BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem Yang Sedang Berjalan

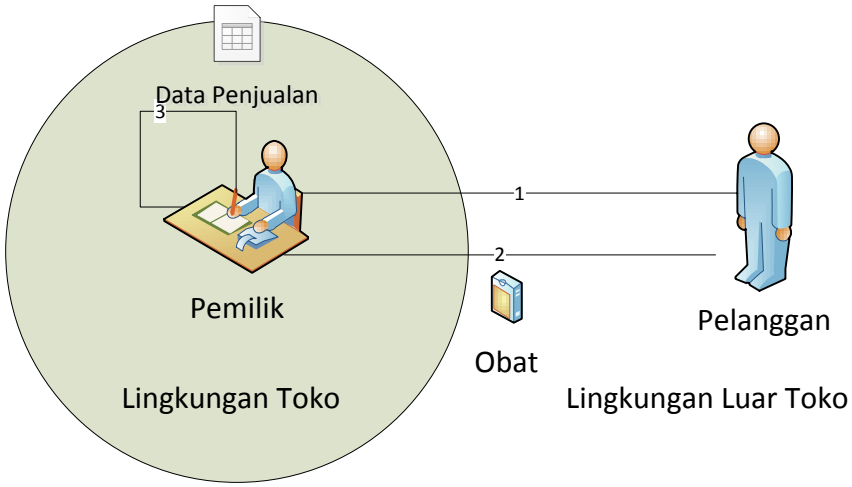
Kegiatan analisis sistem yang berjalan dilakukan dengan analisis yang berorientasi pada objek-objek yang diperlukan oleh sistem yang dirancang, dimaksudkan untuk menitik beratkan kepada fungsi sistem yang berjalan dengan tidak terlalu menitik beratkan kepada alur proses dari sistem. Selanjutnya hasil analisis ini digambarkan dan didokumentasikan dengan metodologi berorientasi objek melalui diagram *use case*, skenario *use case* dan aktifitas diagram. Pertimbangan penggunaan diagram ini karena dianggap diagram-diagram tersebut mewakili secara keseluruhan sistem yang berjalan dan dapat dimengerti oleh *user*.

4.1.1 Analisis Kebutuhan

Sebelum membuat suatu sistem, hendaknya melakukan analisis terlebih dahulu terhadap kebutuhan-kebutuhan apa saja yang diperlukan dengan menggunakan metode-metode yang telah ada.

4.1.1.1 Proses Bisnis

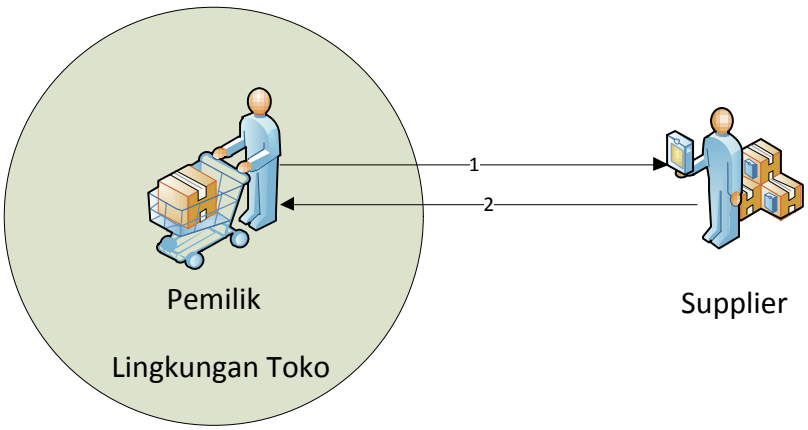
Proses bisnis dari sistem yang sedang berjalan adalah sebagai berikut:



Gambar 4.1 Proses Bisnis sistem yang sedang berjalan untuk penjualan obat

Keterangan:

1. Pelanggan datang ke toko obat Hikmah Jaya dan meminta obat yang ingin dibeli.
2. Penjaga toko memeriksa barang dan menyerahkan obat yang diminta.
3. Penjaga toko mencatat data penjualan barang.



Gambar 4.2 Proses Bisnis sistem yang sedang berjalan untuk penjualan obat

Keterangan:

1. Pemilik membeli obat-obat yang diinginkan (umumnya hanya berdasarkan obat apa yang tidak ada saja, dan datanya didapatkan setelah ada pelanggan yang meminta obat tersebut pada saat kondisi obat kosong atau setelah pengecekan rutin).
2. Supplier memberikan obat yang diminta.

4.1.1.2 Service Time

Sistem penjualan yang ada pada saat ini di Toko Obat Hikmah Jaya masih diniliai kurang efektif, dikarenakan dalam pengecekan ketersediaan barang, penjaga toko masih harus memeriksa secara langsung untuk mencari obat, dan untuk memeriksa tanggal kadaluarsa dari setiap obat masih dilakukan secara manual.

Sistem pembelian barang yang ada juga masih dinilai kurang efektif dikarenakan proses pembelian baru dilakukan pada saat persediaan barang sudah

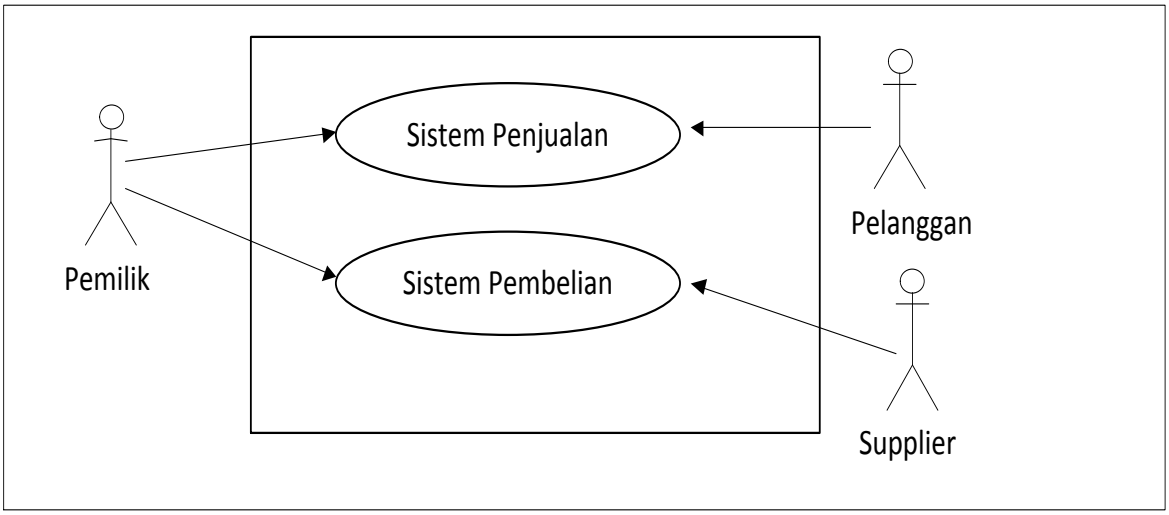
tidak ada ataupun setelah pemeriksaan rutin. Sehingga tidak jarang pada saat pelanggan datang untuk membeli obat, obat tersebut tidak tersedia. Hal ini dikarenakan masih belum adanya sistem yang digunakan untuk menangani pendataan *inventory* obat pada toko.

Proses yang dijalankan masihlah belum efisien, baik dari segi waktu ataupun biaya. Baik proses pembelian stok barang ataupun penjualan yang dilakukan pada saat ini masih berupa manual, sehingga sering ada permasalahan dalam pembuatan laporan, pengecekan ketersediaan barang, dan pembelian stok barang yang terlambat ataupun proses pembelian yang harus dilakukan beberapa kali karena data obat yang kosong umumnya diketahui saat ada pelanggan yang meminta obat tersebut.

4.1.1.3 Use Case Diagram

Use case diagram (diagram *use case*) adalah diagram yang menyajikan interaksi antara *use case* dan *actor*. Dimana *actor* dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai.

Berilkut ini adalah gambar model *Use Case Diagram* Sistem Toko Obat pada Toko Obat Hikmah Jaya yang sedang berjalan :



Gambar 4.3 *Use Case* Sistem yang sedang berjalan

4.1.1.4 Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun tahapan dari skenario *use case* pada sistem yang sedang berjalan adalah sebagai berikut:

Nama *use case*: Sistem Penjualan

Tabel 4. 1 Skenario *use case* sistem penjualan yang sedang berjalan

Judul	Sistem Penjualan
No. *Use case*	UC-TOKOOBAT-01
Deskripsi *Use Case*	Aktor Pelanggan datang pada sistem penjualan dan meminta obat yang diinginkan pada pemilik toko, pemilik kemudian mencari obat yang diinginkan dan memberikannya pada pelanggan.
Aktor	Pemilik, Pelanggan
Trigger	Pelanggan meminta obat pada pemilik toko.

Kondisi Awal	Pemilik toko stand by pada toko.
Skenario Normal	
Aksi- Aktor (Pelanggan)	**Reaksi Sistem (Pemilik)**
1. Aktor Pelanggan meminta obat pada pemilik toko.	
	2. Pemilik toko memeriksa ketersediaan barang.
	3. Sistem (Pemilik) : a. Memberikan obat yang diminta b. Memberitahu bahwa obat yang diminta tidak tersedia.
Kondisi Akhir Skenario Normal: 1. Pelanggan: a. Menerima obat yang diinginkan. b. Menerima informasi obat yang diinginkan kosong.	

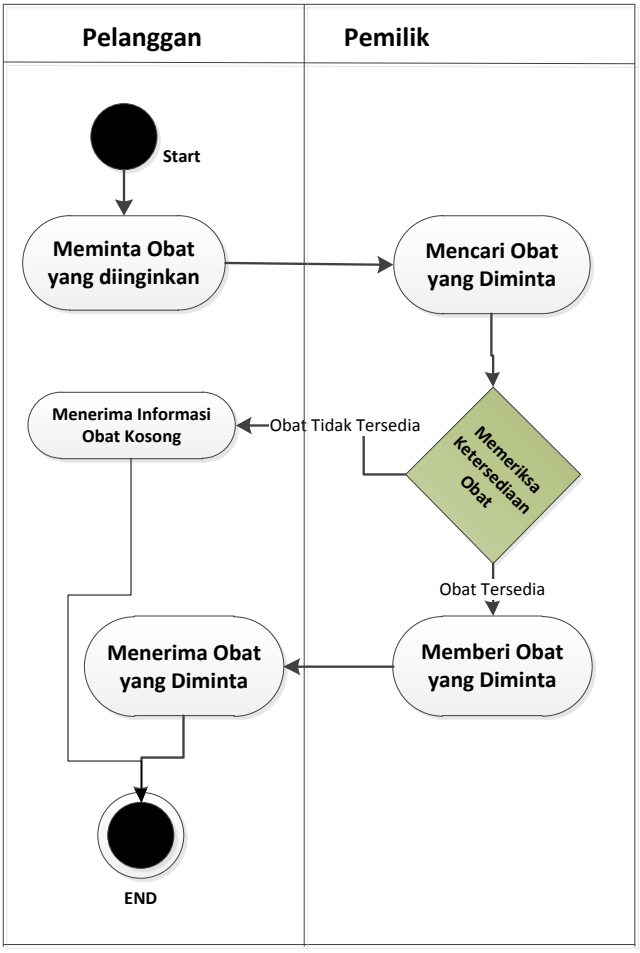
Tabel 4. 2 Skenario *use case* sistem pembelian yang sedang berjalan

Judul	Sistem Pembelian
No. *Use case*	UC-TOKOOBAT-02
Deskripsi *Use Case*	Pemilik toko melakukan pembelian obat baru
Aktor	Pemilik, *Supplier*
Trigger	Stok obat yang tersedia sudah mencapai minimum
Kondisi Awal	Pemilik toko stand by pada toko.
Skenario Normal	
Aksi- Aktor (Pemilik Toko)	**Reaksi Sistem (*Supplier*)**
1. Aktor Pemilik toko meminta obat pada *supplier*.	
	2. *Supplier* memeriksa ketersediaan barang.

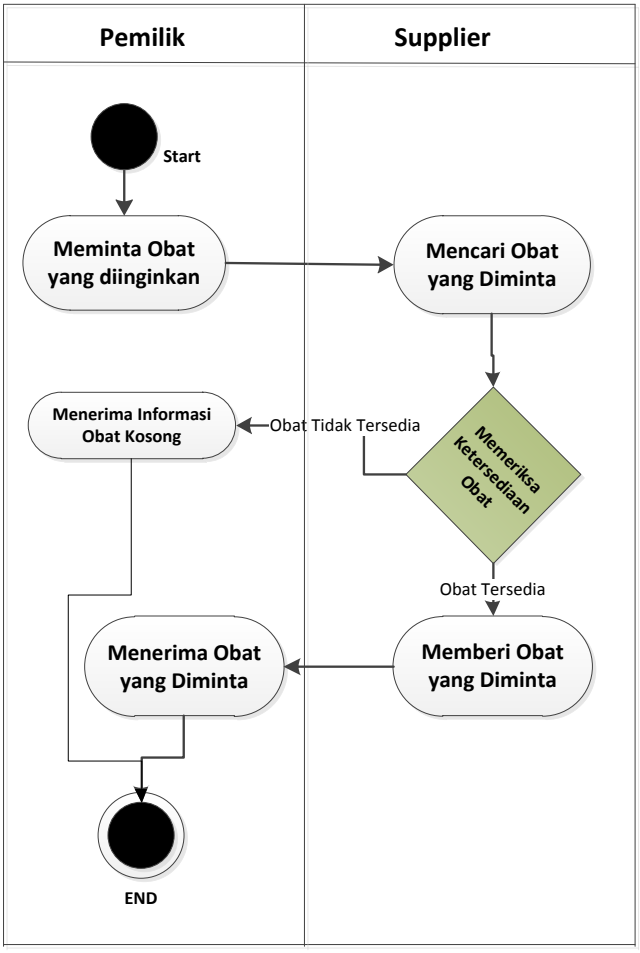
	3. Sistem (*Supplier*) : a. Memberikan obat yang diminta b. Memberitahu bahwa obat yang diminta tidak tersedia.
Kondisi Akhir Skenario Normal: 1. Pemilik Toko: a. Menerima obat yang diinginkan. b. Menerima informasi obat yang diinginkan kosong.	

4.1.1.5 Activity Diagram

Pada bagian ini akan digambarkan dokumentasi alur kerja pada sistem yang sedang berjalan yang bertujuan untuk melihat alur proses sistem yang sedang berjalan.



Gambar 4. 4 *Activity* Diagram Sistem Penjualan yang sedang berjalan



Gambar 4.5 *Activity* Diagram Sistem Pembelian yang sedang berjalan

4.1.2 Evaluasi Sistem Yang Sedang Berjalan

Setelah dilakukan penelitian pada proses penjualan dan pembelian di Toko Obat Hikmah Jaya, terdapat beberapa kelemahan yang ditemukan pada sistem yang sedang berjalan. Kelemahan-kelemahan tersebut adalah sebagai berikut:

1. Pada sistem yang sedang berjalan masih belum ada dibuat sebuah sistem yang menangani database *inventory* obat, dan proses masih dilakukan secara manual, sehingga tidak jarang tidak diketahui bahwa ada data obat yang sudah kosong.

2. Tidak adanya sistem *inventory* menyebabkan sering tidak diketahui barang mana yang baru masuk dan yang sudah lama, sehingga banyak obat yang menjadi kadaluarsa dan merugikan toko.
3. Sistem penjualan yang manual masih belum efisien dari segi waktu, karena pemilik yang bertindak sebagai penjaga toko harus melakukan pemeriksaan secara langsung untuk mengetahui ketersediaan barang.
4. Sistem pembelian yang manual kerap membuat pemilik toko harus melakukan pembelian pada *supplier* yang sama untuk beberapa kali dikarenakan terkadang ada data *supply* obat kosong yang tidak diketahui sampai ada pelanggan yang menanyakan ketersediaan obat.
5. Sistem pembelian dan penjualan yang dilakukan secara manual dan belum adanya sistem pelaporan membuat proses penghitungan keuntungan sering mengalami kesalahan.

Melihat kasus yang terjadi diatas, untuk itu penulis mencoba memberikan solusi untuk mengatasi permasalahan tersebut, yakni dengan membangun sebuah sistem yang dapat menangani:

1. Adanya sistem yang menangani database obat untuk pengetahuan data *inventory* yang ada.
2. Adanya sistem yang mampu menangani sistem transaksi penjualan.
3. Adanya sistem yang mampu menangani perencanaan pembelian obat berdasarkan data stok barang yang ada.
4. Adanya sistem yang mampu menangani data pembelian obat.

5. Adanya sistem yang mampu menangani pembuatan pelaporan, baik pembelian ataupun penjualan barang.
6. Adanya sistem yang menyimpan data dari setiap *supplier* dan jenis obat yang di*supply*.

4.2 Perancangan Sistem

Pada tahap perancangan sistem ini akan dijelaskan mengenai perancangan sistem pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu.

4.2.1 Tujuan Perancangan Sistem yang Diusulkan

Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui sistem yang ada untuk meningkatkan kinerja sistem itu sendiri, agar dapat memenuhi hasil yang diinginkan. Rancangan sistem yang baru, akan diterapkan suatu kegiatan untuk menemukan dan mengembangkan metoda, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai.

Adapun tujuan dari perancangan sistem ini adalah untuk menghasilkan suatu rancangan sistem yang memperbaharui sistem yang sudah ada untuk memaksimalkan proses penjualan, pembelian, dan penanganan *inventory* obat yang sedang berjalan pada Toko Obat Hikmah Jaya.

4.2.2 Gambaran Umum Sistem yang Diusulkan

Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang digunakan untuk mengolah data obat pada Toko Obat Hikmah Jaya dan menangani proses penjualan, pembelian dan pelaporan dengan harapan mampu menangani permasalahan yang ada pada sistem yang sedang berjalan, terkhusus pada optimalisasi waktu yang dibutuhkan dan membantu dalam proses penjualan, pembelian dan pelaporan.

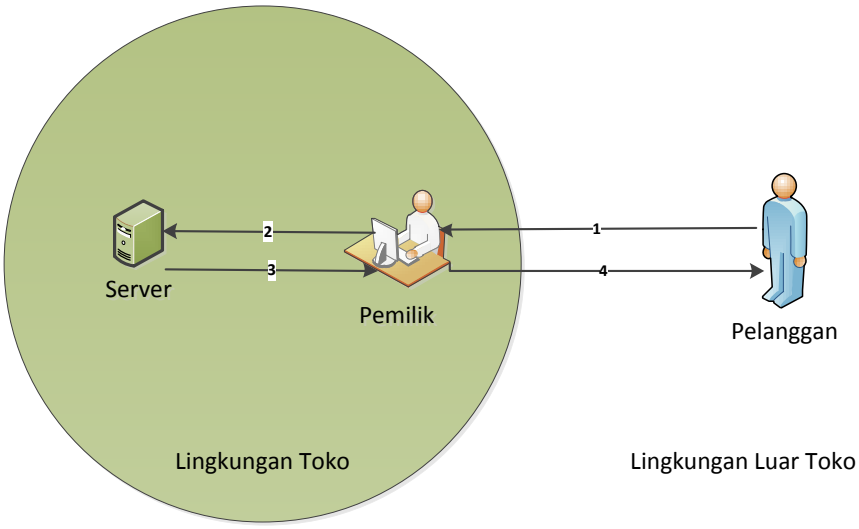
Sistem ini diharapkan akan membantu efektifitas dan optimalisasi waktu serta menghindari adanya ketidaksesuaian data atau kemungkinan data *miss* pada Toko Obat Hikmah Jaya. Sistem informasi ini diharapkan dapat membantu mengatasi masalah yang ada, dan dapat menghasilkan informasi yang cepat, tepat, dan akurat.

4.2.3 Perancangan Prosedur yang Diusulkan

Perancangan Prosedur merupakan awal dari pembuatan sistem yang akan dibuat, dimana dapat dilihat proses-proses apa saja yang nantinya diperlukan dalam pembuatan suatu sistem. Sedangkan perancangan prosedur yang diusulkan merupakan tahap untuk memperbaiki atau meningkatkan efisiensi kerja. Tahap perancangan sistem yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang baik. Sistem yang dirancang tersebut menjadi satu komponen. Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan UML.

4.2.3.1 Proses Bisnis

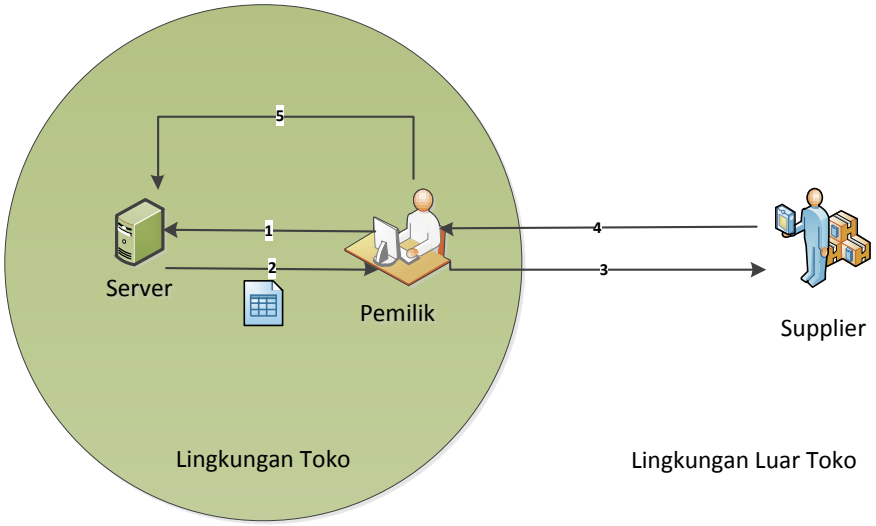
Proses bisnis dari target sistem adalah sebagai berikut:



Gambar 4.6 Proses Bisnis Target Sistem Penjualan

Keterangan:

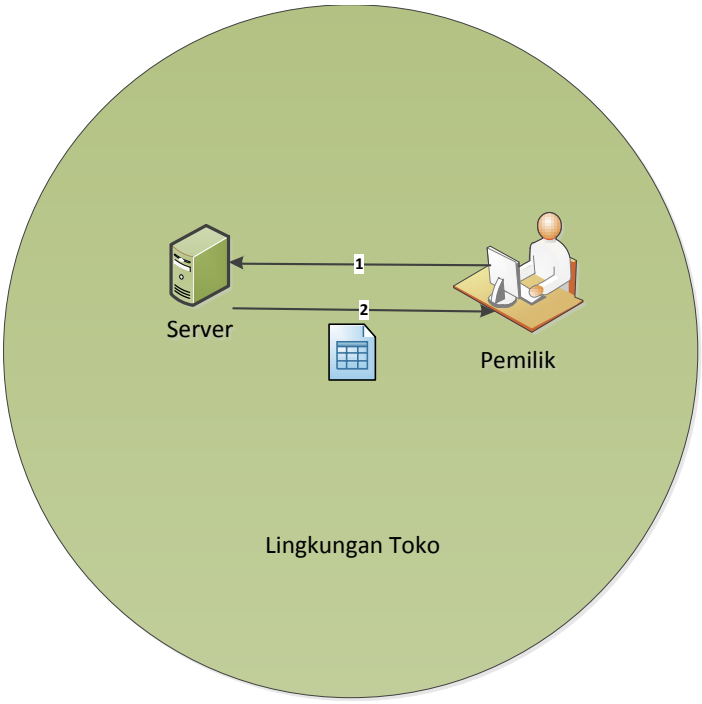
1. Pelanggan meminta obat yang diinginkan.
2. Pemilik memeriksa ketersediaan data obat pada applikasi.
3. Sistem memberikan data ketersediaan data obat yang diminta.
4. Jika obat tersedia, pemilik mengambil obat dan memberikannya pada pelanggan dan menginput data transaksi, jika tidak ada maka pemilik memberitahu pada pelanggan bahwa obat tidak tersedia.



Gambar 4 7 Proses Bisnis Target Sistem Pembelian

Keterangan:

1. Pemilik meminta data perencanaan pembelian pada applikasi.
2. Applikasi menampilkan data perencanaan pembelian pada pemilik.
3. Pemilik meminta obat sesuai dengan data perencanaan pembelian kepada *supplier*.
4. Jika obat tersedia, *supplier* memberikannya pada pemilik, jika tidak ada maka *supplier* memberitahu pada pemilik bahwa obat tidak tersedia.
5. Pemilik menginput data pembelian obat yang baru.



Gambar 4.8 Proses Bisnis Target Sistem Untuk Pengolahan Data

Keterangan:

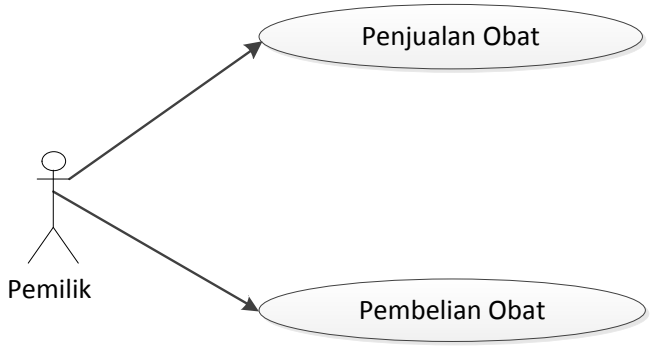
1. Pemilik melakukan pengolahan data, seperti penambahan data obat baru, data *supplier*, atau permintaan untuk laporan.
2. Applikasi menyimpan data baru yang ada atau memberikan laporan sesuai dengan yang diinginkan.

4.2.3.2 Service Time

Dengan sistem transaksi dan *inventory* yang ditargetkan, proses penjualan, pembelian, dan pelaporan akan mampu dioptimalisasikan waktu yang dibutuhkan dalam proses yang ada dan database obat akan tersimpan lebih baik. Hal ini dikarenakan proses transaksi dan data obat sudah dapat dikelola dengan baik.

4.2.3.3 Use Case Diagram

Use case diagram (diagram *use case*) adalah diagram yang menyajikan interaksi antara use case dan actor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai.



Gambar 4.9 *Use Case* Diagram untuk Target Sistem

4.2.3.4 Skenario *Use Case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut.

4.2.3.4.1 Skenario Menambah Data Obat

Tabel 4. 3 Tabel skenario *use case* menambah data Obat

Judul	Menambah data obat
No. *Use case*	UC-TOKOOBAT-01
Deskripsi *Use Case*	Aktor menambahkan data Obat.
Aktor	Pemilik
Trigger	Aktor memilih menu untuk menambahkan data obat.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data obat.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi *field* kosong pada *form* yang data dari obat yang akan ditambahkan.	
2. Aktor men-*submit* data obat yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian obat telah diisi dan sesuai.
	4. Data obat ditambahkan pada *database*.

<table>
<tr><td colspan="2">Kondisi Akhir Skenario Normal:

Data obat yang baru dimasukkan tampil pada list data obat</td></tr>
<tr><td colspan="2">Skenario Abnormal-1</td></tr>
<tr><td>Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Aktor men-submit data dengan menekan sebuah button tanpa mengisi field-field pada form.</td><td>2. Form pengisian data obat ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.</td></tr>
<tr><td></td><td>3. Data obat tidak ditambahkan pada database.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Abnormal-1:

Aplikasi menampilkan form pengisian data obat.</td></tr>
<tr><td colspan="2">Catatan</td></tr>
</table>

4.2.3.4.2 Skenario mengubah data obat

Tabel 4. 4 Skenario *use case* mengubah data obat

Judul	Mengubah data obat
No. *Use case*	UC-TOKOOBAT-02
Deskripsi *Use case*	Aktor mengubah data obat.
Aktor	Pemilik
Trigger	Aktor memilih salah satu dari data obat yang akan dirubah dan menekan tombol edit.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk mengubah data obat.

<table>
<tr><td colspan="2">Skenario Normal</td></tr>
<tr><td>Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Aktor mengubah data pada field yang diinginkan.</td><td></td></tr>
<tr><td>2. Aktor men-submit data obat yang ingin ditambahkan.</td><td>3. Sistem memeriksa apakah field-field pada form pengisian obat telah diisi dan sesuai.</td></tr>
<tr><td></td><td>4. Data obat dirubah pada database.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Normal:

Data obat yang baru dimasukkan tampil pada list data obat</td></tr>
<tr><td colspan="2">Skenario Abnormal-1</td></tr>
<tr><td>Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Aktor men-submit data dengan menekan sebuah button di mana ada field-field pada form yang seharusnya diisi tetapi dikosongkan.</td><td>2. Form pengisian data obat ditampilkan dan ada pesan kesalahan yang menyatakan field harus diisi.</td></tr>
<tr><td></td><td>3. Data obat tidak ditambahkan pada database.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Abnormal-1:

Aplikasi menampilkan form untuk mengubah data obat.</td></tr>
<tr><td colspan="2">Catatan</td></tr>
</table>

4.2.3.4.3 Menambah Data *Supplier*

Tabel 4. 5 Skenario *use case* menambah data *supplier*

<table>
<tr><td>Judul</td><td colspan="2">Menambah data supplier</td></tr>
<tr><td>No. Use case</td><td colspan="2">UC-TOKOOBAT-03</td></tr>
<tr><td>Deskripsi Use Case</td><td colspan="2">Aktor menambahkan data supplier.</td></tr>
<tr><td>Aktor</td><td colspan="2">Pemilik</td></tr>
<tr><td>Trigger</td><td colspan="2">Aktor memilih menu untuk menambahkan data supplier.</td></tr>
<tr><td>Kondisi Awal</td><td colspan="2">Aplikasi menampilkan sebuah form untuk menambahkan data supplier.</td></tr>
<tr><td colspan="3">Skenario Normal</td></tr>
<tr><td colspan="2">Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td colspan="2">1. Aktor mengisi field kosong pada form yang data dari supplier yang akan ditambahkan.</td><td></td></tr>
<tr><td colspan="2">2. Aktor men-submit data supplier yang ingin ditambahkan.</td><td>3. Sistem memeriksa apakah field-field pada form pengisian supplier telah diisi dan sesuai.</td></tr>
<tr><td colspan="2"></td><td>4. Data supplier ditambahkan pada database.</td></tr>
<tr><td colspan="3">Kondisi Akhir Skenario Normal:

Data supplier yang baru dimasukkan tampil pada list data supplier</td></tr>
<tr><td colspan="3">Skenario Abnormal-1</td></tr>
<tr><td colspan="2">Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
</table>

1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi *field-field* pada form.	2. *Form* pengisian data *supplier* ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi.
	3. Data *supplier* tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* pengisian data *supplier*.	
Catatan	

4.2.3.4.4 Mengolah Data *Supplier*

Tabel 4. 6 Skenario *use case* mengolah data supplier

Judul	Mengolah Data *Supplier*
No. *Use case*	UC-TOKOOBAT-04
Deskripsi *Use Case*	Aktor mengolah data *supplier*.
Aktor	Pemilik
Trigger	Aktor memilih salah satu data *supplier* dan menekan tombol edit.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk mengolah data *supplier*.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi *field* yang ingin dirubah pada *form* yang data dari supplier yang akan ditambahkan.	

2. Aktor men-*submit* data *supplier* yang ingin ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian *supplier* telah diisi dan sesuai.
	4. Data *supplier* ditambahkan pada *database*.
Kondisi Akhir Skenario Normal: Data *supplier* yang baru dimasukkan dan disimpan.	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* tanpa mengisi atau mengisi data yang salah *field-field* pada form.	2. *Form* pengisian data *supplier* ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi atau ada data *supplier* yang salah.
	3. Data *supplier* tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Applikasi menampilkan halaman perubahan data *supplier*	
Catatan	

4.2.3.4.5 Menambah data Penjualan

Tabel 4. 7 Skenario *use case* menambah data penjualan

Judul	Menambah data penjualan
No. *Use case*	UC-TOKOOBAT-05
Deskripsi *Use Case*	Aktor menambah data penjualan yang baru.
Aktor	Pemilik
Trigger	Aktor memilih menu untuk menambahkan data transaksi penjualan.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data Penjualan.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi data penjualan pada *field* yang disediakan.	
2. Aktor men-*submit* data penjualan yang baru ditambahkan.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data penjualan telah diisi dan sesuai.
	4. Data transaksi penjualan disimpan pada *database*.
	5. Data stok obat dikurangi sesuai dengan data penjualan baru
Kondisi Akhir Skenario Normal: Data penjualan yang baru dimasukkan tampil pada *list* data penjualan.	
Skenario Abnormal-1	

Aksi- Aktor	Reaksi Sistem
1. Aktor men-*submit* data dengan menekan sebuah *button* di mana ada *field-field* pada form yang seharusnya diisi tetapi dikosongkan.	2. *Form* pengisian data transaksi penjualan ditampilkan dan ada pesan kesalahan yang menyatakan *field* harus diisi atau tidak sesuai.
	3. Data transaksi penjualan tidak ditambahkan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Aplikasi menampilkan *form* untuk menambahkan data transaksi penjualan.	
Catatan	

4.2.3.4.6 Menambah Data Transaksi Pembelian

Tabel 4. 8 Skenario *use case* menambah data transaksi pembelian

Judul	Menambah Data Transaksi Pembelian
No. *Use case*	UC-TOKOOBAT-06
Deskripsi *Use case*	Aktor menambah data transaksi pembelian.
Aktor	Pemilik
Trigger	Aktor memilih menu untuk melakukan transaksi penambahan data pembeliaan.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data pembelian.
Skenario Normal	

Aksi- Aktor	Reaksi Sistem
1. Aktor mengisi *field* pada *form* yang data dari transaksi pembelian yang ingin ditambahkan.	
2. Aktor men-*submit* data pembelian yang baru.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data telah diisi dan sesuai.
	4. Data pembelian ditambahkan pada *database*.
	5. Data stok obat diperbaharui sesuai dengan data pembelian.
Kondisi Akhir Skenario Normal: Data kantor cabang yang baru dimasukkan tampil pada list data kantor cabang	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* di mana ada *field-field* pada form yang seharusnya diisi tetapi dikosongkan atau ada data yang tidak sesuai.	2. *Form* pengisian data pembelian ditampilkan dengan pesan bahwa ada data *field* yang harus diisi namun masih kosong, atau ada data yang tidak sesuai.
	3. Data pembelian baru tidak disimpan pada *database.*
Kondisi Akhir Skenario Abnormal-1: Form penambahan data pembelian ditampilkan	
Catatan	

4.2.3.4.7 Mencetak Rencana Beli

Tabel 4. 9 Skenario *use case Mencetak Rencana Beli*

Judul	Mencetak Rencana Beli
No. *Use case*	UC-TOKOOBAT-07
Deskripsi *Use case*	Aktor meminta sistem untuk mencetak daftar rencana pembelian.
Aktor	Pemilik
Trigger	Aktor memilih menu untuk mencetak daftar rencana pembelian.
Kondisi Awal	Aplikasi menampilkan *form* data obat.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor menekan tombol untuk mencetak daftar rencana pembelian.	
	2. Sistem menampilkan laporan data obat yang sudah kosong atau obat yang sudah mengalami kadaluarsa.
Kondisi Akhir Skenario Normal: Sistem menampilkan daftar rencana pembelian obat.	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**
Kondisi Akhir Skenario Abnormal-1:	

4.2.3.4.8 Mencetak Laporan Penjualan

Tabel 4. 10 Skenario *use case* Mencetak Laporan Penjualan

Judul	Mencetak Laporan Penjualan
No. *Use case*	UC-TOKOOBAT-08
Deskripsi *Use case*	Aktor mencetak laporan penjualan yang terjadi pada *range* waktu yang diinginkan.
Aktor	Pemilik.
Trigger	Aktor memilih menu untuk mencetak laporan penjualan.
Kondisi Awal	Aplikasi menampilkan form untuk menentukan periode laporan yang diinginkan.
Skenario Normal	
Aksi- Aktor	**Reaksi Sistem**
1. Aktor mengisi periode waktu yang diinginkan untuk laporan.	
2. Aktor menekan tombol untuk mencetak laporan	
	3. Sistem mencetak laporan penjualan berdasarkan transaksi penjualan yang terjadi pada periode yang dipilih.
Kondisi Akhir Skenario Normal: Laporan penjualan dicetak.	
Skenario Abnormal-1	
Aksi- Aktor	**Reaksi Sistem**

1. Aktor memilih periode yang tidak memiliki data penjualan samasekali.	
2. Aktor menekan tombol untuk mencetak laporan	
	3. Sistem menampilkan pesan bahwa transaksi penjualan pada periode tersebut kosong.
Kondisi Akhir Skenario Abnormal-1: Sistem menampilkan form untuk melakukan pengisian periode laporan yang diinginkan.	
Catatan	

4.2.3.4.9 Mencetak laporan pembelian

Tabel 4. 11 Skenario *use case* mencetak laporan pembelian

Judul	Mencetak laporan pembelian
No. *Usecase*	UC-TOKOOBAT-09
Deskripsi *Use Case*	Aktor mencetak laoran pembelian pada periode yang diinginkan.
Aktor	Pemilik
Trigger	Aktor memilih menu untuk mencetak laporan pembelian.
Kondisi Awal	Aplikasi menampilkan sebuah *form* untuk menentukan periode laporan penjualan yang diinginkan.

<table>
<tr><td colspan="2">Skenario Normal</td></tr>
<tr><td>Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Aktor mengisi periode laporan pembelian yang diinginkan.</td><td></td></tr>
<tr><td>2. Aktor menekan tombol untuk mencetak laporan pembelian.</td><td>3. Sistem mencetak laporan pembelian sesuai dengan periode yang diinginkan.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Normal:

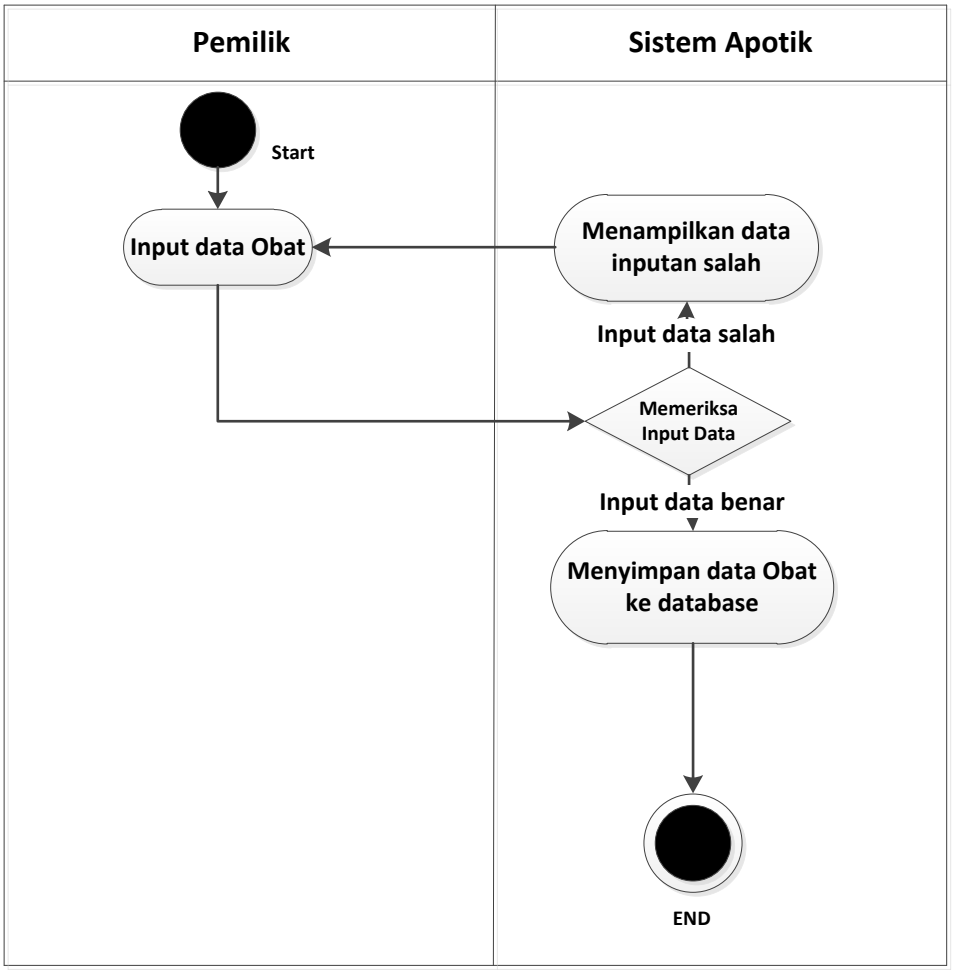
Laporan pembelian yang sesuai dengan periode yang diinginkan akan dicetak.</td></tr>
<tr><td colspan="2">Skenario Abnormal-1</td></tr>
<tr><td>Aksi- Aktor</td><td>Reaksi Sistem</td></tr>
<tr><td>1. Aktor memilih periode yang tidak memiliki data transaksi pembelian.</td><td></td></tr>
<tr><td>2. Aktor menekan tombol untuk mencetak laporan.</td><td>3. Sistem memberikan pesan bahwa transaksi pembelian pada periode yang dipilih tidak ada.</td></tr>
<tr><td></td><td>4. Laporan pembelian tidak dicetak.</td></tr>
<tr><td colspan="2">Kondisi Akhir Skenario Abnormal-1:

Aplikasi menampilkan form untuk pengisian data periode yang diinginkan.</td></tr>
<tr><td colspan="2">Catatan</td></tr>
<tr><td colspan="2"></td></tr>
</table>

4.2.3.5 *Activity Diagram*

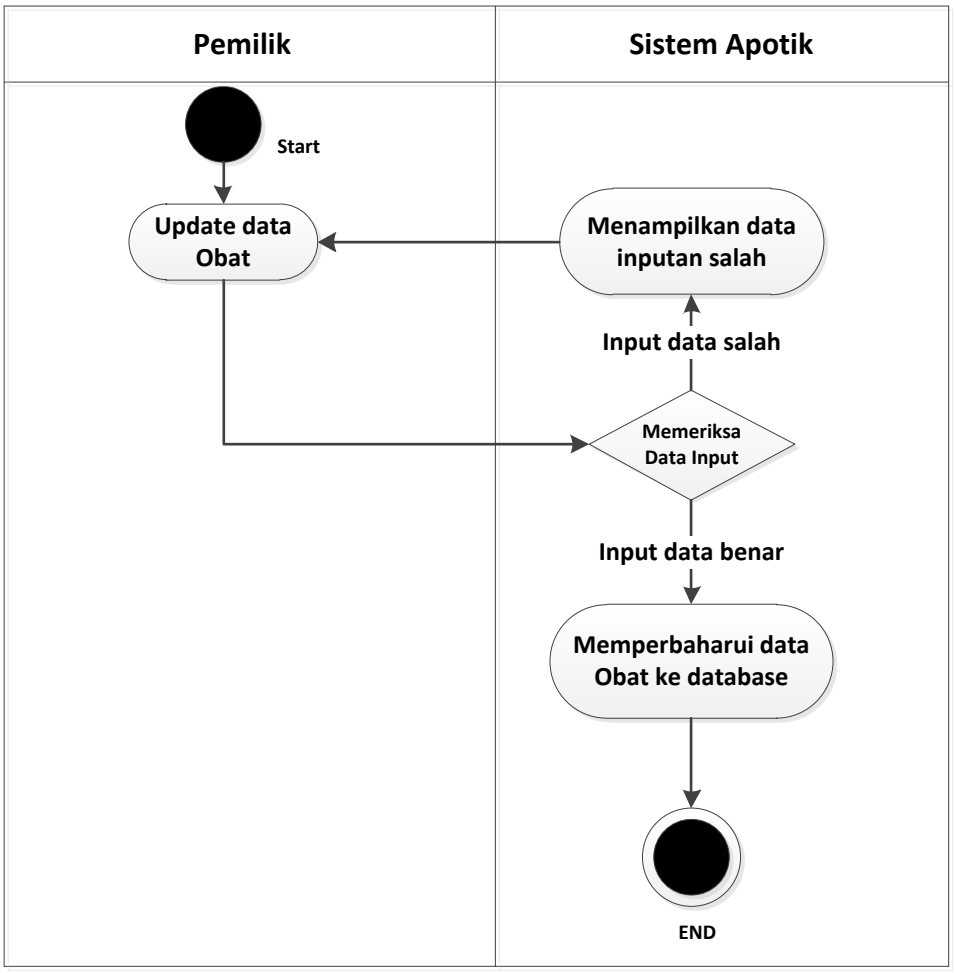
Activity diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Pada tahap pemodelan bisnis, diagram aktivitas dapat digunakan untuk menunjukkan aliran kerja bisnis *(business work flow).* Dapat juga digunakan untuk menggambarkan aliran kejadian *(flow of events).*

4.2.3.5.1 Activity Diagram Menambah Data Obat



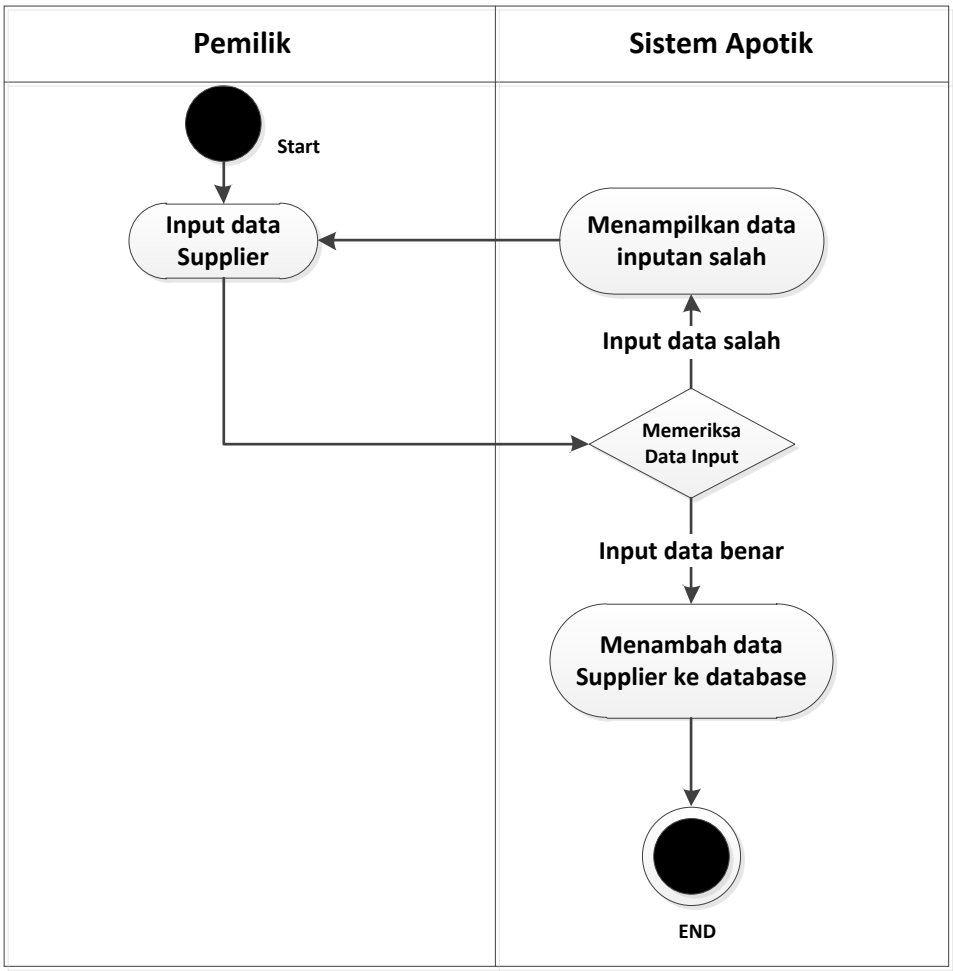
Gambar 4.10 ***Activity Diagram*** **Menambah Data Obat**

4.2.3.5.2 Activity Diagram Mengolah Data Obat



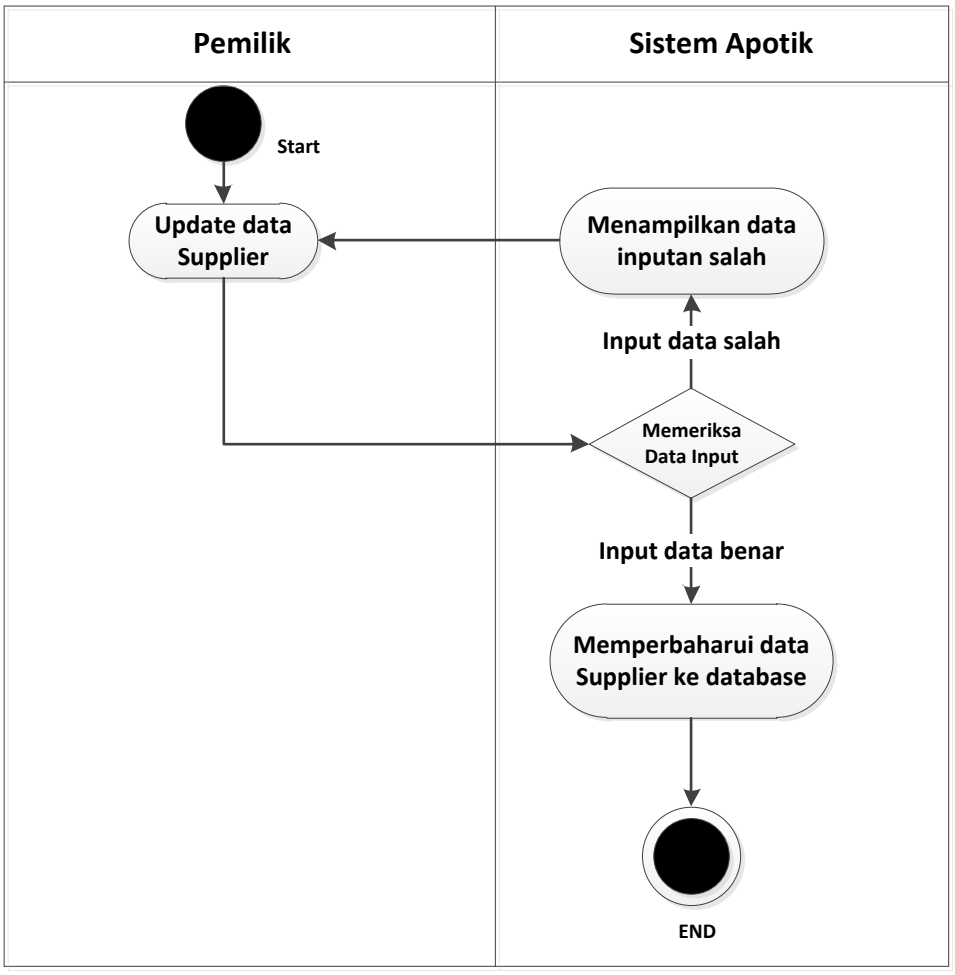
Gambar 4.11 *Activity Diagram* Mengubah Data Obat

4.2.3.5.3 Activity Diagram Menambah Data *Supplier*



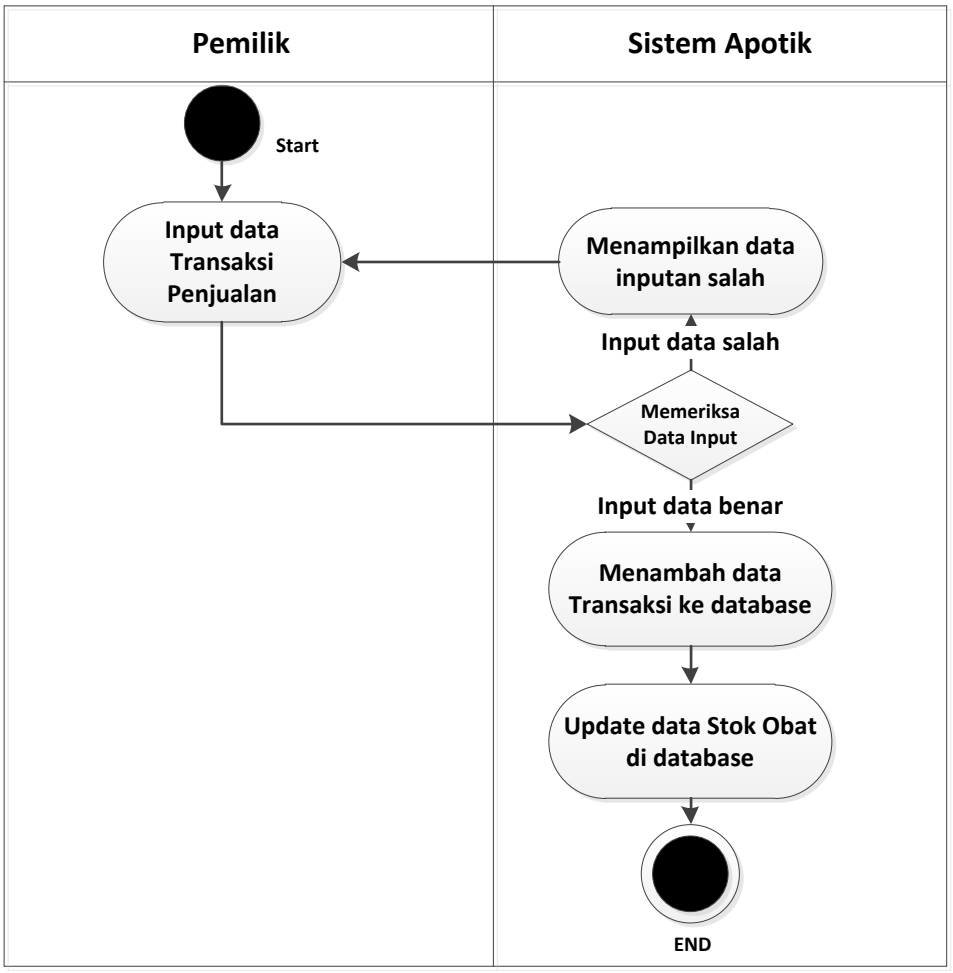
Gambar 4.12 *Activity Diagram* Menambah data *supplier*

4.2.3.5.4 Activity Diagram Mengolah Data *Supplier*



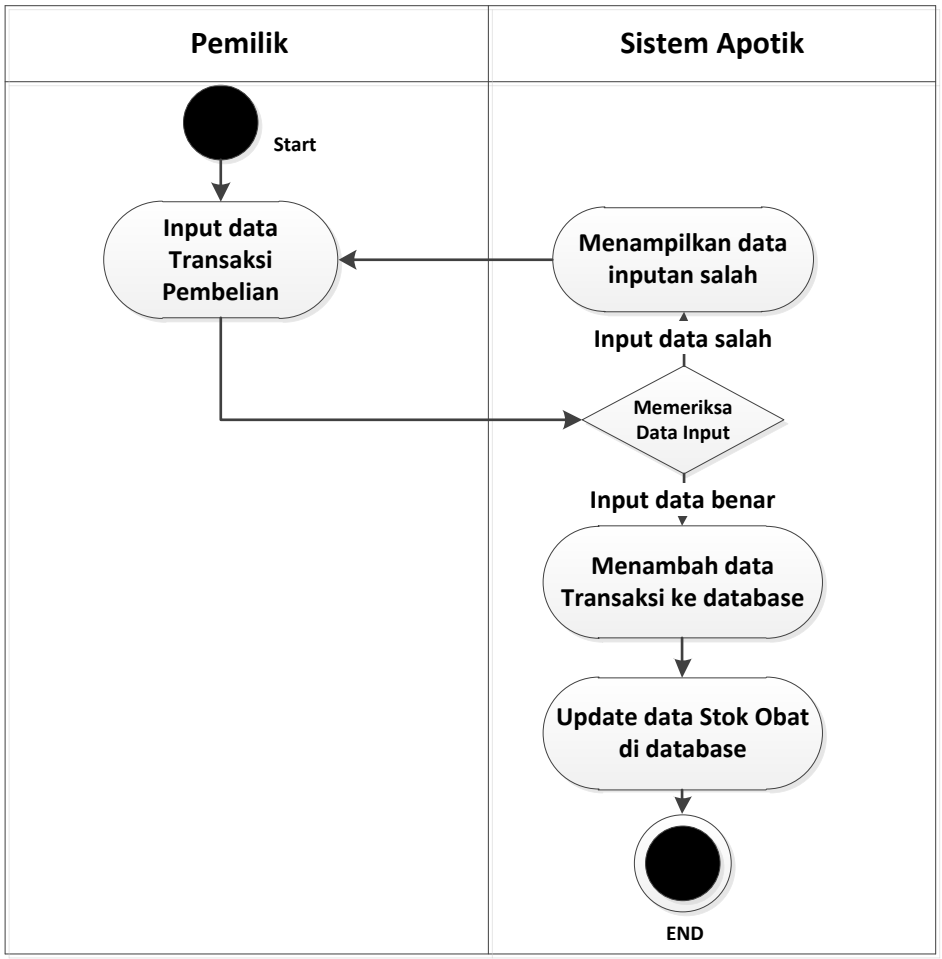
Gambar 4.13 *Activity Diagram* Mengolah Data *Supplier*

4.2.3.5.5 Activity Diagram Menambah Data Penjualan



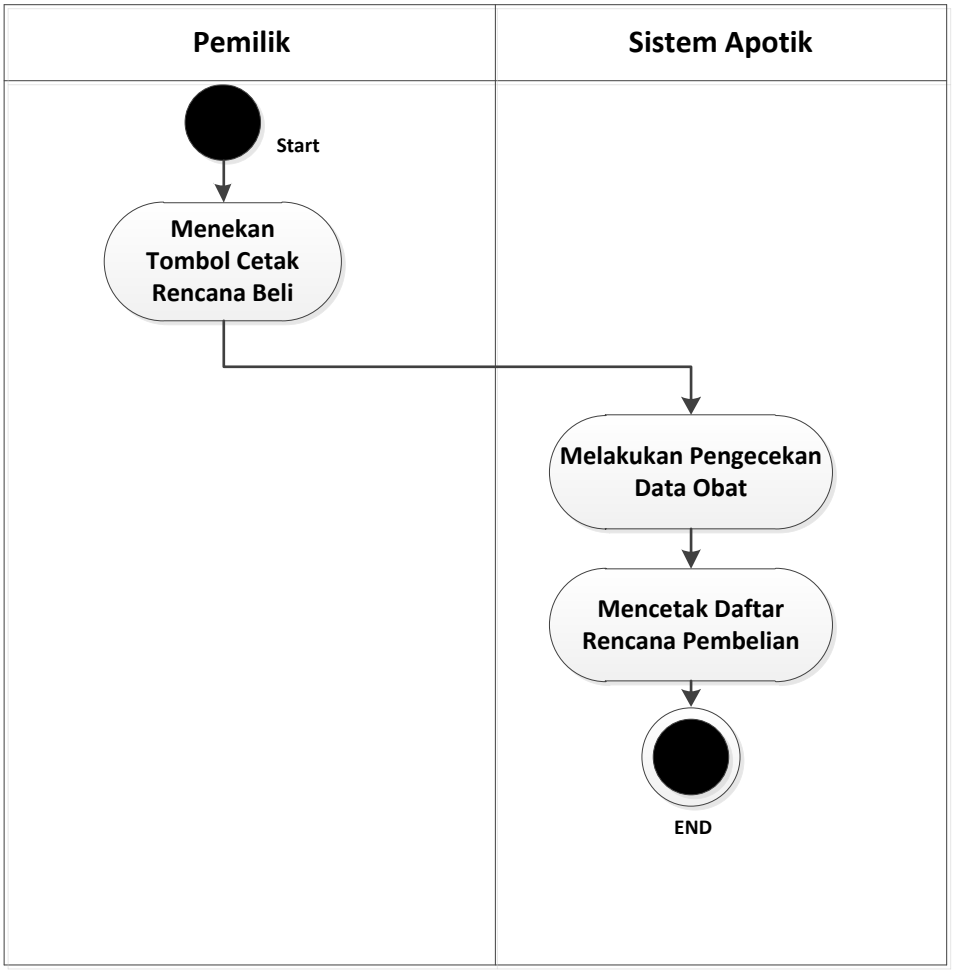
Gambar 4.14 *Activity Diagram* Menambah Data Penjualan

4.2.3.5.6 Activity Diagram Menambah Data Pembelian



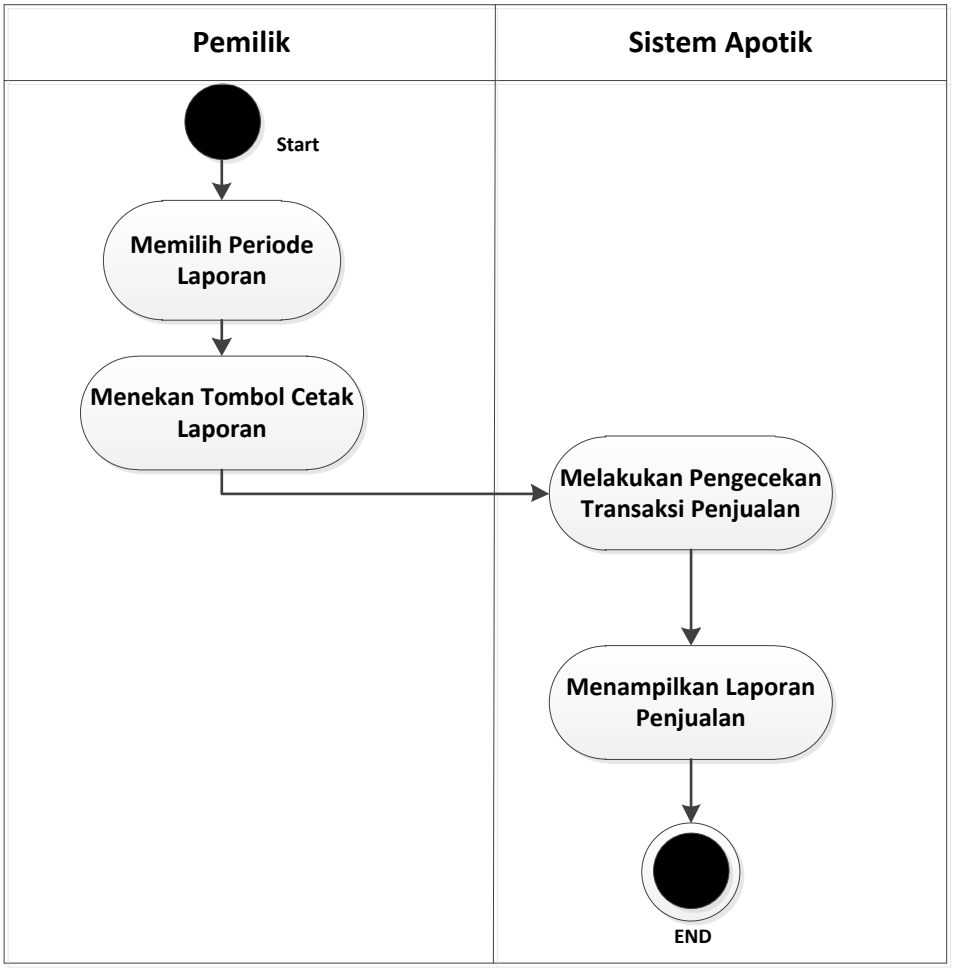
Gambar 4.15 *Activity Diagram* Menambah Data Pembelian

4.2.3.5.7 Activity Diagram Mencetak Daftar Rencana Beli



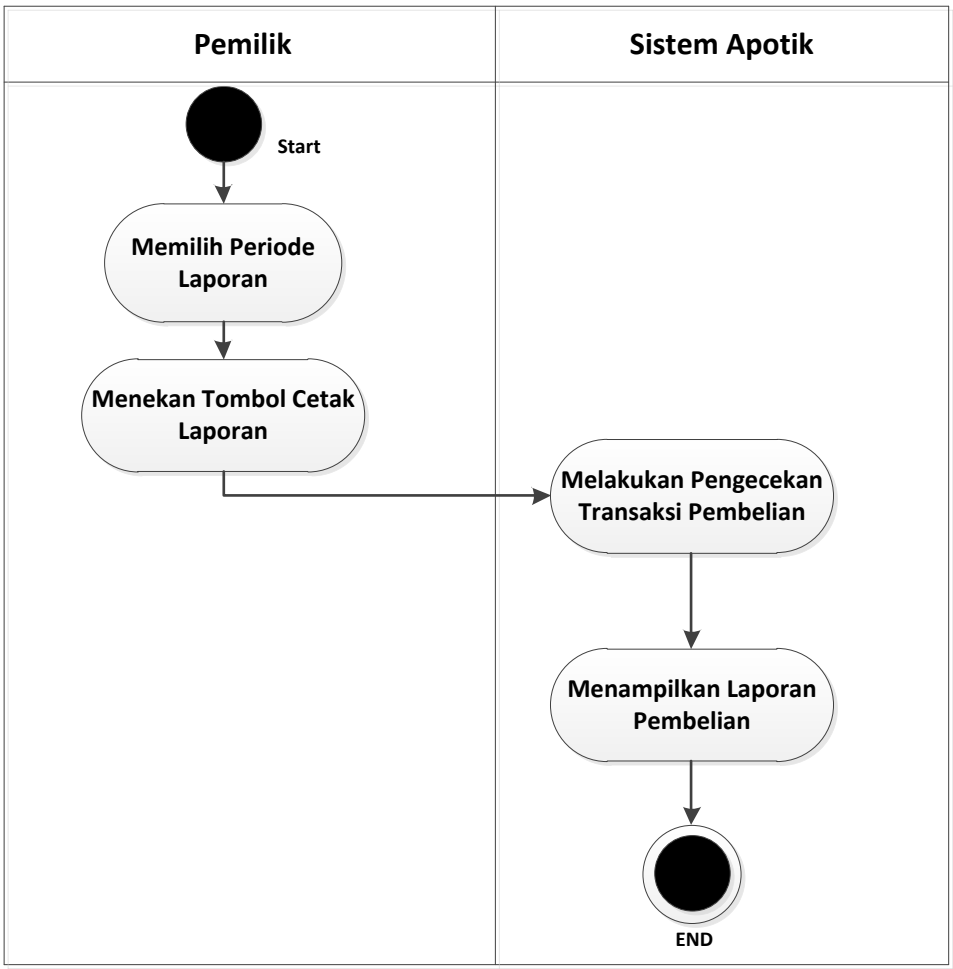
Gambar 4 16 *Activity Diagram* Mencetak Daftar Rencana Pembelian

4.2.3.5.8 Activity Diagram Mencetak Laporan Penjualan



Gambar 4.17 *Activity Diagram* Mencetak Laporan Penjualan

4.2.3.5.9 Activity Diagram Mencetak Laporan Pembelian



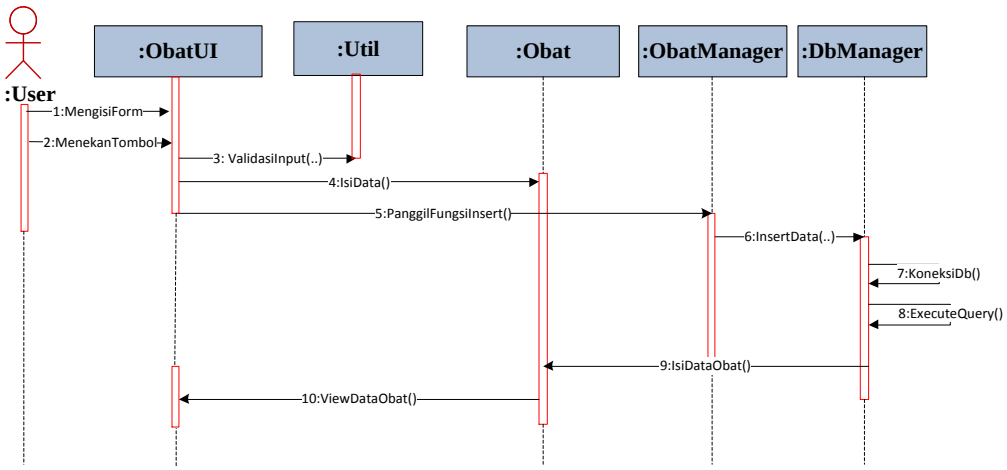
Gambar 4.18 *Activity Diagram* Mencetak Laporan Pembelian

4.2.3.6 Sequence Diagram

Pada setiap *sequence diagram* terdapat aksi aktor yang pertama sekali adalah terhadap *interface*. *Sequence Diagram* digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *sequence diagram* digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar *object*, juga interaksi antar

objek, dan menunjukkan sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama, pesan diwakili oleh garis dengan tanda panah, dan waktu yang ditunjukkan dengan proses vertikal. Berikut adalah *sequence diagram* yang ada pada sistem toko obat yang ditargetkan.

4.2.3.6.1 Sequence Diagram menambah obat



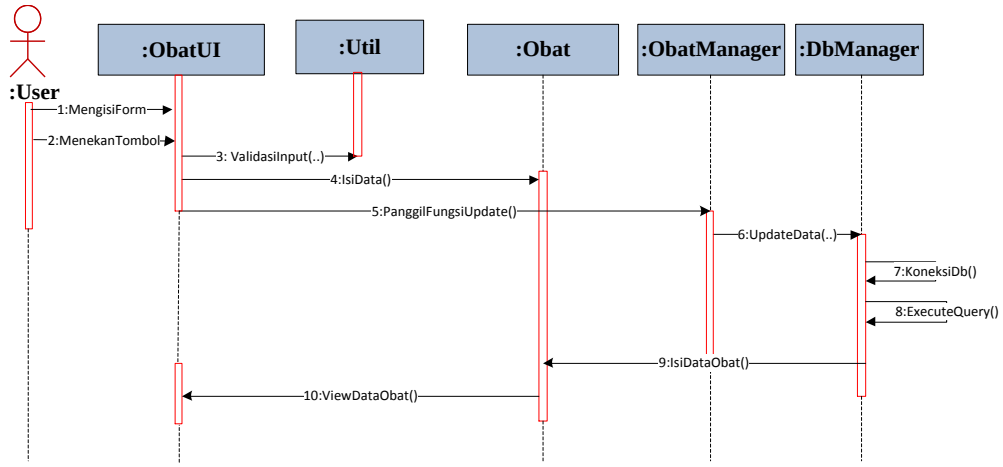
Gambar 4.19 *Sequence Diagram* menambah obat

Keterangan:

Pada Gambar 4.19 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. ObatUI
2. Util
3. Obat
4. ObatManager
5. DbManager

4.2.3.6.2 Sequence Diagram mengolah obat



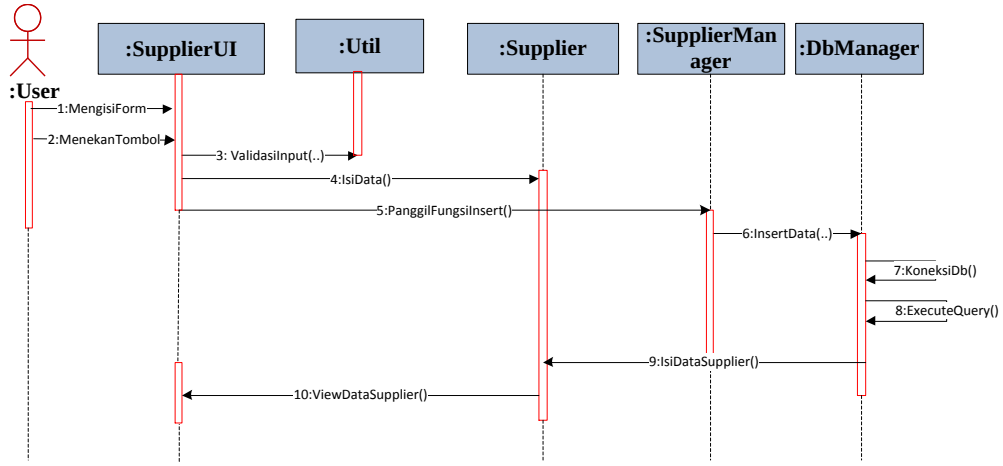
Gambar 4.20 *Sequence Diagram* mengolah data obat

Keterangan:

Pada Gambar 4.20 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. ObatUI
2. Util
3. Obat
4. ObatManager
5. DbManager

4.2.3.6.3 Sequence Diagram menambah *supplier*



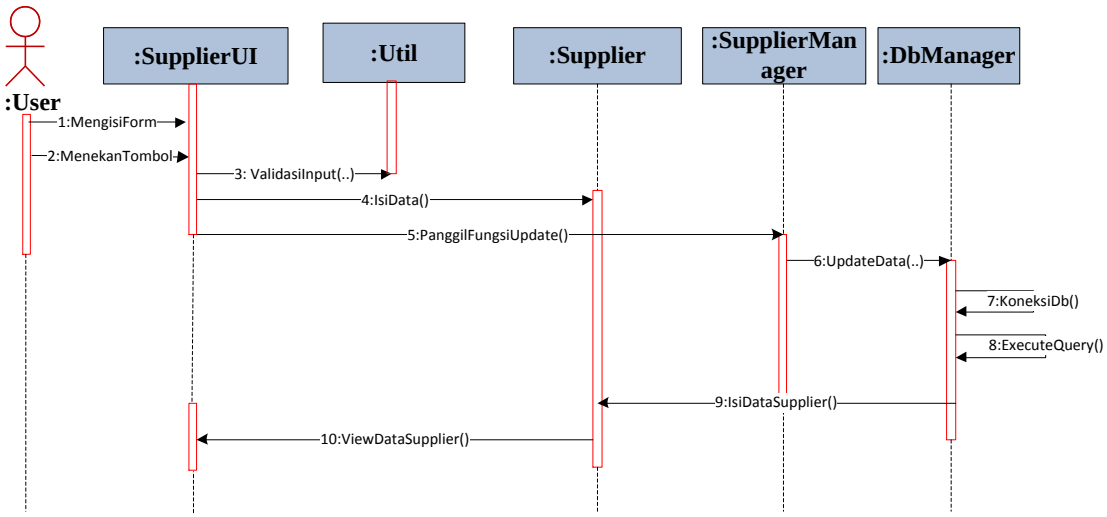
Gambar 4.21 *Sequence Diagram* menambah *supplier*

Keterangan:

Pada Gambar 4.21 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. SupplierUI
2. Util
3. Supplier
4. SupplierManager
5. DbManager

4.2.3.6.4 Sequence Diagram mengolah data *supplier*



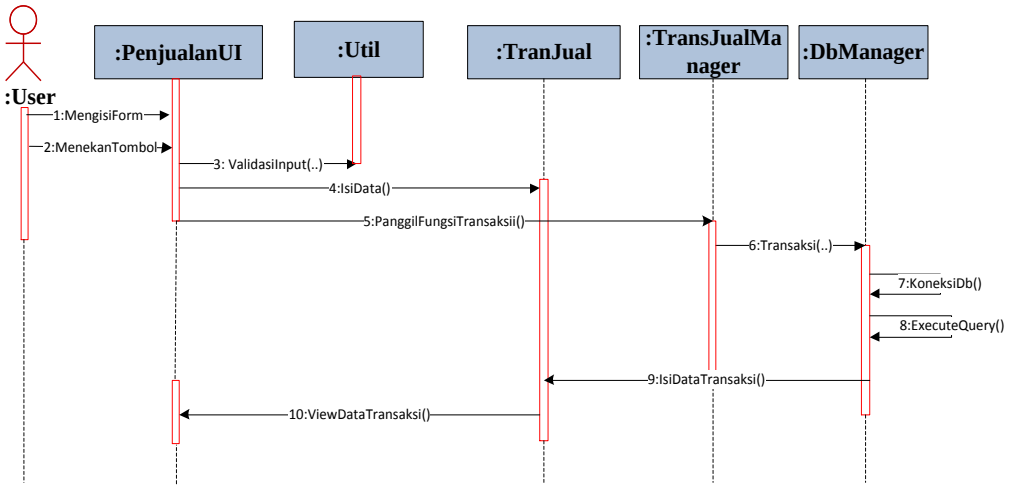
Gambar 4.22 *Sequence Diagram* mengolah data *supplier*

Keterangan:

Pada Gambar 4.22 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. SupplierUI
2. Util
3. Supplier
4. SupplierManager
5. DbManager

4.2.3.6.5 Sequence Diagram menambah penjualan



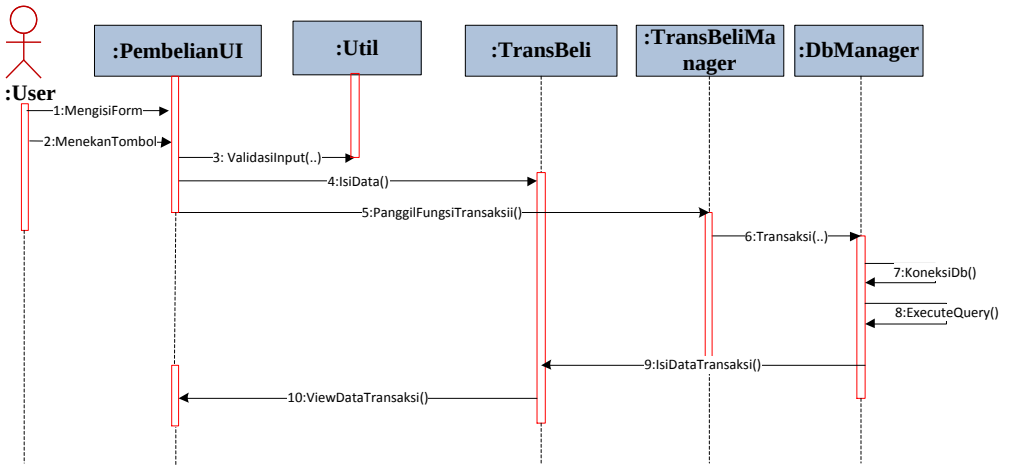
Gambar 4.23 *Sequence diagram* Mengolah Transaksi Penjualan

Keterangan:

Pada Gambar 4.23 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. PenjualanUI
2. Util
3. TransJual
4. TransJualManager
5. DbManager

4.2.3.6.6 Sequence Diagram menambah pembelian



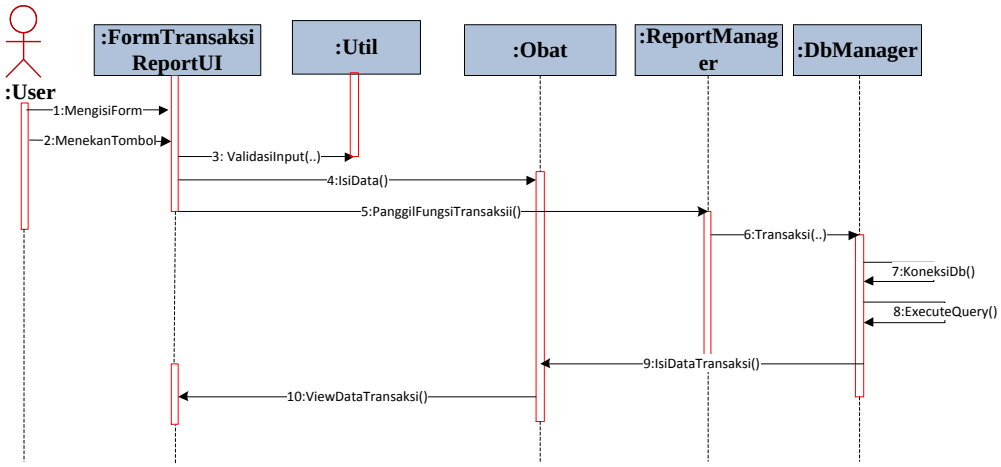
Gambar 4 24 ***Sequence diagram* Mengolah Transaksi pembelian**

Keterangan:

Pada Gambar 4.24 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. PembelianUI
2. Util
3. TransBeli
4. TransBeliManager
5. DbManager

4.2.3.6.7 Sequence Diagram mencetak rencana beli



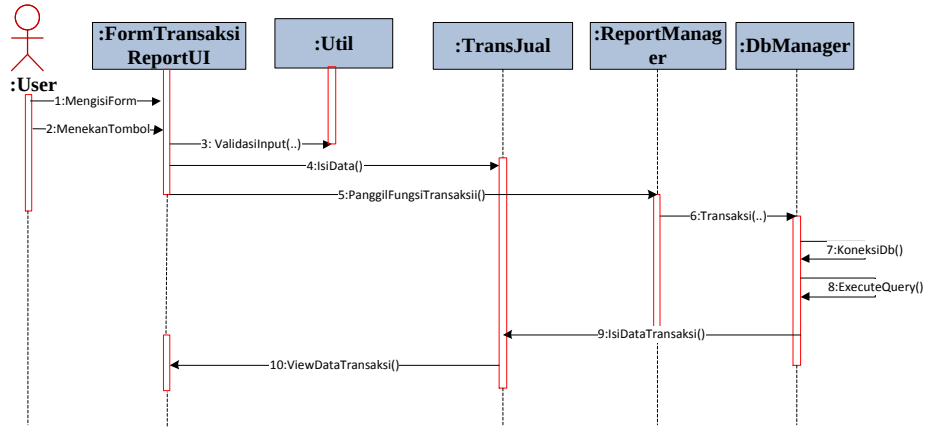
Gambar 4.25 *Sequence diagram* Mencetak rencana beli

Keterangan:

Pada Gambar 4.25 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. FormTransaksiReportUI
2. Util
3. Obat
4. ReportManager
5. DbManager

4.2.3.6.8 Sequence Diagram mencetak laporan pembelian



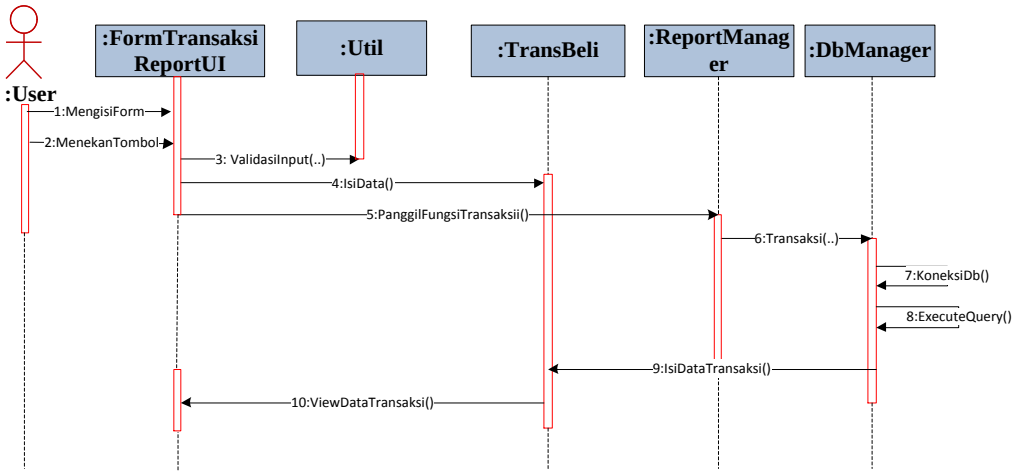
Gambar 4 26 *Sequence diagram* Mencetak rencana beli

Keterangan:

Pada Gambar 4.26 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. FormTransaksiReportUI
2. Util
3. TransJual
4. ReportManager
5. DbManager

4.2.3.6.9 Sequence Diagram mencetak laporan penjualan



Gambar 4 27 *Sequence diagram* Mencetak rencana beli

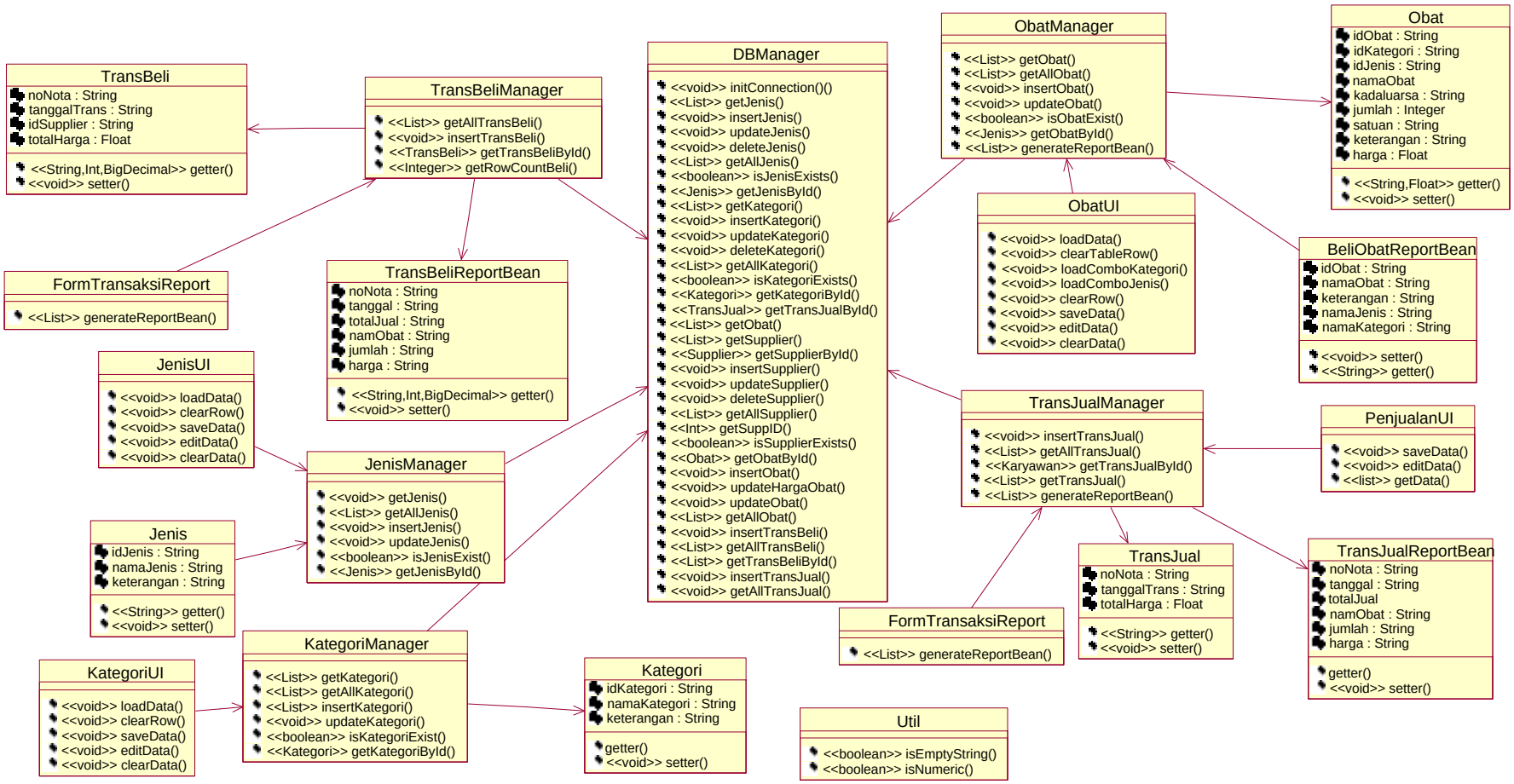
Keterangan:

Pada Gambar 4.27 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. FormTransaksiReportUI
2. Util
3. TransBeli
4. ReportManager
5. DbManager

4.2.3.7 Class Diagram

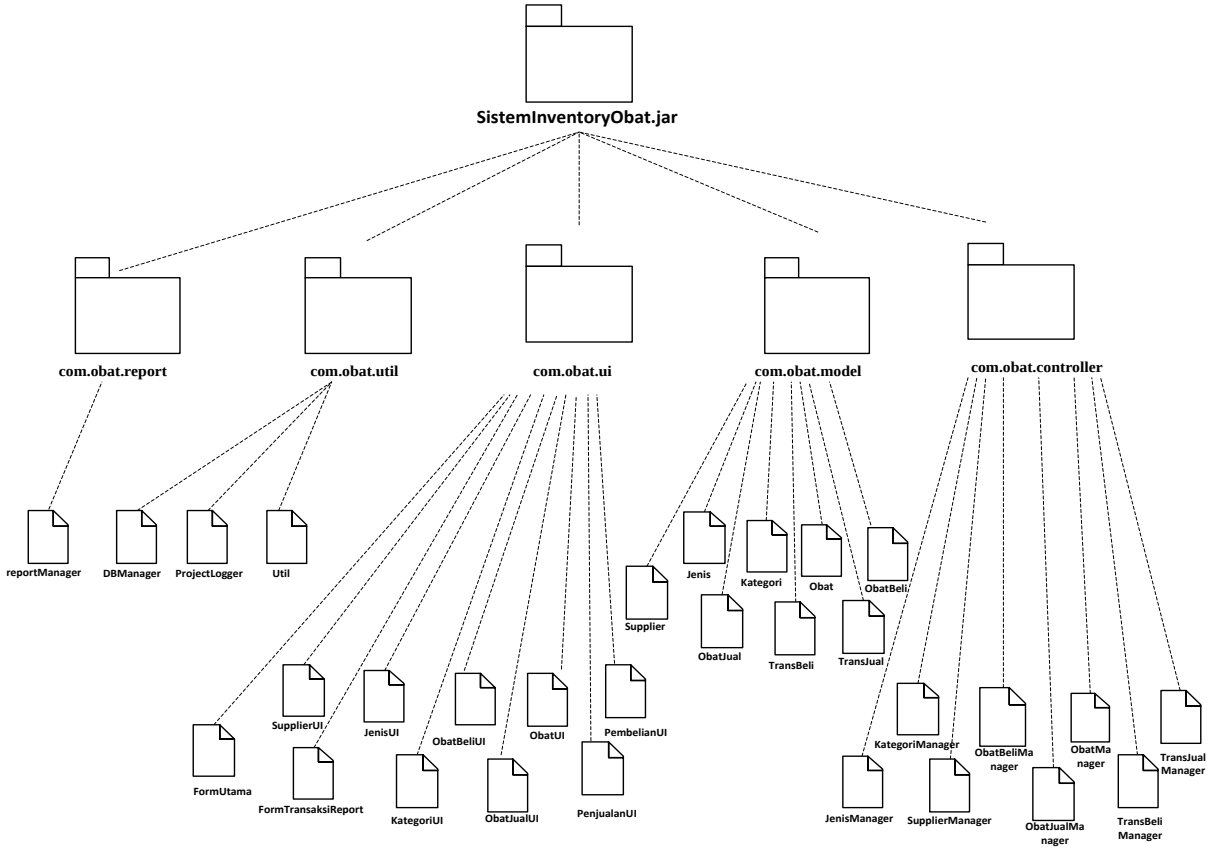
Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem pemasaran produk dari bagian penjualan beserta proses-proses yang terjadi, maka dapat dibuat *Class Diagram* Berikut *Class Diagram* Sistem Toko Obat yang diusulkan pada Toko Obat Hikmah Jaya:



Gambar 4.28 *Class diagram* Sistem Informasi Toko Obat

4.2.3.8 Component Diagram

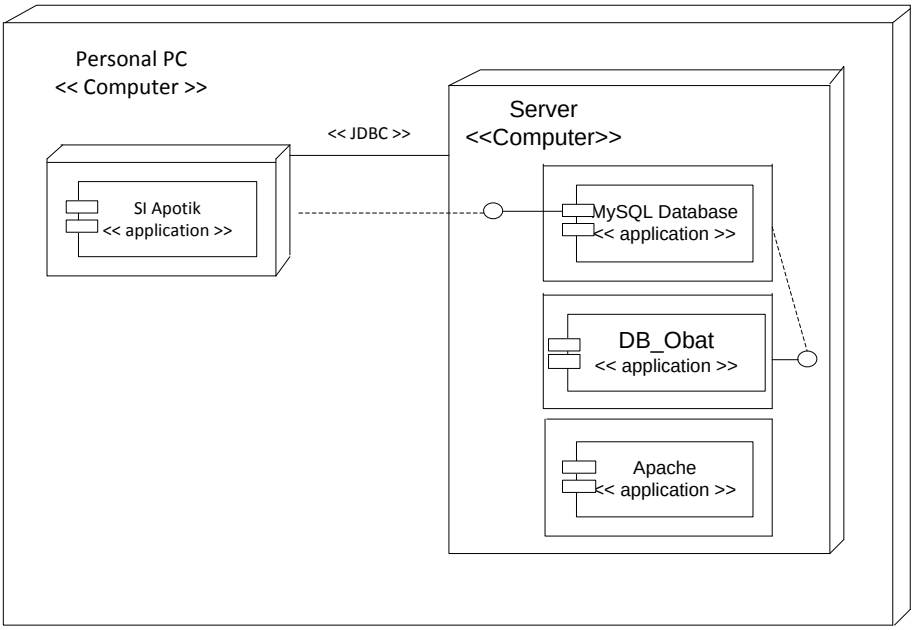
Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*) di antaranya. *Component* piranti lunak adalah modul berisi *code*, baik *berisi source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time.* Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Berikut ini adalah *component diagram* yang dibutuhkan :



Gambar 4.29 *Component diagram* Sistem Informasi Toko Obat

4.2.3.9 Deployment Diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. *Deployment diagram* berhubungan dengan diagram komponen dimana *deployment diagram* memuat satu atau lebih komponen-komponen.



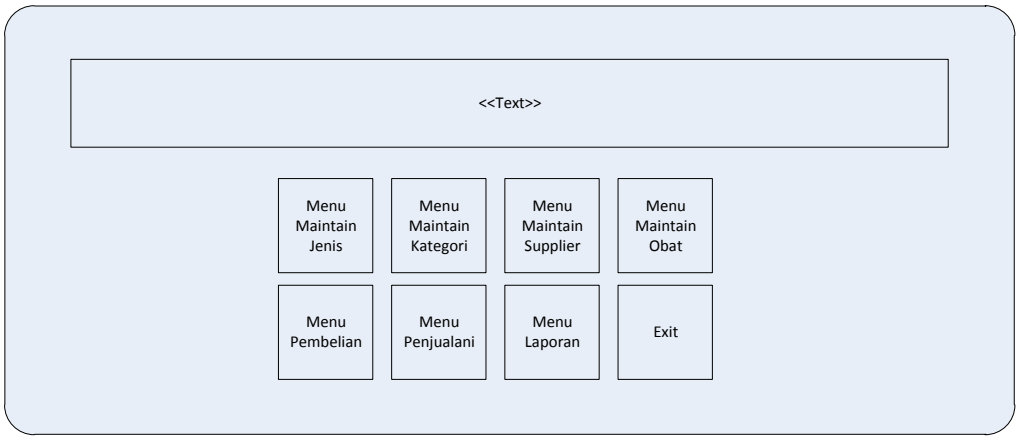
Gambar 4.30 *Deployment diagram* sistem Toko Obat target

4.2.4 Perancangan Antar Muka

Perancangan antar muka merupakan perancangan yang dibuat sebelum program aplikasi dibuat, perancangan antar muka pada sistem informasi Toko Obat untuk Toko Obat Hikmah Jaya adalah sebagai berikut:

4.2.4.1 Perancangan Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian perangkat lunak. Berikut rancangan menu perangkat lunak ini:



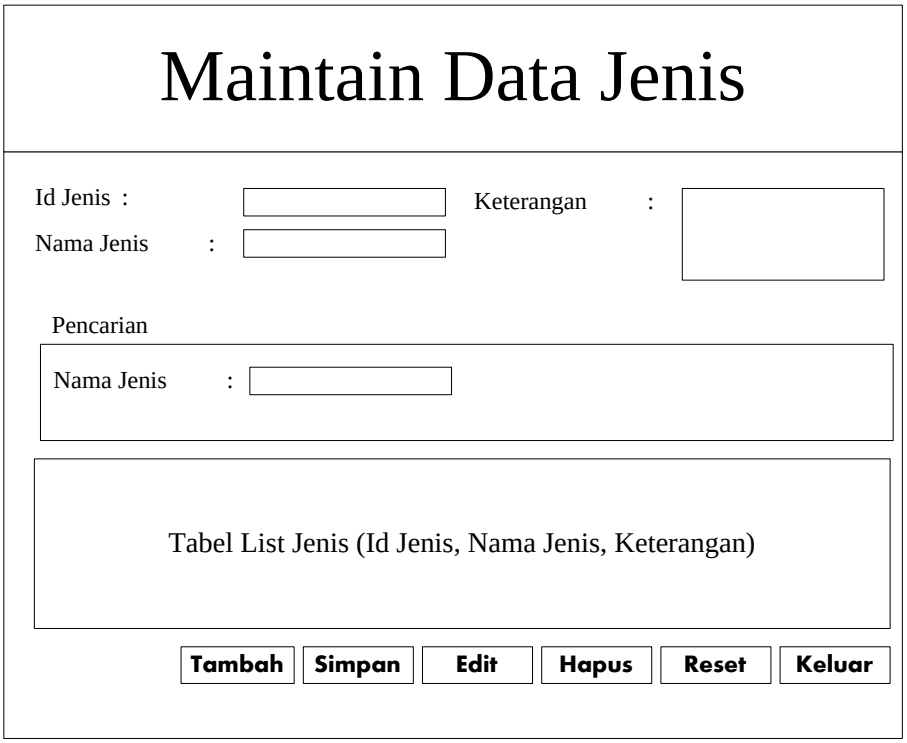
Gambar 4.31 *Perancangan menu* sistem Toko Obat target

4.2.4.2 Perancangan Input

Perancangan input merupakan perancangan tampilan yang akan digunakan guna memasukkan data pada sistem untuk kemudian diproses. Dalam perancangan input ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan input yang ada dalam perancangan ini adalah:

1. Rancangan Maintain Data Jenis

   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data jenis.



Gambar 4.32 Rancangan Maintain data jenis

2. Rancangan Maintain Data Kategori

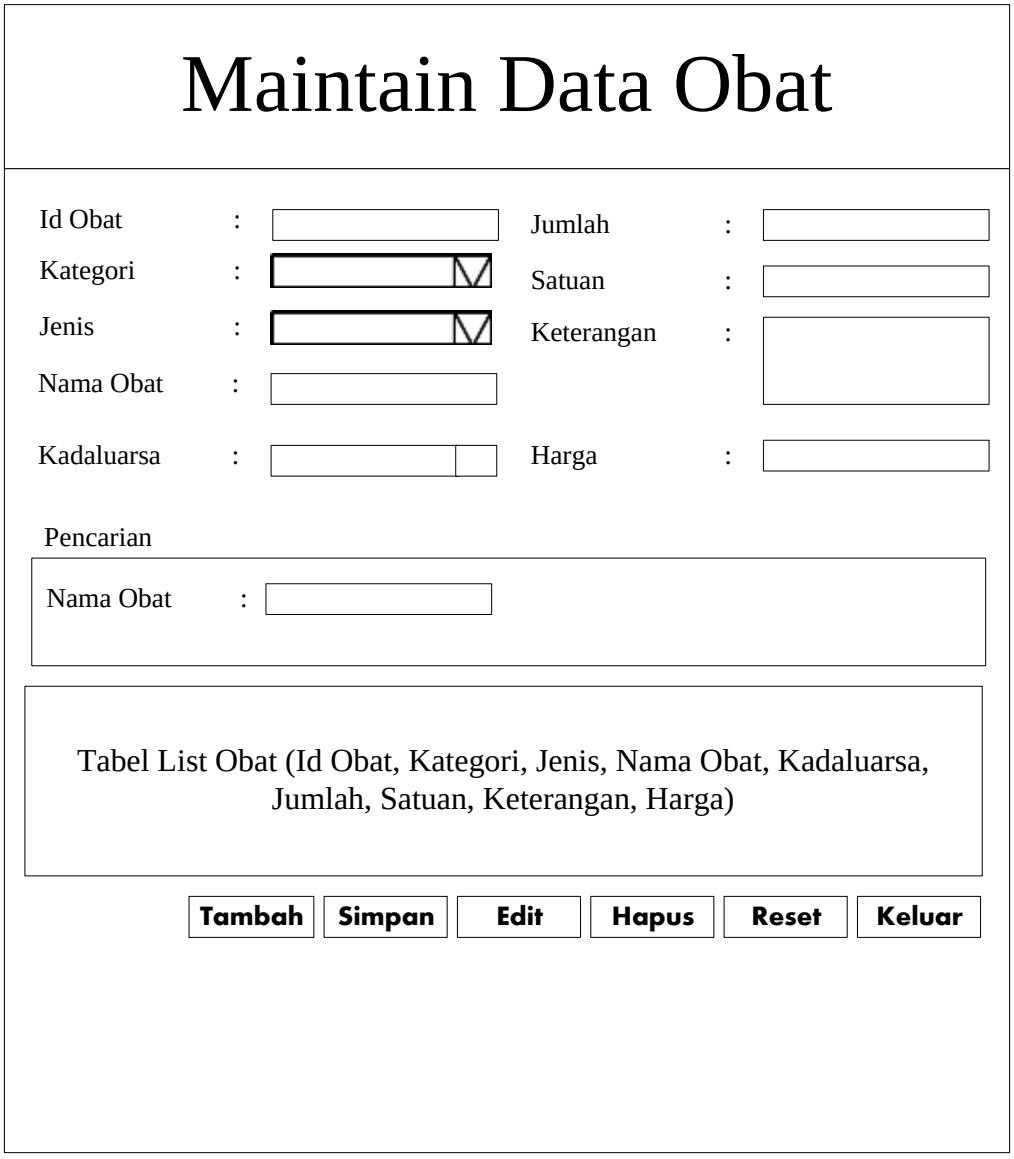
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data kategori.



Gambar 4.33 Rancangan Maintain data kategori

3. Rancangan Maintain Data Obat

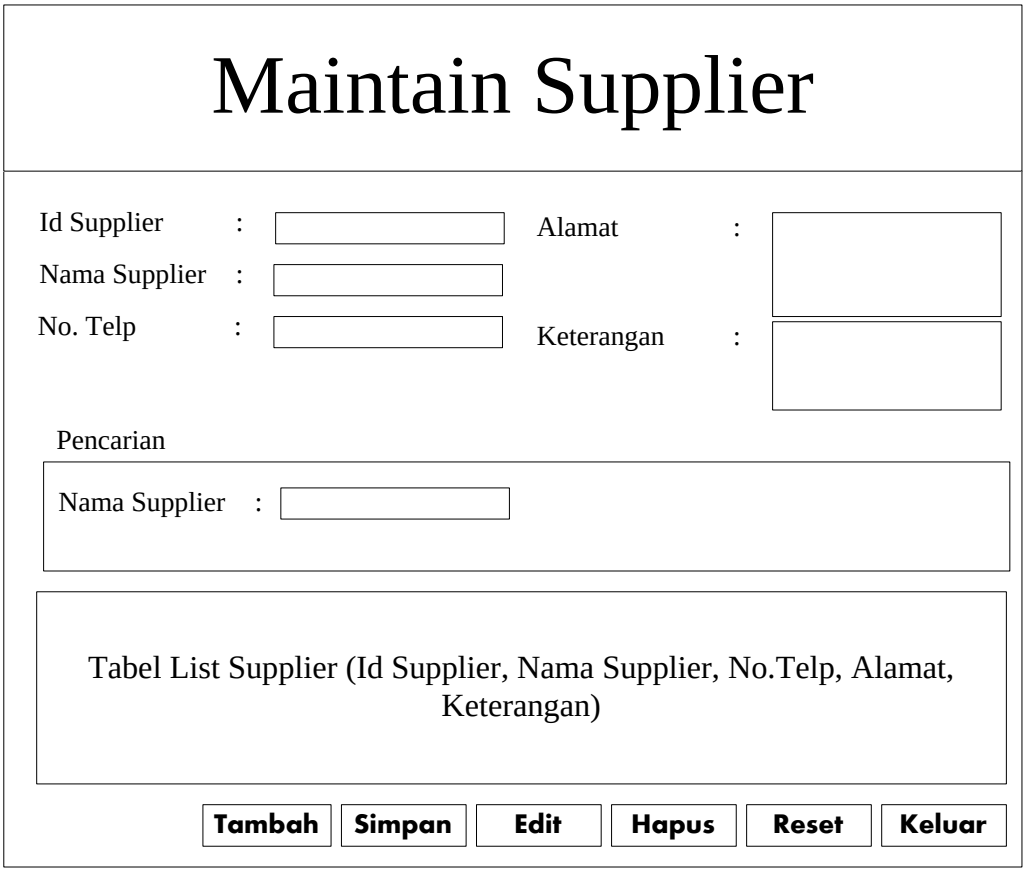
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data obat.



Gambar 4.34 Rancangan Maintain data obat

4. Rancangan Maintain Data *Supplier*

Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data *supplier*.



Gambar 4.35 Rancangan Maintain data *supplier*

5. Rancangan Maintain Data Penjualan

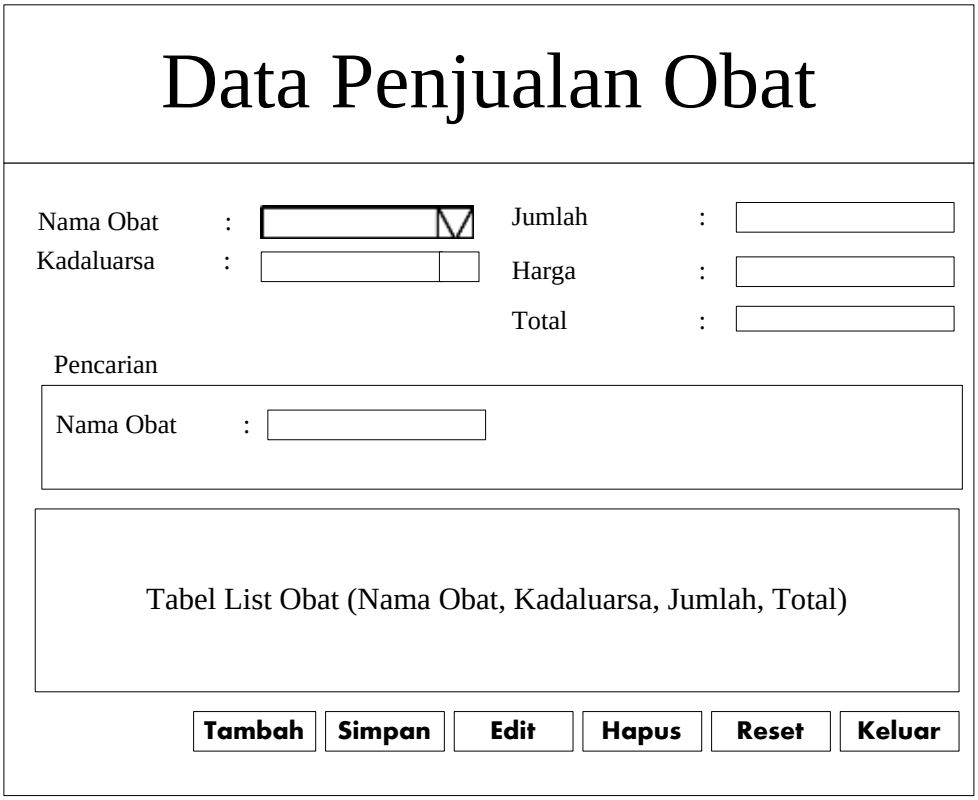
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencaria dan penambahan data penjualan.



Gambar 4.36 Rancangan Maintain data penjualan

6. Rancangan Maintain Data Penjualan Obat

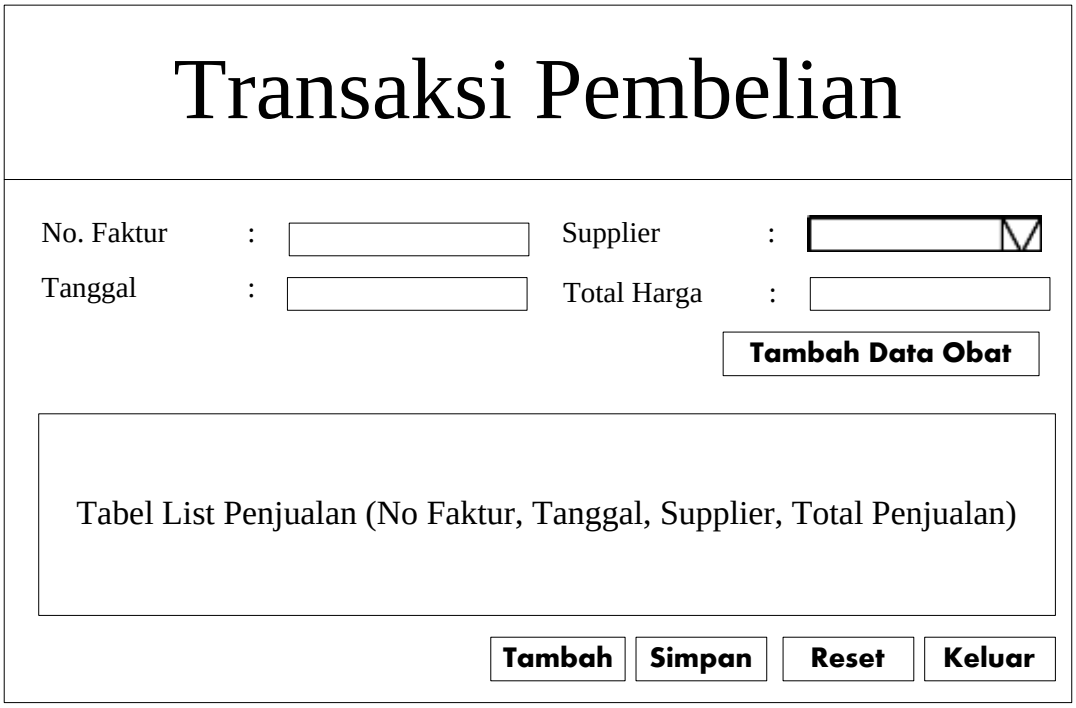
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data penjualan obat.



Gambar 4.37 Rancangan Maintain data penjualan obat

7. Rancangan Maintain Data Pembelian

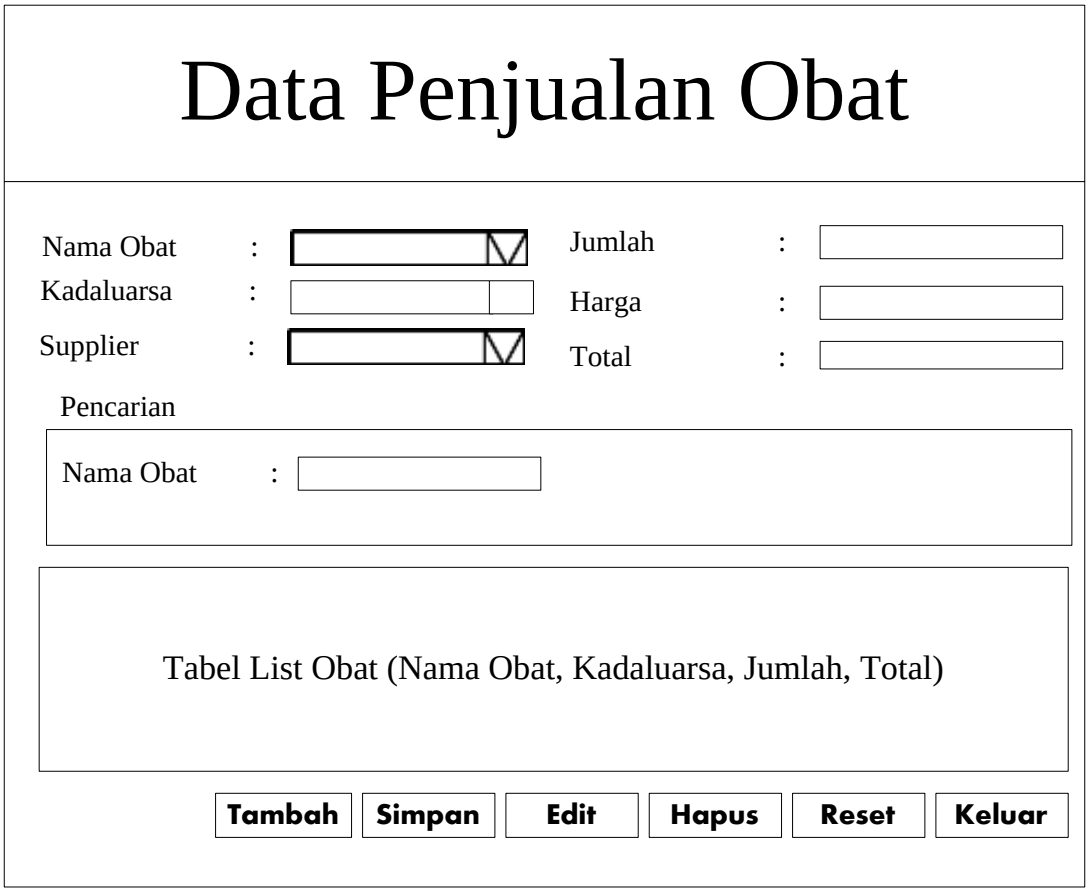
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data pembelian.



Gambar 4.38 Rancangan Maintain data pembelian

8. Rancangan Maintain Data Pembelian Obat

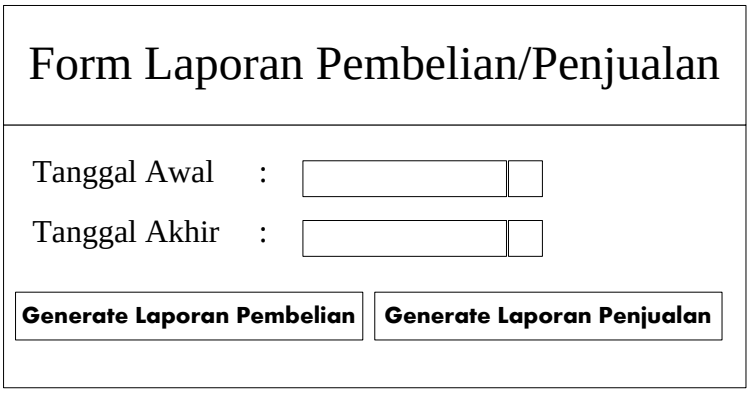
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data pembelian Obat.



Gambar 4.39 Rancangan Maintain data pembelian obat

9. Rancangan Maintain Data Pelaporan

Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan Pencetakan data pelaporan.



Gambar 4.40 Rancangan Maintain data pelaporan