BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Tri Utami (2015) dalam penelitiannya yang berjudul pembangunan sistem informasi penjualan obat pada apotek Punung di Pacitan, Jawa Timur. Apotek punung merupakan salah satu apotek yang mengatur manajemen secara konvensional, dan belum memanfaatkan teknologi komputer secara optimal. Petugas apotek masih melakukan transaksi penjualan obat secara konvensional yang dilakukan diatas secarik kertas sebelum dimasukkan ke dalam buku laporan. Hal tersebut menyulitkan petugas apotek dalam proses pengolahan data. Alat bantu pengembangannya menggunakan *Diagram Alir Data (DAD), ERD diagram* dan *diagram konteks,* dan metode pengembangan sistem yang digunakan adalah *waterfall*.

Deni Eko (2013) dalam penelitiannya yang berjudul pembangunan sistem informasi apotek Pink di Pacitan Jawa Timur. Selama ini di apotek pacitan mengatur manajemen masih dengan cara yang konvensional, dan belum memanfaatkan teknologi komputer secara optimal. Berdasarkan hasil penelitian penulis di Apotek Pink Pacitan petugas Apotek kesulitan dalam proses pengolahan data secara konvensional. Kemungkinan buruk yang bisa terjadi adalah jika kertas tersebut hilang sebelum data transaksi dimasukkan ke dalam buku laporan. Selain hal tersebut permasalahan yang dihadapi adalah petugas membutuhkan waktu yang lama untuk memantau ketersediaan obat, dan petugas juga kesulitan dalam membuat laporan kepada pimpinan. Aplikasi dibangun berbasis *desktop* dengan menggunakan bahasa pemrograman Java, dan *MySQL* sebagai *databasenya*. Metode pengembangan sistem yang digunakan adalah *waterfall*.

Nurdiansyah dan Ramadian (2013) dalam penelitiannya berjudul pembuatan sistem informasi apotek berbasis web pada apotek Tulakan. Apotek Tulakan adalah salah satu apotek yang mengatur manajemen masih dengan cara konvensional. Oleh karena itu dilakukan penelitian ini agar petugas apotek dapat

dengan mudah memproses pengolahan data Apotek, seperti data obat, data penjualan obat, pembuatan kuitansi, pembuatan laporan pada pimpinan dan untuk mempermudah dalam proses pencarian data selain itu memiliki media penyimpanan yang lebih efektif dan lebih besar. Aplikasi yang dibangun berbasis *web* dengan menggunakan Bahasa pemrograman *PHP* dan *MySQL* sebagai databasenya. Alat bantu pengembangannya menggunakan *diagram konteks, flowchart, ERD diagram* dan *DFD (Data Flow Diagram).*

Berdasarkan pengujian dan analisis yang telah dilakukan oleh masing-masing peneliti, ketiga penelitian tersebut membuat sistem informasi apotek berbasis *web* sebagai media yang mempermudah dan mempercepat kinerja apotek. Ketiga penelitian tersebut berkaitan dengan penelitian yang dibuat oleh penulis yaitu membuat sistem informasi berbasis *web* sebagai sarana mempermudah dan mempercepat kinerja apotek, adapun kelebihan dari penelitian ini yaitu adanya 3 tingkatan pengguna untuk kasir, admin, dan super admin. Terdapat juga halaman pembelian obat maupun retur pembelian kepada *supplier* dan bagian laporan dibuat secara detail dan menyeluruh.

2.2 Landasan Teori

2.2.1 Konsep Dasar Sistem

Menurut Fat pengertian sistem adalah sebagai berikut : "Sistem adalah suatu benda nyata atau abstrak (*a set of thing*) yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam satu kesatuan (*Unity*) untuk mencapai tujuan tertentu secara efisien dan efektif".

Pengertian Sistem menurut Murdick, R.G (1991) Suatu sistem adalah seperangkat elemen yang membentuk kumpulan atau prosedur-prosedur atau bagan-bagan pengolahan yang mencari suatu tujuan bagian atau tujuan bersama dengan mengoperasikan data dan atau barang pada waktu rujukan tertentu untuk menghasilkan informasi dan atau energi dan atau barang.

Dengan demikian sistem merupakan kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Maksud dari suatu sistem adalah untuk mencapai suatu tujuan dan

sasaran dalam ruang lingkup yang sempit. Suatu sistem dapat terdiri dari sistem bagian (*subsystem*) dimana masing-masing *subsystem* dapat terdiri dari komponen-komponen yang saling berhubungan dan berinteraksi membentuk suatu kesatuan yang terintegrasi (*integrated*) sehingga tujuan atau sasaran sistem tersebut dapat tercapai.

2.2.2 Pengembangan Sistem Berbasis Web

Pengembangan sistem berbasis web adalah aplikasi yang sejak awal dirancang untuk dieksekusi di lingkungan berbasis web. Definisi ini mengungkapkan dua aspek penting dari aplikasi ini (Simarmata, 2009) sebagai berikut:

1. Suatu aplikasi *web* dirancang dapat berjalan di dalam lingkungan berbasis *web*. Artinya aspek-aspek hipermedia dalam kaitannya dengan *hiperteks* dan multimedia di dalam kombinasi dengan kelola aplikasi tradisional harus diperhitungkan di seluruh hidup aplikasi.
2. Aplikasi *web* adalah suatu aplikasi suatu aplikasi yang tidak hanya berupa sekumpulan halaman-halaman *web*.

2.2.3 *Javascript*

Javascript menurut (Sunyoto,2007:17) adalah bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar *browser* popoler seperti *Internet Explorer* (IE), *Mozilla Firefox, Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *tag SCRIPT*.

Beberapa hal tentang Javascript:

1. *Javascript* didesain untuk menambah interaktif suatu *web*
2. *Javascript* merupakan sebuah bahasa *scripting*.
3. Bahasa *scripting* merupakan bahasa pemograman yang ringan.
4. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*).
5. *Javascript* biasanya disisipkan (*embedded*) dalam halaman HTML.
6. *Javascript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi).

7. Setiap orang dapat menggunakan *Javascript* tanpa membayar lisensi.

2.2.4 CSS *(Cascanding Style Sheet)*

Menurut Bunafit Nugroho (2014:1), *Cascanding Style Sheet* adalah bahasa *stye sheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup.* CSS bekerja sebagai pelengkap pada elemen *HTML* yang kesemuanya itu dapat dikendalikan dengan menggunakan dengan menggunakan sebuah bahasa *script* CSS. Penggunaan CSS dilakukan untuk memperluas kemampuan HTML dalam memformat dokumen *web* atau untuk memperindah tampilan *web*. Penulisan kode CSS disisipkan pada *tak* HTML. Kode CSS ditulis dengan *tak* <*style*> dan <*/style*> dengan mendefinisikan suatu *syle* baru yang kemudian dapat digunakan berulang kali.

2.2.5 XAMPP

Menurut Riyanto (2015:3), XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah data MySQL di komputer lokal. XAMPP berperan sebagai *server web* pada komputer Anda. XAMPP juga dapat disebut sebuah *CPanel server virtual,* yang dapat membantu Anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

2.2.6 MySQL

Menurut Riyanto (2015:28), MySQL adalah RDBMS yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*), di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *close source* atau komersial. Karena sifatnya yang *Open Source,* sehingga komunitas umum turut mengembangkan mesin basis data MySQL dan hal ini menyebabkan kemampuan dan performanya berkembang dengan pesat.

2.2.7 *Unified Modeling Language* (UML)

Menurut Booch (2005:7) UML adalah Bahasa standar untuk membuat rancangan *software*. UML biasanya digunakan untuk menggambarkan dan

membangun, dokumen artifak dari *software –intensive system*. Model UML yang dipakai dalam pembuatan sistem informasi Apotek Kresna antara lain *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, *ER Diagram*.

2.2.8 *Use Case Diagram*

Menurut Whitten, Bentley & Dittman (2004, p. 271) *Use case modeling* adalah sebuah pendekatan yang memfasilitasi pengembangan yang berpusat pada penggunaan. *Use case* diagram digunakan untuk memodelkan proses berdasarkan perspektif pengguna sistem. *Use case* diagram terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. Simbol-simbol yang digunakan dalam *use case diagram* dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Use Case Diagram*

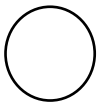
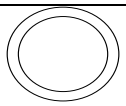
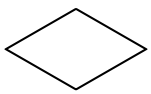
No	Gambar	Nama	Keterangan
1		*Actor*	Menggambarkan sesorang yang beriteraksi dengan sistem, di mana hanya bisa menginputkan informasi dan menerima indormasi dari sistem dan tidak memegang kendali pada use case
2		*Dependency*	Sebuah elemen yang bergantung beberapa cara kepada elemen lainnya.
3		*Generalization*	Sebuah elemen yang menjadi spesialisasi dari elemen yang lain
4		*Include*	Metode yang harus terpenuhi agar sebuah event dapat terjadi.

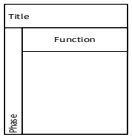
No	Gambar	Nama	Keterangan
5		*Extend*	Metode yang hanya berjalan di bawah kondisi tertentu
6		*Association*	Menghubungkan link antar element

2.2.9 *Activity Diagram*

Menurut Whitten, Bentley & Dittman (2004), *activity diagram* digunakan untuk menggambarkan alur dari proses bisnis atau langkah – langkah *usecase* secara berurutan. Diagram ini juga digunakan untuk menggambar *action* (tindakan) yang akan dieksekusikan ketika suatu proses sedang berjalan dan berserta hasil dari proses eksekusi tersebut. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada Tabel 2.2.

Tabel 2.2 *Activity Diagram*

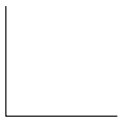
No	**Gambar**	**Nama**	**Keterangan**
1		*Start Point*	Merupakan awal dalam aktifitas
2		*End Point*	Akhir dalam aktivitas
3		*Activities*	Menggambarkan proses kegiatan bisnis
4		*Decision Point*	Menggambarkan pilihan pengambilan keputusan aktifitas.

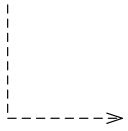
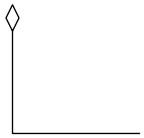
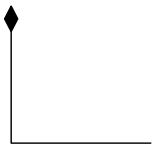
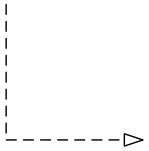
No	Gambar	Nama	Keterangan
5		*Swimlane*	Berfungsi sebagai pembagi activity diagram yang menunjuksiapa yang melakukan aktifitas

2.2.10 *Class Diagram*

Menurut Whitten, Bentley & Dittman (2004), *class diagram* menggambarkan struktur objek yang terdapat pada sebuah sistem. Diagram ini menunjukkan objek – objek yang terdapat pada suatu sistem dan relasi antar objek – objek tersebut. *Class* memiliki tiga area pokok yaitu: Nama, Atribut dan Metode / *Operation*. Atribut dan Metoda dapat mempunyai sifat *Private* (tidak dapat dipanggil dari luar class), *Protected* (Hanya dapat dipanggil oleh class yang bersangkutan dan anak yang mewarisinya) dan *Public* (dapat dipanggil siapa saja). Berikut adalah contoh dari sebuah kelas. *Class Diagram* secara khas meliputi: Nama Kelas (*Class Name*), Atribut (*Attributes*), Operasi (*Operations*), dan Relasi (*Relationships*).

Tabel 2.3 *Class Diagram*

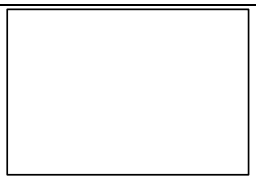
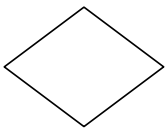
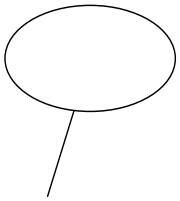
No	Gambar	Nama	Keterangan
1		*Assosiation*	Hubungan statis antar kelas. Asosiasi menggambarkan kelas yang memiliki atribut berupa kelas lain
2		*Directed Assosiation*	Asosiasi dengan makna kelas yang satu digunakan oleh kelas yang lain.
3		*Generalization*	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus) atau untuk menyatakan hubungan *intheritance*

No	Gambar	Nama	Keterangan
4		*Dependency*	Relasi antar kelas dengan makna kebergantungan antar kelas.
5		*Agregation*	Hubungan yang menyatakan bahwa suatu kelas menjadi atribut bagi kelas lain.
6		*Composition*	Bentuk khusus dari agregasi dimana kelas yang menjadi bagian diciptakan setelah kelas menjadi *whole*.
7		*Realization*	Hubungan antar kelas dimana sebuah kelas memiliki keharusan untuk mengikuti aturan yang ditetapkan oleh kelas lainnya

2.2.11 *Entity Relationship Diagram* (ERD)

Menurut Brady dan Loonam (2010). *Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analisis dalam tahap analisis persyaratan proyek pengembangan sistem Terdapat beberapa alat atau tools yang digunakan untuk membuat pemodelan data. Salah satunya adalah *Entity Reliationship Diagram* (ERD) yang merupakan pemodelan data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang digambarkan oleh data tersebut. Untuk menggambarkan digunakan beberapa notasi dan simbol seperti pada tabel 2.4

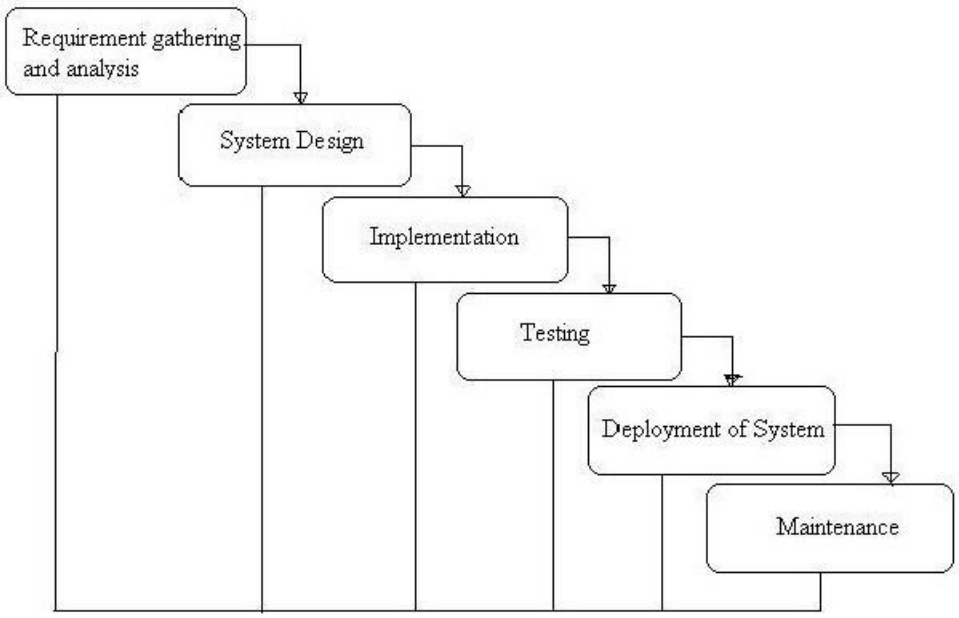
Tabel 2.4 *Entity Relationship Diagram*

No	Gambar	Nama Gambar	Keterangan
1		*Entitas*	Suatu objek yang dapat di identifikasi dalam lingkungan pemakai
2		*Relasi*	Menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda
3		*Atribut*	Berfungsi mendeskripsikan karakter entitas(atribut yang berfungsi sebagai key diberi garis bawah).
4		*Association*	Sebagai penghubung antara relasi dengan entitas, relasi, dan entitas dengan atribut.

2.3

System Development Life Cycle

Dalam alur penelitian, metode yang digunakan adalah model SDLC (*System Development Life Cycle*). SDLC adalah suatu kerangka yang menggambarkan beberapa kegiatan yang dilakukan melalui beberapa tahap dalam pembuatan sebuah *software* (Fatta, 2007). Selain itu, SDLC juga penting untuk proses *maintenance software* itu sendiri. Model SDLC yang dipakai dalam pengembangan aplikasi adalah model *Waterfall. Waterfall model* adalah sebuah contoh dari proses perencanaan dimana semua proses kegiatan harus terlebih dahulu direncanakan dan dijadwalkan sebelum dikerjakan. *Waterfall Model* atau *Classic Life Cycle* merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Disebut *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. (Sommerville, 2011*),*



Gambar 2.1 Ilustrasi Model *Waterfall*

Terdapat 5 tahapan metode *Waterfall* yang dapat dilihat pada gambar 2.1 di mulai dari analisis dan definisi kebutuhan hingga operasi dan pemeliharaan. Berikut penjelasan dari tahapan metode *Waterfall*:

1. Analisis dan definisi kebutuhan. Layanan, batasan, dan tujuan sistem ditentukan melalui konsultasi dengan *user*.
2. Perancangan sistem dan perangkat lunak. Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan, Perancangan melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar.
3. Implementasi dan pengujian unit. Pada tahap ini, perancangan perangkat lunak direalisasikan dengan program atau unit program. Pengujian ini melibatkan verifikasi bahwa setiap unit telah memenuhi spesifikasinya.
4. Integrasi dan pengujian sistem. Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah dipenuhi.
5. Operasi dan pemeliharaan, yaitu mengoperasikan program di lingkungannya dan melakukan pemeliharaan. Biasanya ini merupakan fase siklus hidup yang paling lama.

2.4 Metode Pengujian Sistem

Pengujian sistem menyajikan anomali yang menarik bagi rekayasa perangkat lunak pada proses perangkat lunak, perekayasa berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru kemudian dilakukan pengujian.

2.4.1 Pengujian *Black Box*

Dalam pengujian perangkat lunak ada dua yaitu *white box testing* dan *black box testing*. Dari kedua metode itu, penulis memilih menggunakan *black box testing* karena dianggap lebih tepat dibanding *white box testing*. Perangkat lunak memerlukan tes untuk pencarian kesalahan fungsi-fungsi dalam aplikasi sehingga dalam hal ini *black box testing* lebih sesuai. Pengujian ini digunakan untuk mengetahui apakah fungsi-fungsi dalam perangkat lunak sudah sesuai dengan yang diharapkan.

Menurut Roger S. Pressman (2010), *black box testing* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineer* untuk memperoleh *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

1. Fungsi yang tidak benar atau fungsi yang hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Kesalahan inisialisasi dan pemutusan kesalahan.