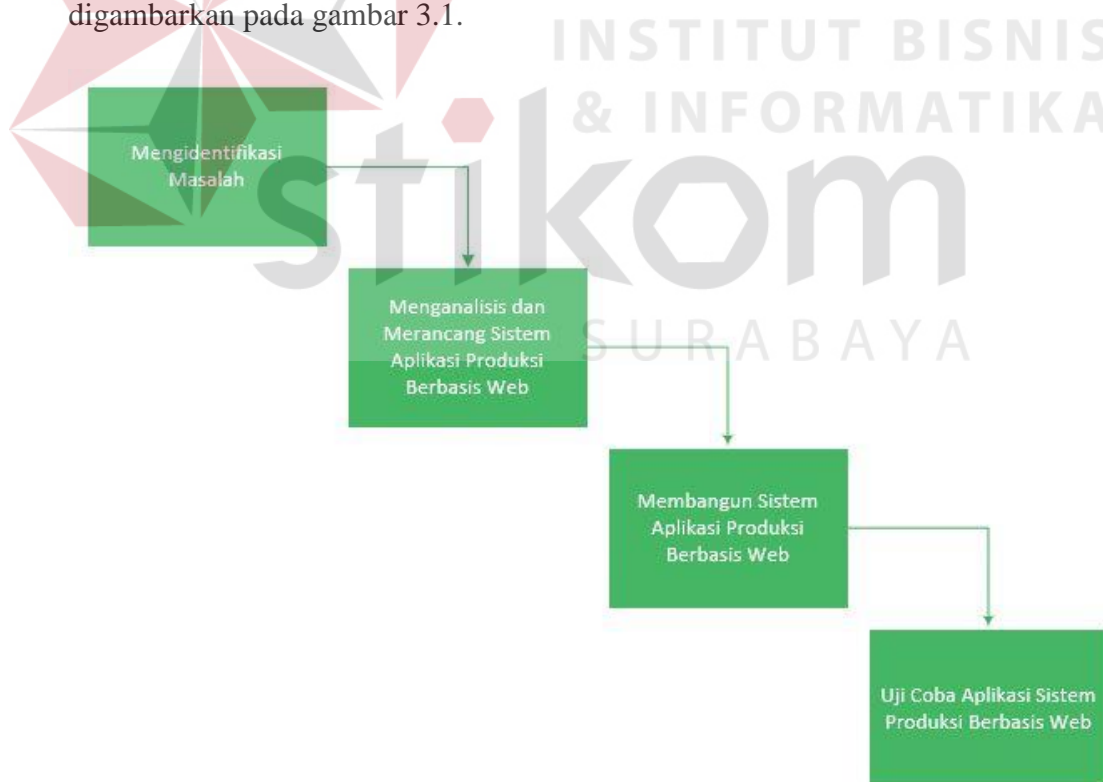


## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan dibahas tentang identifikasi masalah, analisis dan perancangan sistem, rancangan pengujian, dan evaluasi sistem dalam rancang bangun aplikasi perencanaan produksi kebutuhan bahan baku dengan metode *Material Requirement Planning* (MRP) berbasis web pada CV. Mitra Techno Sains. Model pengembangan yang digunakan pada penelitian ini menggunakan teori Pressman (2015) yaitu dengan model *waterfall* dimana melalui tahapan-tahapan yang telah ditentukan, yaitu mengidentifikasi masalah, menganalisis dan merancang sistem, membangun sistem dan uji coba sistem. Secara garis besar digambarkan pada gambar 3.1.



Gambar 3.1 Langkah-Langkah Analisis Sistem

### 3.1 Identifikasi Permasalahan

Untuk mengetahui dan mengidentifikasi masalah yang terjadi pada CV. Mitra Techno Sains, telah dilakukan wawancara dan observasi untuk pengumpulan data. Pengumpulan data ini bertujuan untuk mengidentifikasi permasalahan tentang perencanaan produksi dan kebutuhan bahan baku.

Proses bisnis CV. Mitra Techno Sains dimulai dari pesanan oleh pelanggan, kemudian bagian produksi akan melakukan produksi berdasarkan pesanan dari pelanggan. Pada saat permintaan pesanan sedang meningkat banyak pelanggan yang kecewa karena bagian produksi belum mampu memenuhi pesanan pelanggan dengan tepat waktu atau pesanan dikirim tidak sesuai dengan tanggal yang telah disetujui bersama pelanggan. Hal ini disebabkan karena CV. Mitra Techno Sains belum memiliki perencanaan kebutuhan bahan baku yang baik.

#### 3.1.1 Analisis Permasalahan

Masalah yang sering terjadi di CV. Mitra Techno Sains selama ini adalah banyak pelanggan yang merasa kecewa karena pesanan sering dikirim terlambat. Hal ini karena belum adanya penjadwalan yang baik pada proses perencanaan kebutuhan bahan baku yang digunakan untuk kegiatan produksi.

Dalam menentukan dan merencanakan kebutuhan bahan baku dibutuhkan metode *Material Requirement Planning* (MRP) dengan data nota pesanan yang digunakan sebagai data masukan utama, *Bill of Material* (BoM), *Master Production Schedule* (MPS), dan Status Inventori (SI). Data-data ini kemudian dihitung menggunakan teknik *Lot Sizing* metode *Lot-For-Lot* yang outputnya berupa *MRP Primary Report* dan *MRP Pegging Report*.

### 3.1.2 Analisis Pengguna

Adapun peran serta tanggung jawab dari aktor pada sistem yang ada saat ini di CV. Mitra Techno Sains seperti tabel 3.1.

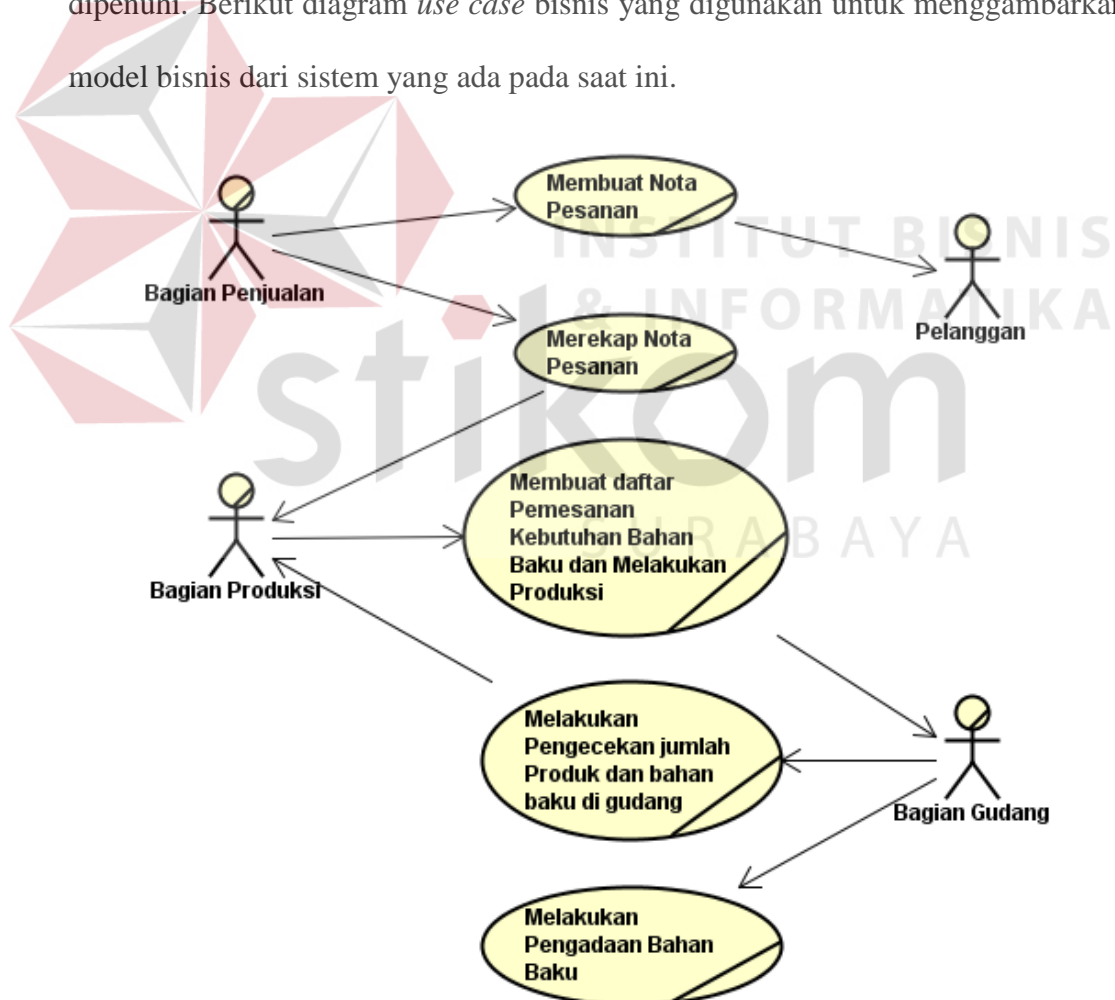
Table 3.1 Analisis Pengguna

No.	Aktor	Peran	Tanggung jawab
1	Bagian Penjualan	Menerima pesanan barang dari pelanggan	Bagian penjualan bertanggung jawab melayani pelanggan dan melakukan pencatatan nota pesanan yang telah disetujui koordinator bagian penjualan
		Merekapitulasi nota pesanan setiap akhir minggu	Bagian penjualan mengumpulkan salinan nota pesanan dan membuat rekapitulasi nota pesanan tiap minggu untuk diberikan kepada bagian produksi
2.	Bagian Produksi	Membuat daftar pemesanan kebutuhan bahan baku dan melakukan kegiatan produksi	Bagian produksi membuat daftar pemesanan bahan baku kepada <i>supplier</i> dan melakukan kegiatan produksi berdasarkan data rekapitulasi nota pesanan dari bagian penjualan
3.	Bagian Gudang	Pengecekan jumlah produk dan bahan baku di gudang	Bagian gudang melakukan pengecekan jumlah produk dan bahan baku di gudang dan memberikan informasi kepada bagian produksi
		Pengadaan bahan baku	Bagian gudang bertanggung jawab atas persediaan bahan baku di gudang dengan melakukan pemesanan bahan

No.	Aktor	Peran	Tanggung jawab
			baku kepada supplier berdasarkan daftar pemesanan kebutuhan bahan baku dari bagian produksi

### 3.2 Analisis Kebutuhan Sistem

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



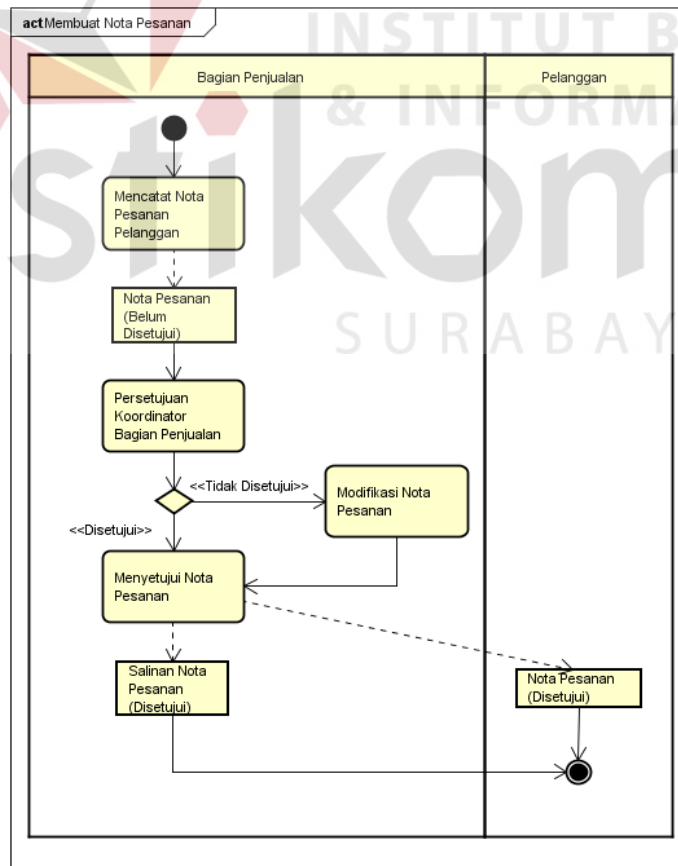
Gambar 3.2 Diagram *Use Case* Bisnis

Diagram *Activity* dibawah akan menjelaskan secara detail masing-masing proses bisnis pada diagram *use case* bisnis diatas.

### 3.2.1 Diagram *Activity*

#### a. Diagram *Activity* Membuat Nota Pesanan

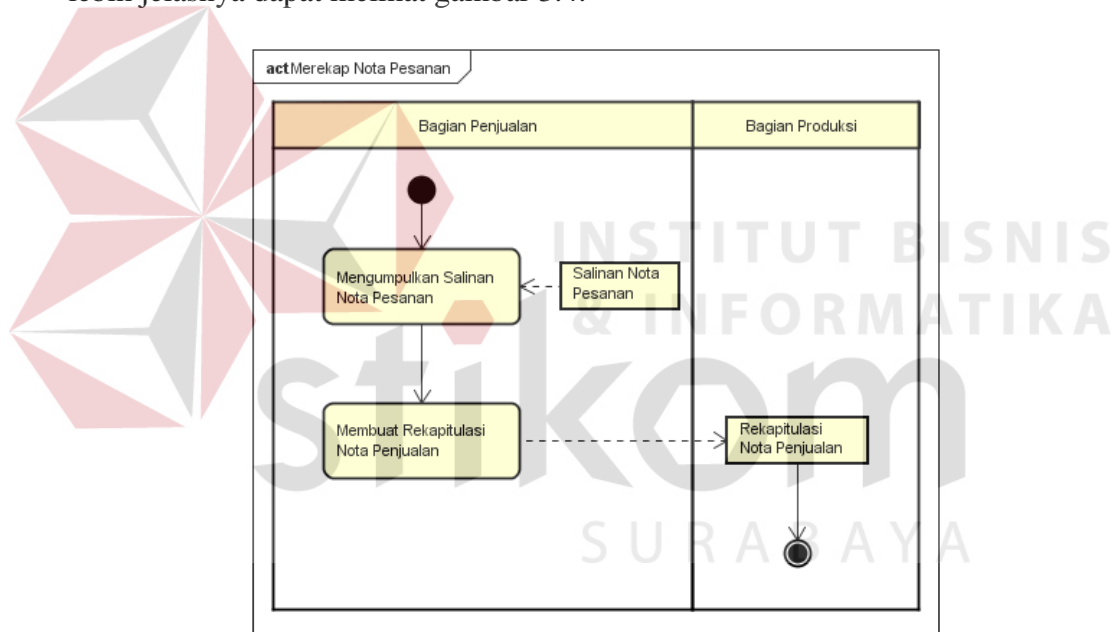
Proses membuat nota pesanan dilakukan oleh bagian penjualan atas permintaan pelanggan. Nota pesanan diisi dengan menentukan produk yang dipesan beserta dengan jumlahnya. Apabila jumlah pesanan dapat dipenuhi maka koordinator bagian penjualan akan menyetujui nota pesanan, jika tidak jumlah pesanan akan dimodifikasi. Kemudian nota pesanan tersebut akan diberikan kepada pelanggan dan salinan nota pesanan berwarna merah akan disimpan. Untuk lebih jelasnya dapat melihat gambar 3.3.



Gambar 3.3 Diagram *Activity* Membuat Nota Pesanan

**b. Diagram Activity Merekap Nota Pesanan**

Proses merekap nota pesanan dilakukan oleh bagian penjualan. Proses ini dilakukan pada setiap hari jumat yang bertujuan untuk merekapitulasi pesanan dalam satu minggu. Proses rekapitulasi nota pesanan sangat dibutuhkan oleh bagian produksi untuk memulai kegiatan produksi dan merencanakan pemesanan kebutuhan bahan baku. Proses merekap nota pesanan dimulai dari mengumpulkan salinan nota pesanan berwarna merah dalam satu minggu. Kemudian membuat laporan rekapitulasi nota pesanan dan menyerahkan kepada bagian produksi. Untuk lebih jelasnya dapat melihat gambar 3.4.

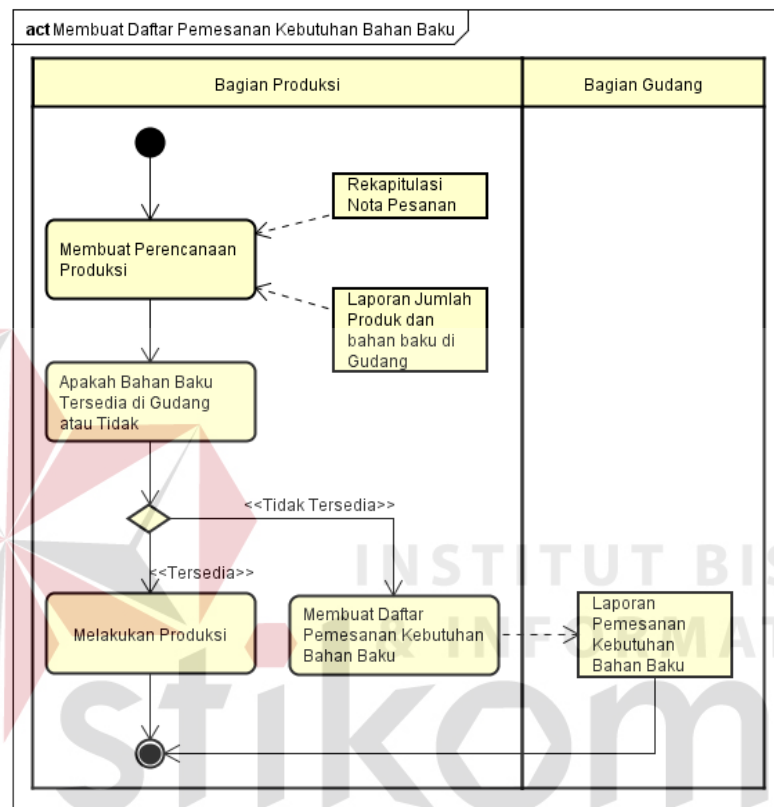


Gambar 3.4 Diagram Activity Merekap Nota Pesanan

**c. Diagram Activity Membuat Daftar Pemesanan Kebutuhan Bahan Baku dan Melakukan Produksi**

Proses membuat daftar pemesanan kebutuhan bahan baku dan melakukan kegiatan produksi dilakukan oleh bagian produksi. Proses ini dilakukan berdasarkan rekapitulasi nota pesanan dan laporan jumlah produk dan bahan baku

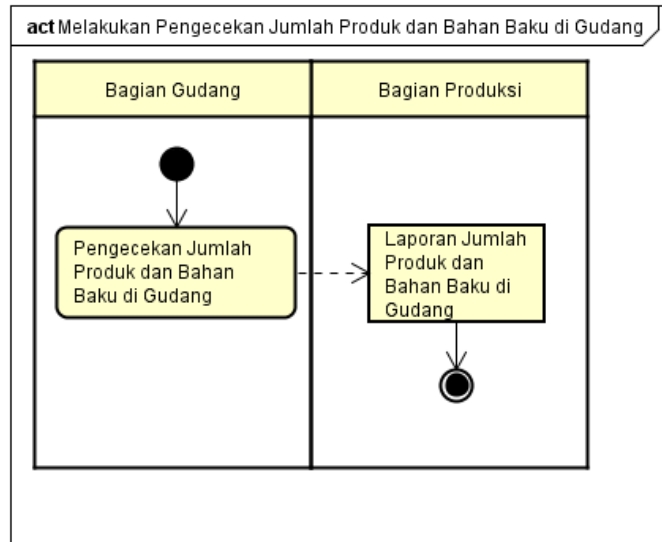
di gudang. Apabila bahan baku tersedia di gudang maka bagian produksi akan melakukan produksi, jika bahan baku tidak tersedia maka bagian produksi akan membuat daftar pemesanan kebutuhan bahan baku dan diberikan ke bagian gudang. Untuk lebih jelasnya dapat melihat gambar 3.5.



Gambar 3.5 Diagram *Activity* Membuat Pemesanan Kebutuhan Bahan Baku

**d. Activity Diagram Melakukan Pengecekan Jumlah Produk dan Bahan Baku di Gudang**

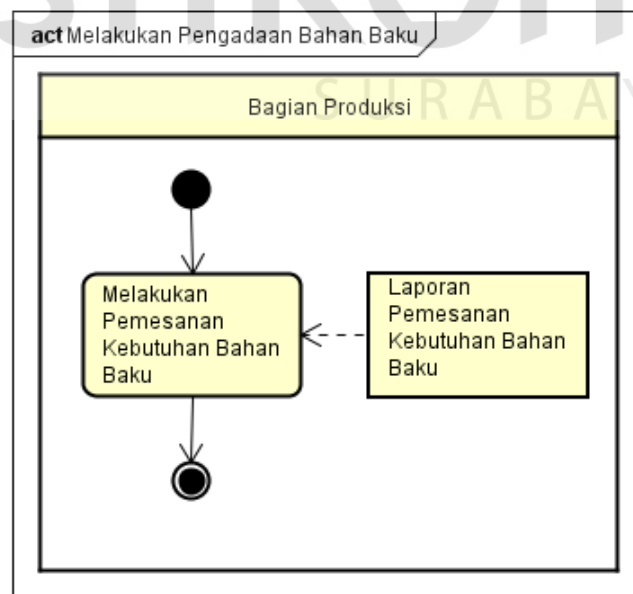
Proses melakukan pengecekan jumlah produk dan bahan baku di gudang dilakukan oleh bagian gudang. Proses ini dilakukan dengan mencatat dan membuat laporan status inventori di gudang yang meliputi jumlah produk dan bahan baku. Untuk lebih jelasnya dapat melihat gambar 3.6.



Gambar 3.6 Diagram *Activity* Membuat Laporan Produk

e. **Diagram *Activity* Melakukan Pengadaan Bahan Baku**

Proses melakukan pengadaan bahan baku dilakukan oleh bagian gudang. Proses ini berfungsi untuk melakukan pemesanan bahan baku kepada *supplier* berdasarkan laporan pemesanan bahan baku yang dilakukan oleh bagian produksi. Untuk lebih jelasnya dapat melihat gambar 3.7.



Gambar 3.7 Diagram *Activity* Melakukan Pengadaan Bahan Baku



### 3.2.2 Hasil Analisis

Dari proses analisis, didapatkan kelemahan-kelemahan sistem yang ada saat ini pada CV. Mitra Techno Sains. Kelemahan-kelemahan sistem saat ini adalah sebagai berikut:

1. Pencatatan pesanan yang masih manual dengan menggunakan nota pesanan. Penggunaan nota pesanan sebagai media pencatatan ini sangat rawan untuk terjadi kehilangan atau kerusakan akibat keteledoran pemilik. Selain itu bagian penjualan sering melakukan kesalahan perhitungan jumlah produk karena prosesnya masih mengandalkan bantuan nota dan kalkulator.
2. Dalam daftar pemesanan kebutuhan bahan baku hanya berisi jumlah bahan baku yang dipesan berdasarkan kebiasaan koordiantor bagian produksi, hal ini dapat menyebabkan proses produksi menjadi terlambat apabila jumlah bahan baku yang dipesan kurang dari yang dibutuhkan.

Untuk mengatasi kelemahan-kelemahan yang ada dalam sistem saat ini, perlu dibuat sebuah rancang bangun aplikasi perencanaan produksi kebutuhan bahan baku yang dapat memenuhi kebutuhan perencanaan kebutuhan bahan baku. Adapun kebutuhan aplikasi sistem produksi yang akan dibangun adalah sebagai berikut:

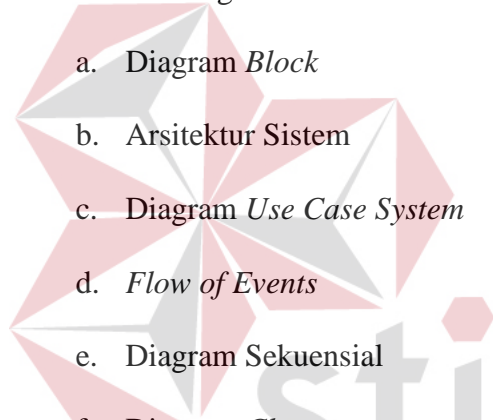
1. Aplikasi sistem produksi yang dibangun akan dapat membantu proses pencatatan nota pesanan dan meminimalkan kesalahan perhitungan jumlah produk yang dipesan.
2. Aplikasi sistem produksi yang dibangun akan dapat membantu dalam proses perencanaan produksi dan bahan baku dengan menggunakan metode *Material Requirement Planning* (MRP).

### 3.3 Perancangan Sistem

Perancangan sistem ini bertujuan untuk mendefinisikan kebutuhan-kebutuhan fungsional yang 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.

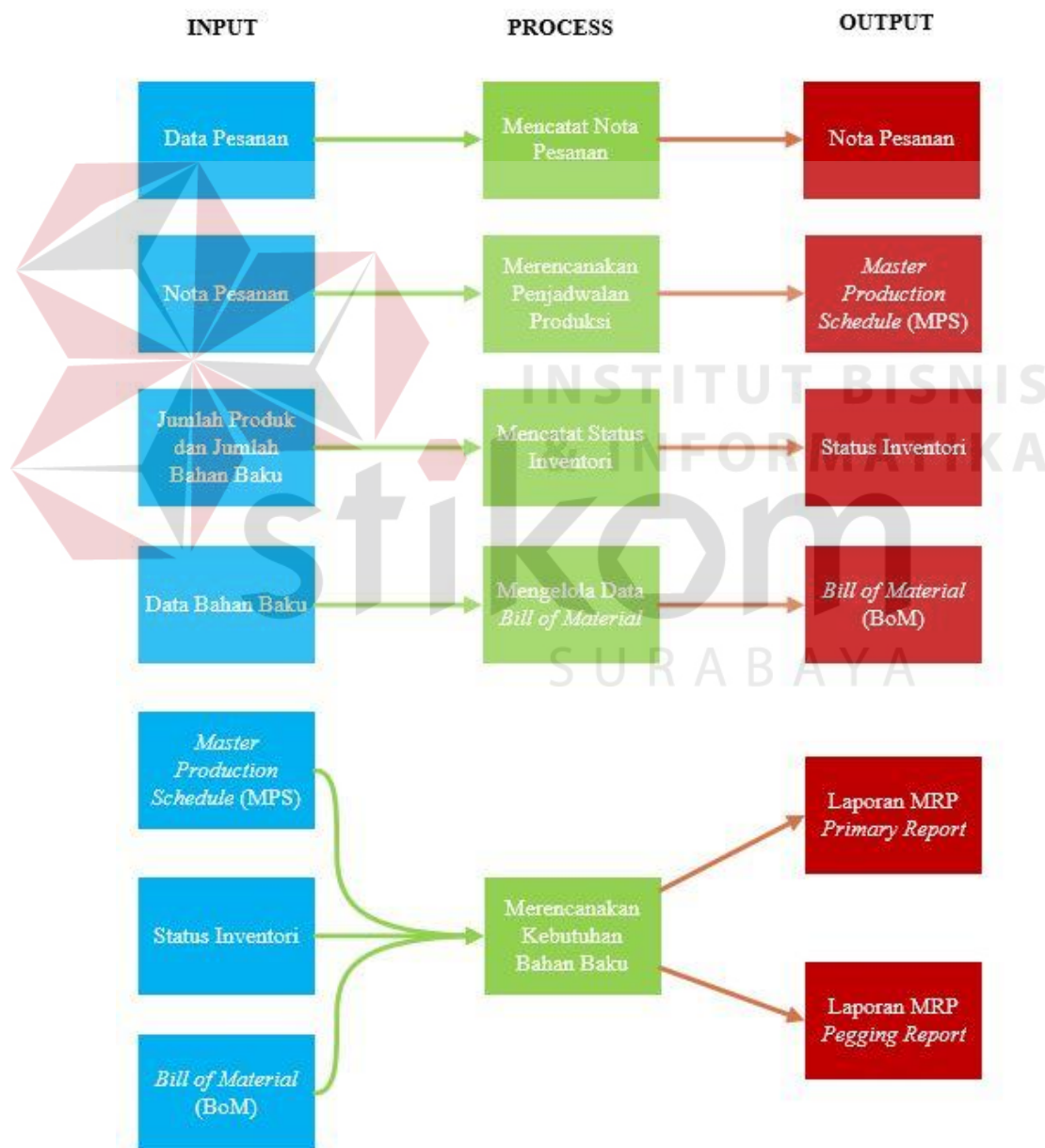
Langkah-langkah operasi dalam perancangan dan pengembangan sistem ini adalah sebagai berikut:

- a. Diagram *Block*
- b. Arsitektur Sistem
- c. Diagram *Use Case System*
- d. *Flow of Events*
- e. Diagram Sekuensial
- f. Diagram *Class*
- g. Diagram *Component*
- h. Diagram *Deployment*
- i. *Physical Data Model* (PDM)
- j. *Sitemap*
- k. Desain *User Interface*
- l. Desain *Input/Output*
- m. Rancangan Pengujian dan Evaluasi Sistem



### 3.3.1 Diagram Block

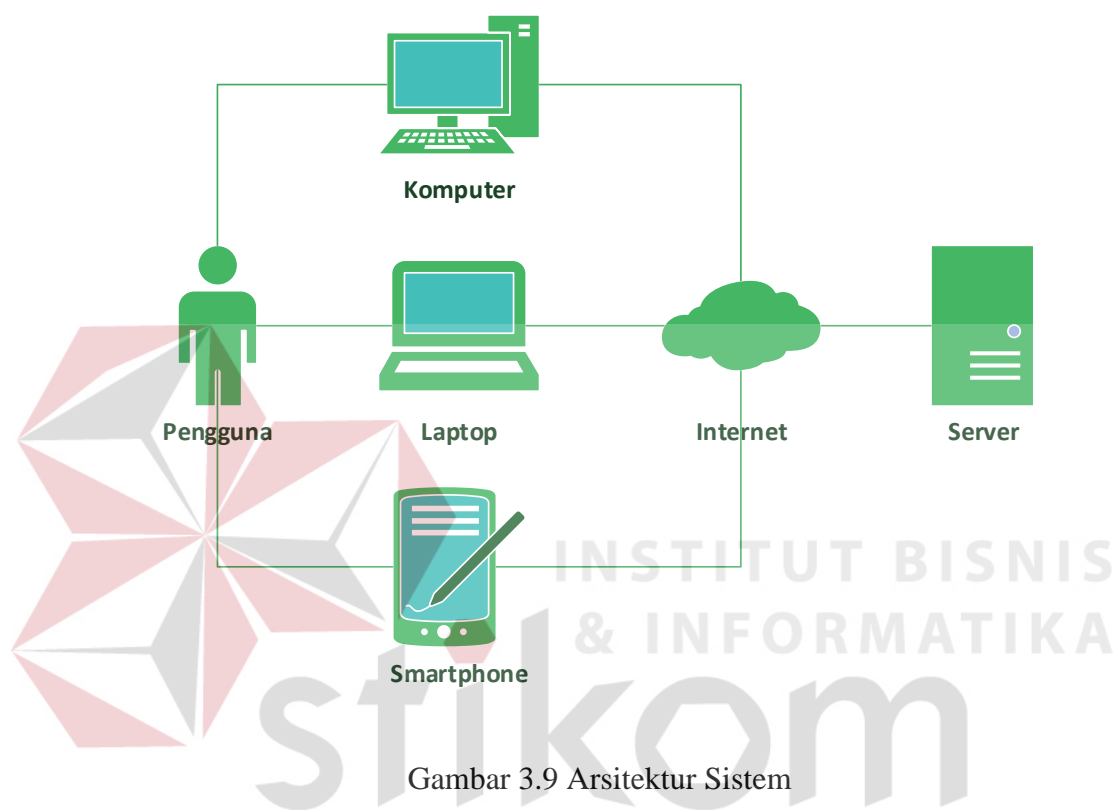
Aplikasi sistem produksi yang dibangun mampu untuk mencatat nota pesanan, merencanakan penjadwalan produksi, mencatat status inventori, mengelola data *Bill of Material*, dan merencanakan kebutuhan bahan baku. Berikut merupakan bentuk diagram *block* aplikasi sistem produksi yang akan dibuat sebagai berikut:



Gambar 3.8 Diagram Block

### 3.3.2 Arsitektur Sistem

Aplikasi sistem produksi yang dibangun merupakan berbasis web sehingga sistem dan data dapat disimpan di data *center* milik penyedia layanan. Desain arsitektur sistem yang akan dibangun adalah sebagai berikut:



Gambar 3.9 Arsitektur Sistem

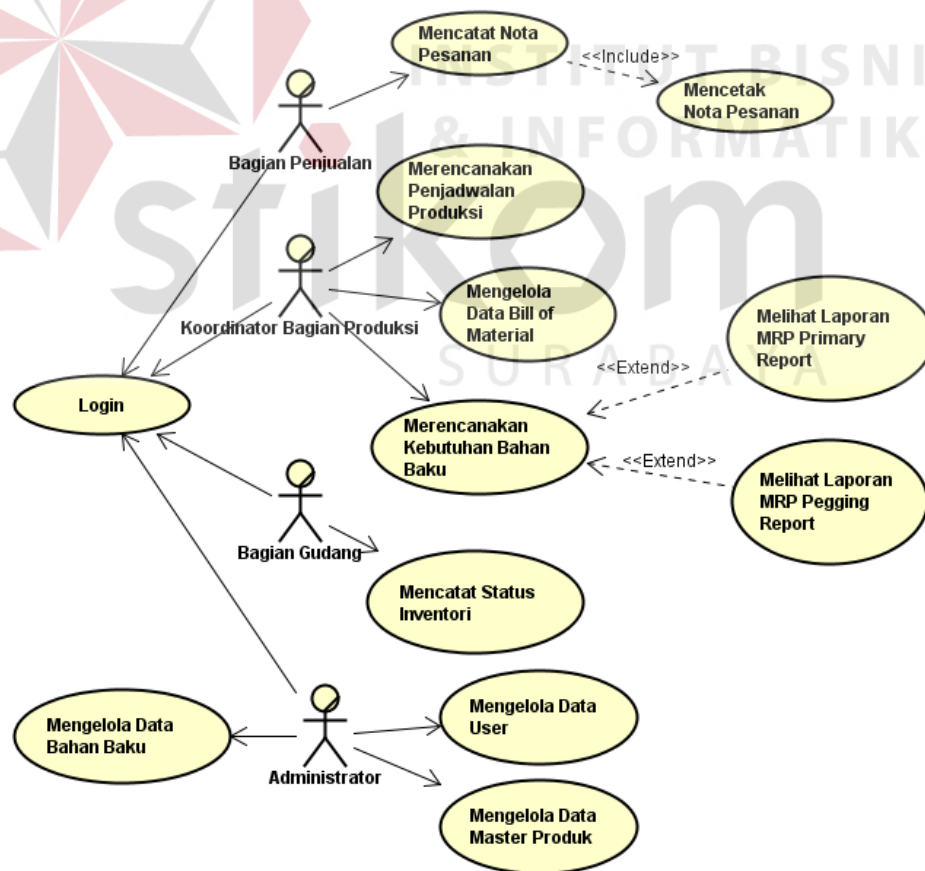
Dengan arsitektur sistem berbasis web pengguna tidak perlu melakukan disibukan melakukan *maintanance* dan perawatan infrastruktur sistem karena hal tersebut merupakan tanggung jawab dari penyedia layanan. Pengguna cukup mendaftarkan diri melalui halaman web penyedia layanan maka aplikasi sistem dapat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

Aktor utama dari sistem didefinisikan sesuai fungsi dan tanggung jawab dapat dimodelkan seperti tabel 3.2.

Tabel 3.2 Aktor Utama

Aktor	Fungsi
<b>Administrator</b>	Mengelola data <i>master</i> produk
	Mengelola data <i>user</i>
	Mengelola data bahan baku
<b>Bagian Penjualan</b>	Mencatat Nota Pesanan
<b>Koordinator Bagian Produksi</b>	Mengelola Data <i>Bill of Material</i> (BoM)
	Merencanakan Penjadwalan Produksi
	Merencanakan Kebutuhan Bahan Baku
<b>Bagian Gudang</b>	Mencatat Status Inventori Produk dan Bahan Baku

### 3.3.3 Diagram Use Case System



Gambar 3.10 Diagram Use Case Sistem

Aplikasi sistem produksi yang dibangun memiliki empat aktor, dimana aktor-aktor dalam *use case* tersebut merupakan semua yang ada diluar lingkup sistem perangkat lunak dan berinteraksi dengan sistem perangkat lunak tersebut. Aktor-aktor tersebut adalah aktor bagian penjualan, aktor koordinator bagian produksi, aktor bagian gudang, dan aktor administrator.

### 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* ini bertujuan untuk mendokumentasikan alur logika dalam *use case* dan menjelaskan secara rinci apa saja yang pemakai akan lakukan dan apa yang sistem itu sendiri lakukan.

#### a. *Flow of Events Login*

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

Tabel 3.3 *Flow of Events Login*

<b>Nama Use Case</b>	<b><i>Login</i></b>
<b>Deskripsi Singkat</b>	<i>Use case login</i> memungkinkan aktor dapat masuk kedalam sistem
<b>Prasyarat</b>	-
<b>Kondisi akhir sukses</b>	Aktor berhasil <i>login</i> dan dapat mengakses halaman utama
<b>Kondisi akhir gagal</b>	Aktor mendapat pesan gagal
<b>Aktor utama</b>	Aktor
<b>Aktor sekunder</b>	Tidak ada

<b>Nama Use Case</b>	<b>Login</b>	
<b>Pemicu</b>	Aktor membuka halaman <i>login</i> web aplikasi sistem produksi	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Aktor membuka halaman login melalui web <i>browser</i>
	2	Aktor mengisi <i>username</i> dan <i>password</i>
	3	Sistem melakukan koneksi ke <i>database</i> untuk melakukan verifikasi <i>username</i> dan <i>password</i> . Jika kombinasi <i>username</i> dan <i>password</i> salah, masuk ke langkah <b>3.1</b> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>3.2</b> .
	4	Sistem menampilkan dashboard sesuai dengan hak akses aktor
	5	<i>Use case</i> berakhir
<b>Alternate Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	3.1	Sistem menampilkan pesan <i>username</i> dan <i>password</i> salah dan meminta aktor mengulangi langkah <b>3</b> .
<b>Error Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	3.2	Sistem menampilkan pesan koneksi <i>database</i> untuk melakukan verifikasi <i>username</i> dan <i>password</i> gagal. Aktor kembali ke langkah <b>3</b> .

**b. Flow of Events Mencatat Nota Pesanan**

Proses mencatat nota pesanan dimulai dari bagian penjualan memasukan nomor nota, data pelanggan, memilih produk, dan jumlah yang dipesan beserta tanggal pengiriman kemudian sistem akan menyimpan dan mencetak nota pesanan

untuk diserahkan kepada pemesan. *Flow of events* mencatat nota pesanan dapat dilihat pada tabel 3.4.

Tabel 3.4 *Flow of Events* Mencatat Nota Pesanan

Nama Use Case	Mencatat Nota Pesanan	
<b>Deskripsi Singkat</b>	<i>Use case</i> mencatat nota pesanan memungkinkan bagian penjualan melakukan pencatatan nota pesanan	
<b>Prasyarat</b>	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai bagian penjualan	
<b>Kondisi akhir sukses</b>	Menyimpan dan mencetak nota pesanan berhasil dilakukan	
<b>Kondisi akhir gagal</b>	1. Jika ada <i>format</i> data yang salah, maka sistem akan memberitahu bahwa data gagal disimpan 2. Jika nota pesanan tidak dapat dicetak, maka sistem akan memberitahu bahwa nota pesanan tidak dapat dicetak	
<b>Aktor utama</b>	Bagian Penjualan	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor membuka halaman Mencatat Nota Pesanan	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Aktor membuka halaman mencatat nota pesanan
	2	Sistem mengarahkan aktor ke halaman mencatat nota pesanan
	3	Aktor memasukkan nomor nota, data pelanggan, produk yang dibeli, jumlah produk, tanggal pesanan, dan tanggal pengiriman.
	4	Sistem memeriksa <i>format</i> masukan dari pengguna. Jika ada <i>format</i> yang salah, masuk ke langkah <b>4.1</b>



Nama <i>Use Case</i>	Mencatat Nota Pesanan	
	5	Sistem melakukan koneksi <i>database</i> untuk menyimpan data nota pesanan. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>5.1</b>
	6	Sistem mengarahkan aktor ke tampilan cetak nota pesanan
	7	<i>Use case</i> berakhir
<i>Alternate Flow</i>	Langkah	Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data nota pesanan salah dan meminta pengguna mengulangi langkah ke <b>3</b>
<i>Error Flow</i>	Langkah	Aksi Percabangan
	5.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data nota pesanan gagal. Pengguna kembali ke langkah <b>2</b>

**c. *Flow of Events* Merencanakan Penjadwalan Produksi**

Proses merencanakan penjadwalan produksi dimulai dari koordinator bagian produksi memilih nama produk dan *range* tanggal pesanan untuk dijadwalkan, kemudian sistem akan menyimpan kedalam *database*. Pada proses ini terdapat satu masukan yaitu data nota pesanan dan dihitung berdasarkan stok inventori produk. *Flow of events* merencanakan penjadwalan produksi dapat dilihat pada tabel 3.5.

Tabel 3.5 *Flow of Events* Merencanakan Penjadwalan Produksi

Nama <i>Use Case</i>		Merencanakan Penjadwalan Produksi
<b>Deskripsi Singkat</b>	<i>Use case</i> merencanakan penjadwalan produksi memungkinkan koordinator bagian produksi menjadwalkan semua pesanan	
<b>Prasyarat</b>	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai koordinator bagian produksi	
<b>Kondisi akhir sukses</b>	Penjadwalan produksi berhasil disimpan	
<b>Kondisi akhir gagal</b>	Jika koneksi database tidak dapat menyimpan data penjadwalan produksi atau <i>Master Production Schedule</i> (MPS)	
<b>Aktor utama</b>	Koordinator bagian produksi	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor membuka halaman <i>Master Production Schedule</i>	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Aktor membuka halaman <i>Master Production Schedule</i>
	2	Sistem mengarahkan aktor ke halaman <i>Master Production Schedule</i>
	3	Aktor menentukan nama produk dan <i>range</i> tanggal pesanan
	4	Sistem melakukan koneksi <i>database</i> untuk mengambil data nota pesanan dan menjadwalkan berdasarkan stok inventori produk lalu menyimpan dalam <i>database</i> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah <b>4.1</b>
	5	<i>Use case</i> berakhir
<b>Alternate Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	-	
<b>Error Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>

Nama Use Case	Merencanakan Penjadwalan Produksi	
	4.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data MPS gagal. Aktor kembali ke langkah 3

d. *Flow of Events* Mengelola Data *Bill of Material* (BoM)

Mengelola data *Bill of Material* berfungsi untuk membuat susunan struktur produk mulai dari tingkatan paling atas (Produk) sampai tingkatan paling bawah (Anak Bahan Baku). Proses ini dimulai dari koordinator bagian produksi menentukan nama produk, nama bahan baku, anak bahan baku, kuantitas, tingkatan, dan *lead time*. Kemudian koordinator bagian produksi akan melakukan proses menyimpan atau memperbarui kedalam *database*. *Flow of events* mengelola data *bill of material* dapat dilihat pada tabel 3.6.

Tabel 3.6 *Flow of Events* Mengelola Data *Bill of Material*

Nama Use Case	Mengelola Data <i>Bill of Material</i> (BoM)	
<b>Deskripsi Singkat</b>	<i>Use case</i> mengelola BoM memungkinkan koordinator bagian produksi untuk mengkonversi data bahan baku kedalam bentuk struktur produk yang terdiri dari bahan baku dan anak bahan baku	
<b>Prasyarat</b>	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai koordinator bagian produksi	
<b>Kondisi akhir sukses</b>	Data <i>Bill of Material</i> berhasil disimpan	
<b>Kondisi akhir gagal</b>	Jika koneksi <i>database</i> tidak dapat menyimpan data <i>Bill of Material</i>	
<b>Aktor utama</b>	Koordinator bagian produksi	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor membuka halaman <i>Bill of Material</i> (BoM)	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Aktor membuka halaman <i>Bill of Material</i>

Nama <i>Use Case</i>	Mengelola Data <i>Bill of Material</i> (BoM)	
	2	Sistem mengarahkan aktor ke halaman <i>Bill of Material</i>
	3	Sistem melakukan koneksi <i>database</i> untuk mengambil data <i>Bill of Material</i> lalu menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>3.1</b>
	4	Aktor menentukan nama produk, nama bahan baku, anak bahan baku, kuantitas, tingkatan, dan <i>lead time</i> . Kemudian aktor melakukan operasi simpan atau perbarui data <i>Bill of Material</i>
	5	Sistem melakukan koneksi <i>database</i> untuk menyimpan semua perubahan yang dilakukan oleh pengguna terhadap data <i>Bill of Material</i> . Jika <i>format</i> data masukan salah, masuk ke langkah <b>5.1</b> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>5.2</b>
	6	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data <i>Bill of Material</i> berhasil
	7	<i>Use case</i> berakhir
	<b>Alternate Flow</b>	<b>Langkah</b>
	5.1	Sistem menampilkan pesan <i>format</i> data yang dimasukkan salah dan meminta aktor untuk kembali ke langkah <b>4</b>
<b>Error Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	3.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data <i>Bill of Material</i> gagal. Aktor kembali ke langkah <b>2</b>

Nama Use Case	Mengelola Data <i>Bill of Material</i> (BoM)	
	5.2	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data <i>Bill of Material</i> gagal. Aktor kembali ke langkah 4

e. *Flow of Events* Merencanakan Kebutuhan Bahan Baku

Proses merencanakan kebutuhan bahan baku berfungsi untuk merencanakan kebutuhan bahan baku. Pada Proses ini terdapat 3 masukan yaitu data penjadwalan produksi atau MPS, data status inventori, dan data BoM. Ketiga data tersebut dihitung dengan MRP memakai teknik *Lot Sizing* metode *Lot-For-Lot*. Proses ini dimulai dari koordinator bagian produksi menentukan nama produk dan *range* tanggal pesanan. *Flow of events* merencanakan kebutuhan bahan baku dapat dilihat pada tabel 3.7.

Tabel 3.7 *Flow of Events* Merencanakan Kebutuhan Bahan Baku

Nama Use Case	Merencanakan Kebutuhan Bahan Baku	
<b>Deskripsi Singkat</b>	<i>Use case</i> merencanakan kebutuhan bahan baku memungkinkan koordinator bagian produksi melakukan perencanaan kebutuhan bahan baku	
<b>Prasyarat</b>	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai koordinator bagian produksi	
<b>Kondisi akhir sukses</b>	Perencanaan kebutuhan bahan baku berhasil dilakukan	
<b>Kondisi akhir gagal</b>	Jika koneksi <i>database</i> tidak dapat menyimpan data MRP	
<b>Aktor utama</b>	Koordinator bagian produksi	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor membuka halaman <i>Material Requirement Planning</i>	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>

Nama <i>Use Case</i>	Merencanakan Kebutuhan Bahan Baku	
	1	Aktor membuka halaman <i>Material Requirement Planning</i>
	2	Sistem mengarahkan aktor ke halaman <i>Material Requirement Planning</i>
	3	Aktor menentukan nama produk dan <i>range</i> tanggal produksi
	4	Sistem akan melakukan perhitungan perencanaan dan koneksi ke <i>database</i> untuk menyimpan data MRP. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk langkah <b>4.1</b>
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data MRP berhasil
	6	<i>Use case</i> berhasil
<b>Alternate Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	-	
<b>Error Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	4.1	Sistem menampilkan pesan koneksi <i>database</i> untuk menyimpan data MRP gagal. Aktor kembali ke langkah <b>3</b>

**f. Flow of Events Mencatat Status Inventori**

Proses mencatat status inventori dimulai dari bagian gudang mengubah data status inventori produk atau bahan baku, kemudian sistem akan menyimpan kedalam *database*. *Flow of events* mencatat status inventori dapat dilihat pada tabel 3.8.

Tabel 3.8 *Flow of Events* Mencatat Status Inventori

Nama <i>Use Case</i>	Mencatat Status Inventori	
<b>Deskripsi Singkat</b>	<i>Use case</i> mencatat status inventori memungkinkan bagian gudang melakukan pencatatan status inventori produk atau bahan baku di gudang	
<b>Prasyarat</b>	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai bagian gudang	
<b>Kondisi akhir sukses</b>	Data status inventori berhasil diperbaharui	
<b>Kondisi akhir gagal</b>	Jika ada <i>format</i> data yang salah, maka sistem akan memberitahu bahwa data gagal disimpan	
<b>Aktor utama</b>	Bagian gudang	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor membuka halaman Status Inventori Produk atau Bahan Baku	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Aktor membuka halaman status inventori
	2	Sistem mengarahkan aktor ke halaman status inventori dan melakukan koneksi <i>database</i> untuk mengambil data status inventori dan menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>2.1</b>
	3	Aktor dapat melakukan operasi mencari, menambah, atau mengurangi data status inventori
4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan pengguna terhadap data. Jika <i>format</i> data masukan salah, masuk ke langkah <b>4.1</b> . Jika terjadi <i>error</i> saat	

Nama Use Case		Mencatat Status Inventori
		melakukan koneksi ke <i>database</i> , masuk ke langkah <b>4.2</b>
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data status inventori berhasil dilakukan
	6	<i>Use case</i> berakhir
Alternate Flow		Langkah Aksi Percabangan
	4.1	Sistem menampilkan pesan <i>format</i> data status inventori yang dimasukkan salah dan meminta pengguna untuk kembali ke langkah 3
Error Flow		Langkah Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data status inventori gagal. Aktor kembali ke langkah 1
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data status inventori gagal. Aktor kembali ke langkah 3

g. *Flow of Events* Mengelola Data Master Produk

Proses mengelola data *master* produk dimulai dari *administrator* memasukan data produk. Proses ini meliputi operasi mencari, menambah, mengubah, dan menghapus data produk. *Flow of events* mengelola data *master* produk dapat dilihat pada tabel 3.9.

Tabel 3.9 *Flow of Events* Mengelola Data Master Produk

Nama Use Case	Mengelola Data Master Produk
Deskripsi Singkat	<i>Use case</i> mengelola data <i>master</i> produk meliputi proses menambah, mengubah, menghapus, dan mencari data produk



<b>Nama Use Case</b>	<b>Mengelola Data Master Produk</b>	
<b>Prasyarat</b>	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai <i>administrator</i>	
<b>Kondisi akhir sukses</b>	Data produk berhasil diperbarui	
<b>Kondisi akhir gagal</b>	Jika ada <i>format</i> data yang salah, maka sistem akan memberitahu bahwa data gagal disimpan	
<b>Aktor utama</b>	<i>Administrator</i>	
<b>Aktor sekunder</b>	Tidak ada	
<b>Pemicu</b>	Aktor membuka halaman <i>Master Produk</i>	
<b>Primary Flow</b>	<b>Langkah</b>	<b>Aksi</b>
	1	Aktor membuka halaman <i>master</i> produk
	2	Sistem mengarahkan aktor ke halaman <i>master</i> produk dan melakukan koneksi <i>database</i> untuk mengambil data produk dan menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>2.1</b>
	3	Aktor melakukan operasi menambah, mengubah, atau menghapus data produk
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan aktor terhadap data produk. Jika <i>format</i> data masukan salah, masuk ke langkah <b>4.1</b> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>4.2</b>
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data produk berhasil dilakukan
	6	<i>Use case</i> berakhir
<b>Alternate Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>

Nama Use Case		Mengelola Data Master Produk
	4.1	Sistem menampilkan pesan <i>format</i> data produk yang dimasukkan salah dan meminta aktor untuk kembali ke langkah 3
Error Flow	Langkah	Aksi Percabangan
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data produk gagal. Aktor kembali ke langkah 1
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data produk gagal. Aktor kembali ke langkah 3

#### h. Flow of Events Mengelola Data User

Proses mengelola data *user* dilakukan oleh *administrator* memasukan atau mengubah data *user* kepada sistem. *Flow of events* mengelola data *user* dapat dilihat pada tabel 3.10.

Tabel 3.10 *Flow of Events* Mengelola Data User

Nama Use Case		Mengelola Data User
Deskripsi Singkat	<i>Use case</i> mengelola data <i>user</i> meliputi proses menambah, mengubah, dan mencari data <i>user</i>	
Prasyarat	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai <i>administrator</i>	
Kondisi akhir sukses	Data <i>user</i> berhasil diperbarui	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberitahu bahwa data gagal disimpan	
Aktor utama	<i>Administrator</i>	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman <i>Master User</i>	
Primary Flow	Langkah	Aksi
	1	Aktor membuka halaman <i>master user</i>

Nama <i>Use Case</i>	Mengelola <i>Data User</i>	
	2	Sistem mengarahkan aktor ke halaman <i>master user</i> , kemudian melakukan koneksi <i>database</i> untuk mengambil data <i>user</i> lalu menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>2.1</b>
	3	Aktor melakukan operasi menambah atau mengubah data <i>user</i>
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan aktor terhadap data <i>user</i> . Jika <i>format</i> data masukan salah, masuk ke langkah <b>4.1</b> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>4.2</b>
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data <i>user</i> berhasil dilakukan
	6	<i>Use case</i> berakhir
	<b>Alternate Flow</b>	<b>Langkah</b>
	4.1	Sistem menampilkan pesan <i>format</i> data user yang dimasukkan salah dan meminta aktor untuk kembali ke langkah <b>3</b>
<b>Error Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data <i>user</i> gagal. Aktor kembali ke langkah <b>1</b>
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data. Aktor kembali ke langkah <b>3</b>

i. *Flow of Events* Mengelola Data Bahan Baku

Proses mengelola data bahan baku dilakukan oleh *administrator* memasukan data bahan baku kedalam sistem. Proses ini meliputi menambah, mengubah, atau menghapus data bahan baku. *Flow of events* mengelola data bahan baku dapat dilihat pada tabel 3.11.

Tabel 3.11 *Flow of Events* Mengelola Data Bahan Baku

Nama Use Case	Mengelola Data Bahan Baku	
Deskripsi Singkat	Use case mengelola data bahan baku meliputi proses menambah, mengubah, menghapus, dan mencari data bahan baku	
Prasyarat	Aktor melakukan <i>login</i> dan memiliki hak akses sebagai <i>administrator</i>	
Kondisi akhir sukses	Data bahan baku berhasil diperbarui	
Kondisi akhir gagal	Jika ada <i>format</i> data yang salah, maka sistem akan memberitahu bahwa data gagal disimpan	
Aktor utama	<i>Administrator</i>	
Aktor sekunder	Tidak ada	
Pemicu	Aktor membuka halaman <i>Master</i> Bahan Baku	
Primary Flow	Langkah	Aksi
	1	Aktor membuka halaman <i>master</i> bahan baku
	2	Sistem mengarahkan aktor ke halaman <i>master</i> bahan baku, kemudian melakukan koneksi <i>database</i> untuk mengambil data bahan baku lalu menampilkannya. Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>2.1</b>
	3	Aktor melakukan operasi menambah, mengubah, atau menghapus data bahan baku

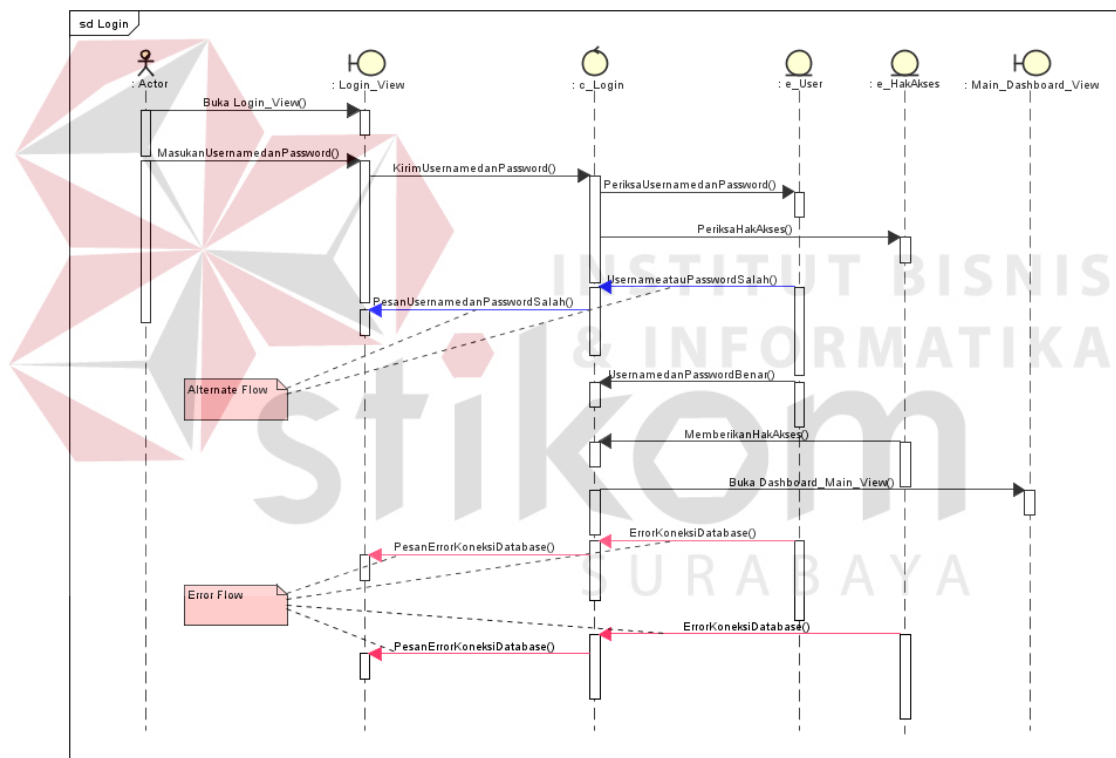
Nama <i>Use Case</i>	Mengelola Data Bahan Baku	
	4	Sistem melakukan koneksi ke <i>database</i> untuk menyimpan semua perubahan yang dilakukan aktor terhadap data bahan baku. Jika <i>format</i> data masukan salah, masuk ke langkah <b>4.1</b> . Jika terjadi <i>error</i> saat melakukan koneksi ke <i>database</i> , masuk ke langkah <b>4.2</b>
	5	Sistem menampilkan pesan bahwa operasi yang dilakukan terhadap data bahan baku berhasil dilakukan
	6	<i>Use case</i> berakhir
<b>Alternate Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	4.1	Sistem menampilkan pesan <i>format</i> data bahan baku yang dimasukkan salah dan meminta aktor untuk kembali ke langkah <b>3</b>
<b>Error Flow</b>	<b>Langkah</b>	<b>Aksi Percabangan</b>
	2.1	Sistem menampilkan pesan koneksi <i>database</i> untuk mengambil data bahan baku gagal. Aktor kembali ke langkah <b>1</b>
	4.2	Sistem menampilkan pesan koneksi <i>database</i> untuk mengubah data bahan baku. Aktor kembali ke langkah <b>3</b>

### 3.3.5 Diagram Sekuensial

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

### a. Diagram Sekuensial *Login*

Proses *login* dimulai dari aktor mengakses *boundary* Login\_View dan memasukkan *username* dan *password*. Kemudian sistem akan meminta *control* c\_Login untuk memverifikasi *username* dan *password* di *entity* e\_User dan e\_HakAkses. Jika *username* atau *password* salah sistem akan menampilkan pesan *username* dan *password* salah dan meminta aktor untuk memasukkan *username* dan *password* kembali. Apabila benar sistem akan mengarahkan aktor ke *boundary* Main\_Dashboard\_View. Diagram sekuensial *login* dapat dilihat pada gambar 3.11.

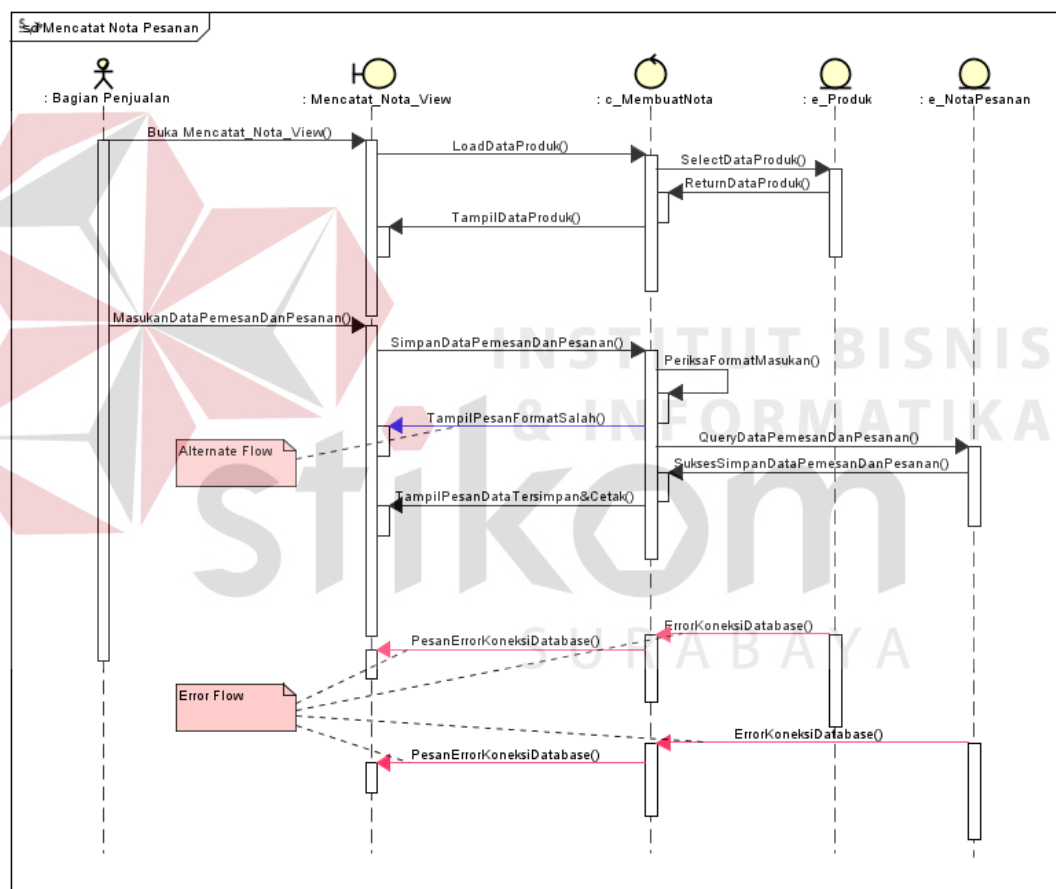


Gambar 3.11 Diagram Sekuensial *Login*

### b. Diagram Sekuensial Mencatat Nota Pesanan

Proses mencatat nota pesanan dimulai dari aktor mengakses *boundary* Mencatat\_Nota\_View. Kemudian *boundary* Mencatat\_Nota\_View meminta *control* c\_MembuatNota mengambil data produk dari *entity* e\_produk untuk

ditampilkan. Setelah itu aktor melakukan operasi memasukkan data pemesan dan pesanan pada *boundary* Mencatat\_Nota\_View. Setelah selesai, sistem akan memeriksa *format* data yang dimasukkan. Jika ada kesalahan, sistem akan menampilkan pesan *format* salah dan meminta aktor untuk mengubahnya. Apabila benar, sistem akan menampilkan pesan data tersimpan dan data nota pesanan disimpan oleh sistem di *entity* e\_NotaPesanan melalui *control* c\_MembuatNota. Diagram sekuensial mencatat nota pesanan dapat dilihat pada gambar 3.12.

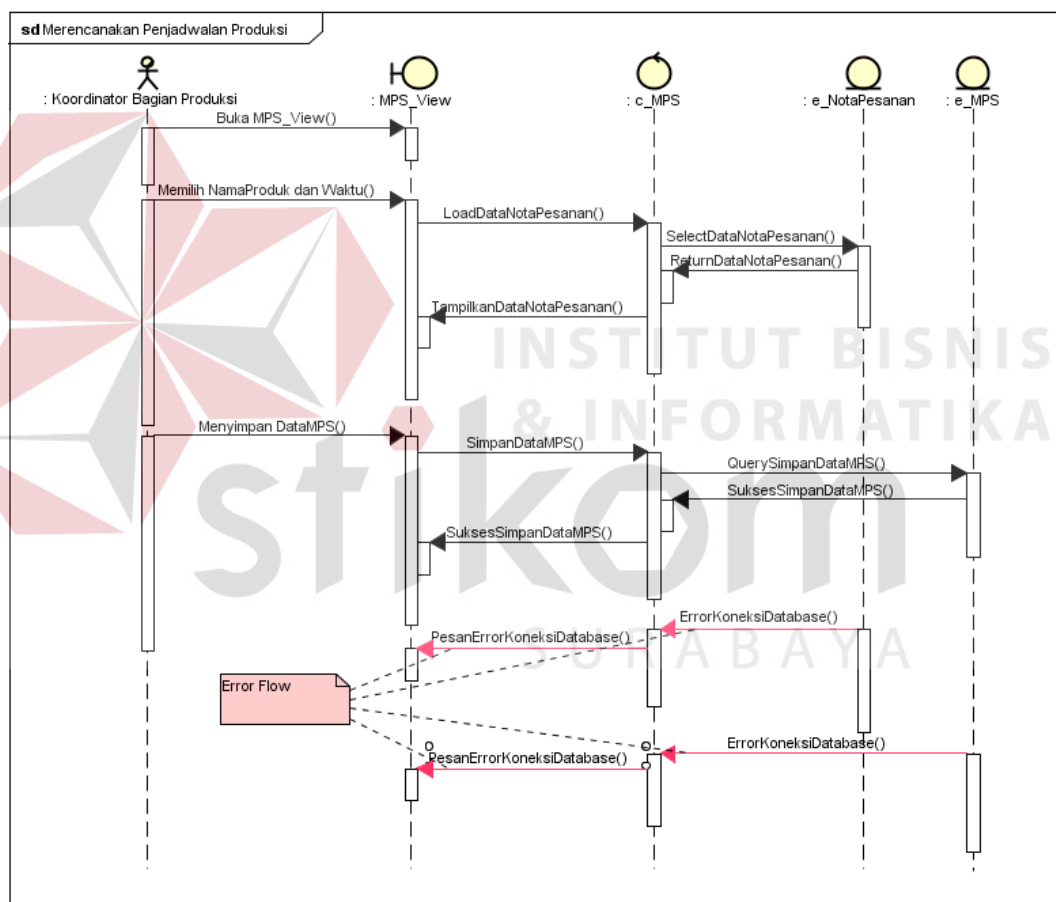


Gambar 3.12 Diagram Sekuensial Mencatat Nota Pesanan

### c. Diagram Sekuensial Merencanakan Penjadwalan Produksi

Proses merencanakan penjadwalan produksi dimulai dari aktor mengakses *boundary* MPS\_View dan memilih nama produk dan waktu penjadwalan produksi.

Kemudian *boundary* MPS\_View meminta *control* c\_MPS mengambil data nota pesanan dari *entity* e\_NotaPesanan untuk ditampilkan. Diproses ini sistem akan merekapitulasi nota pesanan satu hari yang akan dijadwalkan produksinya. Setelah itu aktor akan meminta *boundary* MPS\_View untuk menyimpan kedalam *entity* e\_MPS melalui *control* c\_MPS. Diagram sekuensial merencanakan penjadwalan produksi dapat dilihat pada gambar 3.13.



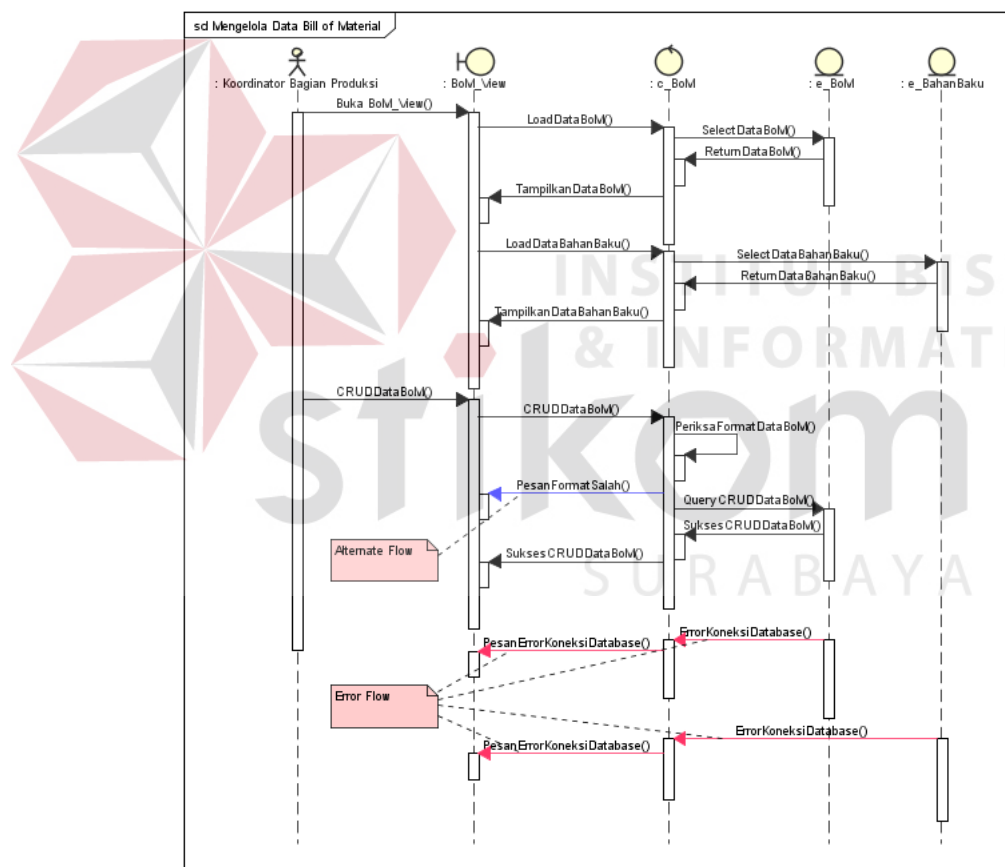
Gambar 3.13 Diagram Sekuensial Merencanakan Penjadwalan Produksi

#### d. Diagram Sekuensial Mengelola Data *Bill of Material*

Proses mengelola data bill of material dimulai dari aktor mengakses *boundary* BoM\_View. Kemudian *boundary* BoM\_View meminta *control* c\_BoM



mengambil data BoM dari *entity* e\_BoM untuk ditampilkan beserta data bahan baku dari *entity* e\_BahanBaku. Setelah itu aktor melakukan operasi memasukan data *bill of material* pada *boundary* BoM\_View. Setelah selesai, sistem akan memeriksa *format* data yang dimasukan. Jika ada kesalahan, sistem akan menampilkan pesan *format* salah dan meminta aktor untuk mengubahnya. Apabila benar, sistem akan menampilkan pesan data tersimpan dan data BoM disimpan oleh sistem di *entity* e\_BoM melalui *control* c\_BoM. Diagram sekuensial mengelola data *bill of material* dapat dilihat pada gambar 3.14.



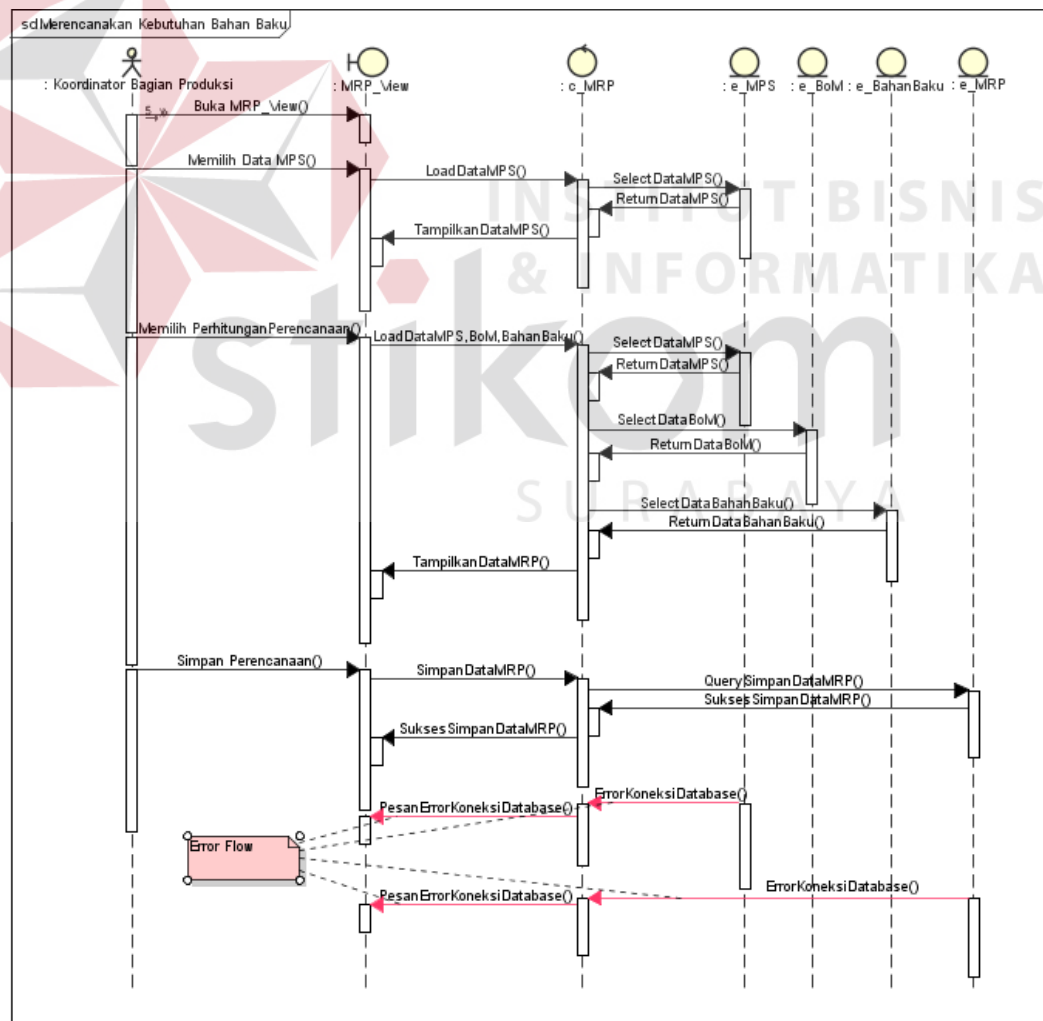
Gambar 3.14 Diagram Sekuensial Mengelola Data *Bill of Material* (BoM)

#### e. Diagram Sekuensial Merencanakan Kebutuhan Bahan Baku

Proses merencanakan kebutuhan bahan baku dimulai dari aktor mengakses *boundary* MRP\_View dan memilih data MPS untuk memilih nama produk dan

waktu yang ingin dijadwalkan kebutuhan bahan bakunya. Kemudian *boundary* MRP\_View meminta *control* c\_MRP mengambil data MPS dari *entity* e\_MPS untuk ditampilkan. Setelah itu aktor akan memilih perhitungan perencanaan MRP untuk menghitung dan merencanakan kebutuhan bahan baku yang dibutuhkan. Proses ini membutuhkan data dalam *entity* e\_MPS, e\_BoM, dan e\_BahanBaku untuk diproses dalam perhitungan MRP dengan bantuan *control* c\_MRP untuk memanggil data dan tampilan. Aktor kemudian akan meminta *boundary* MRP\_View untuk menyimpan kedalam *entity* e\_MRP melalui *control* dari c\_MRP.

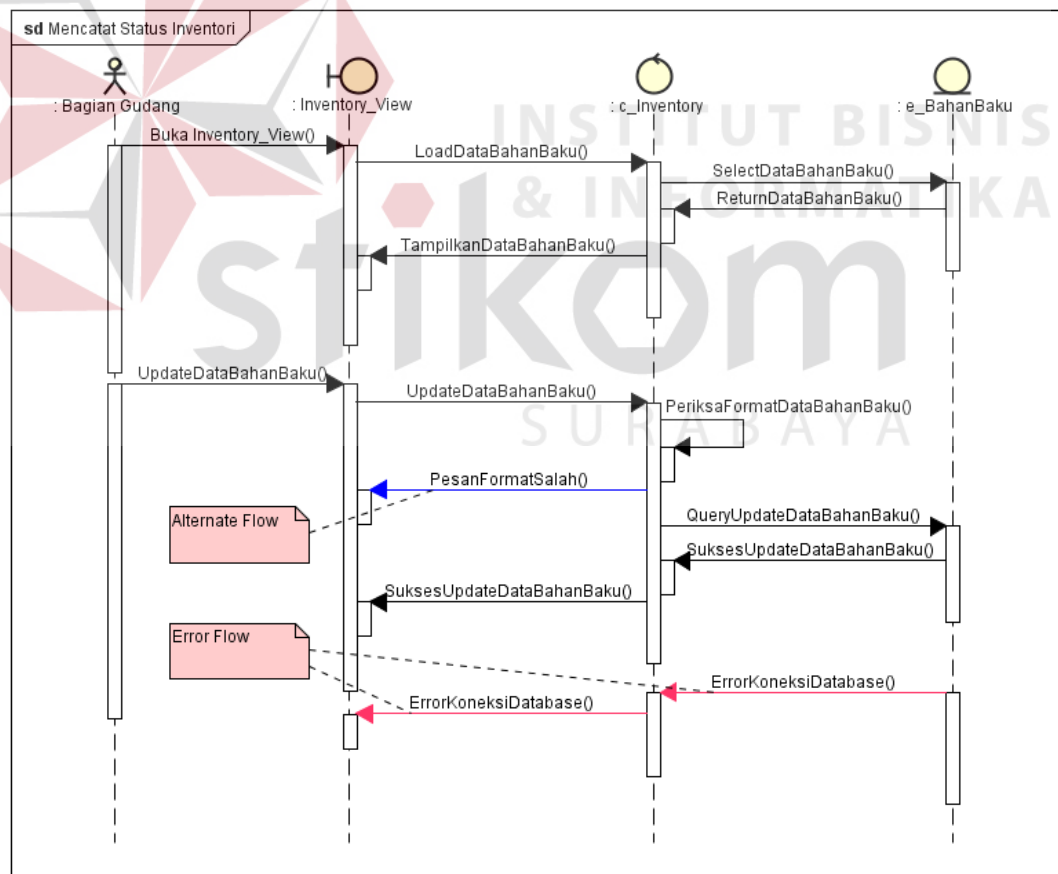
Diagram sekuensial merencanakan kebutuhan baku dapat dilihat pada gambar 3.15.



Gambar 3.15 Diagram Sekuensial Merencanakan Kebutuhan Bahan Baku

### f. Diagram Sekuensial Mencatat Status Inventori

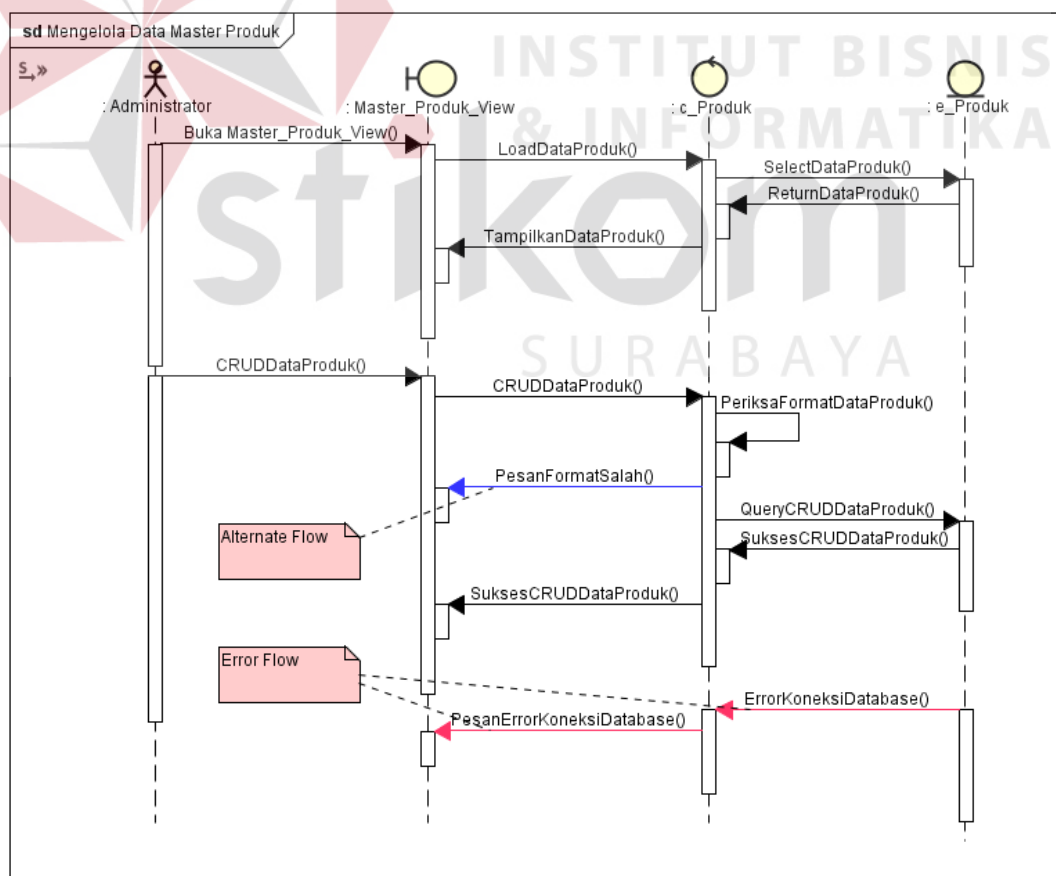
Proses mencatat status inventori dimulai dari aktor mengakses *boundary* Inventory\_View. Kemudian *boundary* Inventory\_View meminta *control* c\_Inventory mengambil data bahan baku dari *entity* e\_BahanBaku untuk ditampilkan. Setelah itu aktor melakukan operasi RU (*Read, Update*) pada *boundary* Inventory\_View. Setelah selesai, sistem akan memeriksa format data yang dimasukan. Jika ada kesalahan, sistem akan menampilkan pesan *format salah* dan meminta aktor untuk mengubahnya. Apabila benar, data bahan baku akan disimpan oleh sistem di *entity* e\_BahanBaku melalui *control* c\_Inventory. Diagram sekuensial mencatat status inventori dapat dilihat pada gambar 3.16.



Gambar 3.16 Diagram Sekuensial Mencatat Status Inventori

### g. Diagram Sekuensial Mengelola Data *Master* Produk

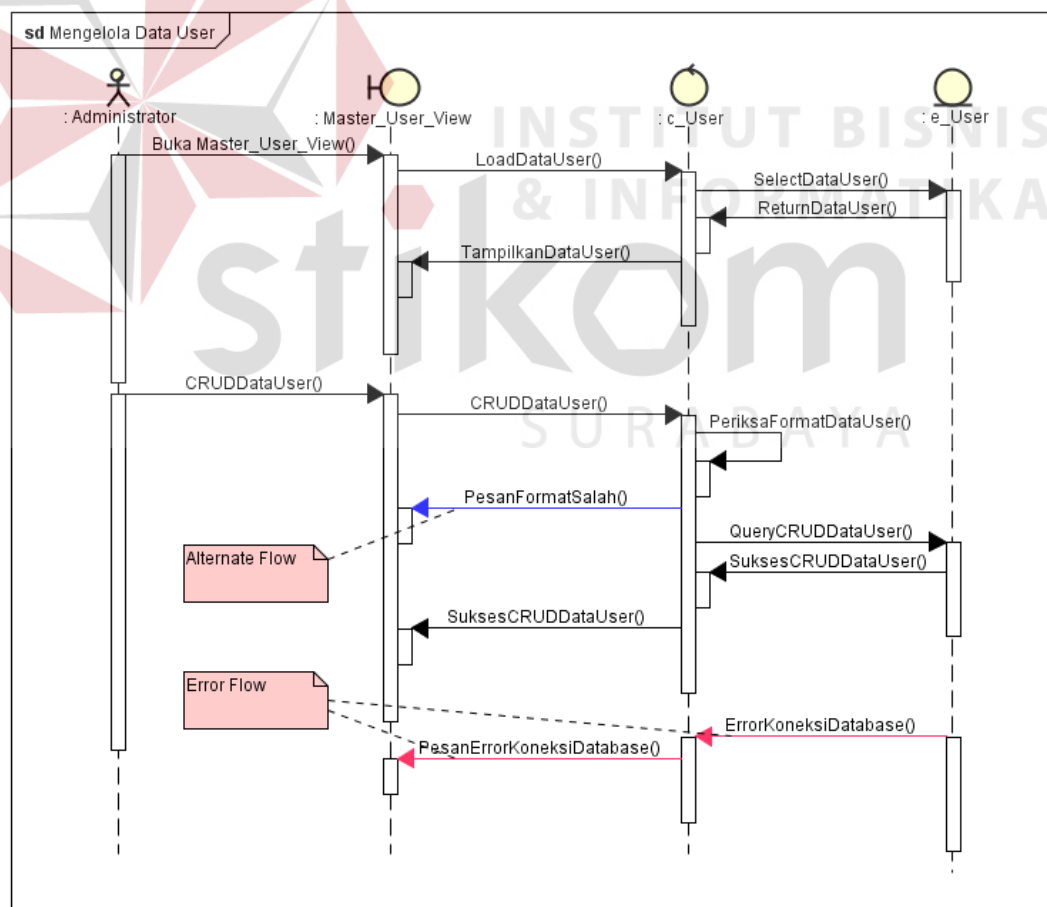
Proses mengelola data *master* produk dimulai dari aktor mengakses *boundary* *Master\_Produk\_View*. Kemudian *boundary* *Master\_Produk\_View* meminta *control* *c\_Produk* mengambil data produk dari *entity* *e\_Produk* untuk ditampilkan. Setelah itu aktor melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* *Master\_Produk\_View*. Setelah selesai, sistem akan memeriksa *format* data yang dimasukkan. Jika ada kesalahan, sistem akan menampilkan pesan *format* salah dan meminta aktor untuk mengubahnya. Apabila benar, data produk akan disimpan oleh sistem di *entity* *e\_Produk* melalui *control* *c\_Produk*. Diagram sekuensial mengelola data *master* produk dapat dilihat pada gambar 3.17.



Gambar 3.17 Diagram Sekuensial Mengelola Data *Master* Produk

### h. Diagram Sekuensial Mengelola Data User

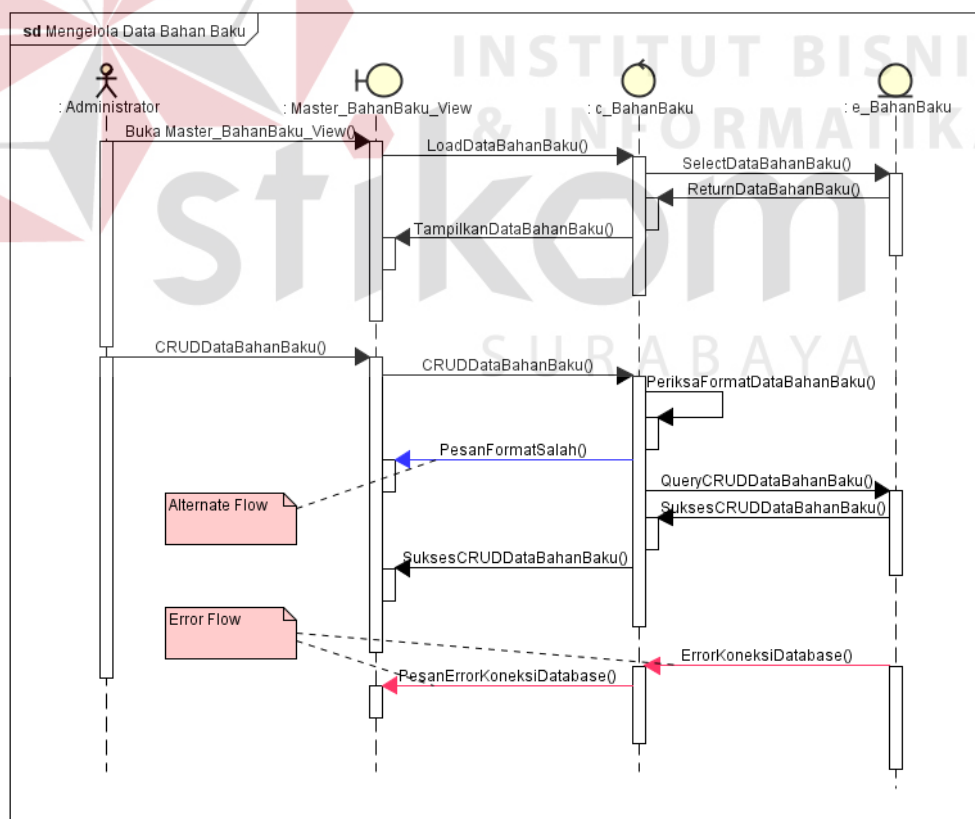
Proses mengelola data *user* dimulai dari aktor mengakses *boundary* *Master\_User\_View*. Kemudian *boundary* *Master\_User\_View* meminta *control* *c\_User* mengambil data *user* dari *entity* *e\_User* untuk ditampilkan. Setelah itu aktor melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* *Master\_User\_View*. Setelah selesai, sistem akan memeriksa *format* data yang dimasukan. Jika ada kesalahan, sistem akan menampilkan pesan *format* salah dan meminta aktor untuk mengubahnya. Apabila benar, data *user* akan disimpan oleh sistem di *entity* *e\_User* melalui *control* *c\_User*. Diagram sekuensial mengelola data *user* dapat dilihat pada gambar 3.18.



Gambar 3.18 Diagram Sekuensial Mengelola Data User

### i. Diagram Sekuensial Mengelola Data Bahan Baku

Proses mengelola data bahan baku dimulai dari aktor mengakses *boundary* Master\_BahanBaku\_View. Kemudian *boundary* Master\_BahanBaku\_View meminta *control* c\_BahanBaku mengambil data bahan baku dari *entity* e\_BahanBaku untuk ditampilkan. Setelah itu aktor melakukan operasi CRUD (*Create, Read, Update, Delete*) pada *boundary* Master\_BahanBaku\_View. Setelah selesai, sistem akan memeriksa *format* data yang dimasukkan. Jika ada kesalahan, sistem akan menampilkan pesan *format* salah dan meminta aktor untuk mengubahnya. Apabila benar, data bahan baku akan disimpan oleh sistem di *entity* e\_BahanBaku melalui *control* c\_BahanBaku. Diagram sekuensial mengelola data bahan baku dapat dilihat pada gambar 3.19.



Gambar 3.19 Mengelola Data Bahan Baku

### 3.3.6 Diagram Class

Diagram *class* adalah diagram interaksi antar kelas dalam sistem. Diagram *class* adalah alat perancangan terbaik untuk membantu tim pengembang dalam mendapatkan pola kelas-kelas kedalam sistem, struktur sistem sebelum menuliskan kode program dan membantu untuk memastikan bahwa sistem adalah rancangan terbaik dari beberapa alternatif. Desain diagram *class* dapat dilihat pada gambar 3.20.



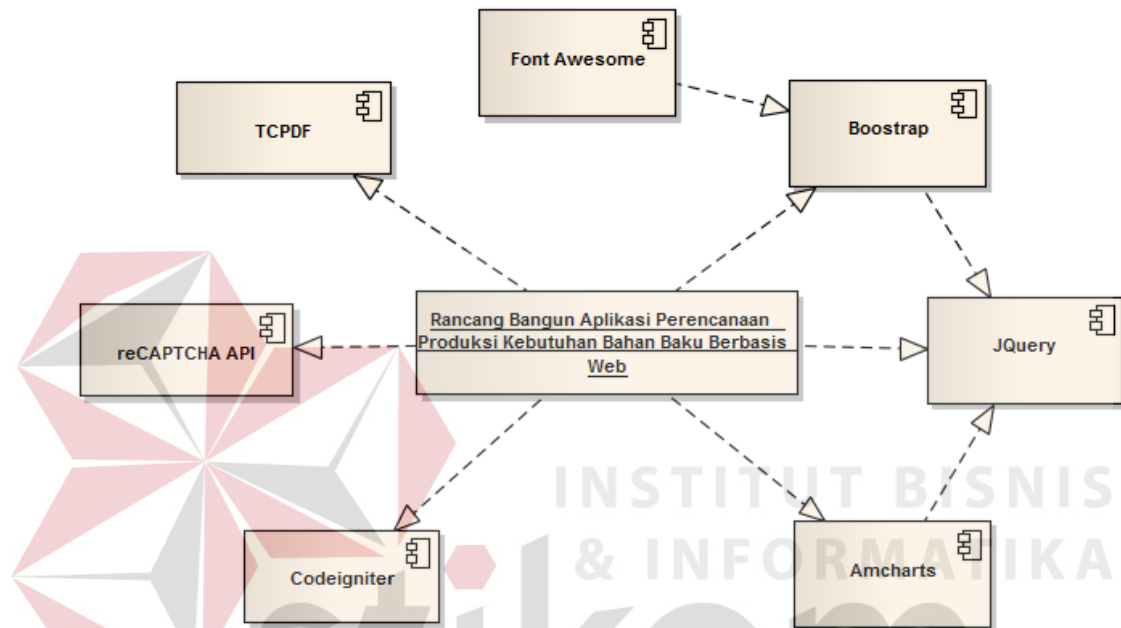


Gambar 3.20 Desain Diagram Class



### 3.3.7 Diagram Component

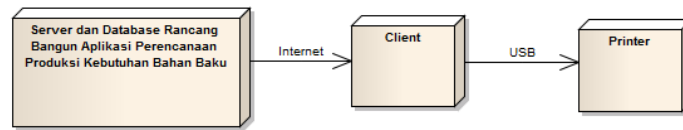
Diagram *component* adalah diagram yang menunjukkan model secara fisik atau komponen perangkat lunak pada sistem hubungannya antar mereka. Desain diagram *component* aplikasi sistem produksi berbasis web dapat dilihat pada gambar 3.21.



Gambar 3.21 Desain Diagram *Component*

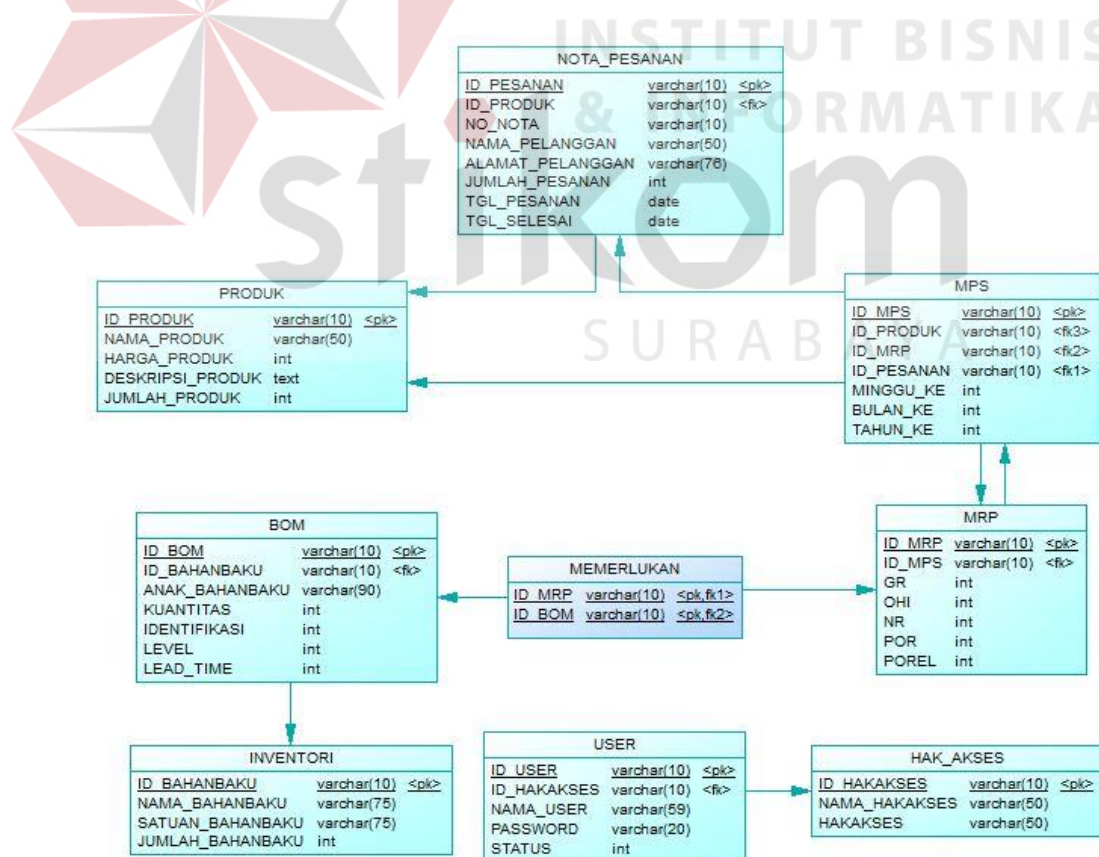
### 3.3.8 Diagram Deployment

Diagram *deployment* adalah diagram yang menampilkan rancangan fisik jaringan tempat berbagai komponen akan diletakan. Diagram ini akan menghubungkan titik-titik alat pemroses dan peralatan lainnya dalam sebuah *layout* fisik. Diagram *deployment* aplikasi sistem produksi berbasis web dapat dilihat pada gambar 3.22.

Gambar 3.22 Desain *Deployment Diagram*

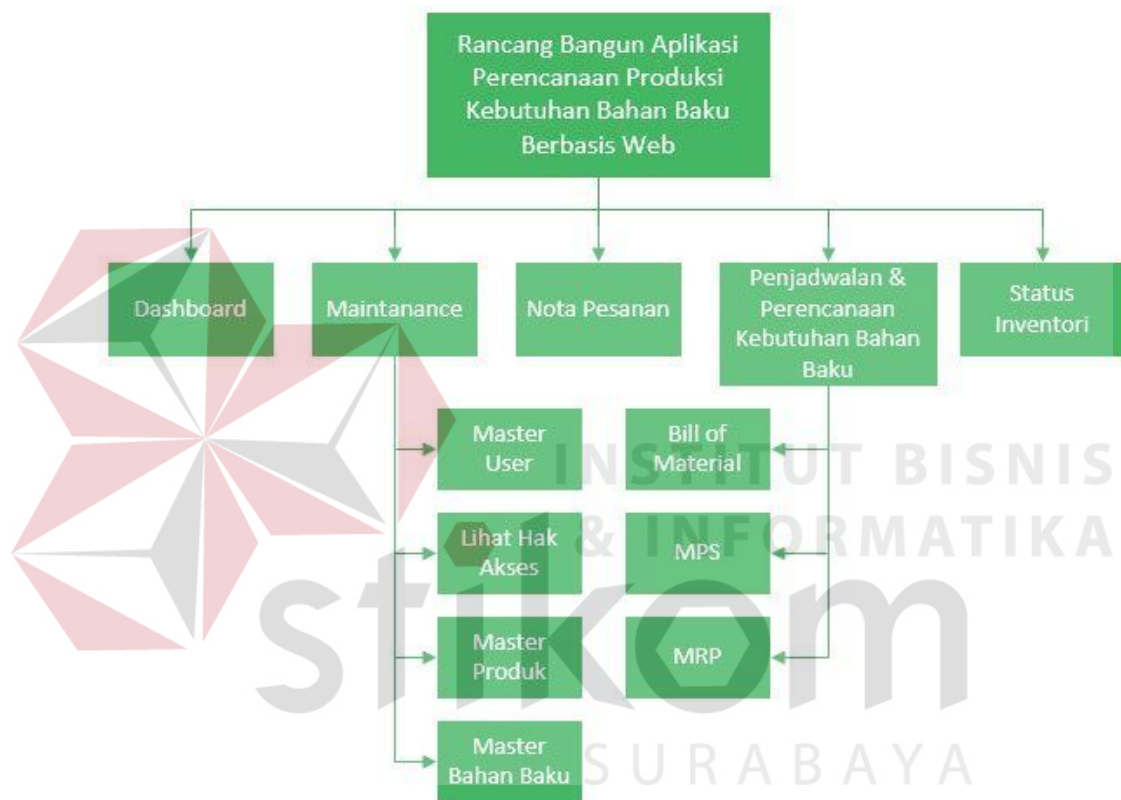
### 3.3.9 Physical Data Model (PDM)

*Physical data model* dapat menggambarkan secara detail tentang konsep struktur basis data yang akan dirancang pada suatu sistem, dimana merupakan hasil *generate* dari *conceptual data model* (CDM). Pada PDM akan tergambar jelas relasi antar tabelnya dengan ditunjukkan *primary key* dan *foreign key* dari masing-masing tabel. *Physical data model* aplikasi sistem produksi dapat dilihat pada gambar 3.23.

Gambar 3.23 Desain *Physical Data Model (PDM)*

### 3.3.10 Sitemap

*Sitemap* merupakan sebuah *file* yang berisi daftar *link* yang terdapat pada sebuah *web/blog* atau dengan kata lain merupakan salah satu alat bantu yang dapat mempermudah dalam pengenalan peta situs dalam suatu *website*. Desain *sitemap* aplikasi sistem produksi dapat dilihat pada gambar 3.24.



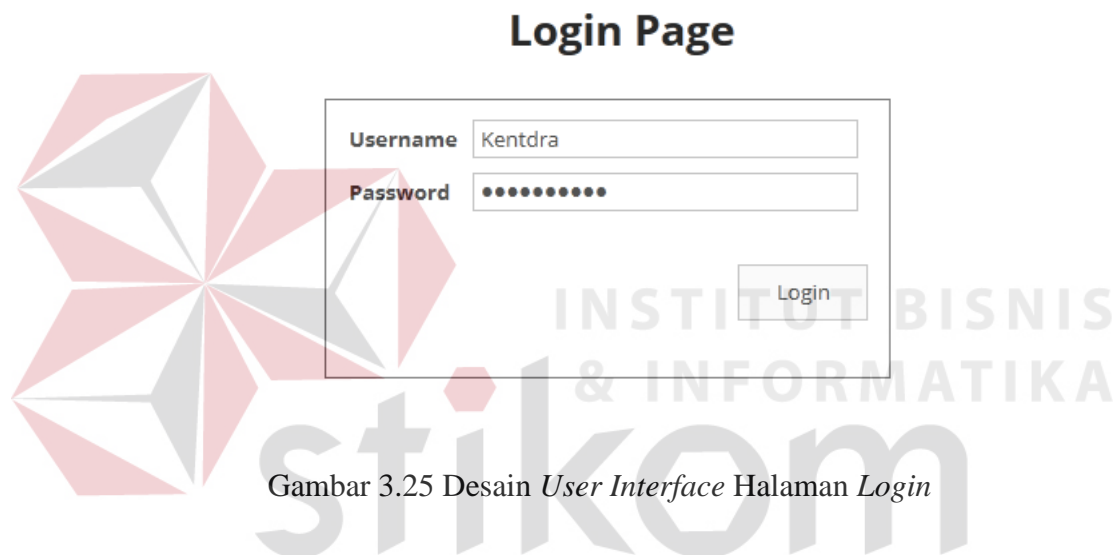
Gambar 3.24 Desain *Sitemap*

### 3.3.11 Desain *User Interface*

Desain *user interface* adalah rancangan *form* yang akan diimplementasikan kedalam sistem dan berfungsi sebagai antarmuka pengguna dengan sistem. Rancangan ini akan menerima masukan data dari pengguna yang dimasukkan dalam *database* dan hasilnya akan berupa laporan.

**a. Desain User Interface Halaman Login**

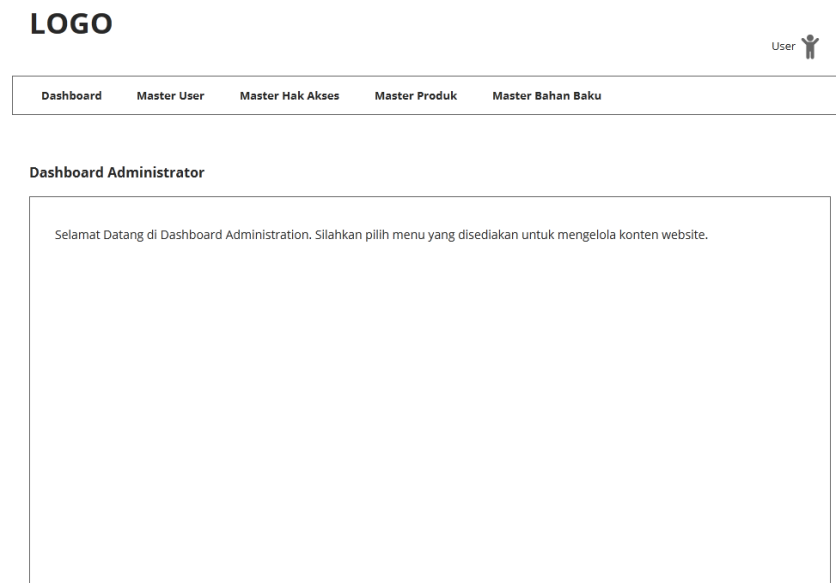
Halaman ini digunakan untuk menjaga keamanan sistem. Tujuannya adalah agar sistem digunakan oleh *user* yang berhak memakai dan berjalan sesuai hak akses yang telah ditentukan. *Field* yang diisi dalam *form* ini adalah *username* dan *password*. Selanjutnya klik tombol *login* yang secara otomatis sistem akan bekerja untuk melakukan validasi *user* dan mencari hak aksesnya. Desain *user interface* halaman *login* ini dapat dilihat pada gambar 3.25.



Gambar 3.25 Desain *User Interface* Halaman *Login*

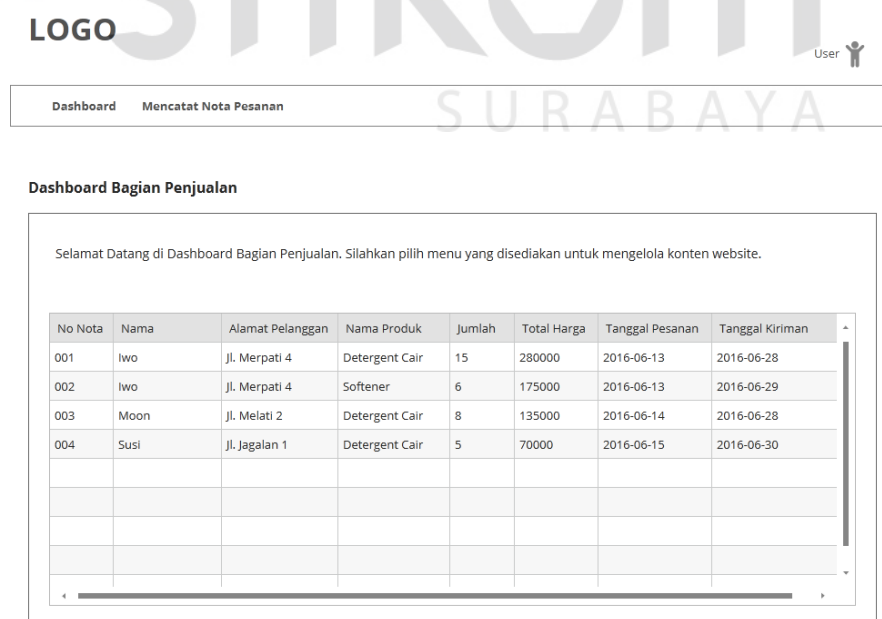
**b. Desain User Interface Halaman Dashboard**

Halaman ini digunakan untuk menampilkan halaman awal apabila *user* berhasil *login* kedalam sistem. Halaman *dashboard* rancang bangun aplikasi perencanaan produksi kebutuhan bahan baku berbasis web dibagi menjadi empat yang dibedakan berdasarkan hak akses masing-masing *user*. Halaman *dashboard* dibawah merupakan halaman *dashboard* yang dapat diakses oleh *user* yang memiliki hak akses sebagai *administrator*. Desain *user interface* halaman *dashboard administrator* dapat dilihat pada gambar 3.26.



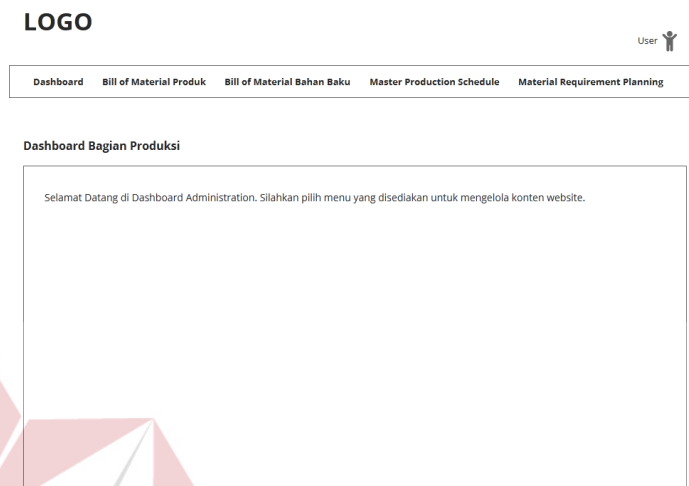
Gambar 3.26 Desain *User Interface* Halaman *Dashboard Administrator*

Halaman *dashboard* dibawah merupakan halaman *dashboard* yang dapat diakses oleh *user* yang memiliki hak akses sebagai bagian penjualan. Desain *user interface* halaman *dashboard* bagian penjualan dapat dilihat pada gambar 3.27.



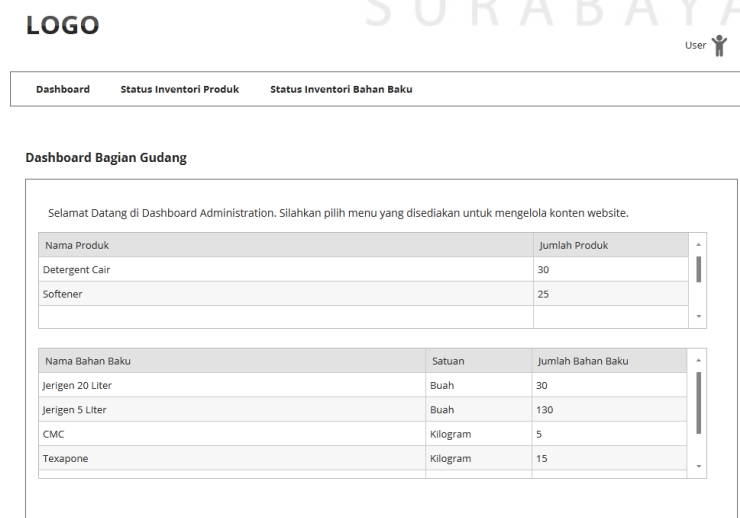
Gambar 3.27 Desain *User Interface* Halaman *Dashboard* Bagian Penjualan

Halaman *dashboard* dibawah merupakan halaman *dashboard* yang dapat diakses oleh *user* yang memiliki hak akses sebagai koordinator bagian produksi. Desain *user interface* halaman *dashboard administrator* dapat dilihat pada gambar 3.28.



Gambar 3.28 Desain *User Interface* Halaman *Dashboard* Koordinator Bagian Produksi

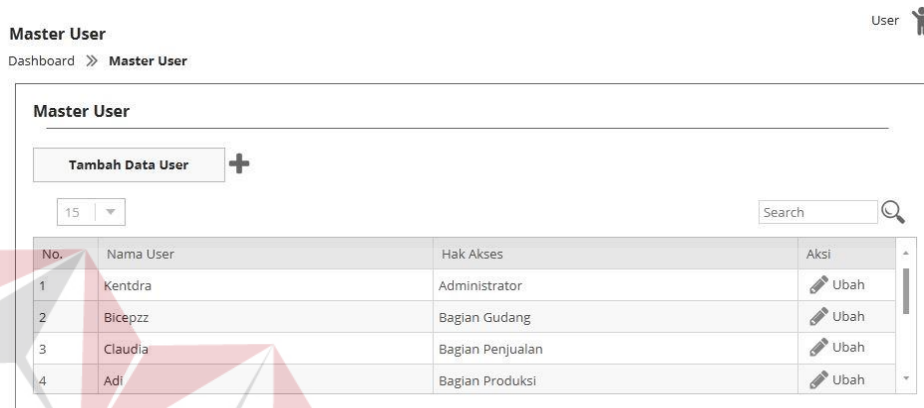
Halaman *dashboard* dibawah merupakan halaman *dashboard* yang dapat diakses oleh *user* yang memiliki hak akses sebagai bagian gudang. Desain *user interface* halaman *dashboard* bagian gudang dapat dilihat pada gambar 3.29.



Gambar 3.29 Desain *User Interface* Halaman *Dashboard* Bagian Gudang

**c. Desain User Interface Halaman Master User**

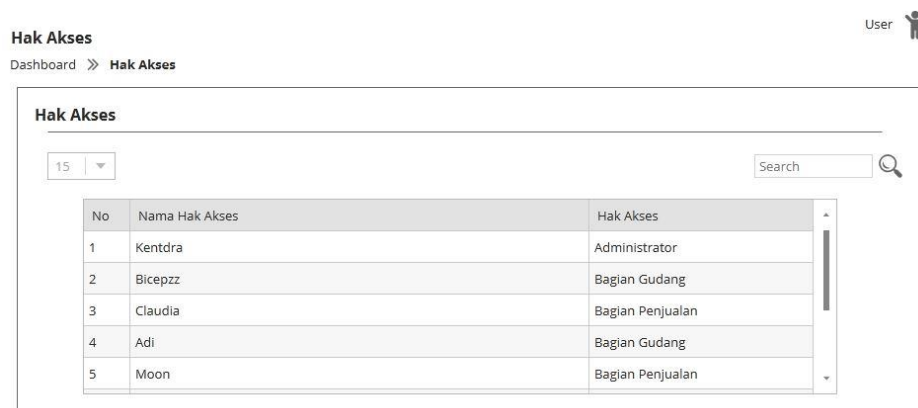
Halaman ini digunakan untuk *maintenance* data *user*, yang bertugas untuk melakukan *maintenance* data *user* ini adalah *administrator*. *Administrator* dapat melakukan *insert* dan *update* data *user*. Desain *user interface* halaman *master user* dapat dilihat pada gambar 3.30.



Gambar 3.30 Desain *User Interface* Halaman *Master User*

**d. Desain User Interface Halaman Hak Akses**

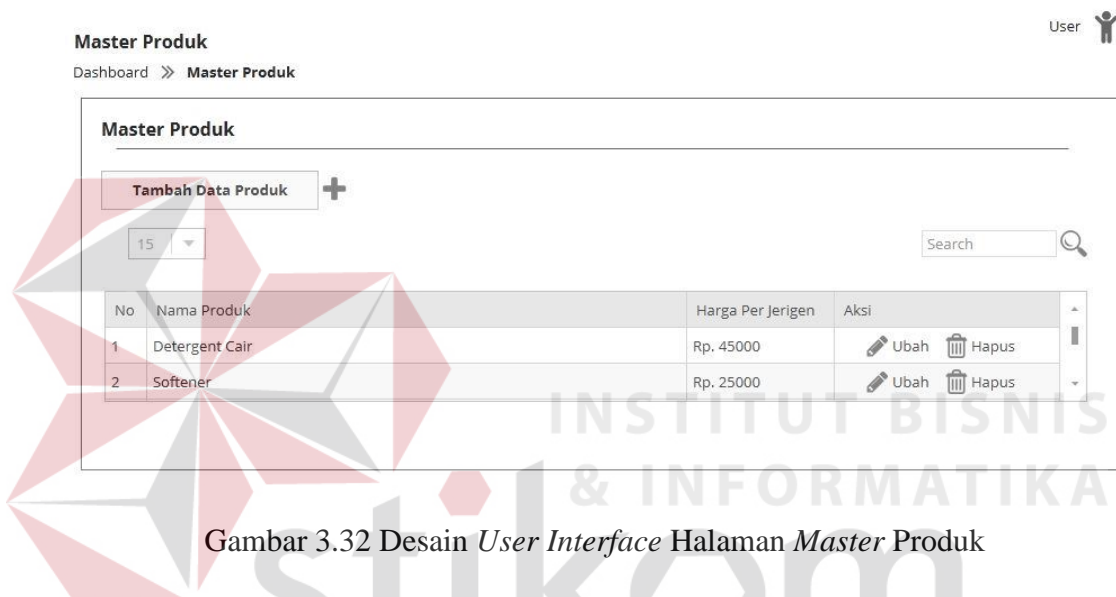
Halaman ini digunakan untuk melihat hak akses dalam sistem, yang dapat melihat hak akses ini adalah *administrator*. Desain *user interface* halaman hak akses ini dapat dilihat pada gambar 3.31.



Gambar 3.31 Desain *User Interface* Halaman Hak Akses

**e. Desain User Interface Halaman Master Produk**

Halaman ini digunakan untuk *maintenance* data produk, yang bertugas untuk melakukan *maintenance* data produk adalah *administrator*. *Administrator* dapat melakukan *insert*, *update*, dan *delete* data produk. Data yang dimasukkan pada halaman ini yaitu nama produk dan harga produk per jerigen. Desain *user interface* halaman *master* produk dapat dilihat pada gambar 3.32.




Gambar 3.32 Desain User Interface Halaman Master Produk

**f. Desain User Interface Halaman Master Bahan Baku**

Halaman ini digunakan untuk *maintenance* data bahan baku, yang bertugas untuk melakukan *maintenance* data bahan baku adalah *administrator*. *Administrator* dapat melakukan *insert*, *update*, dan *delete* data produk. Data yang dimasukkan pada halaman ini yaitu nama bahan baku dan satuan bahan baku. Desain *user interface* halaman *master* bahan baku dapat dilihat pada gambar 3.33.












**Master Bahan Baku** User 

Dashboard » **Master Bahan Baku**

**Master Bahan Baku**

**Tambah Data Bahan Baku** +


15 ▼ Search 

No	Nama Bahan	Satuan	Aksi
1	CMC	Kilogram	 Ubah  Hapus
2	Jerigen 20 Liter	Buah	 Ubah  Hapus
3	Jerigen 5 Liter	Buah	 Ubah  Hapus
4	Aquadest	Liter	 Ubah  Hapus

Gambar 3.33 Desain *User Interface* Halaman Master Bahan Baku

### g. Desain *User Interface* Halaman Mencatat Nota Pesanan


Halaman ini digunakan untuk menyimpan data pesanan dari pelanggan, yang bertugas untuk melakukan pencatatan nota pesanan adalah bagian penjualan. Bagian penjualan dapat melakukan insert dan hapus data nota pesanan. Data yang dimasukkan dalam halaman ini yaitu nomor nota, nama pelanggan, alamat pelanggan, nama produk, jumlah pesanan, tanggal pesanan, dan tanggal pengiriman. Desain *user interface* halaman mencatat nota pesanan dapat dilihat pada gambar 3.34.





**Mencatat Nota Pesanan** User 

Dashboard » **Mencatat Nota Pesanan**

**Nota Pesanan Barang**

**Tambah Data Nota Pesanan** +

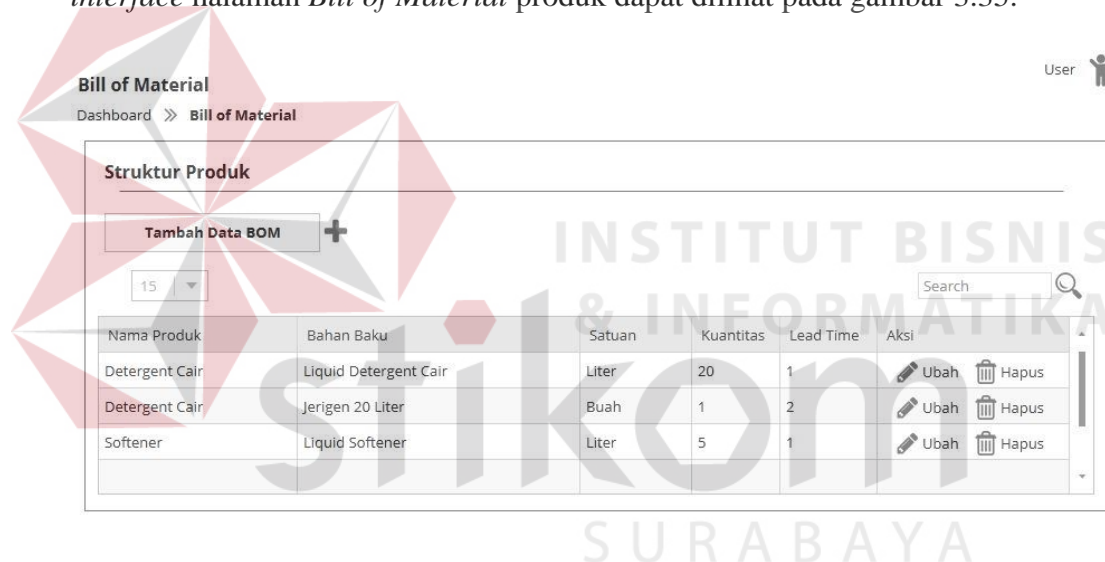
15 ▼ Search 

No Nota	Nama	Alamat Pelanggan	Nama Produk	Jumlah	Total Harga	Tanggal Pesanan	Tanggal Kiriman	Aksi
001	Iwo	Jl. Merpati 4	Detergent Cair	15	280000	2016-06-13	2016-06-28	 Hapus
002	Iwo	Jl. Merpati 4	Softener	6	175000	2016-06-13	2016-06-29	 Hapus
003	Moon	Jl. Melati 2	Detergent Cair	8	135000	2016-06-14	2016-06-28	 Hapus
004	Susi	Jl. Jagalan 1	Detergent Cair	5	70000	2016-06-15	2016-06-30	 Hapus

Gambar 3.34 Desain *User Interface* Halaman Mencatat Nota Pesanan


#### h. Desain *User Interface* Halaman *Bill of Material* (BoM)

Halaman ini digunakan untuk membuat susunan struktur produk mulai dari barang jadi sampai anak bahan baku, yang bertugas untuk bertanggung jawab dalam proses ini adalah koordinator bagian produksi. Dalam desain *user interface* halaman *Bill of Material* dibagi menjadi dua yaitu halaman *Bill of Material* produk dan halaman *Bill of Material* bahan baku. Dalam halaman *Bill of Material* produk koordinator bagian produksi dapat mengatur susunan *Bill of Material* dengan mengisi nama produk, nama bahan baku, kuantitas, dan *lead time*. Desain *user interface* halaman *Bill of Material* produk dapat dilihat pada gambar 3.35.



Gambar 3.35 Desain *User Interface* Halaman *Bill of Material* Produk


Dalam halaman *Bill of Material* bahan baku koordinator bagian produksi dapat mengatur susunan *Bill of Material* dengan mengisi nama bahan baku, anak bahan baku, kuantitas, tingkatan, dan lead time. Desain *user interface* halaman *Bill of Material* bahan baku dapat dilihat pada gambar 3.36.







**Bill of Material** User 

Dashboard » **Bill of Material**

**Struktur Bahan Baku**

+

15 Search 

Bahan Baku	Satuan	Anak Bahan Baku	Satuan	Kuantitas	Lead Time	Aksi
Liquid Detergent Cair	Liter	CMC	Kilogram	0.015	3	 Ubah  Hapus
Liquid Detergent Cair	Liter	Aquadest	Liter	0.685	3	 Ubah  Hapus
Liquid Softener	Liter	Softener Flake	Kilogram	0.05	3	 Ubah  Hapus

Gambar 3.36 Desain *User Interface* Halaman *Bill of Material* Bahan Baku

### i. Desain *User Interface* Halaman *Master Production Schedule* (MPS)

Halaman ini digunakan untuk penjadwalan produksi dengan cara merekap kemudian memasukkan data pesanan dan status inventori produk kedalam sistem untuk direncanakan, yang bertugas adalah koordinator bagian produksi. Koordinator bagian produksi dapat memilih nama produk dan *range* tanggal pesanan. Desain *user interface* halaman *master production schedule* dapat dilihat pada gambar 3.37.

**Master Production Schedule** User 

Dashboard » **Master Production Schedule**


**Penjadwalan Induk Produksi**

1 Friday	2 Saturday	3 Sunday	4 Monday	5 Tuesday	6 Wednesday	7 Thursday
8 Friday	9 Saturday	10 Sunday	11 Monday Nota: 1 = 4 Nota: 2 = 12 Nota: 3 = 4	12 Tuesday Nota: 5 = 10 Nota: 6 = 7	13 Wednesday Nota: 8 = 5 Nota: 9 = 15	14 Thursday Nota: 3 = 10 Nota: 10 = 10
15 Friday Nota: 14 = 8	16 Saturday	17 Sunday	18 Monday	19 Tuesday	20 Wednesday	21 Thursday
22 Friday	23 Saturday	24 Sunday	25 Monday	26 Tuesday	27 Wednesday	28 Thursday
29 Friday	30 Saturday	31 Sunday	32	33	34	35



Gambar 3.37 Desain *User Interface* Halaman *Master Production Schedule*

**j. Desain User Interface Halaman Material Requirement Planning (MRP)**

Halaman ini digunakan untuk merencanakan dan menghitung kebutuhan produksi beserta bahan baku yang akan diproduksi sesuai dengan MPS dan status inventori bahan baku di gudang, yang bertugas adalah koordinator bagian produksi. Koordinator bagian produksi dapat memilih nama produk dan *range* tanggal pesanan. Desain *user interface* halaman *material requirement planning* (MRP) dapat dilihat dari gambar 3.38.

**Material Requirement Planning** User 

Dashboard » Material Requirement Planning » Perencanaan MRP

Cetak MRP Primary Report  Cetak MRP Pegging Report 

Produk : Detergent Cair	Bahan Baku : Liquid (Detergent Cair)
Tanggal Produksi : Saturday, 09/07/2016	Anak Bahan Baku : CMC

Kebutuhan Kotor (GR)	1.2
Persediaan yang Ada (OHI)	0
Kebutuhan Bersih (NR)	1.2
Bahan yang Tersedia (POR)	1.2
Bahan yang Diproduksi (POREL)	1.2

Produk : Detergent Cair	Bahan Baku : Liquid (Detergent Cair)
Tanggal Produksi : Saturday, 09/07/2016	Anak Bahan Baku : Aquadest

Kebutuhan Kotor (GR)	54.8
Persediaan yang Ada (OHI)	0
Kebutuhan Bersih (NR)	54.8
Bahan yang Tersedia (POR)	54.8
Bahan yang Diproduksi (POREL)	54.8

Gambar 3.38 Desain User Interface Halaman MRP

**k. Desain User Interface Halaman Status Inventori**

Halaman ini digunakan untuk memasukan dan memperbarui data bahan baku yang dibutuhkan untuk kegiatan produksi di gudang, yang bertugas adalah bagian gudang. Dalam desain *user interface* halaman status inventori dibagi menjadi dua yaitu halaman status inventori produk dan halaman status inventori bahan baku. Dalam halaman status inventori produk aktor dapat memilih nama

produk dan jumlah produk. Desain *user interface* halaman status inventori produk dapat dilihat pada gambar 3.39.

**Status Inventori** User 

Dashboard » **Status Inventori Produk**

**Data Status Inventori Produk**

**Update Stok Produk** +

15 Search 

Nama Produk	Jumlah Produk
Detergent Cair	30
Softener	25

Gambar 3.39 Desain *User Interface* Halaman Status Inventori Produk

Dalam halaman status inventori bahan baku aktor dapat memilih nama bahan baku dan jumlah bahan baku. Desain *user interface* halaman status inventori bahan baku dapat dilihat pada gambar 3.40.

**Status Inventori** User 

Dashboard » **Status Inventori Bahan Baku**

**Data Status Inventori Bahan Baku**

**Update Stok Bahan Baku** +

15 Search 

Nama Bahan Baku	Satuan	Jumlah Bahan Baku
Jerigen 20 Liter	Buah	30
Jerigen 5 Liter	Buah	130
CMC	Kilogram	5
Texapone	Kilogram	15

Gambar 3.40 Desain *User Interface* Halaman Status Inventori Bahan Baku

### 3.3.12 Desain *Input/Output*

Desain *input/output* digunakan untuk menggambarkan *input* dan *output* dari lingkungan eksternal sistem yang akan dimasukkan kedalam sistem. Desain *input* dalam sistem ini tidak digambarkan karena semua input dimasukkan langsung kedalam sistem. Desain *output* yang akan dihasilkan sistem adalah nota pesanan, MRP *primary report*, dan MRP *pegging report*.

#### a. Desain *Output* Nota Pesanan

Desain nota pesanan adalah *output* untuk setiap transaksi pesanan yang dimasukkan kedalam sistem dihalaman bagian penjualan untuk diberikan kepada pelanggan. Nota pesanan berisi nomor nota, nama pelanggan, alamat pelanggan, tanggal pesanan, tanggal pengiriman, nama produk yang dipesan beserta jumlah, harga dan total harga. Desain *output* nota pesanan dapat dilihat pada gambar 3.41.

**Logo**

**Nota Pesanan CV. Mitra Techno Sains Surabaya**  
 Pondok Wiguna Regency V no.25 Gunung Anyar Surabaya

No Nota : 001  
 Nama Pelanggan : Iwo  
 Alamat Pelanggan : Jalan Merpati 4  
 Tanggal Pesanan : 2016-06-13  
 Tanggal Pengiriman : 2016-06-28

Nama Produk	Harga Produk	Jumlah Pesanan	Harga Total
Detergent Cair	45000	15	280000
Softener	25000	6	175000
<b>Total Harga</b>			<b>: Rp. 400000</b>

Gambar 3.41 Desain *Output* Nota Pesanan

#### b. Desain *Output* MRP *Primary Report*

Desain MRP *primary report* adalah laporan utama dalam rancang bangun aplikasi perencanaan kebutuhan bahan baku. MRP *primary report* akan

menampilkan nama produk, bahan baku, dan anak bahan baku yang dipakai sesuai dengan turunan. Laporan ini ditampilkan berdasarkan periode waktu mingguan.

Desain *output MRP primary report* dapat dilihat pada gambar 3.42.

**Logo MRP Primary Report**  
**Perencanaan Kebutuhan Bahan Baku (MRP) CV Mitra Techno**  
 Pondok Wiguna Regency V no.25 Gunung Anyar Surabaya

Nama Produk : Detergent Cair  
 Tanggal Mulai : 2016-06-18  
 Tanggal Selesai : 2016-06-24

Produk	: Detergent Cair	Bahan Baku	: Liquid (Detergent Cair)
Tanggal Produksi	: Saturday, 09/07/2016	Anak Bahan Baku	: CMC
Kebutuhan Kotor (GR)	1.2		
Persediaan yang Ada (OHI)	0		
Kebutuhan Bersih (NR)	1.2		
Bahan yang Tersedia (POR)	1.2		
Bahan yang Diproduksi (POREL)	1.2		

Produk	: Detergent Cair	Bahan Baku	: Liquid (Detergent Cair)
Tanggal Produksi	: Saturday, 09/07/2016	Anak Bahan Baku	: Aquadest
Kebutuhan Kotor (GR)	54.8		
Persediaan yang Ada (OHI)	0		
Kebutuhan Bersih (NR)	54.8		
Bahan yang Tersedia (POR)	54.8		
Bahan yang Diproduksi (POREL)	54.8		

Produk	: Detergent Cair	Bahan Baku	: Liquid (Detergent Cair)
Tanggal Produksi	: Saturday, 09/07/2016	Anak Bahan Baku	: Texapone
Kebutuhan Kotor (GR)	8		
Persediaan yang Ada (OHI)	0		
Kebutuhan Bersih (NR)	8		
Bahan yang Tersedia (POR)	8		
Bahan yang Diproduksi (POREL)	8		

Gambar 3.42 Desain *Output MRP Primary Report*

### c. Desain *Output MRP Pegging Report*

Desain *MRP pegging report* adalah laporan kebutuhan bahan baku yang berfungsi untuk memberikan informasi bahan baku yang harus dipenuhi untuk proses produksi dan berguna untuk menghindari keterlambatan produksi. Laporan ini ditampilkan berdasarkan periode waktu mingguan. Desain *output MRP pegging report* dapat dilihat pada gambar 3.43.

Logo

**MRP Pegging Report**

Perencanaan Kebutuhan Bahan Baku (MRP) CV Mitra Techno

Pondok Wiguna Regency V no.25 Gunung Anyar Surabaya

Nama Produk : Detergent Cair  
 Tanggal Mulai : 2016-06-18  
 Tanggal Selesai : 2016-06-24

Nama Bahan Baku	Satuan	Jumlah Kekurangan
CMC	Kilogram	20
Aquadest	Liter	24
Jerigen 20 Liter	Buah	10

Gambar 3.43 Desain *Output MRP Pegging Report***3.3.13 Rancangan Pengujian dan Evaluasi Sistem**

Rancang bangun aplikasi sistem perencanaan kebutuhan bahan baku yang telah dirancang dan dibangun harus diuji untuk mengetahui apakah sistem dapat berjalan sesuai dengan yang diharapkan. Uji coba yang dilakukan adalah proses *login*, pengelolaan data *master*, pencatatan nota pesanan, pengelolaan *Bill of Material*, penjadwalan produksi, perencanaan kebutuhan bahan baku, dan status inventori. Pengujian yang dilakukan akan menggunakan metode *Black Box Testing*. Pengujian ini akan berfokus pada apakah unit program telah memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Cara pengujian akan dilakukan dengan menjalankan unit atau modul, kemudian diamati apakah hasil dari unit atau modul sesuai dengan proses yang diinginkan. Hal-hal yang akan diujikan akan ditampilkan secara detail pada tabel 3.12.



Tabel 3.12 Rencana Pengujian Rancang Bangun Aplikasi Sistem Perencanaan Kebutuhan Bahan Baku

<i>Requirement yang diuji</i>	<i>Fungsi yang diuji</i>
<i>Login</i>	Melakukan <i>login</i>
<b>Data User</b>	Melakukan operasi CRUD ( <i>Create, Read, Update, dan Delete</i> ) pada data <i>user</i>
<b>Data Produk</b>	Melakukan operasi CRUD ( <i>Create, Read, Update, dan Delete</i> ) pada data produk
<b>Data Bahan Baku</b>	Melakukan operasi CRUD ( <i>Create, Read, Update, dan Delete</i> ) pada data bahan baku
<b>Data Pesanan</b>	Mencatat transaksi nota pesanan
	Mencetak nota pesanan
<b>Data Penjadwalan Produksi</b>	Mengelola <i>Bill of Material</i> (BoM) Produk dan Bahan Baku
	Melakukan penjadwalan produksi (MPS)
	Melakukan penjadwalan kebutuhan bahan baku produksi (MRP)
	Menampilkan dan mencetak MRP <i>Primary Report</i>
	Menampilkan dan mencetak MRP <i>Pegging Report</i>
<b>Data Status Inventori</b>	Mencatat stok produk dan bahan baku di gudang

a. **Desain Uji Coba *Login***

Desain uji coba *login* bertujuan untuk menguji apakah fungsi *login* dapat berjalan sesuai yang diharapkan. Desain uji coba *login* dapat dilihat pada tabel 3.13.

Tabel 3.13 Desain Uji Coba *Login*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Melakukan <i>login</i>	<i>Username</i> dan <i>Password</i>	<i>Login</i> berhasil dan masuk ke halaman <i>dashboard</i>	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
2	Melakukan <i>login</i> dengan <i>format</i> yang salah	<i>Username</i> dan <i>Password</i>	Muncul pesan “ <i>Username</i> atau <i>Password</i> Anda Salah”	

**b. Desain Uji Coba Data User**

Desain uji coba data *user* bertujuan untuk menguji apakah fungsi CRU (*Create*, *Read*, dan *Update*) pada data *user* dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data *user* dapat dilihat pada tabel 3.14.

Tabel 3.14 Desain Uji Coba Data *User*

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Menampilkan halaman <i>Master User</i>	Memilih menu “ <i>Master User</i> ”	Masuk ke halaman <i>Master User</i>	
2	Memasukan data <i>user</i>	Nama <i>user</i> dan <i>password</i>	Data <i>user</i> berhasil disimpan dan ditampilkan pada tabel <i>user</i>	
3	Mengubah data <i>user</i>	Nama <i>user</i> dan <i>password</i>	Data <i>user</i> berhasil diperbarui dan ditampilkan pada tabel <i>user</i>	

**c. Desain Uji Coba Data Produk**

Desain uji coba data produk bertujuan untuk menguji apakah fungsi CRUD (*Create*, *Read*, *Update*, dan *Delete*) pada data produk dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data produk dapat dilihat pada tabel 3.15.

Tabel 3.15 Desain Uji Coba Data Produk

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Menampilkan halaman <i>Master Produk</i>	Memilih menu “ <i>Master Produk</i> ”	Masuk ke halaman <i>Master Produk</i>	
2	Memasukan data produk	Nama produk dan harga produk	Data produk berhasil disimpan dan ditampilkan pada tabel data produk	
3	Mengubah data produk	Nama produk dan harga produk	Data produk berhasil diperbarui dan ditampilkan pada tabel data produk	

#### d. Desain Uji Coba Data Bahan Baku

Desain uji coba data bahan baku bertujuan untuk menguji apakah fungsi CRUD (*Create, Read, Update, dan Delete*) pada data bahan baku dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data bahan baku dapat dilihat pada tabel 3.16.

Tabel 3.16 Desain Uji Coba Data Bahan Baku

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Menampilkan halaman <i>Master Bahan Baku</i>	Memilih menu “ <i>Master Bahan Baku</i> ”	Masuk ke halaman <i>Master Bahan Baku</i>	
2	Memasukan data bahan baku	Nama bahan dan satuan bahan baku	Data bahan baku berhasil disimpan dan ditampilkan pada tabel data bahan baku	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
3	Mengubah data bahan baku	Nama bahan dan satuan bahan baku	Data bahan baku berhasil diperbarui dan ditampilkan pada tabel data bahan baku	

e. **Desain Uji Coba Data Pesanan**

Desain uji coba data pesanan bertujuan untuk menguji apakah fungsi mencatat, menghapus, dan mencetak pada transaksi data nota pesanan dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data pesanan dapat dilihat pada tabel 3.17.

Tabel 3.17 Desain Uji Coba Data Pesanan

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Menampilkan halaman Mencatat Nota Pesanan	Memilih menu “Mencatat Nota Pesanan”	Masuk ke halaman Mencatat Nota Pesanan	
2	Mencatat transaksi pesanan	Nomor nota, nama pelanggan, alamat pelanggan, nama produk, jumlah pesanan, tanggal pesanan, dan tanggal pengiriman	Pesan transaksi nota pesanan berhasil disimpan dan ditampilkan pada tabel data nota pesanan	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
3	Mencetak nota pesanan barang	No nota	Tampil nota pesanan barang dan dapat dicetak	

**f. Desain Uji Coba Data Penjadwalan Produksi**

Desain uji coba data penjadwalan produksi bertujuan untuk menguji apakah fungsi mengelola *Bill of Material* Produk, *Bill of Material* Bahan Baku, melakukan penjadwalan MPS, melakukan penjadwalan MRP, menampilkan MRP *primary Report*, dan menampilkan MRP *pegging report* dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data penjadwalan dapat dilihat pada tabel 3.18.

Tabel 3.18 Desain Uji Coba Data Penjadwalan Produksi

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Menampilkan halaman <i>Bill of Material</i> Produk	Memilih menu “ <i>Bill of Material</i> Produk”	Masuk ke halaman <i>Bill of Material</i> Produk	
2	Memasukan data <i>Bill of Material</i> Produk	Nama produk, nama bahan baku, kuantitas, <i>lead time</i>	Data struktur produk berhasil disimpan dan ditampilkan pada tabel data BoM	
3	Mengubah data <i>Bill of Material</i> Produk	Nama produk, nama bahan baku, kuantitas, <i>lead time</i>	Data struktur produk berhasil diperbarui dan ditampilkan pada tabel data BoM	

<b>Test Case ID</b>	<b>Tujuan</b>	<b>Input</b>	<b>Output yang diharapkan</b>	<b>Status</b>
4	Menampilkan halaman <i>Bill of Material</i> Bahan Baku	Memilih menu “ <i>Bill of Material</i> Bahan Baku”	Masuk ke halaman <i>Bill of Material</i> Bahan Baku	
5	Memasukan data <i>Bill of Material</i> Bahan Baku	Nama bahan baku, anak bahan baku, kuantitas, tingkatan, <i>lead time</i>	Data struktur produk berhasil disimpan dan ditampilkan pada tabel data BoM	
6	Mengubah data <i>Bill of Material</i> Bahan Baku	Nama bahan baku, anak bahan baku, kuantitas, tingkatan, <i>lead time</i>	Data struktur produk berhasil diperbarui dan ditampilkan pada tabel data BoM	
7	Menampilkan halaman Penjadwalan Produksi (MPS)	Memilih menu “ <i>Master Production Schedule</i> ”	Masuk ke halaman “ <i>Master Production Schedule</i> ”	
8	Merencanakan data penjadwalan produksi (MPS)	Nama produk dan <i>range</i> tanggal pesanan	Data MPS berhasil disimpan dan ditampilkan pada tabel data MPS	
9	Menampilkan halaman Penjadwalan Bahan Baku (MRP)	Memilih menu “ <i>Material Requirement Planning</i> ”	Masuk ke halaman “ <i>Material Requirement Planning</i> ”	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
10	Menghitung data penjadwalan bahan baku (MRP)	Nama produk dan <i>range</i> tanggal produksi	Data MRP berhasil disimpan dan ditampilkan pada tabel data MRP	
11	Menampilkan MRP <i>Primary Report</i>	Memilih tombol “Cetak MRP <i>Primary Report</i> ”	Tampilan MRP <i>Primary Report</i>	
12	Menampilkan MRP <i>Pegging Report</i>	Memilih tombol “Cetak MRP <i>Pegging Report</i> ”	Tampilan MRP <i>Pegging Report</i>	

**g. Desain Uji Coba Data Status Inventori**

Desain uji coba data status inventori bertujuan untuk menguji apakah fungsi mencatat status inventori produk dan bahan baku dapat berjalan sesuai dengan yang diharapkan. Desain uji coba data status inventori dapat dilihat pada tabel 3.19.

Tabel 3.19 Desain Uji Coba Data Status Inventori

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Menampilkan halaman Mencatat Status Inventori Produk	Memilih menu “Status Inventori Produk”	Masuk ke halaman Mencatat Status Inventori Produk	

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
2	Mencatat status inventori Produk	Nama produk dan jumlah produk	Pesan data status inventori berhasil disimpan dan ditampilkan pada tabel status inventori	
3	Menampilkan halaman Mencatat Status Inventori Bahan Baku	Memilih menu “Status Inventori Bahan Baku”	Masuk ke halaman Mencatat Status Inventori Bahan Baku	
4	Mencatat status inventori Bahan Baku	Nama Bahan Baku dan jumlah Bahan Baku	Pesan data status inventori berhasil disimpan dan ditampilkan pada tabel status inventori	

