**SISTEM INFORMASI APOTIK**

Untuk Memenuhi Salah Satu Tugas Mata Kuliah

 Pemrograman II ( Java Fundamental )

Dosen:

**Eka Chandra , M.Kom**

Disusun oleh:

- Mitha Amelia Dianty 1510631170089

- Panji Abujabbar 1510631170111

- Roza Hidayat 1510631170129

-Satria Panji Renova 1510631170133

Kelas III B

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS SINGAPERBANGSA KARAWANG**

**KARAWANG**

**2016**

# KATA PENGATAR

Puji syukur kami panjatkan kepada Allah SWT yang telah memberikan kami kemudahan sehingga dapat menyelesaikan laporan ini.

Laporan ini disusun untuk memenuhi tugas mata kuliah Pemrograman II ( Java Fundamental )yang kami sajikan berdasarkan pengamatan dari berbagai sumber. Laporan ini kami susun dengan segala kemampuan dan usaha dengan berbagai kendala. Namun dengan penuh kesabaran dan terutama pertolongan dari Allah SWT, akhirnya laporan ini dapat kami selesaikan. Laporan ini membahas “Sistem Informasi Apotik”

Kami juga mengucapkan banyak terima kasih kepada Dosen Pemrograman II ( Java Fundamental ) yaitu Bapak Eka Chandra, M. Kom yang telah memberikan tugas ini kepada kami.

Semoga laporan ini dapat memberikan pengetahuan yang lebih luas kepada pembaca. Walaupun laporan yang kami sajikan masih memiliki banyak kekurangan. Oleh karena itu, kami membutuhkan kritik dan saran yang sangat membangun dari pembaca.

Karawang, Desember 2016

Penulis

# DAFTAR ISI

[KATA PENGATAR i](#_Toc468431627)

[DAFTAR ISI ii](#_Toc468431628)

[DAFTAR GAMBAR iii](#_Toc468431629)

[BAB I PENDAHULUAN 1](#_Toc468431630)

[1.1. Latar Belakang 1](#_Toc468431631)

[1.2. Rumusan Masalah 1](#_Toc468431632)

[1.3. Tujuan Penulisan 2](#_Toc468431633)

[1.4. Batasan Masalah 2](#_Toc468431634)

[BAB II LANDASAN TEORI 3](#_Toc468431635)

[2.1. Teori Umum 3](#_Toc468431636)

[2.1.1. Konsep Dasar Sistem 3](#_Toc468431640)

[2.1.2. UML 5](#_Toc468431645)

[2.1.3. Kakas dalam Mendesain Sistem dan Antar Muka 11](#_Toc468431646)

[2.1.4. Java 11](#_Toc468431647)

[2.2. Teori Khusus 12](#_Toc468431648)

[BAB III ANALISIS DAN PENGEMBANGAN 14](#_Toc468431649)

[3.1. Analisa Sistem Berjalan 14](#_Toc468431650)

[3.1.1. Analisis Permasalahan 14](#_Toc468431651)

[3.1.2. Analisis Design 15](#_Toc468431652)

[BAB IV PENUTUP 22](#_Toc468431653)

[4.1. Kesimpulan 22](#_Toc468431654)

[**4.2.** Saran 22](#_Toc468431655)

[DAFTAR PUSTAKA 1](#_Toc468431656)

# DAFTAR GAMBAR

[Gambar 1 Use Case Diagram 1](#_Toc467810404)6

[Gambar 2 Class Diagram 1](#_Toc467810405)7

[Gambar 3 Login 1](#_Toc467810406)8

[Gambar 4 Input Data Obat 1](#_Toc467810407)9

[Gambar 5 Hapus Data Obat 1](#_Toc467810408)9

[Gambar 6 Update Data Obat](#_Toc467810409) 20

[Gambar 7 Transaksi Obat](#_Toc467810410) 20

# BAB I PENDAHULUAN

## Latar Belakang

Pada era modern ini, perkembangan ilmu pengetahuan dan teknologi sangat pesat, apalagi informasi sekarang sangat cepat menyebar ke penjuru dunia. Sejalan dengan hal tersebut permasalahan yang kita hadapi juga semakin kompleks pada masing - masing bidang di kehidupan sehari-hari. Dengan kenyataan itu kita dituntut untuk menyelesaikan permasalahan yang ada dengan memanfaatkan kecanggihan teknologi serta kecepatan, ketepatan dan keakuratan dalam memberi informasi sehingga dalam melaksanakan pekerjaan agar kita mendapat hasil yang optimal. Salah satu solusinya adalah pemanfaatan teknologi komputer. Dengan kemudahan fasilitas yang diberikan komputer akan mempermudah dalam pembuatan dan penyampaian informasi kepada orang yang membutuhkan.

Pada bidang kesehatan pengelolaan data obat pada apotik sudah seharusnya detail dan valid, teknologi computer yang semakin canggih dapat mempermudah pemilik maupun penjaga apotik dalam mengelola data obat. Pencatatan data obat secara manual pun dinilai kurang efektive secara waktu dan tenaga. Ketika kita mencari stok obat, harus sesegera mungkin kita temukan. Ketika membuat laporan bulanan obat keluar dan obat masuk membutuhkan waktu yang lama. Maka perlu adanya software yang membantu pengolahan data obat.

## Rumusan Masalah

Pada saat ini, setelah penulis melakukan penelitian pada sistem yang sudah ada atau yang sedang berjalan, maka permasalahan yang sering terjadi dalam Sistem Informasi Apotik adalah sebagai berikut :

1. Lamanya proses penginputan data obat secara manual,
2. Lamanya mencari dan melihat data obat beserta stoknya, sehingga menyebabkan lamanya informasi yang didapat oleh konsumen,
3. Masih manualnya penyimpanan data sehingga menemui kesulitan diantaranya tidak validnya data obat jika sewaktu-waktu diperlukan.

Dari hambatan / masalah di atas maka perlu dicari jalan keluarnya. Salah satu cara untuk mengatasinya adalah dengan menggunakan komputer sebagai alat bantu untuk memperbaiki sistem manual yang sedang berjalan saat ini. Diharapkan dengan adanya sistem informasi yang baru ini, maka semua kegiatan pencatatan dapat berjalan dengan lebih baik. Alasan lain dilakukannya penggunaan komputer adalah semakin ketatnya persaingan dan perkembangan yang cepat, sehingga diharapkan perusahaan / instansi dapat bergerak lebih cepat dalam segala hal untuk membantu kelancaran sistem yang sedang berjalan.

## Tujuan Penulisan

Selain merupakan salah satu syarat untuk menyelesaikan Tugas Mata Kuliah Java Fundamental, penulisan ini bertujuan untuk menganalisa sistem yang sedang berjalan dan dapat dilihat permasalahannya sehingga dapat dicari sebuah solusi berdasarkan prinsip-prinsip ilmu sistem informasi yang selama ini penulis pelajari.

Solusi yang didapat adalah dengan mencoba membuat suatu rancangan sistem yang telah terkomputerisasi sehingga dapat memberikan solusi yang bertujuan untuk :

1. Mempercepat proses penginputan data obat
2. Mempercepat mencari dan melihat data obat
3. Mempermudah proses pengolahan data sehingga informasi yang didapat konsumen lebih akurat.
4. Meningkatkan mutu keamanan dan pengarsipan data sehingga data dapat disimpan dan terkontrol dengan baik.

## Batasan Masalah

Dalam setiap pembuatan sistem pasti terdapat batasan masalah yang menyangkut seberapa luas jangkauan sistem , dan pembuatan sistem informasi apotik ini memiliki batasan masalah berupa :

* + 1. Sistem informasi apotik hanya dapat menambahkan, dan menghapus stok obat, serta menghitung pembayaran transaksi.
    2. Sistem yang tidak dibuat meliputi: Sistem penghitungan keuntungan tiap bulan, Sistem pengingat Obat Kadaluarsa, dan Sistem pembagian obat berdasarkan jenisnya.

# BAB II LANDASAN TEORI

## Teori Umum



### Konsep Dasar Sistem

#### Pengertian Sistem

Sistem adalah suatu tatanan (keterpaduan) yang terdiri atas sejumlah komponen fungsional (dalam satuan fungsi/tugas khusus) yang saling berhubungan dan bersama-sama bertujuan untuk memenuhi proses atau pekerjaan tertentu. Sistem adalah Sekumpulan hal atau kegiatan atau elemen atau subsistem yang saling bekerja sama atau yang dihubungkan dengan cara-cara tertentu sehingga membentuk satu kesatuan untuk melaksanakan suatu fungsi, guna mencapai suatu tujuan*.*

Dari beberapa definisi mengenai sistem yang diberikan, maka dapat ditarik suatu kesimpulan bahwa sistem merupakan satu kesatuan yang terdiri atas komponen-komponen atau elemen-elemen yang saling berhubungan erat untuk mencapai tujuan dan sasaran tertentu.

#### Pengertian DataBase

Database merupakan kumpulan data yang disimpan secara sistematis didalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak untuk menghasilkan informasi. Database menjadi penting karena dapat mengorganisasi data, menghindari duplikasi data, hubungan antar data yang tidak jelas dan juga update yang rumit.

#### Karakteristik Sistem

Berikut merupakan karakteristik dari suatu sistem :

1. Mempunyai Komponen

Komponen sistem adalah segala sesuatu yang menjadi bagian penyusun sistem. Komponen sistem disebut sebagai sub sistem, dapat berupa orang, benda, hal atau kejadian yang terlibat di dalam sistem.

1. Mempunyai Batas

Batas sistem diperlukan untuk membedakan satu sistem dengan sistem yang lain. Tanpa adanya batas sistem, maka sangat sulit untuk menjelaskan suatu sistem, Batas sistem memberikan batasa *scope* tinjauan sistem.

1. Mempunyai Lingkungan

Lingkungan sistem adalah segala sesuatu yang berada di luar sistem. Lingkungan sistem dapat menguntungkan ataupun merugikan. Umumnya, lingkungan yang menguntungkan akan selalu dipertahankan untuk menjaga keberlangsungan sistem.

1. Mempunyai Penghubung

Penghubung merupakan komponen sistem, yaitu segala sesuatu yang bertugas menjembatani antar komponen dalam sistem.

1. Mempunyai Masukan

Masukan sistem adalah segala sesuatu yang perlu dimasukkan ke dalam sistem sebagai bahan yang akan diolah lebih lanjut untuk menghasilkan keluaran yang berguna. Dalam sistem informasi manajemen, masukan disebut data.

1. Mempunyai Pengolahan

Pengolah sistem berperan untuk mengolah masukan agar menghasilkankeluaran yang berguna bagi para pemakainya. Dalam system informasi manajemen, pengolahan adalah program aplikasi komputer yang dikembangkan untuk keperluan khusus.

1. Mempunyai Keluaran

Keluaran sistem adalah berbagai macam bentuk keluaran yang dihasilkan oleh komponen pengolahan. Dalam sistem informasi manajemen, keluaran adalah informasi yang dihasilkan oleh program aplikasi yang akan digunakan oleh para pemakai sebagai bahan pengambil keputusan.

1. Mempunyai Sasaran

Sasaran sistem adalah apa yang ingin dicapai oleh system untuk jangka waktu yang relatif pendek. Sasaran merupakan hasil pada setiap tahapan tertentu yang medukung upaya pencapaian tujuan.

1. Mempunyai Kendali

Kendali sistem berperan utama menjaga agar proses dalamsistem dapat berlangsung secara normal sesuai batasan yang telah ditetapkan sebelumnya. Kendali dalam sistem dapat berupa validasi masukan, validasi proses, maupun validasi keluaran yang dapat dirancang dan dikembangkan secara terprogram.

1. Mempunyai Umpan Balik

Umpan balik diperlukan oleh bagian kendali sistem untuk mengecek terjadinya penyimpangan proses dalam sistem dan mengembalikannya ke dalam kondisi normal.



### UML

Menurut Adi Nugroho (2005:3), UML (*Unified Modeling Language*) adalah salah satu alat bantu analisis serta perancangan perangkat lunak beroritasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan *visual* yang memungkinkan bagi pengembang *system* untuk membuat cetak biru atau visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

#### Pengertian Use Case

Menurut Satzinger (2010:242), *Use Case* merupakan suatu aktivitas yang dilakukan oleh sistem, bisanya merupakan sebuah respon untuk permintaan dari pengguna sistem. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai.

Urutan langkah-langkah yang menerangkan antara pengguna dan *system* disebut skenario. Setiap skenario mendeskripsikan urutan kejadian. *Use case* diagram menampilkan *actor, use case,* dan hubungan antara meraka:

1. Aktor mana yang menggunakan *use case* mana.
2. *Use case* mana yang memasukkan *use case* lain.

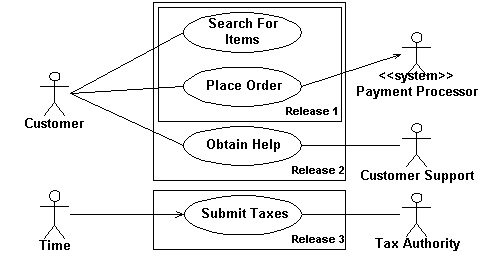
Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML use case. Use Case memiliki dua istilah yaitu :

1. System use case; interaksi dengan sistem.
2. Business use case; interaksi bisnis dengan konsumen atau kejadian nyata

Berikut ini adalah symbol beserta deskripsi symbol yang digunakan dalam use case :

*Tabel a. simbol use case diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Actor | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*. |
| Dependency | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*). |
| *Generalization* | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*). |
| *Include* | Menspesifikasikan bahwa *use case* sumber secara *eksplisit*. |
| *Extend* | Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan. |
| *Association* | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| *System* | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| *Use Case* | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| *Collaboration* | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |
| *Note* | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |



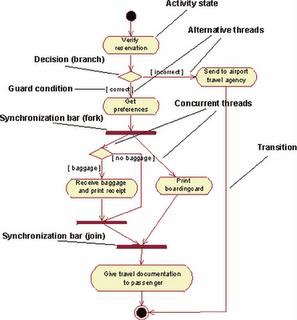
Notasi Use Case Diagram

#### Pengertian Activity Diagram

Menurut Satzinger (2010:141), *Activity diagram* merupakan sebuah tipe dari diagram *workflow* yang menggambarkan tentang aktifitas dari pengguna ketika melakukan setiap kegiatan dan aliran squensial. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedang *flowchart* tidak bisa.

*Tabel b. simbol aktivity diagram*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| *Activity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| *Activity Final Node* | Bagaimana objek dibentuk dan dihancurkan |
| *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |



Notasi Activity Diagram

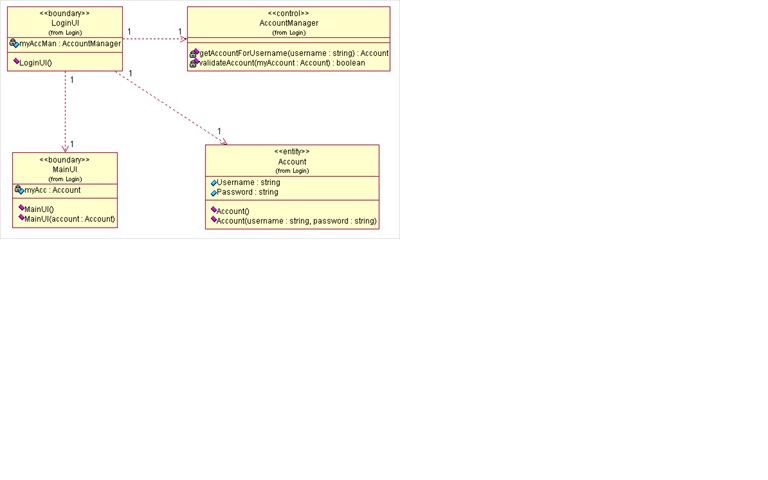
#### Pengertian Class Diagram

*Class* diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang dikembangkan. *Class* diagram memberikan gambaran/diagram statis tentang sistem/perangkat lunak dan relasi-relasi yang ada di dalamnya. Berikut symbol dan deskripsi yang digunakan dalam pembuatan class diagram:

*Tabel c. simbol class diagram*

|  |  |
| --- | --- |
| Simbol | Deskripsi |
| [http://2.bp.blogspot.com/-ONANWqL3deg/VR5DiCMv1pI/AAAAAAAAATk/yo54xjtEtgE/s1600/Snap%2B2015-04-03%2Bat%2B14.23.03.png](http://2.bp.blogspot.com/-ONANWqL3deg/VR5DiCMv1pI/AAAAAAAAATk/yo54xjtEtgE/s1600/Snap+2015-04-03+at+14.23.03.png)Kelas | Kelas pada stuktur sistem. |
| Antarmuka (Interface)  [http://3.bp.blogspot.com/-KrBdyBTW1hc/VR5D8nFNnxI/AAAAAAAAATs/Z2C3ZZSctRE/s1600/Snap%2B2015-04-03%2Bat%2B14.40.20.png](http://3.bp.blogspot.com/-KrBdyBTW1hc/VR5D8nFNnxI/AAAAAAAAATs/Z2C3ZZSctRE/s1600/Snap+2015-04-03+at+14.40.20.png) | Sama dengan konsep interface dalam pemrograman berorientasi objek. |
| Asosiasi (Association)  [http://4.bp.blogspot.com/-zHROgHCpdrs/VR4yxlZgtwI/AAAAAAAAAS4/wPElhz9EWeU/s1600/Snap%2B2015-04-03%2Bat%2B13.13.47.png](http://4.bp.blogspot.com/-zHROgHCpdrs/VR4yxlZgtwI/AAAAAAAAAS4/wPElhz9EWeU/s1600/Snap+2015-04-03+at+13.13.47.png) | Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan multiplicty. |
| [http://1.bp.blogspot.com/-GCiv3UbwSrY/VR5EVrJ5qYI/AAAAAAAAAT0/TN8iV3FI4qM/s1600/Snap%2B2015-04-03%2Bat%2B14.41.14.png](http://1.bp.blogspot.com/-GCiv3UbwSrY/VR5EVrJ5qYI/AAAAAAAAAT0/TN8iV3FI4qM/s1600/Snap+2015-04-03+at+14.41.14.png)Asosiasi berarah (Directed Association) | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai dengan multiplicity. |
| Generalisasi (Generalization)  [http://1.bp.blogspot.com/-cKyffPtNDlo/VR4z51D8JqI/AAAAAAAAATM/sCjErS4-O6g/s1600/Snap%2B2015-04-03%2Bat%2B13.14.52.png](http://1.bp.blogspot.com/-cKyffPtNDlo/VR4z51D8JqI/AAAAAAAAATM/sCjErS4-O6g/s1600/Snap+2015-04-03+at+13.14.52.png) | Relasi antar kelas dengan makna generalisasi - spesialisasi (Umum-khusus) |
| Kebergantungan (Dependency)  [http://3.bp.blogspot.com/-erqoC86gu38/VR5EqWRg5-I/AAAAAAAAAT8/PYCgrIvnajU/s1600/Snap%2B2015-04-03%2Bat%2B14.42.51.png](http://3.bp.blogspot.com/-erqoC86gu38/VR5EqWRg5-I/AAAAAAAAAT8/PYCgrIvnajU/s1600/Snap+2015-04-03+at+14.42.51.png) | Relasi antar kelas dengan makna kebergantungan antar kelas. |
| Agregasi (Aggregation)  [http://1.bp.blogspot.com/-ElOpXzQNKiM/VR5EsB4cWaI/AAAAAAAAAUE/DS8j4PXysQg/s1600/Snap%2B2015-04-03%2Bat%2B14.43.20.png](http://1.bp.blogspot.com/-ElOpXzQNKiM/VR5EsB4cWaI/AAAAAAAAAUE/DS8j4PXysQg/s1600/Snap+2015-04-03+at+14.43.20.png) | Relasi antar kelas dengan makna semua - bagian (Whole-part) |

Class memiliki tiga area pokok :

* 1. Nama (dan stereotype)
  2. Atribut
  3. Metoda

Notasi class diagram

### Kakas dalam Mendesain Sistem dan Antar Muka

#### Star UML

*Star UML* adalah platform pemodelan perangkat lunak yang mendukung *UML* (*Unified Modeling Language*). Star*UML* yang berbasiskan pada *UML* versi 1.4, menyediakan sebelas jenis Diagram yang berbeda, dan mendukung notasi *UML* 2.0.Star *UML* juga secara aktif mendukung konsep *UML* profile.

*Star UML* unggul dalam hal kustomisasi lingkungan kerja pengguna, dan memiliki ekstensibilitas tinggi pada fungsionalitasnya. *Star UML* mengklaim diri sebagai salah satu alat pemodelan perangkat lunak termuka yang menjamin dapat memaksimalkan produktivitas dan kualitas proyek perangkat lunak anda.

#### Notepad++

*Notepad++* adalah sebuah aplikasi penyuting teks dan penyuting kode sumber yang berjalan di sistem operasi Windows. Notepad++ menggunakan komponen Scintilla untuk dapat menampilkan dan menyuntung teks dan berkas kode sumber berbagai bahasa pemrograman.

#### MySQL dengan Xampp

*MySQL (Structured Query Language*) merupakan bahasa terstruktur yang difungsikan untuk mengelola database. MySQL dapat digunakan untuk membuat dan mengellola database beserta isinya. MySQL juga dapat digunakan untuk menambahkan, mengubah, dan menghapus data didalam database.

#### CMD (Command Prompt)

*CMD atau DOS Prompt* adalah sebuah command line interface (CLI) pada sistem operasi Windows untuk mengeksekusi file dengan cara memasukkan perintah-perintah menggunakan keyboard.

### Java

Java adalah bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, Simula). Java diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991. Karakteristik Java antara lain : Sederhana (Simple), Berorientasi Objek (Object Oriented), Terdistribusi (Distributed), Interpreted, Robust, Aman (Secure), Architecture Neutral, Portable, Performance, Multithreaded, dan Dinamis

#### Macam-macam Java 2 Software Developer Kit (J2SDK) antara lain : J2SE (Java 2 Standard Edition), J2EE (Java 2 Enterprise Edition), dan J2ME (Java 2 Micro Edition)

1. Aplikasi-aplikasi java antara lain : Stand alone, Applets, Aplikasi berbasis Windows (GUI), Java Servlet, Java Server Pages (JSP), Java Beans dan Enterprise Java, Beans, dan Java Micro Edition

## Teori Khusus

* + 1. Pengertian Apotik

1. Menurut Keputusan Menkes RI No.1332/Menkes/SK/X/2002 Apotek merupakan suatu tempat tertentu untuk melakukan pekerjaan kefarmasian dan penyaluran obat kepada masyarakat.
2. Definisi apotek menurut PP 51 Tahun 2009. Apotek merupakan suatu tempat atau terminal distribusi obat perbekalan farmasi yang dikelola oleh apoteker sesuai standar dan etika kefarmasian.
   * 1. Graphical User Interface ( GUI )

GUI atau Graphical User Interface merupakan sistem yang memungkinkan para pengguna untuk melakukan interaksi dengan komputer. GUI merupakan sebuah desain antar muka pada komputer, selain GUI ada pula CLI atau Command Line Interface, yang berbasis text dimana para penguna dapat melakukan interaksi dengan komputer melalui perintah text atau Command Line. Istilah GUI bukan hal yang lumrah pada saat awal kemunculan komputer. Namun setelah komputer generasi keempat mulai diciptakan, munculnya televisi berwarna (yang mendorong pada penciptaan layar monitor berwarna) serta evolusi pada perangkat penampil gambar (graphic adapter atau graphic card atau video card) membuat komputer mulai mendapatkan suatu sistem baru.

Secara sederhana, GUI adalah suatu media virtual yang dapat membuat pengguna memberikan perintah tertentu pada komputer tanpa mengetik perintah atau Command. Dengan GUI, perintah dapat dikonversi menjadi ikon dalam layar monitor yang dapat diklik untuk memulai fungsinya.

GUI merupakan antarmuka pada system yang menggunakan menu grafis. Menu grafis ini maksudnya terdapat tampilan yang lebih ditekankan untuk membuat system yang user-friendly agar para pengguna lebih nyaman menggunakannya.

Program GUI (GraphicalUserInterface) merupakan metode pendesainan program kedunia nyata (realworld) atau visual bukan dalam bentuk console. Paket yang dapat digunakan unuk mengembangkan program-program GUI dalam java salah satunya adalah AWT dan Swing.

# BAB III ANALISIS DAN PENGEMBANGAN

## Analisa Sistem Berjalan

Analisa sistem merupakan proses memilah-milah suatu permasalahan menjadi elemen-elemen yang lebih kecil untuk dipelajari guna mempermudah permasalahan dari suatu sistem informasi. Hasil akhir dari analisa sistem merupakan cara pemecahan suatu masalah yang terjadi dalam spesifikasi sistem baru. Pada tahap analisa diperlukan suatu pendekatan analisa guna menghindari kesalahan-kesalahan yang mungkin muncul pada tahap berikutnya, yaitu perancangan sistem baru. Karena pada tahap ini merupakan tahapan yang sangat penting, pendekatan yang dilakukan adalah mendefinisikan masalah pada sistem yang sedang berjalan dan sekaligus melakukan evaluasi setiap cara kerja sistem yang sedang berjalan berdasarkan prosedur-prosedur yang ada. Sehingga akan diketahui permasalahannya serta kesulitan apa saja yang dihadapi oeh sistem yang sedang berjalan, apa saja pengaruhnya dan harus diperhatikan validasinya terhadap sasaran sistem yang dirancang sebelum dilakukan perbaikan.

### Analisis Permasalahan

Untuk memecahkan permasalahan pengelolaan data dan pemantauan Klinik yang dihadapi, maka penulis menggunakan kerangka pemecahan masalah dengan menggunakan kerangka PIECES (Performance, Information, Economic, Control, Eficiency, Service). Adapun tahapan-tahapan dalam kerangka PIECES tersebut adalah sebagai berikut :

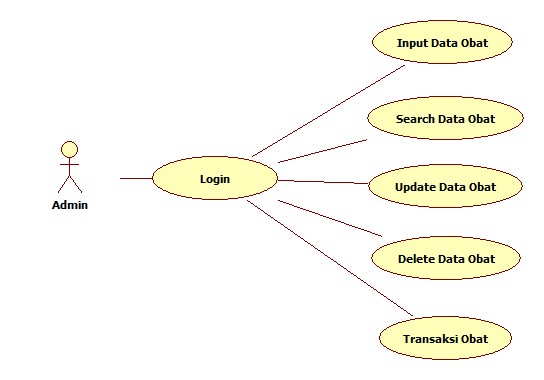
|  |  |
| --- | --- |
| **P** | ***The Need To Improve Perfomance***  Lambatnya informasi mengenai data pasien, sehingga tindak lanjut mengenai keadaan pasien menjadi lambat terlaksana. |
| **I** | ***The Need To Improve Information And Data***  ***Output :***  Informasi yang diperoleh terkadang tidak sesuai dengan yang dibutuhkan dan terkadang tidak akurat karena proses penyimpanan data yang belum terkomputerisasi.  ***Input :***  a. Data yang disimpan tidak fleksibel.  b. Data yang disimpan juga masih dalam bentuk berkas sehingga kemungkinan data hilang atau rusak. |
| **E** | ***The Need To Improve Economic, Control Cost, Or Increase Profits***  Besarnya biaya operasional yang dikeluarkan oleh Klinik, dikarenakan masih banyaknya penggunaan kertas dalam penyimpanan arsip-arsip data. |
| **C** | ***The Need To Improve Control Or Security***  Penyimpanan data dalam bentuk arsip tidak aman, karena ada kemungkinan orang yang tidak berkepentingan bisa melihat data yang tersimpan, bahkan mengubahnya, dan juga data-data tersebut dapat hilang bahkan rusak. |
| **E** | ***The Need To ImproveEfficiency Of PeopleAnd Processes***  Pencarian data-data Pasien, yang berupa arsip menyebabkan pemborosan waktu kerja yang seharusnya dapat dilakukan untuk pekerjaan lainnya. |
| **S** | ***The Need To Improve Service***  Belum adanya suatu sistem yang terkomputerisasi dengan baik yang sesuai dengan kriteria kebutuhan dalam sistem Klinik, sehingga menyebabkan sulitnya pengelolaan data-data dan pemantauan Klinik terhadap keadaan pasien pada umumya. |

### Analisis Design

Tahap analisis Design bertujuan untuk mendefenisikan kebutuhan dari sistem yang akan dikembangkan. Untuk menganalisis kebutuhan sistem yang akan dikembangkan maka dalam penelitian ini penulis menggunakan pemodelan *use case, class diagram* dan *activity diagram.*

#### *Use Case* Diagram

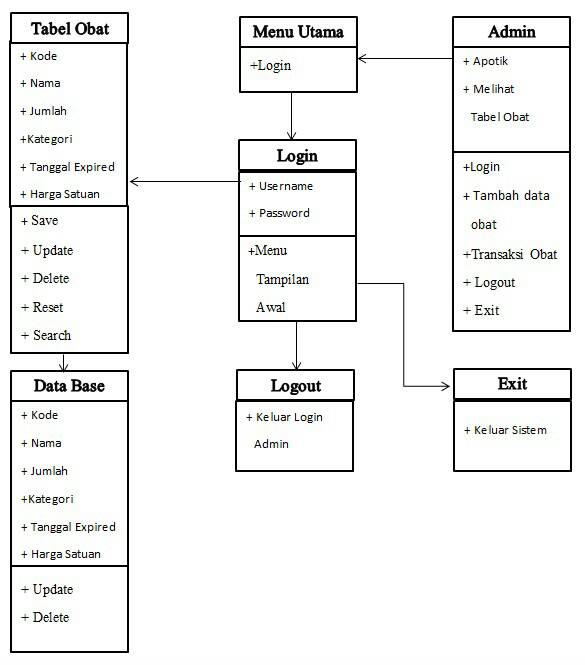
Permodelan ini dimaksudkan untuk mengambarkan kegiatan - kegiatan dan hubungan yang terjadi antara para aktor dan use case didalam sistem yang sedang berjalan. Kegiatan yang dilakukan dan hubungan antara para aktor dalam sistem yang sedang berjalan digambarkan menggunakan *use case* diagram yang terlebih dahulu digambarkan melalui *use case* sekenario dimana telah terwakili oleh gambaran umum sistem yang sedang berjalan.



Gambar 1 Use Case Diagram

#### *Class* Diagram

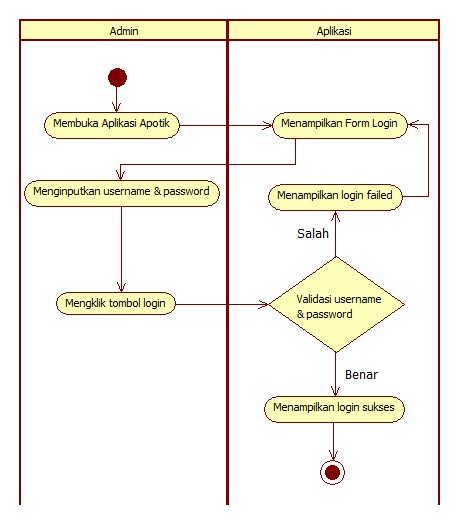
Diagram yang menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang dikembangkan.



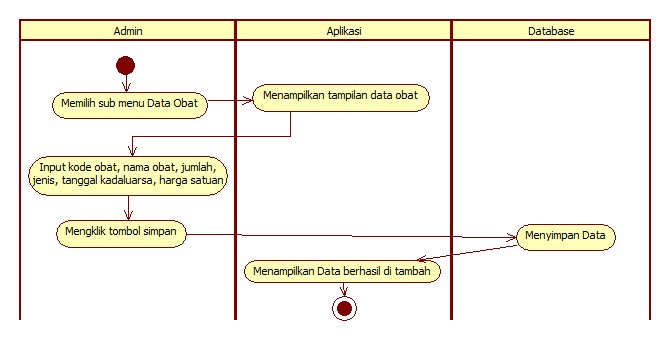
Gambar 2 Class Diagram

#### *Activity* Diagram

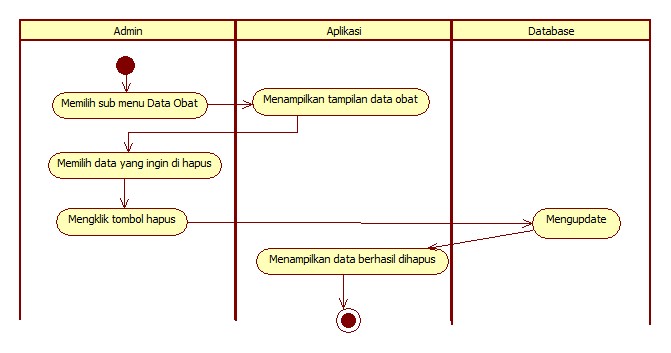
Pada bagian ini akan digambarkan dokumentasi alur kerja pada sistem yang sedang berjalan yang bertujuan untuk melihat alur proses sistem yang sedang berjalan dan sekaligus memperjelas *use case* sistem yang sedang berjalan.



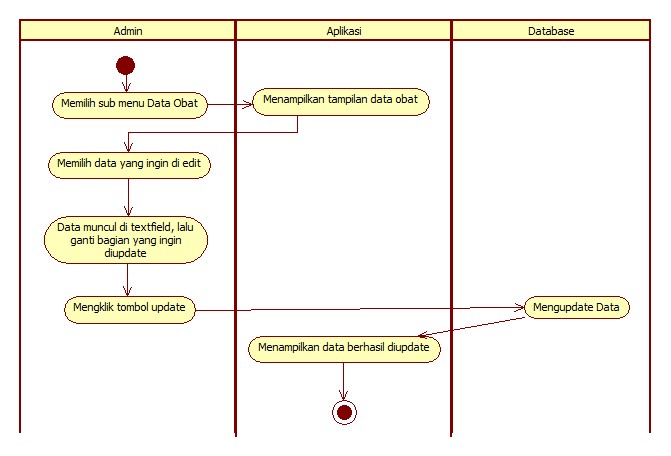
Gambar 3 Login



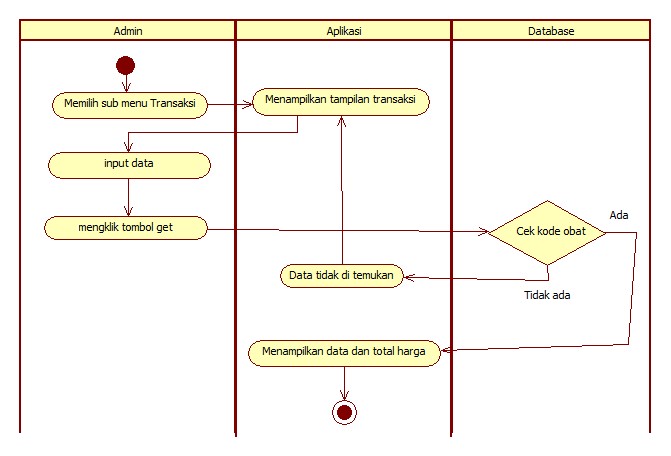
Gambar 4 Input Data Obat



Gambar 5 Hapus Data Obat



Gambar 6 Update Data Obat



Gambar 7 Transaksi Obat

# BAB IV PENUTUP

## Kesimpulan

Aplikasi system informasi apotik ini sebagai jawaban atas kelemahan sistem yang ada di apotik. Dimana sebelum sering terjadi beberapa kekurangan seperti: Lamanya proses penginputan data obat secara manual,lamanya mencari dan melihat data obat beserta stoknya, sehingga menyebabkan lamanya informasi yang didapat oleh konsumen, dan masih manualnya penyimpanan data sehingga menemui kesulitan diantaranya tidak validnya data obat jika sewaktu-waktu diperlukan.

Dengan dibuatnya sistem ini kami berharap agar pelayanan Apotik menjadi lebih maksimal, dan pelanggan pun mendapatkan apa yang mereka inginkan serta membantu petugas Apotik dalam melakukan pendataan obat.

* 1. Saran

Berdasarkan kesimpulan yang diambil maka penulis membuat rekomendasi

kepada yang menggunakan perangkat lunak ini, sebagai berikut :

1. Bagi peneliti lain yang meneliti suatu aplikasi untuk Apotik diharapkan dapat mengembangkan tampilan dan pengolahan data secara otomatisasi.
2. Untuk pengguna aplikasi ini menggadakan penelitian bagi para pasien untuk bisa lebih mengutamakan kenyamatan bagi para pelanggan.
3. Agar peroses atau perogram ini dapat di oprasikan dengan baik maka maka dilakukan pemeliharaan perangkat lunak (maintenance).
4. Memberikan pelatihan kepada petugas, khususnya karyawan baru dalam pengoperasian aplikasi agar dapat berjalan dengan baik sesuai yang dibutuhkan.

# DAFTAR PUSTAKA

Ladjamudin Al – Bahra. 2005. Analisis Sistem Informasi.Yogyakarta : Graha Ilmu

Haviluddin. 2011. Memahami Penggunaan UML (*Unified Modelling Language.* Samarinda:Universitas Mulawarman

<http://dosen.gufron.com/digital/notepad-v653/51/>

<http://www.termasmedia.com/65-pengertian/69-pengertian-database.html>

<http://www.transiskom.com/2015/07/pengertian-command-prompt-cmd-dan.html?m=1>

htttp://www.dosenpendidikan.com/pengertian-komponen-dan-fungsi-xampp-lengkap-dengan-penjelasannya/