporan pendapatan	Tampilan laporan pendapatan	Tampilan laporan pendapatan dalam bentuk grafik	
30	Menampilkan laporan stok	Tampilan laporan, nama produk dan jumlah stok yang tersedia	Tampilan laporan stok	
31	Menampilkan laporan retur penjualan dan pembelian	Tampilan laporan, nama produk, nama <i>supplier</i> dan periode laporan	Tampilan laporan retur penjualan dan pembelian	

K. Desain Uji Coba Tampilan Responsif

Desain uji coba tampilan responsif dilakukan terhadap salah satu halaman yang terdapat pada sistem yaitu halaman *dashboard*. Desain uji coba ini bertujuan apakah tampilan sistem dapat menyesuaikan dengan berbagai macam ukuran layar dari *device* yang dimiliki oleh pengguna. Untuk menguji tampilan ini penulis menggunakan bantuan layanan dari <http://responsivetest.net/>. Desain uji coba tampilan responsif dapat dilihat pada Tabel 3.27.

Tabel 3.27 Desain Uji Coba Tampilan Responsif

<i>Test Case ID</i>	Tujuan	<i>Input</i>	Output yang diharapkan	Status
32	Menampilkan	Memilih menu	Tampilan sistem	

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
	sistem pada ukuran layar 360x640	"Dashboard"	dapat menyesuaikan sesuai dengan ukuran layar <i>device</i> pengguna	
33	Menampilkan sistem pada ukuran layar 1280x800	Memilih menu "Dashboard"	Tampilan sistem dapat menyesuaikan sesuai dengan ukuran layar <i>device</i> pengguna	
34	Menampilkan sistem pada ukuran layar 1366x768	Memilih menu "Dashboard"	Tampilan sistem dapat menyesuaikan sesuai dengan ukuran layar <i>device</i> pengguna	
35	Menampilkan sistem pada ukuran layar 1600x900	Memilih menu "Dashboard"	Tampilan sistem dapat menyesuaikan sesuai dengan ukuran layar <i>device</i> pengguna	
36	Menampilkan sistem pada ukuran layar 1920x1080	Memilih menu "Dashboard"	Tampilan sistem dapat menyesuaikan sesuai dengan ukuran layar <i>device</i> pengguna	
37	Menampilkan sistem pada ukuran layar 1920x1200	Memilih menu "Dashboard"	Tampilan sistem dapat menyesuaikan sesuai dengan ukuran layar <i>device</i> pengguna	

L. Desain Uji Coba Implementasi Untuk Beberapa Jenis Toko

Desain uji coba implementasi untuk beberapa jenis toko dilakukan untuk mengetahui apakah sistem yang telah dibuat mampu mengakomodir kebutuhan dari berbagai macam jenis UMKM. Jenis toko yang dipilih untuk menguji sistem ini adalah toko *fashion* yang menjual baju dan sepatu, toko kue almond crispy,

dan toko yang menjual aneka jenis sambal. Desain uji coba tampilan responsif dapat dilihat pada Tabel 3.28.

Tabel 3.28 Desain Uji Coba Implementasi Untuk Beberapa Jenis Toko

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
38	Melakukan operasi CRUD (<i>Create, Read, Update, Delete</i>) pada data produk dengan berbagai jenis kategori	Nama, kategori, <i>supplier</i> , minimal stok, harga beli, harga jual, deskripsi produk dan gambar produk	Data produk berhasil disimpan dan ditampilkan pada tabel data produk	
39	Mencatat transaksi penjualan untuk berbagai jenis produk	Nama produk, jumlah, diskon satuan, pajak, diskon keseluruhan dan uang yang dibayarkan	Total harga, pesan transaksi penjualan berhasil disimpan dan nota penjualan	
40	Mencatat transaksi penerimaan produk untuk berbagai jenis produk	Id atau nama produk, jumlah dan nama <i>supplier</i>	Pesan transaksi penerimaan berhasil disimpan	

M. Desain Evaluasi Kesesuaian Hasil Akhir Sistem

Evaluasi kesesuaian hasil akhir sistem adalah sebuah pengujian untuk mengetahui apakah sistem yang dibangun telah memenuhi kebutuhan-kebutuhan yang terdapat pada latar belakang permasalahan. Sistem yang dibangun juga harus mengatasi permasalahan-pemersalahan sistem penjualan yang telah digunakan sebelumnya. Desain uji coba kesesuaian hasil akhir sistem dapat dilihat pada Tabel 3.29.

Tabel 3.29 Desain Evaluasi Hasil Akhir Sistem

	Kebutuhan	Hasil	Bukti
1	Sistem dapat mencatat transaksi penjualan, mencatat retur penjualan dan pembelian		
2	Sistem dapat menampilkan informasi tentang omset penjualan dan pendapatan, stok terkini dan produk paling laris tanpa perlu melakukan rekap		
3	Sistem dapat menampilkan produk ke dalam bentuk web katalog		
4	Sistem dapat menangani berbagai macam jenis toko		
5	Biaya <i>maintenance</i> , perangkat teknologi serta SDM pengelola sistem yang murah		
6	Sistem dapat diakses kapanpun dan dimanapun dengan koneksi internet melalui <i>web browser</i> , termasuk <i>mobile browser</i>		