

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

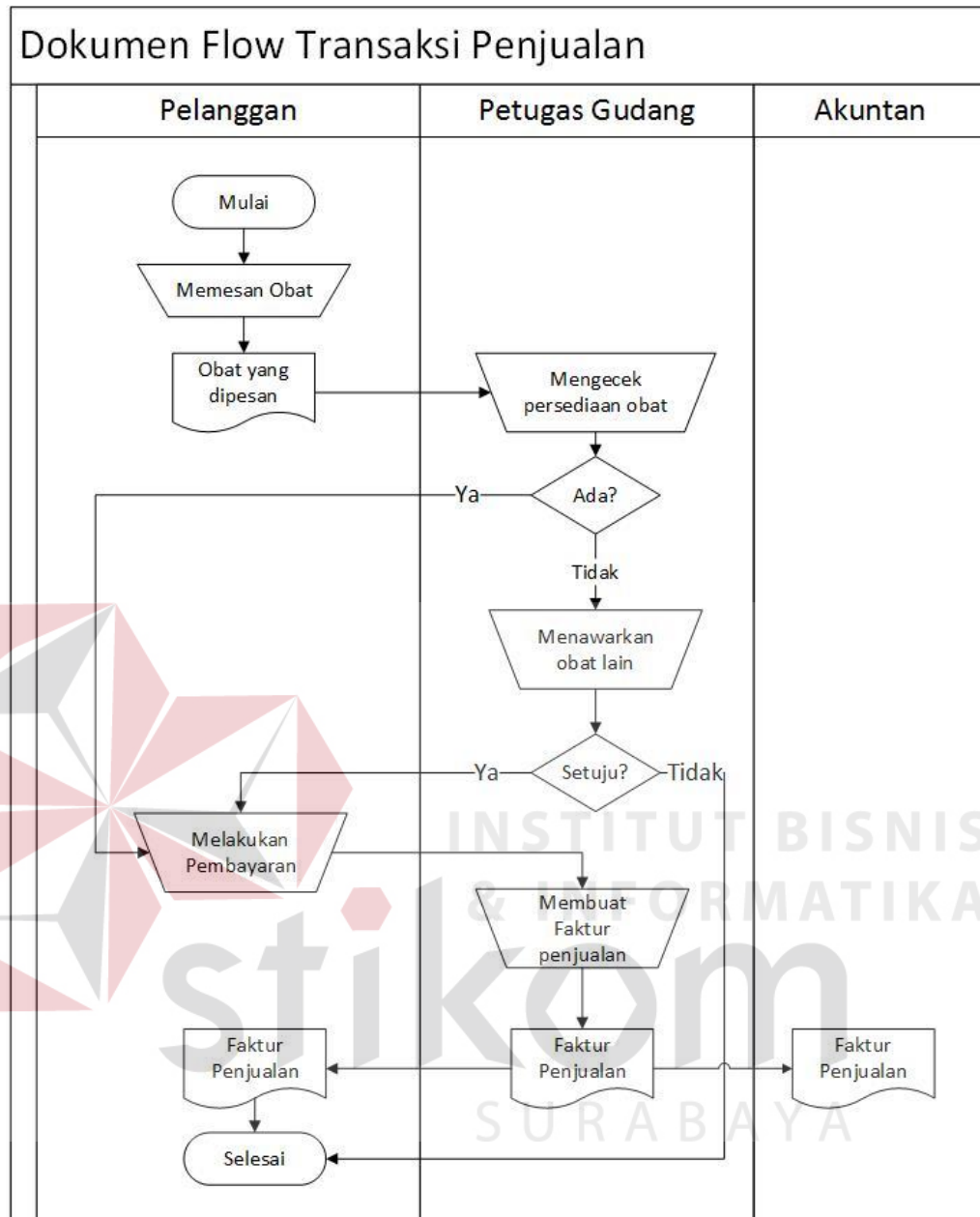
Pada tahap ini akan dijelaskan mengenai analisis permasalahan yang didapatkan dari pemahaman terhadap proses bisnis pada BSM, untuk selanjutnya dibuat rancangan desain sistem dari aplikasi penjualan obat hewan ternak pada BSM berdasarkan proses bisnis tersebut sebagai solusi terhadap permasalahan yang terjadi. Pada sub bab ini akan dimulai dari tahap identifikasi masalah dan analisis kebutuhan.

3.1.1 Identifikasi Masalah

Identifikasi masalah dilakukan setelah data didapatkan dari hasil wawancara dan observasi di BSM. Dari kegiatan identifikasi masalah didapatkan bahwa gudang BSM saat ini belum terdapat aplikasi yang dapat membantu dalam menyimpan dan mengolah transaksi penjualan setiap harinya sehingga menyulitkan petugas gudang dalam melayani setiap transaksi, melakukan pencatatan dan menghitung seluruh transaksi.

Semua transaksi penjualan yang terjadi setiap harinya dicatat ke dalam buku gudang berdasarkan faktur penjualan yang diterima, dan dari sanalah petugas gudang menghitung rekapitulasi penjualan obat untuk dijadikan laporan penjualan serta menghitung jumlah obat yang keluar untuk kemudian dicocokkan dengan jumlah persediaan dalam gudang.

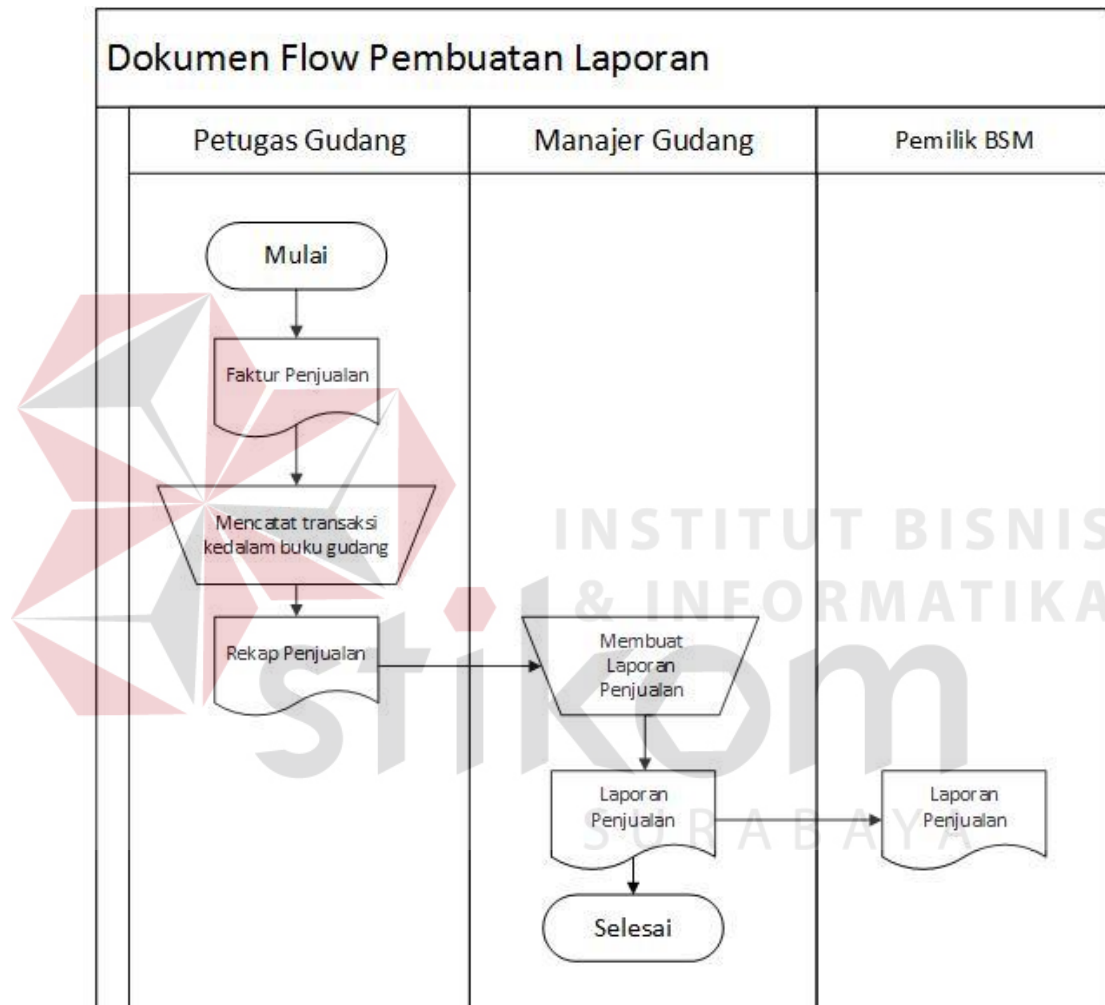
Proses transaksi penjualan yang berjalan pada gudang BSM saat ini dapat digambarkan dalam *document flow* yang ditunjukkan pada gambar 3.1.



Gambar 3.1 *Document Flow* Transaksi Penjualan.

Transaksi penjualan terjadi di gudang BSM dan dilakukan oleh dua entitas yaitu pelanggan dan petugas gudang itu sendiri. Pelanggan melakukan pemesanan obat kepada petugas gudang. Jika obat tersedia dalam gudang, maka pelanggan langsung melakukan pembayaran dan oleh petugas gudang pelanggan diberikan faktur penjualan. Jika obat tidak tersedia dalam gudang, maka petugas gudang

menawarkan obat alternatif lain kepada pelanggan, dan bila pelanggan setuju maka pelanggan melakukan pembayaran bila tidak setuju pelanggan tidak melakukan transaksi di BSM. Sedangkan proses pembuatan laporan penjualan pada BSM digambarkan *document flow* pada gambar 3.2.



Gambar 3.2 *Document Flow* Pembuatan Laporan Penjualan.

Petugas gudang mencatat seluruh transaksi yang terjadi ke dalam buku gudang berdasarkan faktur yang didapat setiap harinya. Dari buku gudang itulah petugas gudang membuat laporan penjualan yang dicocokkan dengan faktur untuk dilaporkan kepada manajer gudang.

Berdasarkan hasil pengamatan dan wawancara yang dilakukan dapat diidentifikasi beberapa permasalahan sebagai berikut:

1. Data pelanggan, data obat, dan data transaksi tidak tersimpan dalam *database* sehingga petugas gudang melakukan kesulitan mencari persediaan obat dan data pelanggan setiap harinya untuk membuat faktur atau membuat laporan penjualan.
2. Petugas gudang harus bekerja dua kali dengan mencatat ke dalam buku gudang dan menyalinnya ke dalam Microsoft Office Excel untuk dibuat laporan, sehingga tidak efektif dan membutuhkan waktu lama.
3. Seluruh data transaksi penjualan dihitung secara manual dan dicocokkan dengan buku gudang sehingga membutuhkan waktu lama dalam membuat laporan setiap bulannya.

Sistem yang baru diperlukan untuk mengatasi kesulitan petugas gudang dalam mencatat, menyimpan dan menghitung seluruh transaksi penjualan setiap harinya. Manajer gudang juga bisa dengan mudah mendapatkan laporan transaksi penjualan yang cepat dan dengan mudah mendapatkan informasi keluarannya obat serta obat yang paling diminati oleh pelanggan.

3.1.2 Analisis Kebutuhan

Dalam sub bab ini menjelaskan kebutuhan informasi yang dibutuhkan oleh pengguna berdasarkan identifikasi masalah di atas, termasuk kebutuhan fungsional maupun *non-fungsional*. Pengguna yang berhubungan langsung dengan aplikasi ini adalah administrator, kasir, dan manajer gudang. Secara general kebutuhan fungsional manajer gudang dan administrator dapat dilihat pada tabel 3.1.

Table 3.1 Kebutuhan Fungsional Berdasarkan Pengguna

Pengguna	Fungsi
Administrator	Master <i>Supplier</i> Master Pelanggan Master Kategori Master Satuan Master Obat Master Pengguna
Kasir	Transaksi Penjualan
Manajer	Stok Obat Laporan <i>History</i> Penjualan Laporan Penjualan Laporan Profit Penjualan Laporan Peringkat Penjualan

A. Administrator

Kebutuhan fungsional dan *non-fungsional* administrator yang pertama adalah Master Data *Supplier*, dijelaskan pada tabel 3.2 di bawah ini :

Tabel 3.2 Fungsi Master Data *Supplier*

Nama Fungsi	Master Data <i>Supplier</i>	
Stakeholder	Administrator	
Deskripsi	Fungsi ini bertujuan untuk mengelola master data <i>supplier</i> yang bekerja sama dengan BSM, seperti memasukkan data <i>supplier</i> baru dan mengubah data <i>supplier</i> lama.	
Kondisi Awal	<i>Login</i> sebagai administrator	
Alur Normal	Aksi Pengguna	Respon Sistem
	1. Pengguna memilih menu master data <i>supplier</i>	1. Menampilkan <i>form</i> master data <i>supplier</i>

	<p>2. Pengguna memasukkan data <i>supplier</i></p> <p>3. Pengguna menekan tombol simpan</p>	<p>2. ID <i>supplier</i> otomatis tampil</p> <p>3. Data <i>supplier</i> masuk ke dalam <i>database</i> dan menampilkan pesan berhasil disimpan</p>
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	<p>1. Pengguna mencari nama <i>supplier</i></p> <p>2. Pengguna memilih data <i>supplier</i> yang ingin diubah</p> <p>3. Pengguna memasukkan data <i>supplier</i></p> <p>4. Pengguna menekan tombol ubah</p>	<p>1. Menampilkan data <i>supplier</i></p> <p>2. Menampilkan data <i>supplier</i> terpilih</p> <p>3. Data <i>supplier</i> masuk ke dalam <i>database</i> dan menampilkan pesan berhasil diubah</p>
Kondisi Akhir	<p>1. Fungsi dapat menyimpan data <i>supplier</i> dalam <i>database</i></p> <p>2. Fungsi dapat menampilkan data <i>supplier</i></p> <p>3. Fungsi dapat mengubah data <i>supplier</i> dalam <i>database</i></p>	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	Correctness	1. Kolom telepon dan kodepos harus diisi angka

		2. Data yang dimasukkan harus lengkap diisi
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* master data pelanggan yang dijelaskan pada tabel 3.3.

Tabel 3.3 Fungsi Master Data Pelanggan

Nama Fungsi	Master Data Pelanggan
Stakeholder	Administrator
Deskripsi	Fungsi ini bertujuan untuk mengelola master data pelanggan yang melakukan transaksi di BSM, seperti memasukkan data pelanggan baru dan mengubah data pelanggan lama.
Kondisi Awal	<i>Login</i> sebagai Administrator

Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu master data pelanggan 2. Pengguna memasukkan data pelanggan 3. Pengguna menekan tombol simpan 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> master data pelanggan 2. ID Pelanggan otomatis tampil 3. Data pelanggan masuk ke dalam <i>database</i> dan menampilkan pesan berhasil disimpan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna mencari nama pelanggan 2. Pengguna memilih data pelanggan yang ingin diubah 3. Pengguna memasukkan data pelanggan 4. Pengguna menekan tombol ubah 	<ol style="list-style-type: none"> 1. Menampilkan data pelanggan 2. Menampilkan data pelanggan terpilih 3. Data pelanggan masuk ke dalam <i>database</i> dan menampilkan pesan berhasil diubah
Kondisi Akhir	<ol style="list-style-type: none"> 1. Fungsi dapat menyimpan data pelanggan dalam <i>database</i> 2. Fungsi dapat menampilkan data pelanggan 3. Fungsi dapat mengubah data pelanggan dalam <i>database</i> 	

Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	Correctness	1. Kolom telepon dan kodepos harus diisi angka 2. Data yang dimasukkan harus lengkap diisi
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* master data kategori yang dijelaskan pada tabel 3.4.

Tabel 3.4 Fungsi Master Data Kategori

Nama Fungsi	Master Data Kategori
Stakeholder	Administrator
Deskripsi	Fungsi ini bertujuan untuk mengelola master data kategori obat dalam menu master obat

Kondisi Awal	<i>Login sebagai Administrator</i>	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu master kategori 2. Pengguna memasukkan data kategori 3. Pengguna menekan tombol simpan 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> master kategori 2. ID kategori otomatis tampil 3. Data kategori masuk ke dalam <i>database</i> dan menampilkan pesan berhasil disimpan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna mencari nama kategori 2. Pengguna memilih data kategori yang ingin diubah 3. Pengguna memasukkan data kategori 4. Pengguna menekan tombol ubah 	<ol style="list-style-type: none"> 1. Menampilkan data kategori 2. Menampilkan data kategori terpilih 3. Data kategori masuk ke dalam <i>database</i> dan menampilkan pesan berhasil diubah
Kondisi Akhir	<ol style="list-style-type: none"> 1. Fungsi dapat menyimpan data kategori dalam <i>database</i> 2. Fungsi dapat menampilkan data kategori 	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator

	Correctness	-
	Interface	<ol style="list-style-type: none"> 1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* master data kategori yang dijelaskan pada tabel 3.5.

Tabel 3.5 Fungsi Master Data Satuan

Nama Fungsi	Master Data Satuan	
Stakeholder	Administrator	
Deskripsi	Fungsi ini bertujuan untuk mengelola master data satuan obat dalam menu master obat	
Kondisi Awal	<i>Login</i> sebagai Administrator	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu master satuan 2. Pengguna memasukkan data satuan 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> master satuan 2. ID satuan otomatis tampil

	3. Pengguna menekan tombol simpan	3. Data satuan masuk ke dalam <i>database</i> dan menampilkan pesan berhasil disimpan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	1. Pengguna mencari nama satuan 2. Pengguna memilih data satuan yang ingin diubah 3. Pengguna memasukkan data satuan 4. Pengguna menekan tombol ubah	1. Menampilkan data satuan 2. Menampilkan data satuan terpilih 3. Data satuan masuk ke dalam <i>database</i> dan menampilkan pesan berhasil diubah
Kondisi Akhir	1. Fungsi dapat menyimpan data satuan dalam <i>database</i> 2. Fungsi dapat menampilkan data satuan	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	Correctness	-
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang

	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik
--	-----------------	--

Selanjutnya kebutuhan fungsional dan *non-fungsional* master data obat yang dijelaskan pada tabel 3.6.

Tabel 3.6 Fungsi Master Data Obat

Nama Fungsi	Master Data Obat	
Stakeholder	Administrator	
Deskripsi	Fungsi ini bertujuan untuk mengelola master data Obat yang dijual di BSM, seperti memasukkan data obat baru dan mengubah data obat lama	
Kondisi Awal	<i>Login</i> sebagai Administrator	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu master data obat 2. Pengguna memasukkan data obat 3. Pengguna menekan tombol simpan 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> master data obat 2. ID Obat otomatis tampil 3. Data obat masuk ke dalam <i>database</i> dan menampilkan pesan berhasil disimpan
Alur Eksepsi	Aksi Pengguna	Respon Sistem

	<ol style="list-style-type: none"> 1. Pengguna mencari nama obat 2. Pengguna memilih data obat yang ingin diubah 3. Pengguna memasukkan data obat 4. Pengguna menekan tombol ubah 	<ol style="list-style-type: none"> 1. Menampilkan data obat 2. Menampilkan data obat terpilih 3. Data obat masuk ke dalam <i>database</i> dan menampilkan pesan berhasil diubah
Kondisi Akhir	<ol style="list-style-type: none"> 1. Fungsi dapat menyimpan data obat dalam <i>database</i> 2. Fungsi dapat menampilkan data obat 	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	Correctness	Data isi, harga, dan stok minimal harus diisi angka
	Interface	<ol style="list-style-type: none"> 1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* master data obat yang dijelaskan pada tabel 3.7.

Tabel 3.7 Fungsi Master Data Pengguna

Nama Fungsi	Master Data Pengguna	
Stakeholder	Administrator	
Deskripsi	Fungsi ini bertujuan untuk mengelola master data pengguna, yang akan menggunakan aplikasi	
Kondisi Awal	<i>Login</i> sebagai Administrator	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu master pengguna 2. Pengguna memasukkan data pengguna 3. Pengguna menekan tombol simpan 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> master pengguna 2. ID pengguna otomatis tampil 3. Data pengguna masuk ke dalam <i>database</i> dan menampilkan pesan berhasil disimpan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna mencari nama pengguna 2. Pengguna memilih data pengguna yang ingin diubah 	<ol style="list-style-type: none"> 1. Menampilkan data pengguna 2. Menampilkan data pengguna terpilih 3. Data pengguna masuk ke dalam <i>database</i> dan

	3. Pengguna memasukkan data pengguna 4. Pengguna menekan tombol ubah	menampilkan pesan berhasil diubah
Kondisi Akhir	1. Fungsi dapat menyimpan data pengguna dalam <i>database</i> 2. Fungsi dapat menampilkan data pengguna	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	<i>Correctness</i>	Data <i>password</i> harus disamarkan
	<i>Interface</i>	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

B. Kasir

Kebutuhan fungsional dan *non-fungsional* transaksi penjualan yang dijelaskan pada tabel 3.8.

Tabel 3.8 Transaksi Penjualan

Nama Fungsi	Transaksi Penjualan	
Stakeholder	Administrator	
Deskripsi	Fungsi ini digunakan dalam proses transaksi penjualan yang terjadi dalam gudang BSM	
Kondisi Awal	<i>Login</i> sebagai Administrator	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu transaksi penjualan 2. Pengguna memasukkan data pembeli 3. Pengguna memasukkan data transaksi penjualan 4. Pengguna menekan tombol bayar 5. Pengguna menekan tombol cetak 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> transaksi penjualan 2. Menampilkan data pembeli 3. Menampilkan data obat 4. Menghitung jumlah transaksi 5. Menghitung jumlah persediaan obat dan menyimpannya 6. Menyimpan transaksi penjualan dalam <i>database</i> 7. Menampilkan faktur penjualan 8. Mencetak faktur penjualan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Persediaan mencapai <i>Safetystock</i> 	<ol style="list-style-type: none"> 1. Menampilkan pesan obat mencapai <i>Safetystock</i>

	2. Persediaan obat tidak mencukupi	2. Menampilkan pesan persediaan obat tidak mencukupi
Kondisi Akhir	1. Fungsi dapat menghitung dan menyimpan data jumlah persediaan obat 2. Fungsi dapat menghitung dan menyimpan transaksi 3. Fungsi dapat mencetak faktur penjualan	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	<i>Correctness</i>	Jenis data yang dimasukkan untuk diproses dalam perhitungan harus berbentuk angka
	<i>Interface</i>	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

C. Manajer Gudang

Kebutuhan fungsional dan *non-fungsional* manajer gudang adalah stok obat yang dijelaskan pada tabel 3.9.

Tabel 3.9 Fungsi Stok Obat

Nama Fungsi	Stok Obat	
Stakeholder	Administrator, Manajer Gudang	
Deskripsi	Fungsi ini bertujuan untuk memasukkan jumlah obat yang baru masuk dalam gudang BSM	
Kondisi Awal	<i>Login</i> sebagai Administrator	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu stok obat 2. Pengguna memasukkan jumlah obat yang baru dibeli 3. Pengguna menekan tombol simpan 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> stok obat 2. Menampilkan jumlah obat dalam persediaan 3. Menghitung jumlah obat yang ada dalam gudang dengan jumlah obat yang baru dibeli 4. Data persediaan obat masuk ke dalam <i>database</i> dan menampilkan pesan berhasil tersimpan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	-	-

Kondisi Akhir	1. Fungsi dapat menyimpan data jumlah persediaan obat 2. Fungsi dapat menghitung jumlah persediaan obat	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai Administrator
	Correctness	Jenis data yang dimasukkan untuk diproses dalam perhitungan harus berbentuk angka
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu memasukkan data dalam <i>database</i> kemudian menampilkannya dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* laporan *history* penjualan, dijelaskan pada tabel 3.10.

Tabel 3.10 Fungsi Laporan *History* Penjualan

Nama Fungsi	Laporan <i>history</i> Penjualan
Stakeholder	Manajer Gudang

Deskripsi	Fungsi ini digunakan untuk menampilkan dan mencetak laporan <i>history</i> penjualan	
Kondisi Awal	<i>Login</i> sebagai manajer gudang	
Alur Normal	Aksi Pengguna	Respon Sistem
	1. Pengguna memilih menu <i>history</i> penjualan	1. Menampilkan <i>form history</i> penjualan
	2. Pengguna memilih nama pelanggan	2. Menampilkan transaksi penjualan sesuai nama pelanggan
	3. Pengguna menekan tombol tampil 4. Pengguna menekan tombol <i>export</i>	3. <i>Export</i> laporan ke Microsoft excel
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	Pengguna memilih periode transaksi	Menampilkan laporan <i>history</i> sesuai periode transaksi penjualan
Kondisi Akhir	1. Fungsi dapat menampilkan laporan <i>history</i> transaksi penjualan 2. Fungsi dapat mencetak laporan <i>history</i> transaksi penjualan	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai manajer gudang
	Correctness	1. Periode laporan harus diisi

		2. Nama pelanggan harus diisi
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu menampilkan laporan dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* laporan penjualan yang dijelaskan pada tabel 3.11.

Tabel 3.11 Fungsi Laporan Penjualan

Nama Fungsi	Laporan Penjualan	
Stakeholder	Manajer Gudang	
Deskripsi	Fungsi ini digunakan untuk menampilkan dan mencetak laporan penjualan sesuai periode yang dimasukkan	
Kondisi Awal	<i>Login</i> sebagai manajer gudang	
Alur Normal	Aksi Pengguna	Respon Sistem
	1. Pengguna memilih menu laporan penjualan 2. Pengguna memilih periode harian	1. Menampilkan <i>form</i> laporan penjualan 2. Menampilkan laporan penjualan sesuai periode bulan yang dimasukkan 3. Mencetak laporan penjualan

	<p>3. Pengguna memasukkan periode harian</p> <p>4. Pengguna menekan tombol tampil</p> <p>5. Pengguna menekan tombol cetak</p>	
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	<p>1. Pengguna memilih periode mingguan</p> <p>2. Pengguna memasukkan periode mingguan</p> <p>3. Pengguna memilih periode bulanan</p> <p>4. Pengguna memasukkan periode bulanan</p>	<p>1. Menampilkan laporan penjualan per Minggu sesuai periode yang dimasukkan</p> <p>2. Menampilkan laporan penjualan per bulan sesuai periode yang dimasukkan</p>
Kondisi Akhir	<p>1. Fungsi dapat menghitung jumlah penjualan dan pendapatan sesuai periode</p> <p>2. Fungsi dapat menampilkan laporan penjualan sesuai periode yang dimasukkan</p> <p>3. Fungsi dapat mencetak laporan penjualan</p>	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai manajer gudang

	Correctness	Periode laporan harus dipilih dan diisi
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu menampilkan laporan dalam waktu maksimal 5 detik

Selanjutnya kebutuhan fungsional dan *non-fungsional* laporan profit penjualan yang dijelaskan pada tabel 3.12.

Tabel 3.12 Fungsi Laporan Profit Penjualan

Nama Fungsi	Laporan Profit Penjualan	
Stakeholder	Manajer Gudang	
Deskripsi	Fungsi ini digunakan untuk menampilkan dan mencetak laporan profit penjualan	
Kondisi Awal	<i>Login</i> sebagai manajer gudang	
Alur Normal	Aksi Pengguna	Respon Sistem
	1. Pengguna memilih menu laporan profit penjualan 2. Pengguna memasukkan periode laporan	1. Menampilkan <i>form</i> laporan profit penjualan 2. Menghitung jumlah profit penjualan

	3. Pengguna menekan tombol tampil 4. Pengguna menekan tombol cetak	3. Menampilkan laporan profit penjualan sesuai periode 4. Mencetak laporan profit penjualan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	-	-
Kondisi Akhir	1. Fungsi dapat menghitung jumlah profit penjualan 2. Fungsi dapat menampilkan laporan profit penjualan 3. Fungsi dapat mencetak laporan profit penjualan	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai manajer gudang
	Correctness	Periode laporan harus diisi
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu menampilkan laporan dalam waktu maksimal 5 detik

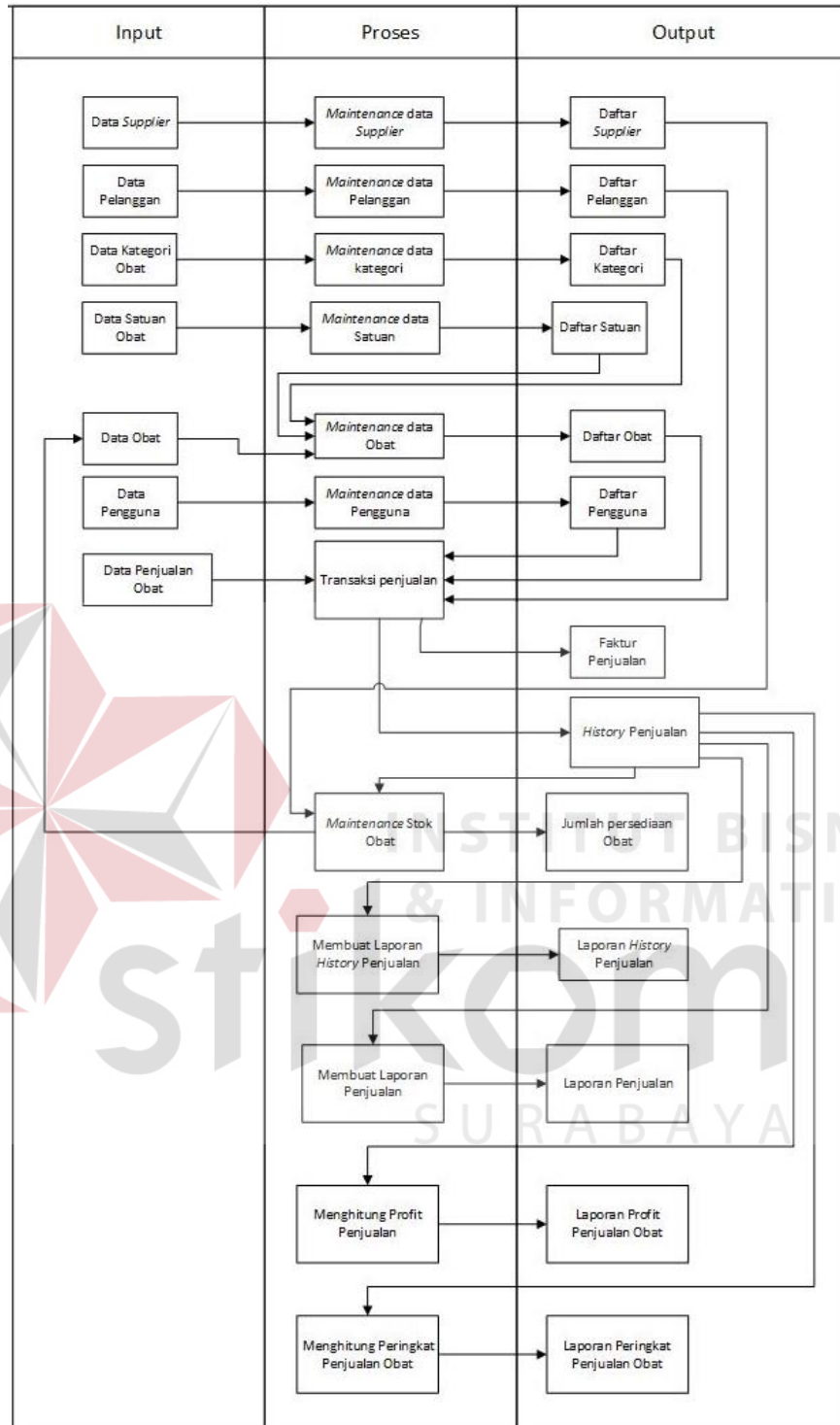
Selanjutnya kebutuhan fungsional dan *non-fungsional* laporan peringkat penjualan yang dijelaskan pada tabel 3.13.

Tabel 3.13 Fungsi Laporan Peringkat penjualan

Nama Fungsi	Laporan Peringkat penjualan	
Stakeholder	Manajer Gudang	
Deskripsi	Fungsi ini digunakan untuk membuat laporan peringkat penjualan berdasarkan profit penjualan atau profit penjualan terbanyak	
Kondisi Awal	<i>Login</i> sebagai manajer gudang	
Alur Normal	Aksi Pengguna	Respon Sistem
	<ol style="list-style-type: none"> 1. Pengguna memilih menu laporan peringkat penjualan 2. Pengguna memilih peringkat berdasarkan pendapatan 3. Pengguna memasukkan periode laporan 4. Pengguna menekan tombol tampil 5. Pengguna menekan tombol cetak 	<ol style="list-style-type: none"> 1. Menampilkan <i>form</i> laporan peringkat penjualan 2. Menghitung peringkat penjualan 3. Menampilkan laporan peringkat pendapatan penjualan sesuai periode 4. Mencetak laporan peringkat penjualan
Alur Eksepsi	Aksi Pengguna	Respon Sistem
	Pengguna memilih peringkat berdasarkan profit penjualan	Menampilkan laporan peringkat profit penjualan sesuai periode

Kondisi Akhir	1. Fungsi dapat menghitung peringkat penjualan obat 2. Fungsi dapat menampilkan laporan peringkat penjualan 3. Fungsi dapat mencetak laporan peringkat penjualan	
Kebutuhan Non-Fungsional	Keamanan	Hanya dapat diakses oleh orang yang <i>login</i> sebagai manajer gudang
	Correctness	Periode laporan harus diisi
	Interface	1. Menu tersedia dalam bahasa 2. Ukuran huruf besar dan jelas 3. Warna tampilan tidak terlalu terang
	Performa	Mampu menampilkan laporan dalam waktu maksimal 5 detik

Setelah didapatkan kebutuhan informasi berdasarkan pengguna, selanjutnya berdasarkan identifikasi masalah di atas juga didapatkan beberapa kebutuhan sistem, yang selanjutnya digunakan dalam perangkat lunak (*software*). Kebutuhan sistem tersebut secara umum digambarkan dalam diagram blok. Secara umum blok diagram aplikasi penjualan obat hewan ternak dikelompokkan menjadi tiga bagian yaitu *input*, proses, *output*. Dalam blok diagram digambarkan secara umum proses yang ada dalam aplikasi, serta masukan yang dibutuhkan oleh proses, dan *output* yang dihasilkan oleh proses tersebut. Lebih jelasnya blok diagram aplikasi penjualan obat hewan ternak pada BSM dapat dilihat pada gambar 3.3.



Gambar 3.3 Blok Diagram Aplikasi Penjualan Obat Hewan Ternak pada BSM

Dalam kelompok *input* terdapat data *supplier*, data pelanggan, data pengguna, data kategori, data satuan, data obat, dan data penjualan obat. Data

pelanggan berisi data pribadi pelanggan yang melakukan transaksi di BSM seperti nama, alamat, dan lain-lain. Data pengguna berisi data petugas gudang BSM yang memiliki akses terhadap aplikasi. Data satuan dan data kategori merupakan masukan yang diperlukan untuk mengolah data obat selanjutnya. Data Obat berisi obat yang dijual di BSM beserta harga dan jenis obat tersebut. Data penjualan obat dimasukkan saat pelanggan melakukan transaksi di gudang BSM berdasarkan daftar obat, dan daftar pelanggan yang telah tersimpan di *database*.

Dalam kelompok proses terdapat proses *Maintenance* data *supplier* yang menghasilkan daftar *supplier*. *Maintenance* data pelanggan yang menghasilkan daftar pelanggan. *Maintenance* data kategori dan *maintenance* data satuan yang menghasilkan daftar kategori dan daftar satuan untuk menjadi masukan di proses *maintenance* data obat. *Maintenance* data obat yang menghasilkan daftar obat. Proses *Maintenance* data pengguna yang menghasilkan daftar pengguna yang memiliki akses terhadap aplikasi. Proses transaksi penjualan dilakukan saat pelanggan melakukan transaksi dengan petugas gudang BSM, masukan yang dibutuhkan dalam proses ini adalah data penjualan obat, data obat, data pelanggan yang membeli dari *database*, dan data pengguna yang melayani transaksi penjualan tersebut. Keluaran yang dihasilkan oleh proses ini adalah faktur penjualan yang dan *history* transaksi penjualan yang disimpan dalam *database*.

Kemudian terdapat proses menghitung jumlah persediaan obat yang menghasilkan jumlah persediaan obat terkini yang digunakan dalam transaksi penjualan dan diserahkan pada manajer gudang untuk mengetahui saatnya melakukan pembelian obat pada produsen. Proses membuat laporan yang dilakukan untuk menghasilkan laporan penjualan yang nantinya akan disajikan per hari, per

minggu, dan per bulan. Masukan yang dibutuhkan adalah *history* transaksi penjualan. Proses menghitung profit penjualan untuk menghasilkan laporan profit penjualan selama sebulan membutuhkan masukan *history* transaksi penjualan. Terakhir proses menghitung peringkat penjualan obat menghasilkan keluaran laporan peringkat penjualan obat selama sebulan. Proses ini membutuhkan masukan *history* transaksi penjualan. Seluruh laporan di atas akan disampaikan kepada manajer gudang.

3.2 Perancangan Sistem

Berdasarkan dari analisis permasalahan yang ada, tahap berikutnya dari siklus pengembangan sistem adalah perancangan sistem. Tujuan dari desain sistem ini adalah membuat kerangka dasar dalam melakukan implementasi ke aplikasi yang dibuat. Pada tahap ini terdapat aktivitas pendefinisian kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun hingga implementasi dari sistem. Pada tahap ini akan membahas tentang perancangan sistem yang meliputi *System Flow* yang menunjukkan alur sistem yang akan dibuat, *Data Flow Diagram* (DFD) yang merupakan diagram aliran data pada sistem yang akan dibuat, dan *Entity Relationship Diagram* (ERD) yang menggambarkan diagram relasi antar tabel-tabel. Pada ERD terdapat *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) yang menggambarkan tabel-tabel dan relasinya secara detil, juga dilengkapi dengan perancangan *interface* yang menggambarkan tampilan *interface* program yang akan dibuat.

3.2.1 *System Flow*

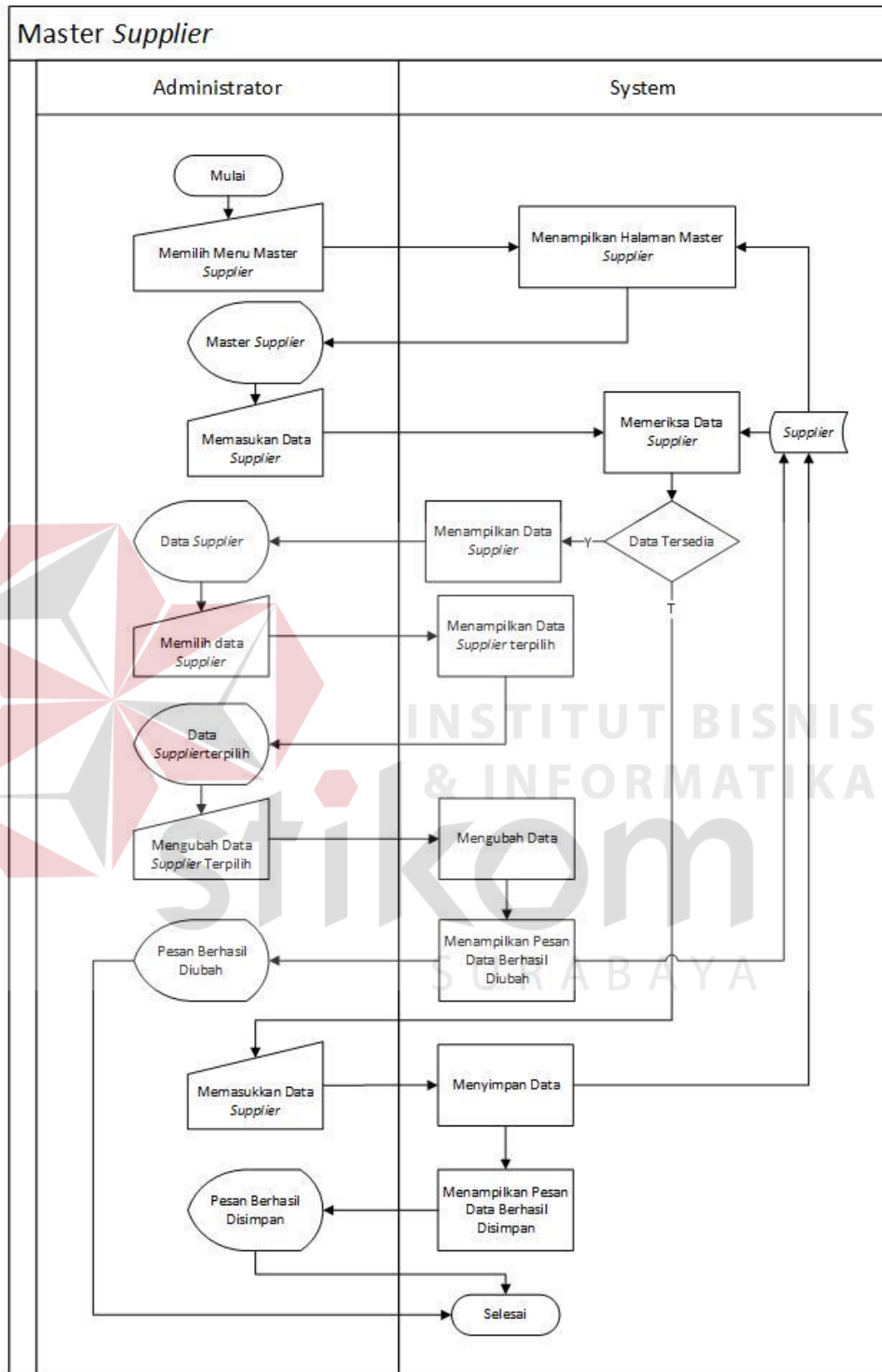
System flow adalah bagan yang menunjukkan arus pekerjaan secara menyeluruh dari suatu sistem di mana bagan ini menjelaskan urutan prosedur-prosedur yang ada di dalam sistem (Jogiyanto, 1990). Dalam *system flow* juga digambarkan fungsi-fungsi yang melaksanakan atau bertanggung jawab terhadap sub-sub sistem. Proses - proses yang akan digambarkan dalam *System Flow* adalah sebagai berikut :

A. *Maintenance Data*

Pada proses ini administrator memasukkan data pelanggan, data obat, data pegawai, data pengguna, data kategori, dan data satuan yang akan digunakan sebagai masukan dalam proses-proses selanjutnya. Proses *maintenance* data terdiri dari master data *supplier*, master data pelanggan, master data obat, stok obat.

A.1 Master Data *Supplier*

Administrator memilih menu master data *supplier* kemudian sistem akan menampilkan *form* master data *supplier*, kemudian administrator memasukkan nama *supplier* ke dalam sistem. Sistem akan mengecek apakah data yang di masukan oleh administrator sudah ada atau belum, bila data sudah ada maka sistem menampilkan data tersebut, bila data belum ada maka administrator akan memasukkan data *supplier* baru untuk kemudian disimpan oleh sistem ke dalam *database supplier*. Data *supplier* yang sudah tersimpan dapat di ubah oleh administrator dengan memilih data *supplier* yang akan diubah, kemudian data akan ditampilkan oleh sistem untuk dapat diubah oleh administrator dan disimpan ke dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.4.



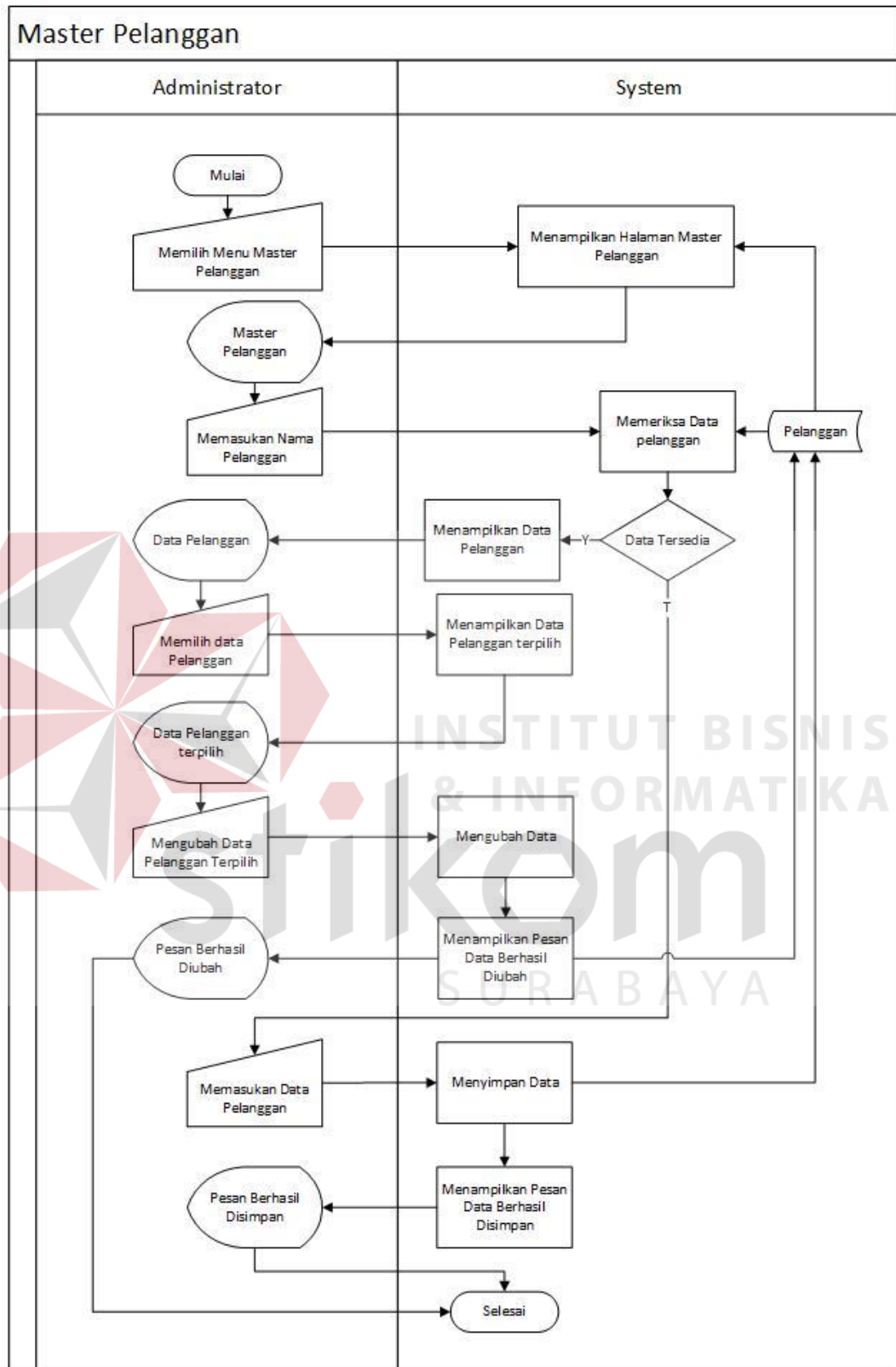
Gambar 3.4 System Flow Master Data Supplier.

A.2 Master Data Pelanggan

Administrator memilih menu master data pelanggan kemudian sistem akan menampilkan *form* master data pelanggan, kemudian administrator memasukkan nama pelanggan ke dalam sistem. Sistem akan mengecek apakah data yang di masukan oleh administrator sudah ada atau belum, bila data sudah ada maka sistem menampilkan data tersebut, bila data belum ada maka administrator akan memasukkan data pelanggan baru untuk kemudian disimpan oleh sistem ke dalam *database* pelanggan. Data pelanggan yang sudah tersimpan dapat di ubah oleh administrator dengan memilih data pelanggan yang akan diubah, kemudian data akan ditampilkan oleh sistem untuk dapat diubah oleh administrator dan disimpan ke dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.5.

A.3 Master Data Kategori

Proses ini diawali dengan administrator memilih menu master data kategori kemudian sistem akan menampilkan *form* master data kategori, kemudian administrator memasukkan nama kategori ke dalam sistem. Sistem akan mengecek apakah data yang di masukan oleh administrator sudah ada atau belum, bila data sudah ada maka sistem menampilkan data tersebut, bila data belum ada maka administrator akan memasukkan data kategori baru untuk kemudian disimpan oleh sistem ke dalam *database* kategori. Data kategori yang sudah tersimpan dapat di ubah oleh administrator dengan memilih data kategori yang akan diubah, kemudian data akan ditampilkan oleh sistem untuk dapat diubah oleh administrator dan disimpan ke dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.6.



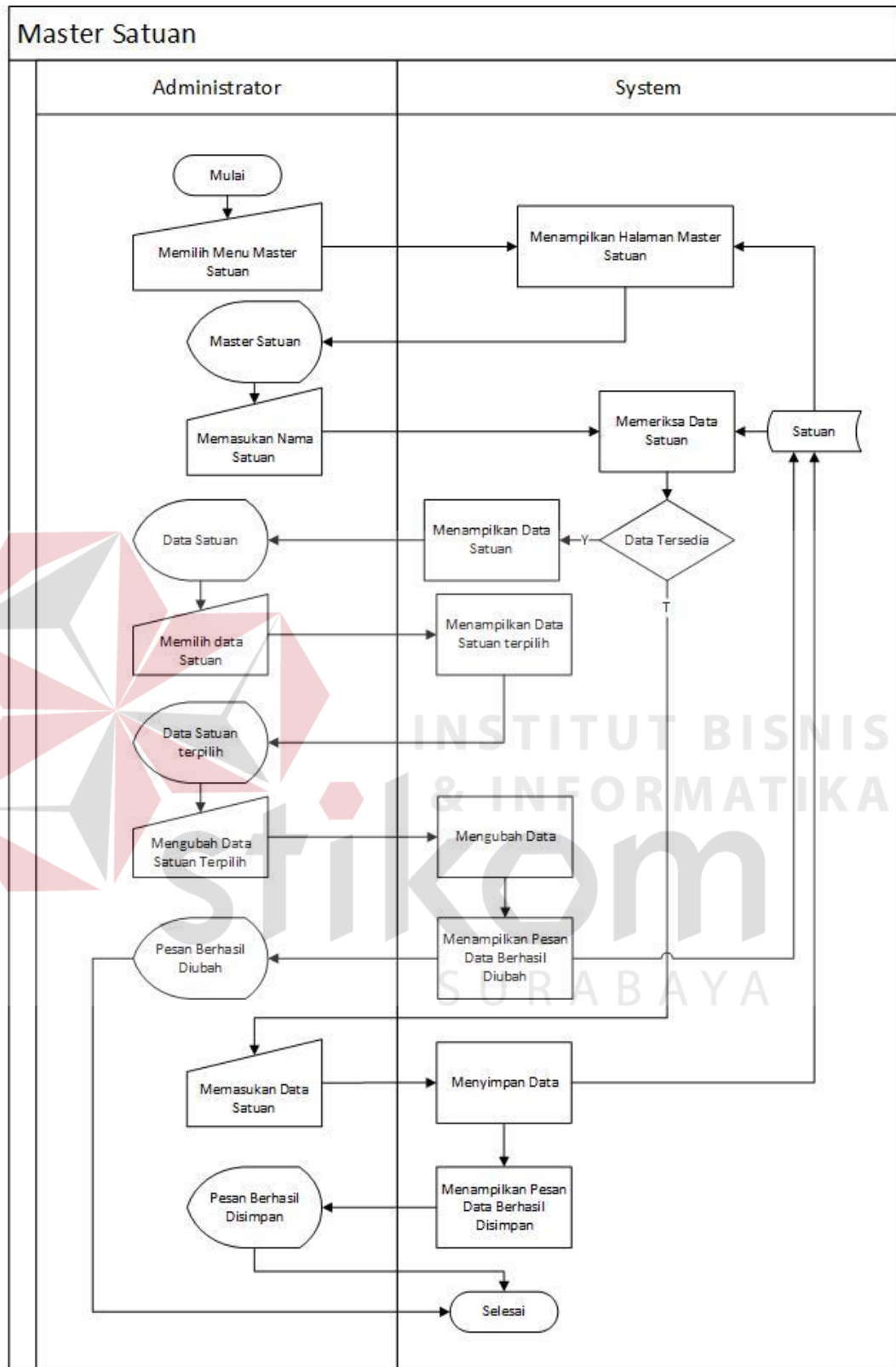
Gambar 3.5 System Flow Master Data Pelanggan.

A.4 Master Data Satuan

Proses ini diawali dengan administrator memilih menu master data satuan kemudian sistem akan menampilkan *form* master data satuan, kemudian administrator memasukkan nama satuan ke dalam sistem. Sistem akan mengecek apakah data yang di masukan oleh administrator sudah ada atau belum, bila data sudah ada maka sistem menampilkan data tersebut, bila data belum ada maka administrator akan memasukkan data satuan baru untuk kemudian disimpan oleh sistem ke dalam *database* satuan. Data satuan yang sudah tersimpan dapat di ubah oleh administrator dengan memilih data pengguna yang akan diubah, kemudian data akan ditampilkan oleh sistem untuk dapat diubah oleh administrator dan disimpan ke dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.7.

A.5 Master Data Obat

Proses ini diawali dengan administrator memilih menu master data obat kemudian sistem akan menampilkan *form* master data obat, kemudian administrator memasukkan nama obat ke dalam sistem. Sistem akan mengecek apakah data yang di masukan oleh administrator sudah ada atau belum, bila data sudah ada maka sistem menampilkan data tersebut, bila data belum ada maka administrator akan memasukkan data obat baru untuk kemudian disimpan oleh sistem ke dalam *database*. Data obat yang sudah tersimpan dapat di ubah oleh administrator dengan memilih data obat yang akan diubah, kemudian data akan ditampilkan oleh sistem untuk dapat diubah oleh administrator dan disimpan ke dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.8.

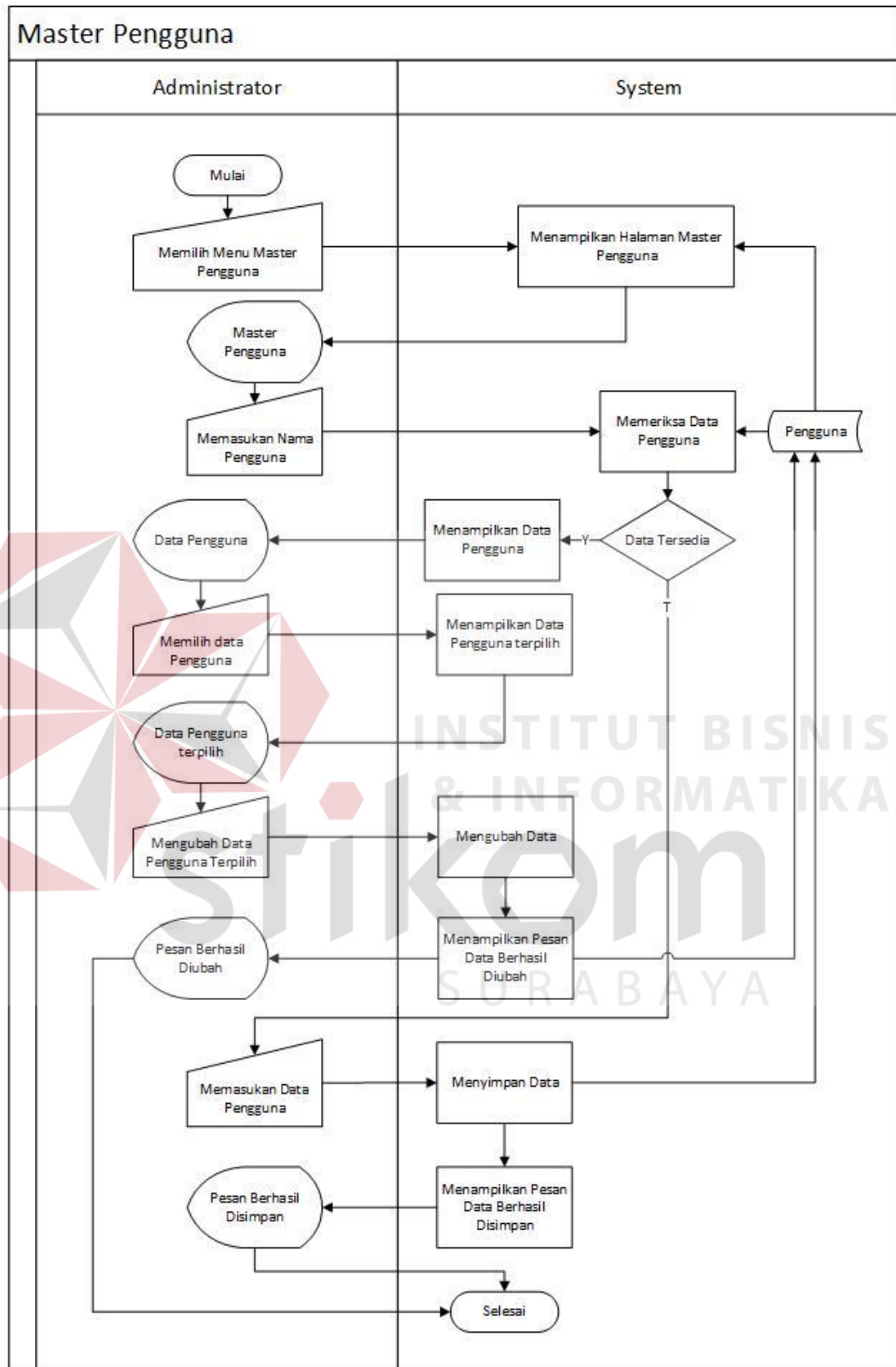


Gambar 3.7 System Flow Master Data Satuan

A.6 Master Data Pengguna

Tujuan dari proses ini adalah menyimpan dan *update* data pengguna aplikasi penjualan obat hewan ternak. Pengguna aplikasi ini digolongkan menjadi tiga grup, yaitu grup administrator, grup kasir, dan grup manajer. Proses master data pengguna ini meliputi menggolongkan pengguna berdasarkan tiga grup di atas, sehingga masing-masing pengguna dapat mengakses aplikasi sesuai fungsi grupnya. Bagi pengguna yang tidak terdaftar dalam *database* tidak dapat mengakses aplikasi.

Proses ini diawali dengan administrator memilih menu master data pengguna kemudian sistem akan menampilkan *form* master data pengguna. Selanjutnya administrator memasukkan nama pengguna ke dalam sistem. Sistem akan mengecek apakah data yang di masukan oleh administrator sudah ada atau belum, bila data sudah ada maka sistem menampilkan data tersebut, bila data belum ada maka administrator akan memasukkan data pengguna baru untuk kemudian disimpan oleh sistem ke dalam *database* pengguna. Data pengguna yang sudah tersimpan dapat di ubah oleh administrator dengan memilih data pengguna yang akan diubah, kemudian data akan ditampilkan oleh sistem untuk dapat diubah oleh administrator dan disimpan ke dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.9.



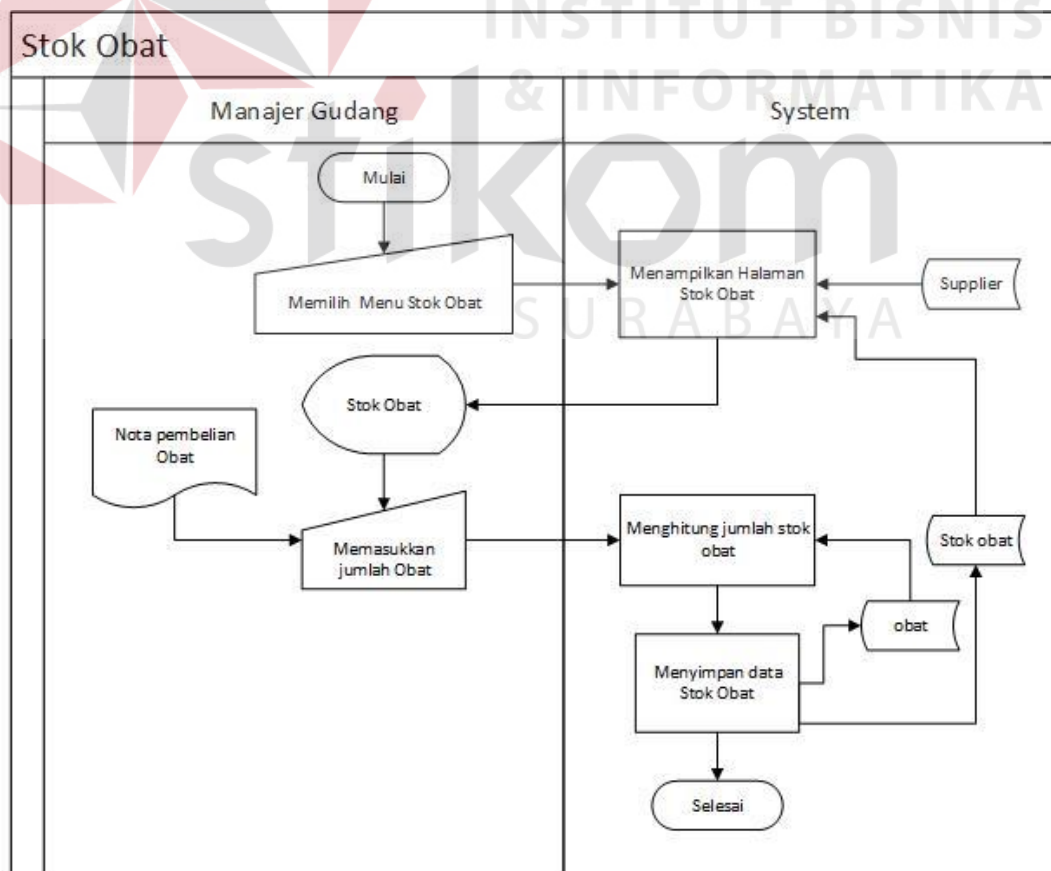
Gambar 3.9 System Flow Master Data Pengguna

B. Transaksi

Proses transaksi pada sistem ini terdiri dari dua transaksi yaitu transaksi stok obat dan transaksi penjualan.

B.1 Stok Obat

Tujuan dari proses ini adalah dapat menyimpan dan *update* data persediaan obat yang terdapat dalam gudang BSM. Administrator memilih menu stok obat kemudian sistem akan menampilkan *form* stok obat, kemudian administrator memasukan data jumlah obat yang baru dibeli. Sistem akan menjumlah data yang baru dimasukkan dengan data pada tabel obat yang ada untuk kemudian disimpan dalam *database*. Proses digambarkan dalam *system flow* pada gambar 3.10.



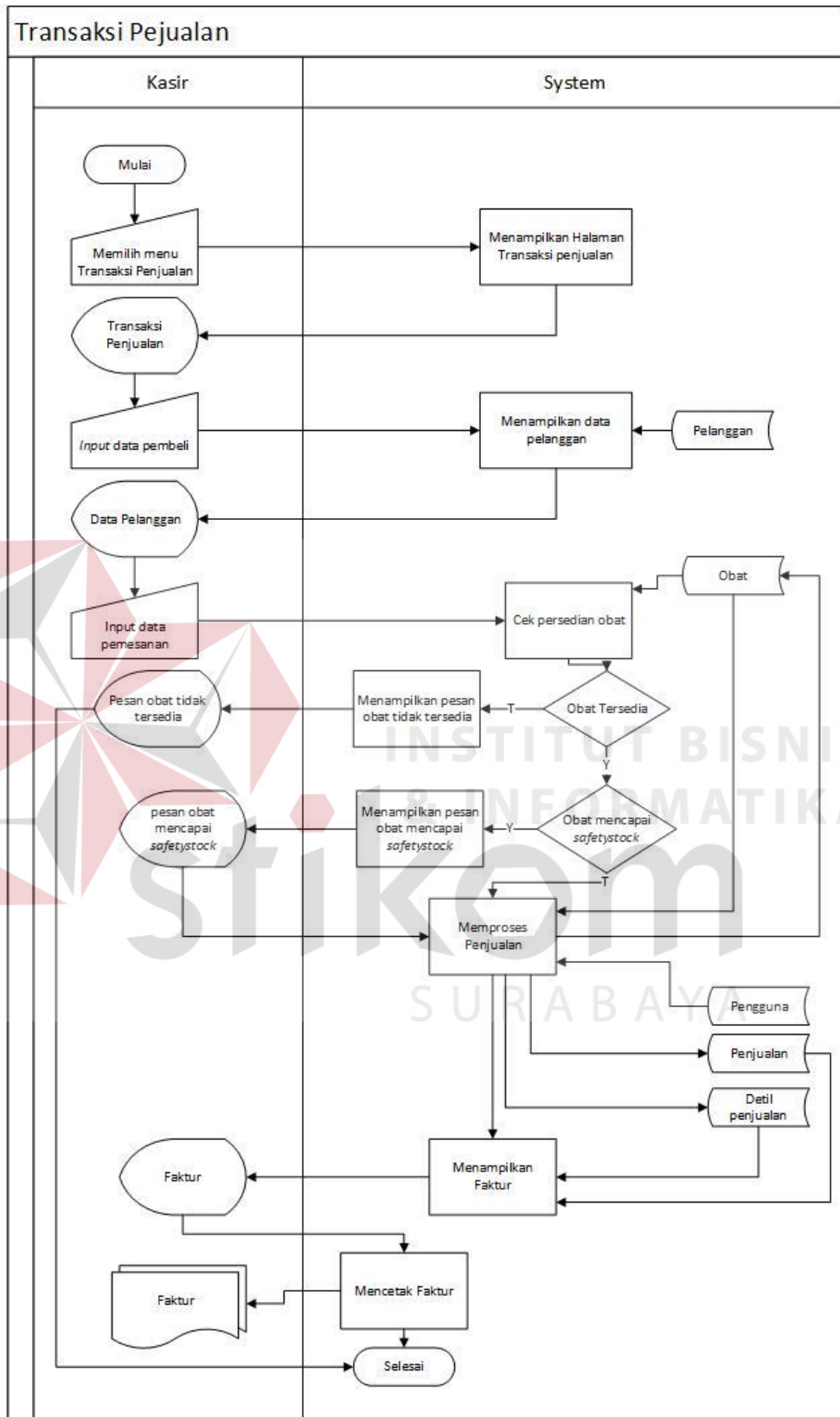
Gambar 3.10 System Flow Stok Obat.

B.2 Transaksi Penjualan

Pada proses ini kasir melayani kegiatan transaksi penjualan yang terjadi di BSM. Proses ini bertujuan menghitung transaksi yang terjadi dan menyimpannya dalam *database* untuk kemudian digunakan dalam pembuatan laporan. Proses ini dimulai dengan kasir memilih menu transaksi penjualan, kemudian sistem menampilkan *form* transaksi penjualan.

Kasir memasukkan data pelanggan terlebih dulu yang telah tersimpan di tabel pelanggan dan selanjutnya petugas memasukkan permintaan obat pelanggan ke dalam sistem, kemudian sistem akan memeriksa apakah persediaan obat yang diminta masih tersedia atau tidak yang tersimpan di tabel stok obat dan obat.

Bila persediaan tidak mencukupi, maka sistem akan menampilkan pesan obat tidak tersedia dan tidak melanjutkan ke proses selanjutnya, bila persediaan mencukupi maka sistem akan melanjutkan ke proses transaksi penjualan dan mencetak faktur penjualan. Bila jumlah persediaan mencapai *safetystock* maka akan ditampilkan pesan obat mencapai *safetystock* sehingga manajer bisa melakukan pembelian obat. Setelah menampilkan pesan obat mencapai *safetystock*, proses akan dilanjutkan ke transaksi penjualan. Jumlah persediaan obat dalam tabel obat juga akan langsung di *update* dan disimpan. Faktur penjualan dicetak rangkap 3 untuk diberikan pada pelanggan, petugas gudang, dan bagian akuntansi. Lebih jelasnya proses digambarkan dalam *system flow* pada gambar 3.11.



Gambar 3.11 System Flow Transaksi Pejualan.

C. Pembuatan laporan

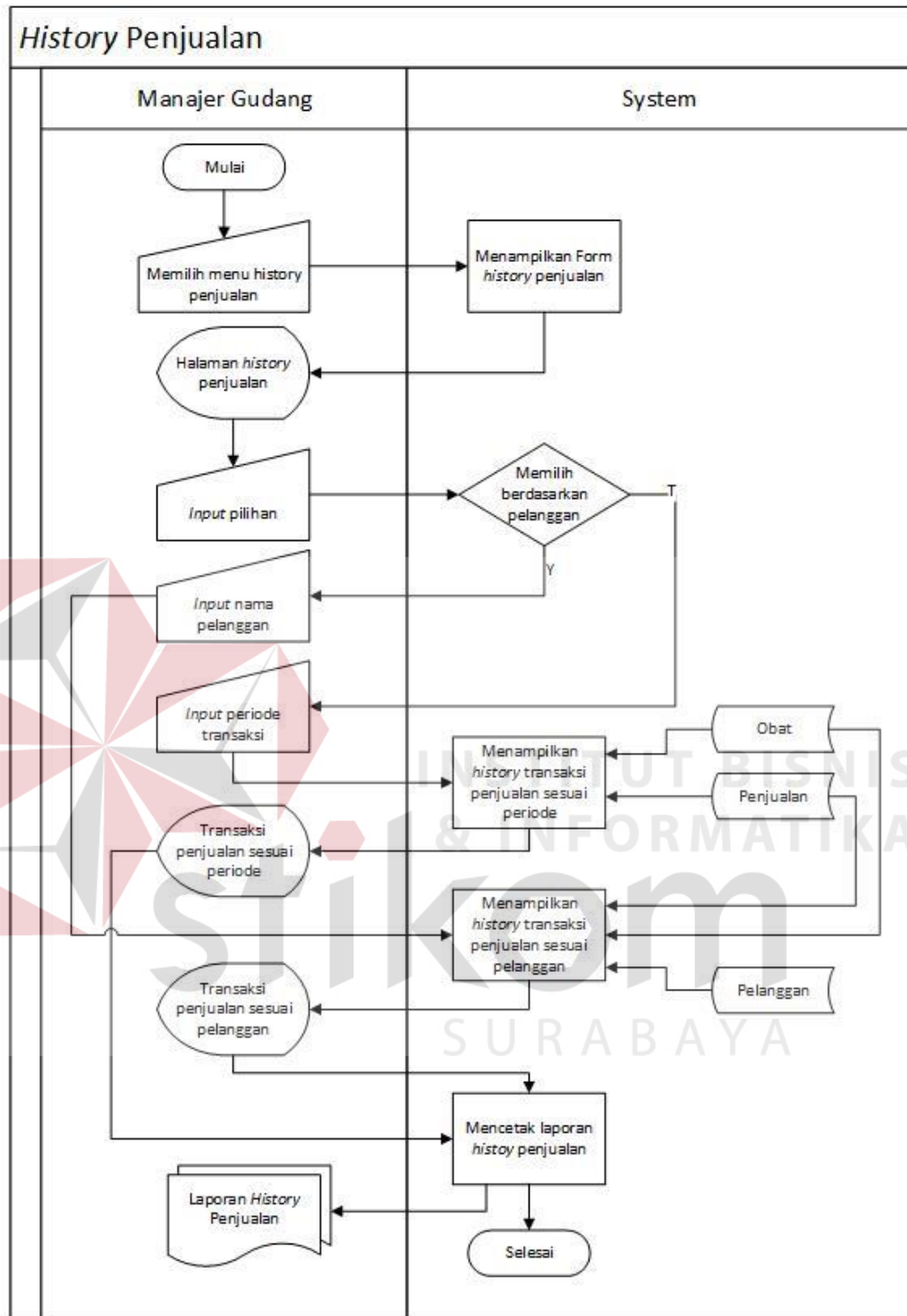
Proses pembuatan laporan pada sistem ini terdiri dari empat laporan yaitu laporan *history* penjualan, laporan penjualan, laporan profit penjualan, dan laporan peringkat penjualan obat.

C.1 Laporan *history* penjualan

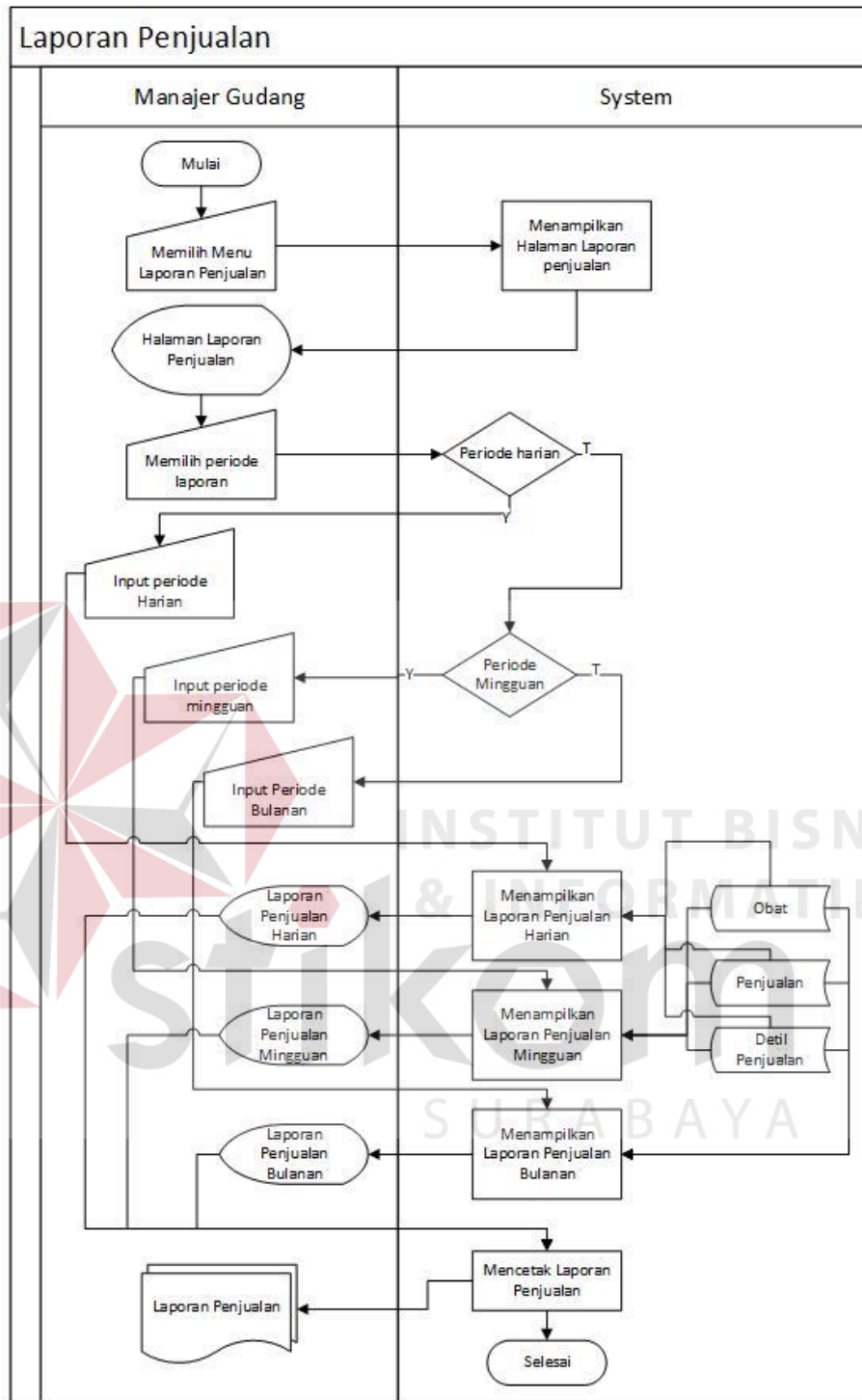
Proses ini bertujuan menampilkan laporan *history* penjualan obat selama periode yang di masukan dalam sistem atau sesuai nama pelanggan yang dimasukkan ke dalam sistem oleh manajer gudang. Manajer gudang memasukkan periode penjualan atau nama pelanggan yang akan dilaporkan ke dalam sistem. Kemudian sistem akan menampilkan laporan *history* penjualan sesuai masukan yang telah dimasukkan. Proses ini membutuhkan data yang diambil dari tabel obat, pelanggan, detail penjualan dan penjualan yang tersimpan. Selanjutnya laporan *history* penjualan dapat di *export* ke Microsoft excel. Proses ini secara umum digambarkan pada gambar 3.12.

C.2 Laporan Penjualan

Proses ini bertujuan menampilkan laporan penjualan obat selama periode yang di masukan ke dalam sistem oleh manajer gudang. Manajer gudang memilih jenis periode yang akan ditampilkan dalam laporan, kemudian memasukkan periode penjualan yang akan dilaporkan ke dalam sistem. Sistem akan menampilkan laporan penjualan sesuai masukan yang telah dimasukkan. Proses ini membutuhkan data yang diambil dari tabel obat, detail penjualan dan penjualan yang tersimpan. Selanjutnya laporan penjualan dapat dicetak. Proses ini secara umum digambarkan pada gambar 3.13.



Gambar 3.12 *System Flow* Laporan *History* Penjualan

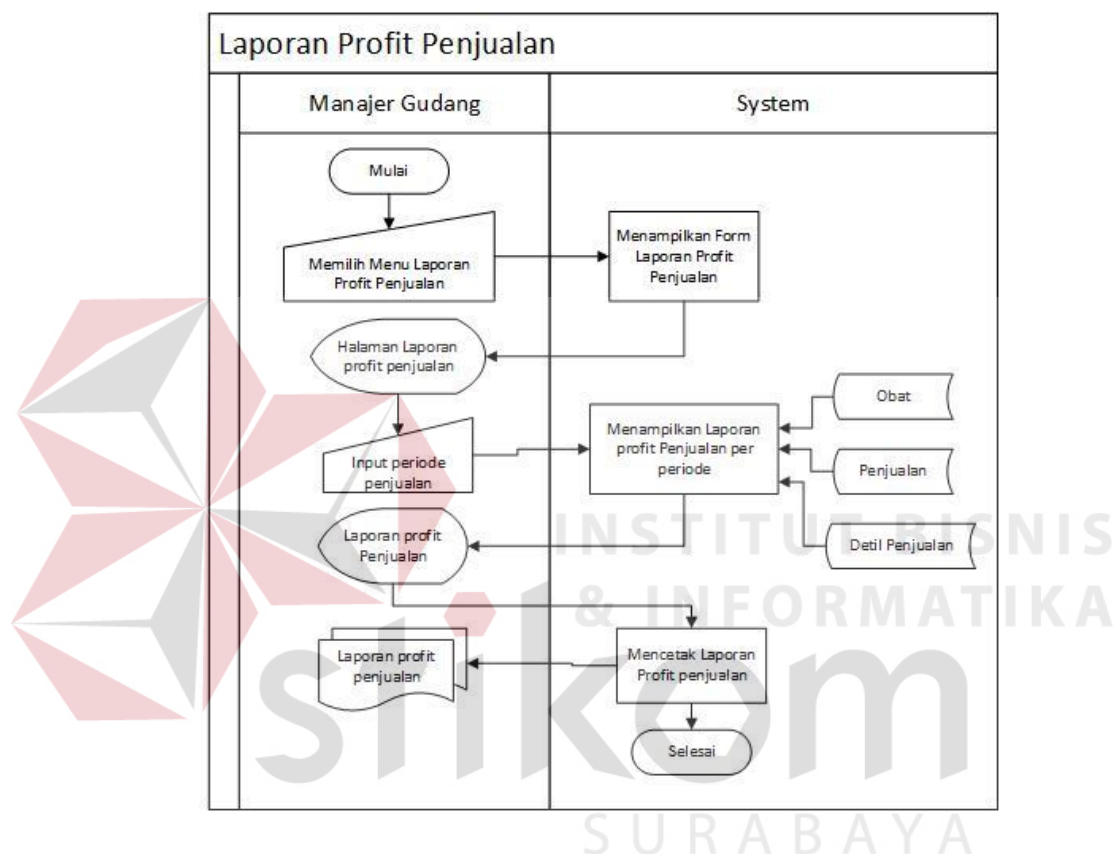


Gambar 3.13 *System Flow* Laporan Penjualan

C.3. Laporan Profit Penjualan

Proses ini dimulai dengan manajer gudang memasukan periode penjualan dan jenis obat yang akan dilaporkan ke dalam sistem. Kemudian sistem akan

menghitung profit dari hasil penjualan obat selama periode tersebut dan menampilkan laporan profit. Proses ini membutuhkan data yang diambil dari tabel obat, detail penjualan dan penjualan yang tersimpan. Selanjutnya laporan profit dapat dicetak. Proses ini secara umum digambarkan pada gambar 3.14.

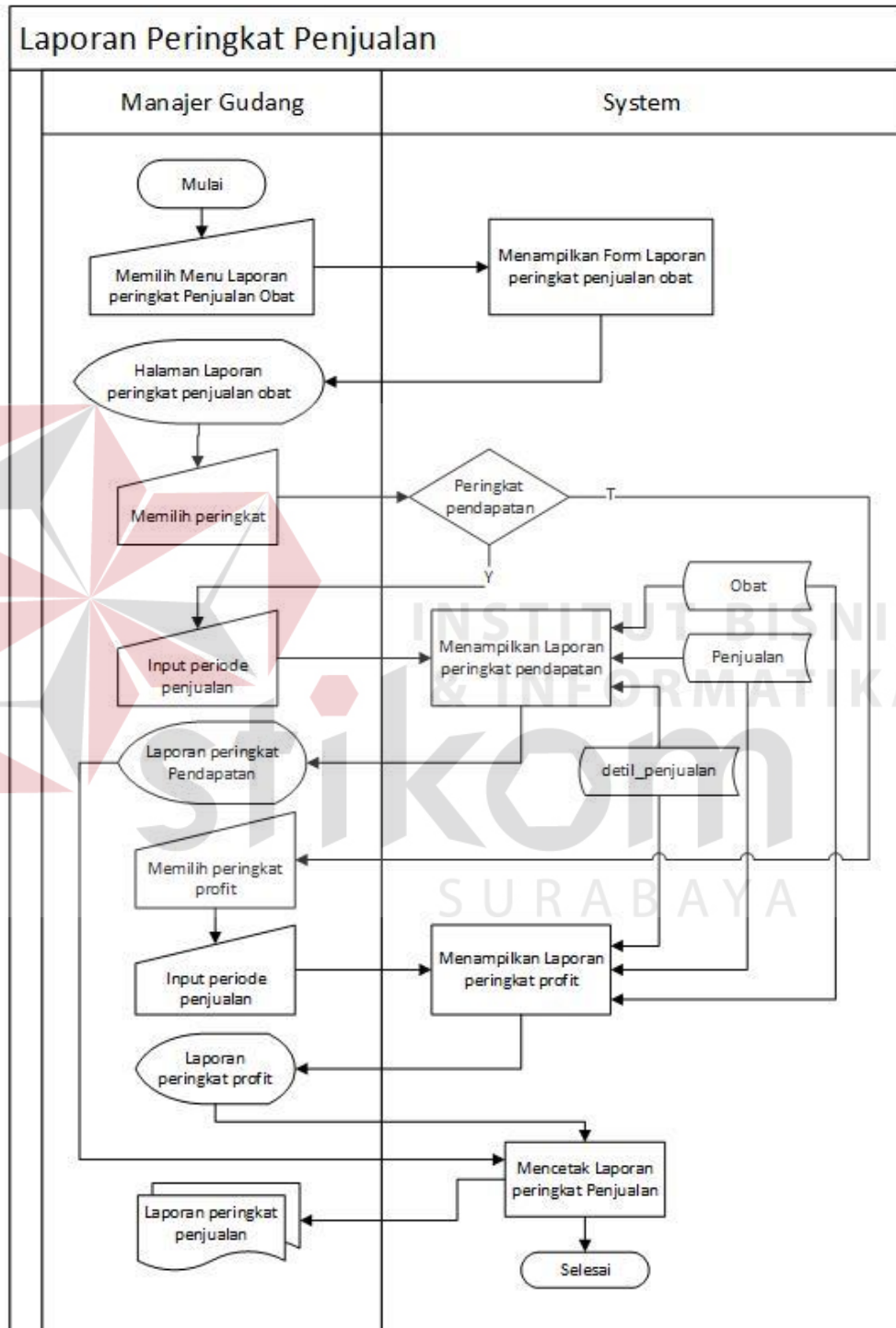


Gambar 3.14 *System Flow* Laporan Profit Penjualan

B.4. Laporan Peringkat Penjualan Obat

Peringkat penjualan ditampilkan berdasarkan dua hal yaitu peringkat profit penjualan dan peringkat pendapatan penjualan. Manajer gudang memilih peringkat yang akan ditampilkan kemudian memasukkan periode penjualan. Kemudian sistem akan menampilkan peringkat penjualan obat dari transaksi yang telah terjadi selama periode tersebut. Proses ini membutuhkan data yang diambil dari database obat dan penjualan yang tersimpan. Selanjutnya laporan

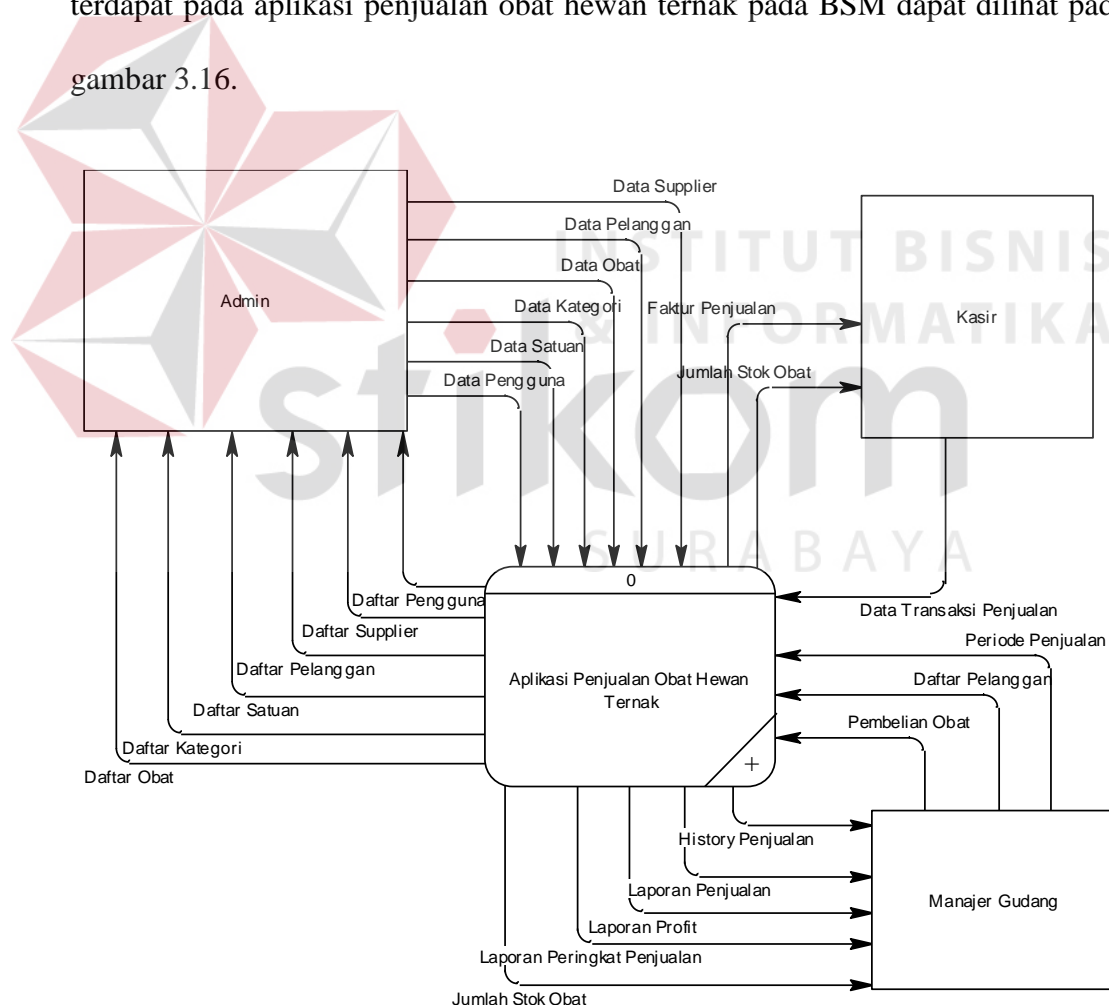
peringkat penjualan dapat dicetak. Lebih jelasnya dapat dilihat pada gambar 3.15.



Gambar 3.15 *System Flow* Laporan Peringkat Penjualan Obat

3.2.2 Context Diagram

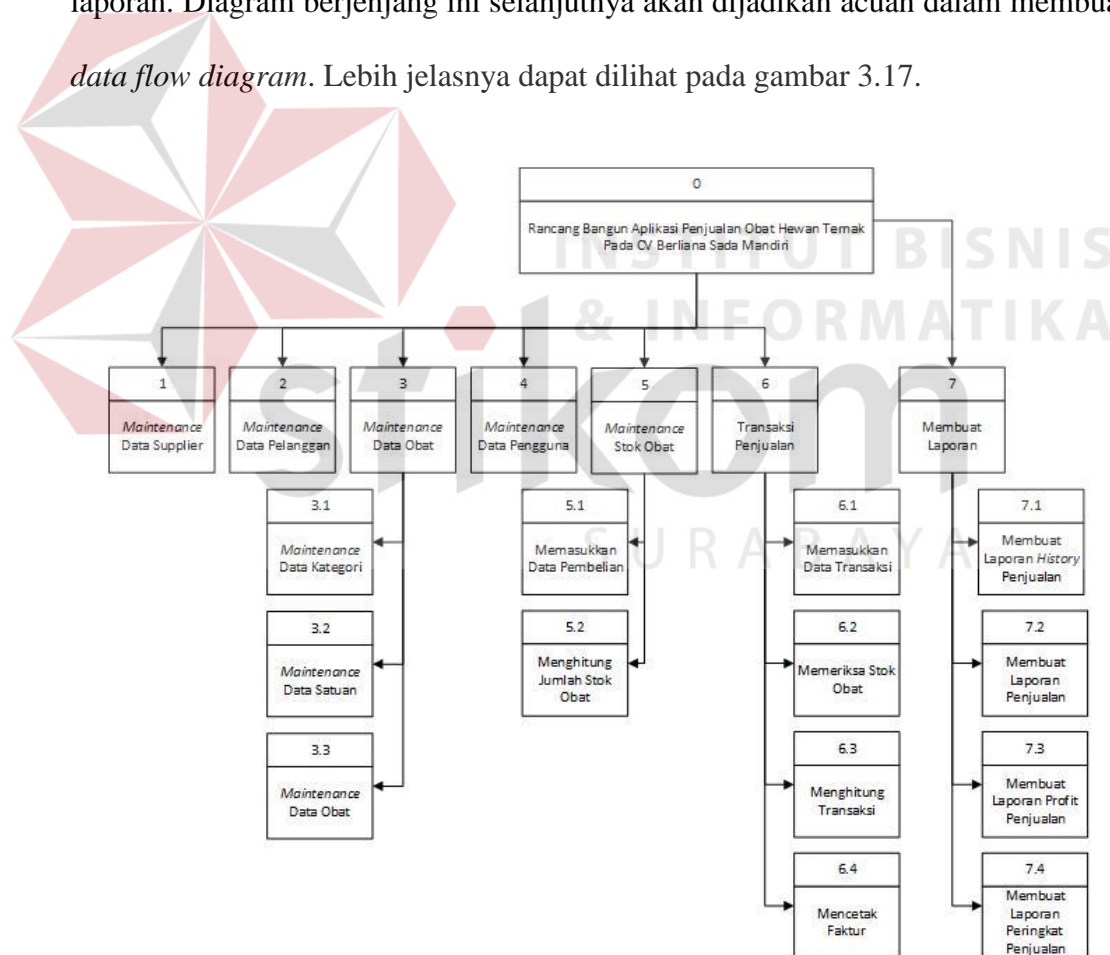
Context diagram merupakan diagram yang bertujuan untuk menggambarkan keluar masuknya data dalam sebuah aplikasi dan menggambarkan entitas yang bertanggung jawab atas data tersebut. Pada aplikasi ini, melibatkan tiga *external entity* yaitu manajer gudang, admin, dan kasir. Aliran data yang keluar dari masing-masing *external entity* mempunyai arti bahwa data tersebut berasal dari *external entity* tersebut. Sedangkan aliran data yang masuk mempunyai arti ditujukan untuk *external entity* tersebut. Gambaran lebih detail dari proses yang terdapat pada aplikasi penjualan obat hewan ternak pada BSM dapat dilihat pada gambar 3.16.



Gambar 3.16 *Context Diagram* Aplikasi Penjualan Obat Hewan Ternak pada BSM

3.2.3 Diagram Berjenjang

Diagram berjenjang adalah sebuah bagan yang menggambarkan proses-proses yang ada dalam sebuah aplikasi. Pada diagram berjenjang aplikasi penjualan obat hewan ternak pada BSM memiliki dua level. Pada level pertama terdapat tujuh proses yaitu *maintenance data supplier*, *maintenance data pelanggan*, *maintenance data obat*, *maintenance data pengguna*, *maintenance stok obat*, transaksi penjualan, dan membuat laporan. Pada level dua terdapat sub – sub proses dari proses *maintenance data obat*, *maintenance stok obat*, transaksi penjualan dan membuat laporan. Diagram berjenjang ini selanjutnya akan dijadikan acuan dalam membuat *data flow diagram*. Lebih jelasnya dapat dilihat pada gambar 3.17.



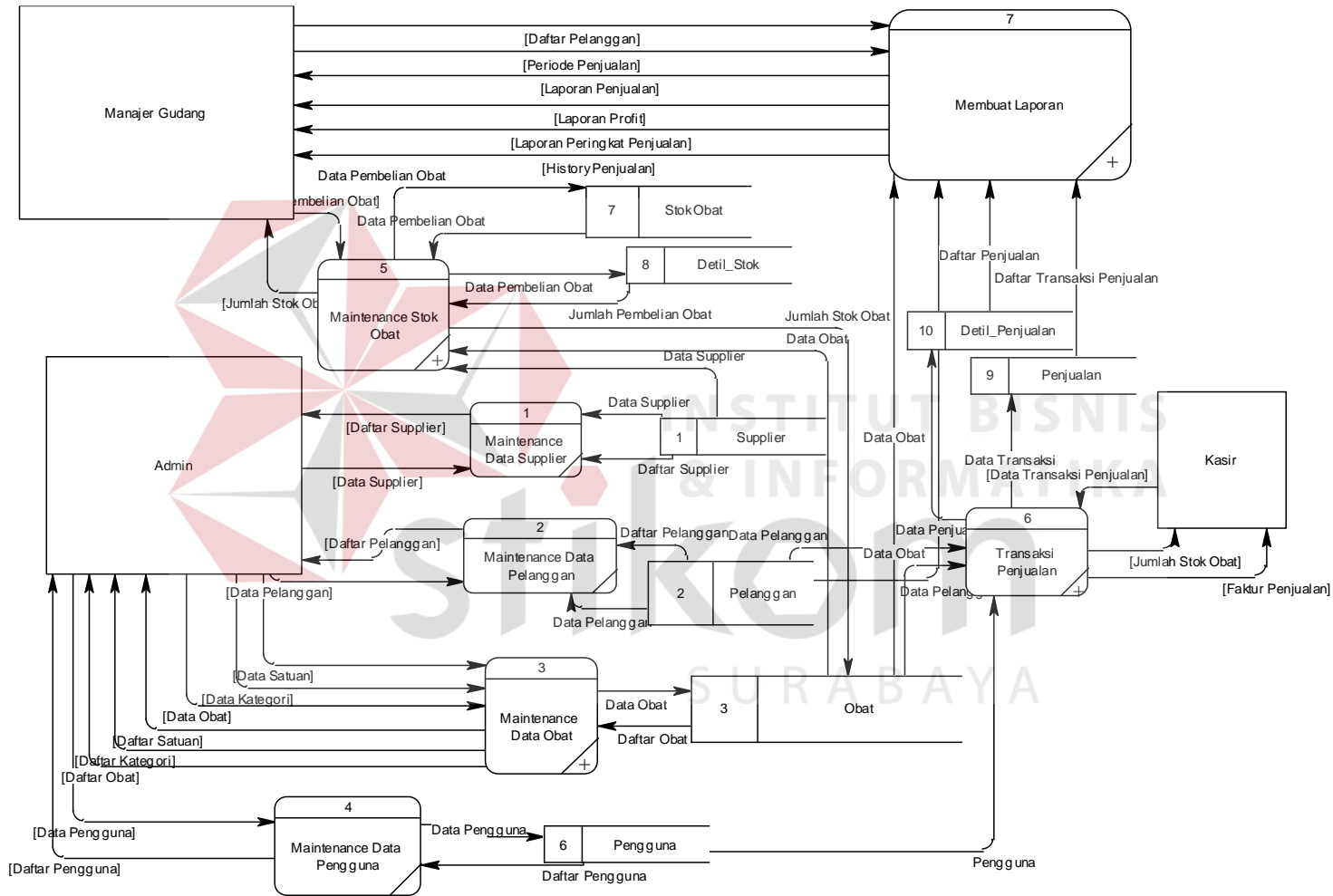
Gambar 3.17 Diagram Berjenjang Aplikasi Penjualan Obat Hewan Ternak

3.2.4 Data Flow Diagram (DFD)

Data flow diagram merupakan perangkat yang digunakan pada metodologi pengembangan sistem yang terstruktur. DFD dapat menggambarkan seluruh kegiatan yang terdapat pada sistem secara jelas. Selain itu, DFD juga mampu menggambarkan komponen dan aliran data antar komponen yang terdapat pada sistem yang akan dikembangkan. Penggambaran sistem menggunakan DFD dimulai dari *context diagram* seperti yang dapat dilihat pada gambar 3.16. Dari *context diagram* dapat di dekomposisi lagi menjadi level yang lebih rendah untuk menggambarkan sistem lebih rinci.

a. DFD Level 0 Aplikasi Penjualan Obat Hewan Ternak pada BSM

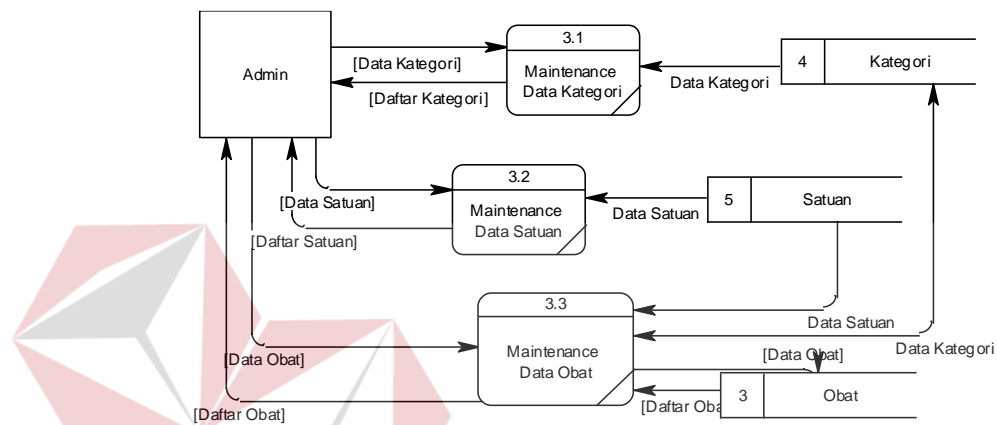
Pada DFD Level 0 Aplikasi Penjualan Obat Hewan Ternak pada BSM terdapat tujuh proses utama terdiri dari *maintenance data supplier*, *maintenance data pelanggan*, *maintenance data obat*, *maintenance data pengguna*, *maintenance stok obat*, transaksi penjualan, dan membuat laporan. Pada level ini terlihat tabel yang dibutuhkan dalam membangun aplikasi penjualan obat hewan ternak. Terdapat delapan tabel yang dibutuhkan dalam aplikasi ini, yaitu tabel pelanggan, tabel pengguna, tabel obat, tabel *supplier*, tabel *stok_obat*, tabel *detil_stok*, tabel penjualan, dan tabel *detil_penjualan*. Lebih jelasnya DFD level 0 dapat di lihat pada gambar 3.18.



Gambar 3.18 DFD Level 0 Aplikasi Penjualan Obat Hewan Ternak pada BSM

b. DFS Level 1 *Maintenance* Data Obat

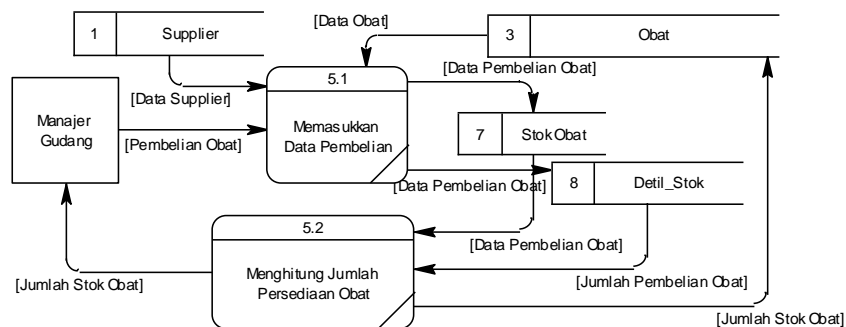
DFD level 1 *maintenance* data obat terdiri dari tiga proses yaitu memasukkan data satuan, memasukkan data kategori, dan *maintenance* data obat. Proses ini menghasilkan daftar obat. Lebih jelasnya DFD level 1 *maintenance* data obat dapat dilihat pada gambar 3.19.



Gambar 3.19 DFD Level 1 *Maintenance* Data Obat

c. DFD Level 1 *Maintenance* Stok Obat

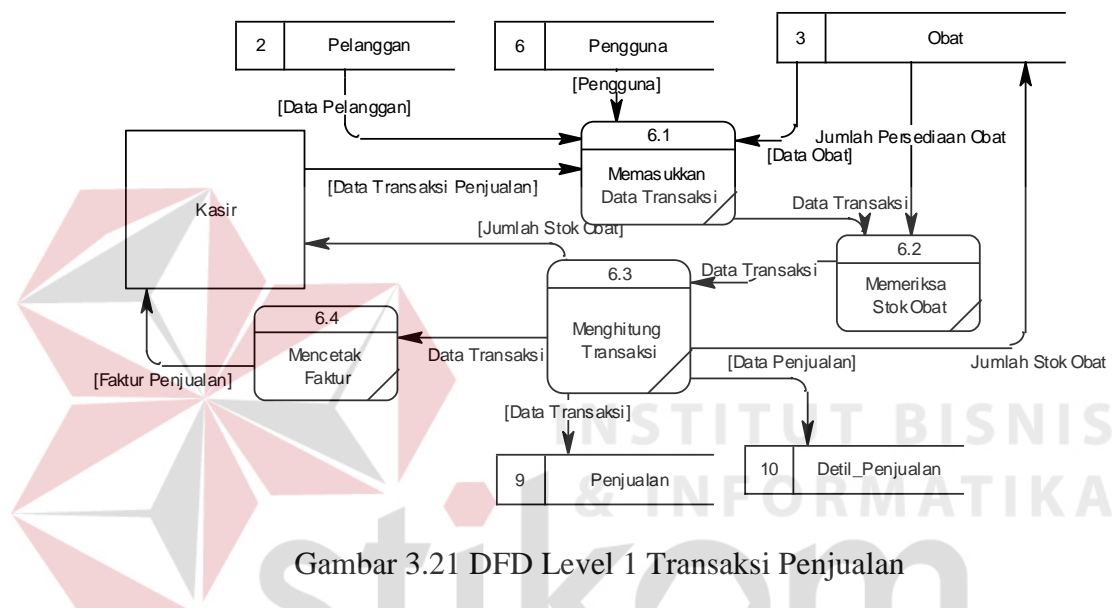
DFD level 1 *maintenance* stok obat terdiri dari dua proses yaitu memasukkan data pembelian dan menghitung jumlah persediaan obat. Proses ini menghasilkan jumlah stok obat. Lebih jelasnya DFD level 1 *maintenance* stok obat dapat dilihat pada gambar 3.20.



Gambar 3.20 DFD Level 1 *Maintenance* stok obat

d. DFD Level 1 Transaksi Penjualan

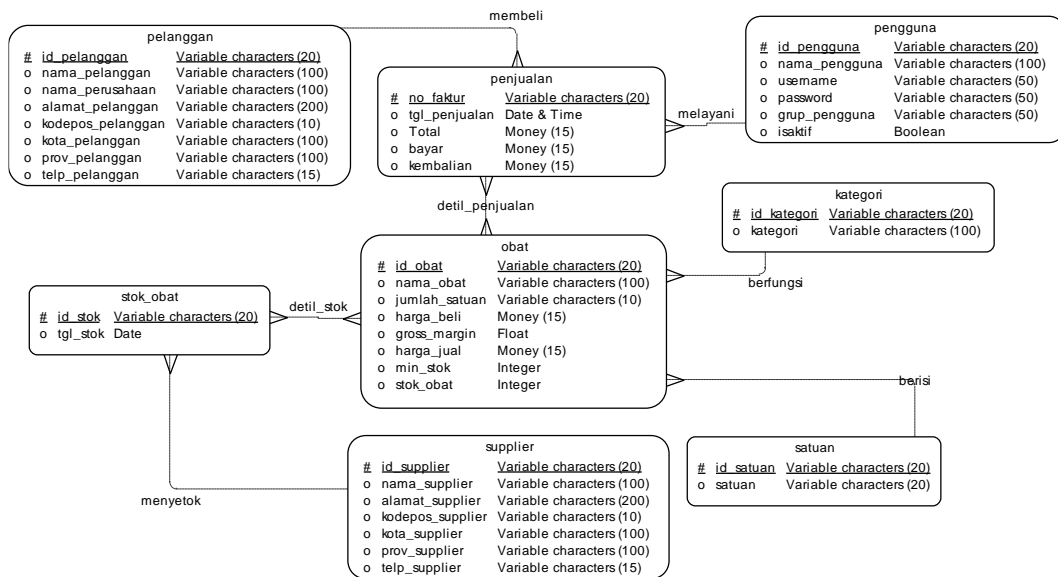
DFD level 1 Transaksi penjualan terdiri dari empat proses yaitu memasukkan data transaksi, memeriksa stok obat, menghitung transaksi, dan mencetak faktur. Proses ini menghasilkan faktur penjualan. Lebih jelasnya DFD level 1 transaksi penjualan dapat dilihat pada gambar 3.21.



Gambar 3.21 DFD Level 1 Transaksi Penjualan

e. DFD Level 1 Membuat Laporan

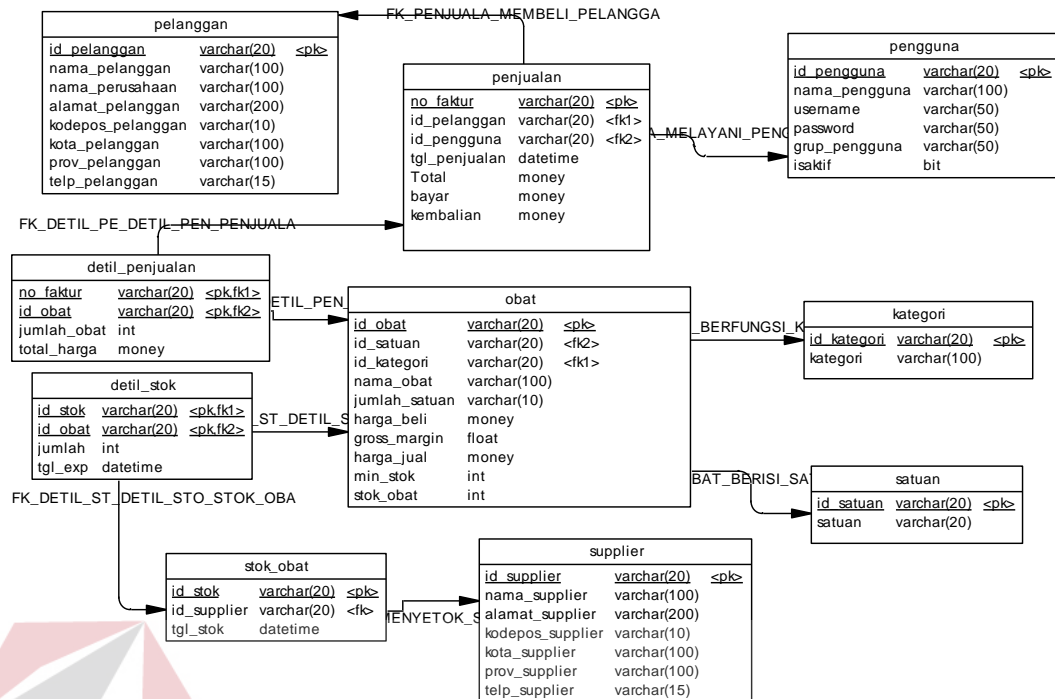
DFD level 1 membuat laporan terdiri dari empat proses yaitu membuat laporan *history* penjualan, membuat laporan penjualan, membuat laporan profit penjualan, dan membuat laporan peringkat penjualan. Proses ini menghasilkan laporan *history* penjualan, laporan penjualan, laporan profit penjualan, dan laporan peringkat penjualan. Lebih jelasnya DFD level 1 membuat laporan dapat dilihat pada gambar 3.22.



Gambar 3.23 CDM Aplikasi Penjualan Obat Hewan Ternak

b. *Physical Data Model (PDM)*

Physical Data Model (PDM) merupakan hasil *generate* dari CDM sebelumnya yang berfungsi menggambarkan keseluruhan tabel yang dibutuhkan dalam membangun sebuah aplikasi beserta hubungan antar tabel tersebut. Setelah CDM aplikasi penjualan obat hewan ternak di *generate* menghasilkan PDM aplikasi penjualan obat hewan ternak yang memiliki sepuluh tabel yaitu tabel *supplier*, tabel *pengguna*, tabel *pelanggan*, tabel *obat*, tabel *stok_obat*, tabel *detail_stok*, tabel *penjualan*, tabel *detail_penjualan*, tabel *kategori*, dan tabel *satuan*. PDM inilah yang digunakan dalam membangun aplikasi penjualan obat hewan ternak. PDM dari aplikasi penjualan obat hewan ternak dapat dilihat pada gambar 3.24.



Gambar 3.24 PDM Aplikasi Penjualan Obat Hewan Ternak

3.2.6 Struktur Tabel

Struktur tabel aplikasi penjualan obat hewan ternak pada BSM adalah sebagai berikut:

1. Tabel pengguna

Primary Key : id_pengguna

Foreign Key : -

Fungsi : Menyimpan data pengguna

Tabel 3.14 Struktur Pengguna

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_pengguna	varchar	20	Primary Key
nama_pengguna	varchar	100	
username	varchar	50	
password	varchar	50	
grup_pengguna	varchar	50	
isaktif	boolean		

2. Tabel Obat

Primary Key : id_obat

Foreign Key : id_satuan, id_kategori

Fungsi : Menyimpan data obat

Tabel 3.15 Struktur Obat

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_obat	varchar	20	<i>Primary Key</i>
id_satuan	varchar	20	<i>Foreign Key</i>
id_kategori	varchar	20	<i>Foreign Key</i>
nama_obat	varchar	100	
jumlah_satuan	varchar	10	
harga_beli	money	10	
gross_margin	int		
harga_jual	float		
min_stok	int		
stok_obat	inti		

3. Tabel *Supplier*

Primary Key : id_supplier

Foreign Key : -

Fungsi : Menyimpan data *supplier*

Tabel 3.16 Struktur *Supplier*

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_supplier	varchar	20	<i>Primary Key</i>
nama_supplier	varchar	100	
alamat_supplier	varchar	200	
kodepos_supplier	varchar	10	
kota_supplier	varchar	100	
prov_supplier	varchar	100	
telp_supplier	varchar	15	

4. Tabel Pelanggan

Primary Key : id_pelanggan

Foreign Key : -

Fungsi : Menyimpan data pelanggan

Tabel 3.17 Struktur Pelanggan

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_pelanggan	varchar	20	<i>Primary Key</i>
nama_pelanggan	varchar	100	
alamat_pelanggan	varchar	200	
nama_perusahaan	varchar	100	
kodepos_pelanggan	varchar	10	
kota_pelanggan	varchar	100	
prov_pelanggan	varchar	100	
telp_pelanggan	varchar	15	

5. Tabel Stok_Obat

Primary Key : id_stok

Foreign Key : id_supplier

Fungsi : Menyimpan data obat yang baru masuk gudang

Tabel 3.18 Struktur Stok Obat

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_stok	varchar	20	<i>Primary Key</i>
id_supplier	varchar	20	<i>Foreign Key</i>
tanggal_stok	datetime		

6. Tabel Penjualan

Primary Key : no_faktur

Foreign Key : id_pengguna, id_pelanggan

Fungsi : Menyimpan data penjualan

Tabel 3.19 Struktur Penjualan

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
no_faktur	varchar	20	<i>Primary Key</i>
id_pengguna	varchar	20	<i>Foreign Key</i>
id_pelanggan	varchar	20	<i>Foreign Key</i>
tgl_penjualan	datetime		
total	money		
bayar	money		
kembalian	money		

7. Tabel Detil_Penjualan

Primary Key : id_obat, no_faktur

Foreign Key : -

Fungsi : Menyimpan detil data penjualan

Tabel 3.20 Struktur Detil Penjualan

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_obat	varchar	20	<i>Primary Key</i>
no_faktur	varchar	20	<i>Primary Key</i>
jumlah_obat	int		
total_harga	money		

8. Tabel Satuan

Primary Key : id_satuan

Foreign Key : -

Fungsi : Menyimpan data satuan

Tabel 3.21 Struktur Satuan

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_satuan	varchar	20	<i>Primary Key</i>
satuan	varchar	20	

9. Tabel Kategori

Primary Key : id_kategori

Foreign Key : -

Fungsi : Menyimpan data kategori

Tabel 3.22 Struktur Kategori

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_kategori	varchar	20	<i>Primary Key</i>
kategori	varchar	100	

10. Tabel Detil_Stok

Primary Key : id_stok, id_obat

Foreign Key : -

Fungsi : Menyimpan detil data stok

Tabel 3.23 Struktur Detil Stok

<i>Field</i>	<i>Type Data</i>	<i>Length</i>	<i>Constraint</i>
id_stok	varchar	20	<i>Primary Key</i>
id_obat	varchar	20	<i>Primary Key</i>
jumlah	inti		
tol_exp	datetime		

3.2.7 Perancangan *Interface*

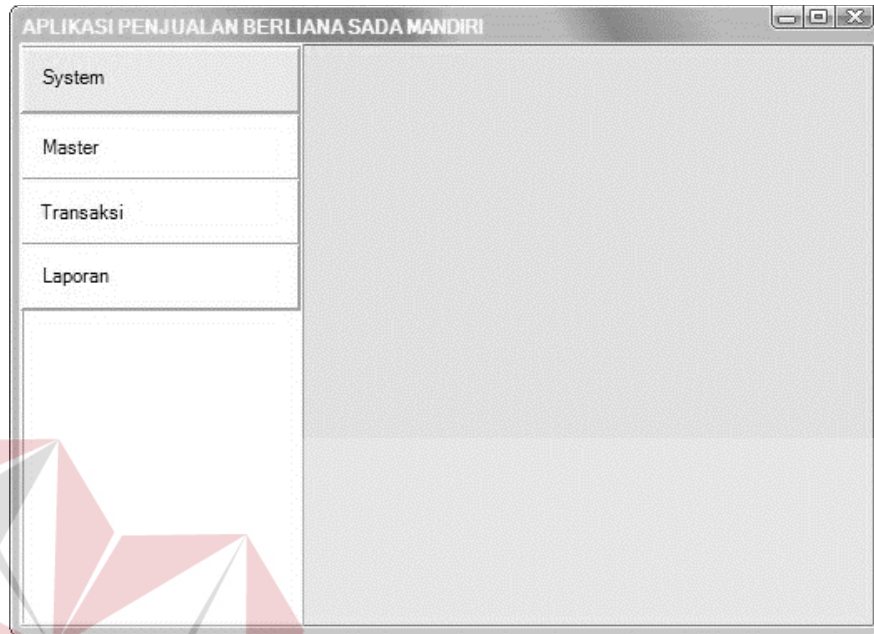
Perancangan *interface* digunakan untuk memberikan gambaran awal terhadap desain *form* yang akan dibangun pada aplikasi penjualan obat hewan ternak. Berikut ini desain *interface* dari aplikasi penjualan obat hewan ternak :

a. *Form* Menu Utama

Form ini merupakan tampilan awal dari aplikasi ketika aplikasi dijalankan.

Pengguna tidak bisa membuka menu selain menu *System* sebelum melakukan

proses *login*. Untuk lebih jelasnya, rancangan *form* menu utama dapat dilihat pada gambar 3.25.



Gambar 3.25 Rancangan *Form* Menu Utama

b. *Form Login*

Form login digunakan sebagai autentifikasi pengguna yang ingin masuk ke dalam sistem. Pada *form* ini pengguna diharuskan memasukkan *username*, *password* dan memilih grup pengguna, baru kemudian menekan tombol “Masuk”. Apabila *username*, *password*, dan grup yang dipilih tersebut sesuai dengan data yang tersimpan, maka pengguna dapat masuk ke dalam aplikasi dan membuka menu sesuai dengan grup pengguna yang terpilih. Tombol “Keluar” digunakan untuk membatalkan pengguna untuk masuk ke aplikasi. Untuk lebih jelasnya, rancangan *form login* dapat dilihat pada gambar 3.26.

Gambar 3.26 Rancangan *Form Login*

c. *Form Ubah Password*

Form ubah *password* digunakan untuk mengganti *password* pengguna. Pada *form* ini pengguna akan memasukkan *password* lama, *password* baru, dan konfirmasi *password* baru kemudian menekan tombol “Ubah”. Apabila *password* lama sesuai dengan *password* lama yang tersimpan dalam database, serta konfirmasi *password* baru sesuai dengan *password* baru yang dimasukkan dalam kolom, maka akan tampil pesan “Password Berhasil Diubah”. Tombol “Batal” digunakan untuk membatalkan pengguna untuk mengubah *password*. Untuk lebih jelasnya, rancangan *form* ubah *password* dapat dilihat pada gambar 3.27.

Gambar 3.27 Rancangan *Form Ubah Password*

d. *Form Master Supplier*

Form master *Supplier* digunakan untuk mengelola data *supplier* yang bekerja sama dengan BSM. Dalam *form* ini terdapat tombol "Baru" untuk menambah data baru, tombol "Simpan" untuk menyimpan data baru, tombol "Ubah" untuk mengubah data lama, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* master *supplier*. *Form* master *supplier* juga memiliki tabel data grid untuk menampilkan data *supplier* yang telah tersimpan. Dalam *form* ini juga terdapat kolom untuk mencari data *supplier* tertentu sesuai *filter* yang disediakan. Untuk lebih jelasnya, rancangan *form* master *supplier* dapat dilihat pada gambar 3.28.

The screenshot shows a window titled 'APLIKASI PENJUALAN BERLIANA SADA MANDIRI' with a sub-window 'MASTER SUPPLIER'. The form contains the following elements:

- Input Fields:** ID Supplier (value: SUPY00001), Nama, Kota, Alamat, Provinsi, Kode Pos, and No Telp.
- Buttons:** Baru, Simpan, Ubah, Batal, and Keluar.
- Data Grid:** A table with columns: ID Supplier, Nama, Alamat, Kode Pos, Kota, Provinsi, and No Telp.
- Search Section:** Cari: ID Supplier (dropdown), [input field], Cari (button).

Gambar 3.28 Rancangan *Form* Master *Supplier*

e. *Form* Master Pelanggan

Form master pelanggan digunakan untuk mengelola data pelanggan yang bertransaksi di gudang BSM. Dalam *form* ini terdapat tombol "Baru" untuk

menambah data baru, tombol "Simpan" untuk menyimpan data baru, tombol "Ubah" untuk mengubah data lama, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* master pelanggan. *Form* master pelanggan juga memiliki tabel data grid untuk menampilkan data pelanggan yang telah tersimpan. Dalam *form* ini juga terdapat kolom untuk mencari data pelanggan tertentu sesuai *filter* yang disediakan. Untuk lebih jelasnya, rancangan *form* master pelanggan dapat dilihat pada gambar 3.29.

The screenshot shows a window titled "APLIKASI PENJUALAN BERLIANA SADA MANDIRI" with a sub-header "MASTER PELANGGAN". The form contains the following elements:

- ID Pelanggan:** CUST00001
- Input Fields:** Nama, Alamat, Kode Pos, Kota, Provinsi, No. Telepon, Nama Perusahaan, Promo.
- Buttons:** Baru, Simpan, Ubah, Batal, Keluar.
- Search Section:** Cari: ID Pelanggan (dropdown), [input field], Cari (button).
- Data Grid Table:**

ID Pelanggan	Tanggal ...	Nama	Alamat	Kode Pos	Kota	Provinsi	Nama P...	Alamat...	Kode Po...

Gambar 3.29 Rancangan *Form* Master Pelanggan

f. *Form* Master Kategori

Form master kategori digunakan untuk mengelola data kategori yang digunakan sebagai salah satu masukan data obat. Dalam *form* ini terdapat tombol "Baru" untuk menambah data baru, tombol "Simpan" untuk

menyimpan data baru, tombol "Ubah" untuk mengubah data lama, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* master kategori. *Form* master kategori juga memiliki tabel *data grid* untuk menampilkan data kategori yang telah tersimpan. Dalam *form* ini juga terdapat kolom untuk mencari data kategori tertentu sesuai *filter* yang disediakan. Untuk lebih jelasnya, rancangan *form* master kategori dapat dilihat pada gambar 3.30.



Gambar 3.30 Rancangan *Form* Master Kategori

g. *Form* Master Satuan

Form master satuan digunakan untuk mengelola data satuan yang digunakan sebagai salah satu masukan dalam data obat. Dalam *form* ini terdapat tombol "Baru" untuk menambah data baru, tombol "Simpan" untuk menyimpan data baru, tombol "Ubah" untuk mengubah data lama, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* master satuan. *Form* master satuan juga memiliki tabel

data grid untuk menampilkan data satuan yang telah tersimpan. Dalam *form* ini juga terdapat kolom untuk mencari data satuan tertentu sesuai *filter* yang disediakan. Untuk lebih jelasnya, rancangan *form* master satuan dapat dilihat pada gambar 3.31.

Gambar 3.31 Rancangan *Form* Master Satuan

h. *Form* Master Obat

Form master obat digunakan untuk mengelola data obat yang dijual di gudang BSM. Dalam *form* ini terdapat tombol "Baru" untuk menambah data baru, tombol "Simpan" untuk menyimpan data baru, tombol "Ubah" untuk mengubah data lama, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* master obat. *Form* master obat juga memiliki tabel data grid untuk menampilkan data obat yang telah tersimpan. Dalam *form* ini juga terdapat kolom untuk mencari data obat tertentu sesuai *filter* yang disediakan. Untuk lebih jelasnya, rancangan *form* master obat dapat dilihat pada gambar 3.32.

Gambar 3.32 Rancangan *Form* Master Obat

i. *Form* Master Pengguna

Form master pengguna digunakan untuk mengelola data pengguna yang menggunakan aplikasi penjualan obat hewan ternak. Dalam *form* ini terdapat tombol "Baru" untuk menambah data baru, tombol "Simpan" untuk menyimpan data baru, tombol "Ubah" untuk mengubah data lama, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* master pengguna. *Form* master pengguna juga memiliki tabel data grid untuk menampilkan data pengguna yang telah tersimpan. Dalam *form* ini juga terdapat kolom untuk mencari data pengguna tertentu sesuai *filter* yang disediakan. Untuk lebih jelasnya, rancangan *form* master pengguna dapat dilihat pada gambar 3.33.

The image shows a software window titled "APLIKASI PENJUALAN BERLIANA SADA MANDIRI" with a sub-window titled "MASTER PENGGUNA". The form contains the following fields and controls:

- ID Pengguna:** Text field containing "PENG00001".
- Provinsi:** Text field.
- Nama:** Text field.
- Kode Pos:** Text field.
- Alamat:** Text field with a "Combo Box" dropdown menu.
- Kota:** Text field.
- Buttons:** "Baru", "Simpan", "Ubah", "Batal", and "Keluar".
- Table:** A table with columns: "ID Pengguna", "Nama", "Username", and "Grup Pengguna". The table is currently empty.
- Search Section:** "Cari:" followed by a dropdown menu set to "ID Pengguna", an input field, and a "Cari" button.

Gambar 3.33 Rancangan *Form* Master Pengguna

j. *Form* Stok Obat

Form stok obat digunakan untuk memasukkan data persediaan obat baru di dalam gudang BSM. Dalam *form* ini terdapat tombol "Baru" untuk menambah data baru, tombol "Simpan" untuk menyimpan data baru, tombol "Batal" untuk membatalkan proses menyimpan atau mengubah data, dan tombol "Keluar" untuk keluar dari *form* stok obat. Untuk lebih jelasnya, rancangan *form* master obat dapat dilihat pada gambar 3.34.

The image shows a software window titled "APLIKASI PENJUALAN BERLIANA SADA MANDIRI". Inside the window, there is a section titled "STOKOBAT". This section contains several input fields: "ID Stok", "Tanggal", "Supplier", "Obat", "Jumlah Obat", and "Tanggal Expired". Below these fields are four buttons: "Baru", "Simpan", "Batal", and "Keluar". At the bottom of the window, there is a table with the following columns: "Nama Obat", "jumlah Obat", and "Tanggal Exp...". The table is currently empty.

Gambar 3.34 Rancangan *Form* Stok Obat

k. *Form* Transaksi Penjualan

Form transaksi penjualan digunakan untuk mengolah dan menyimpan data transaksi penjualan yang terjadi di BSM. Dalam *form* ini terdapat tombol "Baru" untuk menambah data transaksi baru, tombol "Lanjut" untuk melanjutkan ke proses memasukkan obat, tombol "Tambah" untuk menambah obat yang dibeli, tombol "Hapus" untuk membatalkan obat yang akan dibeli, kemudian tombol "Bayar" untuk menyimpan dan menghitung transaksi. Untuk lebih jelasnya, rancangan *form* master obat dapat dilihat pada gambar 3.35.

m. *Form* Laporan Penjualan

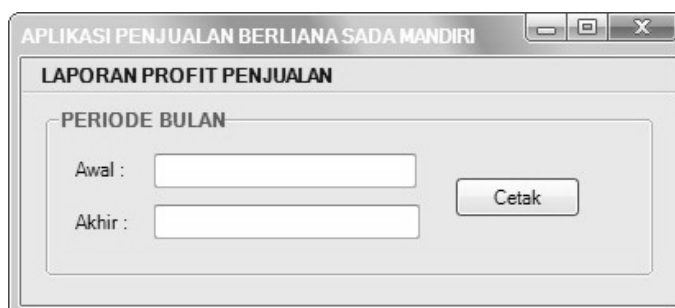
Form laporan penjualan digunakan untuk mencetak laporan penjualan. Terdapat tiga pilihan cetak laporan yaitu cetak laporan berdasarkan periode bulanan, periode harian, dan periode mingguan. Lebih jelasnya dapat dilihat pada gambar 3.37.

The image shows a software window titled "APLIKASI PENJUALAN BERLIANA SADA MANDIRI" with a sub-header "LAPORAN PENJUALAN". The window contains three distinct sections for selecting reporting periods. The first section, "PERIODE BULAN", has two text input fields labeled "Awal" and "Akhir", followed by a "Cetak" button. The second section, "PERIODE MINGGUAN", also has "Awal" and "Akhir" input fields and a "Cetak" button. The third section, "PERIODE HARIAN", has "Awal" and "Akhir" input fields and a "Cetak" button. A large, semi-transparent watermark logo is overlaid on the left side of the window.

Gambar 3.37 Rancangan *Form* Laporan Penjualan

n. *Form* Laporan Profit Penjualan

Form laporan profit penjualan digunakan untuk mencetak laporan profit penjualan. Pengguna diharuskan memasukkan periode bulan yang diinginkan lebih dulu kemudian tekan tombol cetak. Lebih jelasnya dapat dilihat pada gambar 3.38.

The image shows a software window titled "APLIKASI PENJUALAN BERLIANA SADA MANDIRI" with a sub-header "LAPORAN PROFIT PENJUALAN". The window contains a single section titled "PERIODE BULAN" which includes two text input fields labeled "Awal" and "Akhir", and a "Cetak" button. A large, semi-transparent watermark logo is overlaid on the left side of the window.

Gambar 3.38 Rancangan *Form* Laporan Profit Penjualan

o. *Form* Laporan Peringkat Penjualan

Form laporan peringkat penjualan digunakan untuk mencetak laporan peringkat penjualan. Terdapat dua pilihan cetak yaitu berdasarkan peringkat pendapatan dan peringkat profit penjualan. Lebih jelasnya dapat dilihat pada gambar 3.39.

The image shows a screenshot of a software application window titled "APLIKASI PENJUALAN BERLIANA SADA MANDIRI". Inside the window, there is a form titled "LAPORAN PERINGKAT PENJUALAN". The form is divided into two main sections: "PERINGKAT PENDAPATAN" and "PERINGKAT PROFIT". Each section contains two input fields labeled "Awal" and "Akhir" for specifying date ranges, and a "Cetak" button to print the report. A large, semi-transparent watermark logo for "stikom" (Institut Bisnis & Informatika) is overlaid on the image.

Gambar 3.39 Rancangan *Form* Laporan Peringkat Penjualan

3.2.8 Perancangan *Output*

Perancangan *output* merupakan perancangan laporan hasil dari proses data yang terjadi, yang tersimpan pada *database* yang kemudian akan diolah sedemikian rupa menjadi informasi yang berguna bagi pengguna sistem informasi. Terdapat empat jenis laporan yang dihasilkan oleh aplikasi, yaitu laporan *history* penjualan, laporan penjualan, laporan profit penjualan, dan laporan peringkat penjualan.

a. Desain Cetak Faktur Penjualan

Faktur penjualan adalah bukti pembayaran yang diberikan BSM kepada pelanggan setelah melakukan transaksi. Desain cetak faktur penjualan dapat dilihat pada gambar 3.40.

Berliana Sada Mandiri
 Perum Wisma Tropodo JL. Kiay Kholil
 NO.3 Waru Sidoarjo
 Telp.031 8665690

LAPORAN PENJUALAN
 Periode: xx/xxxx sampai xx/xxxx
 Bulan xx/xxxx

PRODUK	QUANTITY (in pack)	PENDAPATAN

TOTAL:

Gambar 3.41 Desain Laporan Penjualan Bulanan

Berliana Sada Mandiri
 Perum Wisma Tropodo JL. Kiay Kholil
 NO.3 Waru Sidoarjo
 Telp.031 8665690

LAPORAN PENJUALAN
 Periode: xx/xx/xxxx sampai xx/xx/xxxx
 Tanggal xx/xx/xxxx

PRODUK	QUANTITY (in pack)	PENDAPATAN

TOTAL:

Gambar 3.42 Desain Laporan Penjualan Harian

Berliana Sada Mandiri
 Perum Wisma Tropodo JL. Kiay Kholil
 NO.3 Waru Sidoarjo
 Telp.031 8665690

LAPORAN PENJUALAN
 Periode: xx/xx/xxxx sampai xx/xx/xxxx
 Minggu xx/xx/xxxx

PRODUK	QUANTITY (in pack)	PENDAPATAN
TOTAL:	<input type="text"/>	<input type="text"/>

Gambar 3.43 Desain Laporan Penjualan Mingguan

Berliana Sada Mandiri
 Perum Wisma Tropodo JL. Kiay Kholil
 NO.3 Waru Sidoarjo
 Telp.031 8665690

LAPORAN PROFIT PENJUALAN
 Periode: xx/xxxx sampai xx/xxxx
 Bulan xx/xxxx

PRODUK	QUANTITY (in pack)	PROFIT
TOTAL:	<input type="text"/>	<input type="text"/>

Gambar 3.44 Desain Laporan Profit Penjualan

d. Desain Laporan Peringkat Penjualan

Laporan peringkat penjualan menampilkan peringkat obat berdasarkan pendapatan penjualan terbanyak dan peringkat obat dengan profit penjualan

terbanyak. Dalam laporan peringkat penjualan menampilkan jenis produk, jumlah pendapatan yang diperoleh per produk, dan jumlah profit yang didapatkan per produk. Lebih jelasnya desain laporan peringkat penjualan dapat pada gambar 3.45.

Berliana Sada Mandiri Perum Wisma Tropodo J.L. Kay Kholil NO.3 Waru Sidoarjo Telp.031 8665690			
LAPORAN PERINGKAT PENJUALAN Periode: xx/xxxx sampai xx/xxxx Bulan xx/xxxx			
PERINGKAT	PRODUK	PENDAPATAN	PROFIT
1		Rp.	Rp.
2		Rp.	Rp.
3		Rp.	Rp.
4		Rp.	Rp.
5		Rp.	Rp.

Gambar 3.45 Desain Laporan Peringkat Penjualan

3.2.9 Perancangan Uji Coba

Pengujian sistem dilakukan dengan cara melakukan berbagai percobaan terhadap beberapa menu utama untuk membuktikan bahwa aplikasi telah berjalan sesuai dengan tujuan. Pengujian sistem ini menggunakan metode *Black Box Testing*. Berikut ini adalah perancangan uji coba pada aplikasi penjualan obat hewan ternak.

a. Rancangan Uji Coba Menu Utama

Rancangan uji coba menu utama dapat dilihat pada tabel 3.24. Menu utama digunakan untuk memilih menu-menu yang ada pada sistem aplikasi penjualan obat hewan ternak.

Tabel 3.24 Rancangan Uji Coba Menu Utama

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Form menu utama dengan menu <i>login</i> dan keluar yang dapat dipilih.
2	Mengetahui respon sistem setelah proses <i>login</i> .	Memasukkan <i>username</i> dan <i>password</i> yang valid pada <i>form login</i> .	menu utama tampil dan dapat dapat dipilih.

b. Rancangan Uji Coba Menu *Login*

Rancangan uji coba menu *login* dapat dilihat pada tabel 3.26. Untuk melakukan uji coba digunakan data pada tabel 3.25.

Tabel 3.25 Data Uji Coba Menu *Login*

<i>Username</i>	<i>Password</i>	<i>Group</i>
admin	admin	Administrator
(kosong)	(kosong)	Manajer
(kosong)	(kosong)	(kosong)

Tabel 3.26 Rancangan Uji Coba Menu *Login*

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Form <i>login</i> tampil.
2	Mengetahui respon sistem terhadap <i>username</i> dan <i>password</i> pengguna yang valid.	Memasukkan data <i>login</i> : <i>username</i> = admin, <i>password</i> = admin, <i>group</i> = Administrator.	Muncul pesan “Selamat datang, admin” dan tampil menu-menu yang dapat dipilih.
3	Mengetahui respon sistem terhadap <i>username</i> dan <i>password</i> pengguna yang tidak valid.	Memasukkan data <i>login</i> : <i>username</i> = admin, <i>password</i> = admin, <i>group</i> = Manager.	Muncul pesan “Anda tidak mempunyai hak akses untuk login..!!”.
4	Mengetahui respon sistem terhadap <i>username</i> dan	Memasukkan data <i>login</i> : <i>username</i> =	Muncul pesan “Username tidak boleh kosong!”.

No.	Tujuan	Input	Output yang Diharapkan
	<i>password</i> pengguna yang tidak valid.	..., <i>password</i> = ..., <i>group</i> =	

c. Perancangan Uji Coba Menu Ubah *Password*

Rancangan uji coba menu ubah *password* dapat dilihat pada tabel 3.28. Untuk melakukan uji coba digunakan data pada tabel 3.27.

Tabel 3.27 Data Uji Coba Menu Ubah *Password*

<i>Password</i> Lama	<i>Password</i> Baru	Konfirmasi <i>Password</i>
admin	1234	1234
-	-	-
admin2	1234	1234
admin	1234	12345

Tabel 3.28 Rancangan Uji Coba Menu Ubah *Password*

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Form ubah <i>password</i> tampil.
2	Deskripsi perubahan <i>password</i> yang valid.	Memasukkan data <i>password</i> lama = admin, <i>password</i> baru= 1234, konfirmasi <i>password</i> = 1234.	Muncul pesan "Password Berhasil Diubah...!!!".
3	Deskripsi perubahan <i>password</i> yang tidak valid (Data tidak lengkap).	Memasukkan data <i>password</i> lama = "", <i>password</i> baru= "", konfirmasi <i>password</i> = "".	Muncul pesan "Data Belum Lengkap...".
4	Deskripsi perubahan <i>password</i> yang tidak valid (<i>password</i> tidak sesuai).	Memasukkan data <i>password</i> lama = admin2, <i>password</i> baru= 1234, konfirmasi <i>password</i> = 1234.	Muncul pesan "Password Salah...!!!".
5	Deskripsi perubahan <i>password</i> yang tidak valid (konfirmasi	Memasukkan data <i>password</i> lama = admin, <i>password</i> baru= 1234,	Muncul pesan "Salah Input Data...!!! (Password baru =

No.	Tujuan	Input	Output yang Diharapkan
	<i>password</i> baru tidak sesuai).	konfirmasi <i>password</i> = 12345.	Konfirmasi <i>password</i> baru)".

d. Perancangan Uji Coba Menu Master *Supplier*

Menu master *supplier* digunakan untuk mengolah data *supplier*. Rancangan uji coba menu master *supplier* dapat dilihat pada tabel 3.29.

Tabel 3.29 Rancangan Uji Coba Menu Master *Supplier*

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> master <i>supplier</i> dengan data <i>supplier</i> yang tersimpan
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol "Baru", memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol "Simpan".	Muncul pesan "Data Berhasil Disimpan..!!!" dan data tersimpan dalam <i>database</i> .
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol "Baru", beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol "Simpan".	Muncul pesan "Data Belum Lengkap..!!!".
4	Melakukan perubahan data pada <i>database</i> dengan data yang valid.	Memilih data yang akan diubah pada tabel <i>grid control</i> , tekan tombol "Ubah", melakukan perubahan pada beberapa data, kemudian tekan tombol "Simpan".	Muncul pesan "Data Berhasil Diubah..!!!" dan data yang terdapat dalam <i>database</i> sudah berubah.

e. Perancangan Uji Coba Menu Master Pelanggan

Menu master pelanggan digunakan untuk mengolah data pelanggan. Rancangan uji coba menu master pelanggan dapat dilihat pada tabel 3.30.

Tabel 3.30 Rancangan Uji Coba Menu Master Pelanggan

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> master pelanggan dengan data pelanggan yang tersimpan
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol “Simpan”.	Muncul pesan “Data Berhasil Disimpan..!!!” dan data tersimpan dalam <i>database</i> .
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol “Baru”, beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol “Simpan”.	Muncul pesan “Data Belum Lengkap..!!!”.
4	Melakukan perubahan data pada <i>database</i> dengan data yang valid.	Memilih data yang akan diubah pada tabel <i>grid control</i> , tekan tombol “Ubah”, melakukan perubahan pada beberapa data, kemudian tekan tombol “Simpan”.	Muncul pesan “Data Berhasil Diubah..!!!” dan data yang terdapat dalam <i>database</i> sudah berubah.

f. Perancangan Uji Coba Menu Master Obat

Menu master *supplier* digunakan untuk mengolah data obat. Rancangan uji coba menu master *supplier* dapat dilihat pada tabel 3.31.

Tabel 3.31 Rancangan Uji Coba Menu Master Obat

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> master obat dengan data obat yang tersimpan
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol “Simpan”.	Muncul pesan “Data Berhasil Disimpan..!!!” dan data tersimpan dalam <i>database</i> .

No.	Tujuan	Input	Output yang Diharapkan
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol “Baru”, beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol “Simpan”.	Muncul pesan “Data Belum Lengkap..!!!”.
4	Melakukan perubahan data pada <i>database</i> dengan data yang valid.	Memilih data yang akan diubah pada tabel <i>grid control</i> , tekan tombol “Ubah”, melakukan perubahan pada beberapa data, kemudian tekan tombol “Simpan”.	Muncul pesan “Data Berhasil Diubah..!!!” dan data yang terdapat dalam <i>database</i> sudah berubah.

g. Perancangan Uji Coba Menu Master Kategori

Menu master kategori digunakan untuk mengolah data kategori yang dibutuhkan untuk mengolah data obat. Rancangan uji coba menu master kategori dapat dilihat pada tabel 3.32.

Tabel 3.32 Rancangan Uji Coba Menu Master Kategori

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> master kategori dengan data kategori yang tersimpan
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol “Simpan”.	Muncul pesan “Data Berhasil Disimpan..!!!” dan data tersimpan dalam <i>database</i> .
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol “Baru”, beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol “Simpan”.	Muncul pesan “Data Belum Lengkap..!!!”.
4	Melakukan perubahan data pada <i>database</i>	Memilih data yang akan diubah pada tabel <i>grid control</i> , tekan tombol	Muncul pesan “Data Berhasil Diubah..!!!” dan data yang terdapat

No.	Tujuan	Input	Output yang Diharapkan
	dengan data yang valid.	“Ubah”, melakukan perubahan pada beberapa data, kemudian tekan tombol “Simpan”.	dalam <i>database</i> sudah berubah.

h. Perancangan Uji Coba Menu Master Satuan

Menu master satuan digunakan untuk mengolah data satuan yang dibutuhkan untuk mengolah data obat. Rancangan uji coba menu master satuan dapat dilihat pada tabel 3.33.

Tabel 3.33 Rancangan Uji Coba Menu Master Satuan

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> master satuan dengan data satuan yang tersimpan
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol “Simpan”.	Muncul pesan “Data Berhasil Disimpan..!!!” dan data tersimpan dalam <i>database</i> .
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol “Baru”, beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol “Simpan”.	Muncul pesan “Data Belum Lengkap..!!!”.
4	Melakukan perubahan data pada <i>database</i> dengan data yang valid.	Memilih data yang akan diubah pada tabel <i>grid control</i> , tekan tombol “Ubah”, melakukan perubahan pada beberapa data, kemudian tekan tombol “Simpan”.	Muncul pesan “Data Berhasil Diubah..!!!” dan data yang terdapat dalam <i>database</i> sudah berubah.

i. Perancangan Uji Coba Menu Master Pengguna

Menu master pengguna digunakan untuk mengolah data pengguna yang menggunakan aplikasi. Rancangan uji coba menu master pengguna dapat dilihat pada tabel 3.34.

Tabel 3.34 Rancangan Uji Coba Menu Master Pengguna

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> master pengguna dengan data pengguna yang tersimpan
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol “Simpan”.	Muncul pesan “Data Berhasil Disimpan..!!!” dan data tersimpan dalam <i>database</i> .
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol “Baru”, beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol “Simpan”.	Muncul pesan “Data Belum Lengkap..!!!”.
4	Melakukan perubahan data pada <i>database</i> dengan data yang valid.	Memilih data yang akan diubah pada tabel <i>grid control</i> , tekan tombol “Ubah”, melakukan perubahan pada beberapa data, kemudian tekan tombol “Simpan”.	Muncul pesan “Data Berhasil Diubah..!!!” dan data yang terdapat dalam <i>database</i> sudah berubah.

j. Perancangan Uji Coba Menu Stok Obat

Menu stok obat digunakan untuk memasukkan data persediaan obat dalam gudang. Rancangan uji coba menu stok obat dapat dilihat pada tabel 3.35.

Tabel 3.35 Rancangan Uji Coba Menu Stok Obat

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> stok obat dengan jumlah data obat yang tersimpan di gudang
2	Menambah data baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada semua <i>textbox</i> , kemudian menekan tombol “Simpan”.	Muncul pesan “Data Berhasil Disimpan..!!!” dan data tersimpan dalam <i>database</i> .
3	Menambah data baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol “Baru”, beberapa <i>textbox</i> tidak diisi, kemudian menekan tombol “Simpan”.	Muncul pesan “Data Belum Lengkap..!!!”.

k. Perancangan Uji Coba Menu Transaksi Penjualan

Menu transaksi penjualan digunakan untuk memasukkan data transaksi penjualan yang terjadi kemudian menghitungnya dan menyimpannya dalam *database*. Rancangan uji coba menu transaksi penjualan dapat dilihat pada tabel 3.36.

Tabel 3.36 Rancangan Uji Coba Menu Transaksi Penjualan

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Menampilkan <i>form</i> transaksi penjualan, dan <i>textbox</i> transaksi tidak bisa digunakan
2	Menambah data pelanggan baru ke dalam <i>database</i> dengan data yang valid.	Menekan tombol “Baru”, memasukkan data pada <i>textbox</i> pelanggan	<i>textbox</i> obat dapat digunakan
3	Menambah data pelanggan baru ke dalam <i>database</i>	Menekan tombol “Baru”, <i>textbox</i> pelanggan tidak diisi	<i>textbox</i> obat tidak dapat digunakan

No.	Tujuan	Input	Output yang Diharapkan
	dengan data tidak valid.		
4	Menambah data transaksi baru ke dalam <i>database</i> dengan data yang valid.	Mengisi seluruh <i>textbox</i> transaksi kemudian menekan tombol tambah	Muncul total pesanan, dan data transaksi masuk dalam tabel <i>grid</i>
5	Menambah data transaksi baru ke dalam <i>database</i> dengan data yang tidak valid (tidak lengkap).	Menekan tombol "Tambah", tapi <i>textbox</i> transaksi tidak diisi	Muncul pesan "Data Belum Lengkap..!!!".
6	Memeriksa jumlah persediaan obat dengan data obat yang valid(jumlah obat mencukupi)	Menekan "Enter" / "Tambah" setelah mengisi <i>textbox</i> obat dan jumlah	Muncul total pesanan, dan data transaksi masuk dalam tabel <i>grid</i>
7	Memeriksa jumlah persediaan obat dengan data obat yang valid(jumlah obat tidak mencukupi)	Menekan tombol "Enter"/"Tambah" dengan jumlah obat melebihi stok	Muncul pesan "Stok tidak mencukupi"
8	Mengurangi jumlah obat yang telah masuk dalam <i>data grid</i>	menekan salah satu data di <i>data grid</i> kemudian mengisi jumlah yang ingin dikurangi dan menekan tombol kurang	jumlah obat berkurang dalam <i>data grid</i>
9	Menghapus obat yang telah masuk dalam <i>data grid</i>	menekan salah satu data di <i>data grid</i> kemudian mengisi jumlah obat sesuai dengan jumlah obat dalam <i>data grid</i> dan menekan tombol kurang	obat yang ada dalam <i>data grid</i> terhapus
10	Memberikan informasi obat dengan tanggal <i>expired</i> tercepat yang harus diambil terlebih dulu	Menekan tombol rak	menampilkan <i>message box</i> informasi obat, tanggal <i>expired</i> , dan jumlahnya
11	Mencetak faktur penjualan dengan data yang valid	Jumlah pembayaran dimasukkan dalam <i>textbox</i> bayar, kemudian tekan tombol "Bayar"	Menyimpan seluruh data transaksi dan menampilkan faktur penjualan

No.	Tujuan	Input	Output yang Diharapkan
12	Mencetak faktur penjualan dengan data yang tidak valid(tidak lengkap)	Jumlah pembayaran tidak diisi dalam <i>textbox</i> bayar, kemudian tekan tombol "Bayar"	Muncul pesan "Data Belum Lengkap..!!!".

1. Perancangan Uji Coba Menu Laporan *History* Penjualan

Menu laporan *history* penjualan digunakan untuk mencetak laporan *history* transaksi penjualan yang terjadi sesuai data yang diujikan pada tabel 3.37.

Rancangan uji coba menu transaksi penjualan dapat dilihat pada tabel 3.38.

Tabel 3.37 Data Uji Coba Menu Laporan *History* Penjualan

Kategori yang dipilih	Tanggal
Periode	01-07-2016 s/d 31-07-2016
Pelanggan	Ali

Tabel 3.38 Rancangan Uji Coba Menu Laporan *History* Penjualan

No.	Tujuan	Input
1	Mengetahui respon sistem pertama kali dijalankan.	-
2	Mengetahui respon sistem setelah menekan tombol "Tampil" dengan data valid	Memasukkan nama pelanggan pada <i>textbox</i> pelanggan
3	Mengetahui respon sistem setelah menekan tombol "Tampil" dengan data tidak valid	Memasukkan periode tanggal pada <i>textbox</i>
4	Mengetahui respon sistem setelah menekan tombol "Tampil" dengan data valid	Tidak memasukkan nama pelanggan pada <i>textbox</i> pelanggan
5	Mengetahui respon sistem setelah menekan tombol "Tampil" dengan data tidak valid	Tidak memasukkan periode pada <i>datetimepicker</i>
6	Mengetahui respon sistem setelah menekan tombol "Cetak" dengan data valid	menekan tombol "Cetak" setelah menampilkan laporan ke <i>datagrid view</i>

No.	Tujuan	Input
7	Mengetahui respon sistem setelah menekan tombol "Cetak" dengan data tidak valid	menekan tombol "Cetak" tanpa memilih data

m. Perancangan Uji Coba Menu Laporan Penjualan

Menu laporan penjualan digunakan untuk mencetak laporan penjualan sesuai masukkan data yang terlihat pada tabel 3.39. Rancangan uji coba menu transaksi penjualan dapat dilihat pada tabel 3.40.

Tabel 3.39 Data Uji Coba Menu Laporan Penjualan

Kategori yang dipilih	Tanggal
Periode Bulanan	01-06-2016 s/d 31-08-2016
Periode Harian	01-06-2016 s/d 31-07-2016
Periode Mingguan	01-06-2016 s/d 31-08-2016

Tabel 3.40 Rancangan Uji Coba Menu Laporan Penjualan

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Form laporan penjualan tampil.
2	Mengetahui respon sistem setelah menekan tombol "Cetak" pada <i>groupbox</i> Bulanan	Memasukkan periode tanggal pada <i>groupbox</i> bulan	Form laporan penjualan dengan <i>crystal report viewer</i> sesuai periode bulan yang dimasukkan.
3	Mengetahui respon sistem setelah menekan tombol "Cetak" pada <i>groupbox</i> harian	Memasukkan periode tanggal pada <i>groupbox</i> harian	Form laporan penjualan dengan <i>crystal report viewer</i> sesuai periode hari yang dimasukkan.

No.	Tujuan	Input	Output yang Diharapkan
4	Mengetahui respon sistem setelah menekan tombol "Cetak" pada <i>groupbox</i> mingguan	Memasukkan periode tanggal pada <i>groupbox</i> mingguan	<i>Form</i> laporan penjualan dengan <i>crystal report viewer</i> sesuai periode yang dimasukkan.

n. Perancangan Uji Coba Menu Laporan Profit Penjualan

Menu laporan profit penjualan digunakan untuk mencetak laporan profit penjualan sesuai masukkan yang dimasukkan oleh pengguna. Rancangan data uji coba menu transaksi penjualan dapat dilihat pada tabel 3.41. Rancangan uji coba menu transaksi penjualan dapat dilihat pada tabel 3.42.

Tabel 3.41 Data Uji Coba Menu Laporan Profit Penjualan

Kategori yang dipilih	Tanggal
Periode Bulanan	01-06-2016 s/d 31-08-2016

Tabel 3.42 Rancangan Uji Coba Menu Laporan Profit Penjualan

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	<i>Form</i> laporan profit penjualan tampil.
2	Mengetahui respon sistem setelah menekan tombol "Cetak"	Memasukkan periode tanggal pada <i>textbox</i> bulan	<i>Form</i> laporan penjualan dengan <i>crystal report viewer</i> sesuai periode bulan yang dimasukkan.

o. Perancangan Uji Coba Menu Laporan Peringkat Penjualan

Menu laporan peringkat penjualan digunakan untuk mencetak laporan peringkat penjualan obat terbanyak sesuai masukkan periode yang dapat dilihat pada tabel 3.43. Peringkat penjualan disajikan dalam dua jenis laporan

yaitu berdasarkan pendapatan terbanyak dan berdasarkan profit terbanyak.

Rancangan uji coba menu transaksi penjualan dapat dilihat pada tabel 3.44.

Tabel 3.43 Data Uji Coba Menu Laporan Peringkat Penjualan

Kategori yang dipilih	Tanggal
Pendapatan	01-06-2016 s/d 31-08-2016
Profit	01-06-2016 s/d 31-08-2016

Tabel 3.44 Rancangan Uji Coba Menu Laporan Peringkat Penjualan

No.	Tujuan	Input	Output yang Diharapkan
1	Mengetahui respon sistem pertama kali dijalankan.	-	Form laporan pendapatan penjualan tampil.
2	Mengetahui respon sistem setelah menekan tombol "Cetak"	Memasukkan periode tanggal pada <i>groupbox</i> pendapatan	Form laporan penjualan dengan <i>crystal report viewer</i> sesuai periode yang dimasukkan.
3	Mengetahui respon sistem setelah menekan tombol "Cetak"	Memasukkan periode tanggal pada <i>groupbox</i> profit	Form laporan penjualan dengan <i>crystal report viewer</i> sesuai periode yang dimasukkan.