

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

A. Konsep Dasar Sistem Informasi

1. Metode *Waterfall*

Menurut Pressman (2010:36) Sekuensial linier atau lebih sering disebut metode *waterfall* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut:

a. Rekayasa dan pemodelan sistem atau informasi.

Karena perangkat lunak selalu merupakan bagian dari sistem (bisnis) yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut. Pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain seperti perangkat lunak manusia, dan *database*. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

b. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak. Untuk memahami sifat program yang dibangun, perekayasa perangkat lunak (analisis) harus memahami domain informasi,

tingkah laku, unjuk kerja, dan antarmuka (*interface*) yang diperlukan. Kebutuhan baik untuk sistem maupun perangkat lunak didokumentasikan dan dilihat lagi dengan pelanggan.

c. Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda, struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural. Proses desain menerjemahkan syarat atau kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi perangkat lunak.

d. Generasi kode

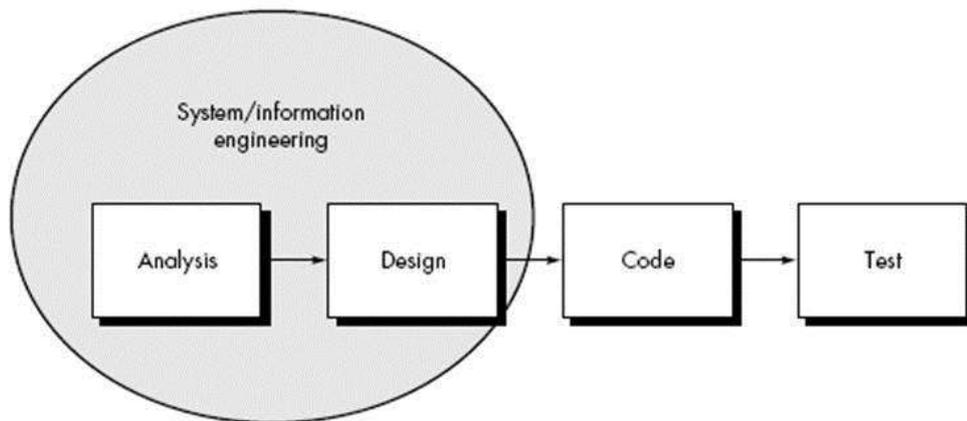
Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

e. Pengujian

Sekali kode dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

f. Pemeliharaan

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah perangkat lunak yang diletakkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya, atau karena pelanggan membutuhkan perkembangan fungsional atau untuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap *fase* program sebelumnya dan tidak membuat yang baru lagi.



Gambar II.1

Linear Sequential Model

Sumber: Roger S. Pressman, Ph. D (2010:36)

2. *Black Box Testing*

Menurut Pressman, (459: 2010) “Pengujian *Black Box* adalah pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak”. Metode ini digunakan untuk mengetahui apakah perangkat lunak

berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi perangkat lunak. Data uji dieksekusi pada perangkat lunak dan kemudian keluar dari perangkat lunak dicek apakah telah sesuai yang diharapkan.

Menurut Pressman (2010:459) “Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori diantaranya” :

- a. Fungsi-fungsi yang tidak benar atau hilang.
- b. Kesalahan interface.
- c. Kesalahan dalam struktur data atau akses database eksternal.
- d. Kesalahan kinerja.
- e. Inisialisasi dan kesalahan terminasi.



Sumber: Roger S. Pressman, Ph. D (2010:459)

Gambar II.2

Black Box Testing

Keunggulan *Black Box* yaitu bisa memilih subset *test* secara efektif dan efisien, dapat menemukan cacat, dan bisa memaksimalkan testing investment. Sedangkan kelemahan dari *Black Box* yaitu *tester* tidak pernah yakin apakah perangkat lunak tersebut benar-benar lolos uji.

Menurut Pressman (2010: 459) “Langkah-langkah dalam pengujian *Black Box*, yaitu” :

- a. Analisa kebutuhan dan spesifikasi
- b. Pemilihan input
- c. Pemilihan outputnya
- d. Seleksi Input
- e. Pengujian
- f. Revisi hasil
- g. Evaluasi

3. Pengertian Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen atau variabel-variabel yang terorganisir, saling berinteraksi, saling ketergantungan satu sama lainnya dan terpadu. Sistem juga merupakan suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai tujuan. Ada beberapa pakar yang berlainan pendapat dalam mendefinisikan suatu sistem yaitu sebagai berikut :

Menurut Jogiyanto (2005:1) bahwa “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran yang tertentu”.

Menurut Jogiyanto (2005:2) bahwa “Suatu sistem adalah pendekatan sistem yang lebih menekankan pada elemen atau komponennya, mendefinisikan sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu”.

Menurut Romley (2006:2) bahwa “Sistem adalah rangkaian dari dua atau lebih komponen-komponen yang saling berhubungan, yang berinteraksi untuk mencapai suatu tujuan”.

Menurut Winarno (2006:5) bahwa “Sistem adalah sekumpulan komponen yang saling bekerja sama untuk mencapai satu tujuan. masing-masing komponen memiliki fungsi berbeda dengan yang lain,tetapi tetap dapat bekerja sama”.

4. Pengertian Sistem Informasi

Menurut Alter dalam Kadir dan Triwahyuni (2014:546) “sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi”.

Menurut Burch dan Grudnitski dalam Kadir dan Triwahyuni (2014:546) Kualitas informasi ditentukan oleh tiga (3) faktor, yaitu:

- a. Relevansi
- b. Tepat waktu, dan
- c. Akurasi.

Akurasi berarti bahwa informasi bebas dari kesalahan.Relevansi berarti bahwa informasi benar-benar berguna bagi suatu tindakan, putusan yang dilakukan oleh seorang.Tepat waktu berarti bahwa informasi datang pada saat dibutuhkan sehingga bermanfaat untuk pengambilan keputusan.

B. *Persediaan Barang (Inventory)*

Menurut M. Munandar dalam Marihot Manullang dan Dearlina Sinaga (2008 : 50) bahwa yang dimaksud dengan inventory adalah persediaan barang-barang yang menjadi objek usaha pokok perusahaan, bagi perusahaan perdagangan barang-barang tersebut berupa persediaan barang dagangan, sedangkan bagi perusahaan yang memproduksi (industri) berupah persediaan

barang mentah, persediaan bahan pembantu, persediaan barang yang sedang diproses dan persediaan barang jadi.

C. *Unified Modeling Language*(UML)

1. Pengertian UML

Menurut Nugroho (2010:6) “UML (*Unified Modelling Language*) adalah bahasa pemodelan sistem atau perangkat lunak yang berparadigma berorientasi objek”. Pengembangan UML dimulai dari kerjasama Grady Booch dan James Rumbaugh pada tahun 1994 untuk mengkombinasikan 2 metologi terkenal Booch dan OMT (*Object Management Group*). Notasi UML terutama diurutkan dari 3 notasi yang telah ada sebelumnya, yakni Grady Booch OOD (*Object Oriented Design*), Jim Rumbough OMT (*Object Modelling Technique*) dan Ivar jacobson OOSE (*Object Oriented Software Engineering*).

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari sistem perangkat lunak. Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standardisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Seperti yang kita ketahui bahwa menyatukan banyak kepala untuk menceritakan sebuah ide dengan tujuan untuk memahami hal yang sama tidaklah mudah, oleh karena itu diperlukan sebuah bahasa pemodelan perangkat lunak yang dapat dimengerti oleh banyak orang.

2. Diagram UML

Menurut Nugroho (2010:5) “Berikut ini adalah definisi mengenai 5 (lima) diagram UML”:

a. *Use Case Diagram*

Use case diagram secara grafis menggambarkan interaksi antara sistem, sistem eksternal dan pengguna. Dengan kata lain use case diagram secara grafis mendeskripsikan siapa yang akan menggunakan sistem dan dalam cara apa pengguna (user) mengharapkan interaksi dengan sistem itu. Use case secara naratif digunakan untuk secara tekstual menggambarkan sekuensi langkah-langkah dari setiap interaksi.

b. *Class Diagram*

Menggambarakan struktur *object* sistem. Diagram ini menunjukkan *class object* yang menyusun sistem dan juga hubungan antara *class object* tersebut.

c. *Sequence Diagram*

Secara grafis menggambarkan bagaimana objek berinteraksi dengan satu sama lain melalui pesan pada sekuensi sebuah use case atau operasi.

d. *State Chart Diagram*

Digunakan untuk memodelkan behavior objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan event-event (kejadian) yang menyebabkan objek beralih dari satu state ke state yang lain.

e. *Activity Diagram*

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun use case. Activity diagram dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut.

D. *Entity Relationship Diagram*(ERD)

Menurut Nugroho (2010:57) “Model E-R (ERD) adalah perincian yang merupakan organisasi atau area tertentu”.ERD digunakan untuk pemodelan basis data relasional. Sehingga jika penyimpanan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD.

Komponen yang terdapat dalam *Entity Relationship Diagram* menurut Nugroho (2010:57) adalah sebagai berikut:

1. Entity (Entitas)

Pada ER diagram, entity digambarkan dengan sebuah bentuk persegi panjang. Entity adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu didalamnya).

2. *Relationship*

Pada ER diagram, Relationship dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya

(bisa dengan kalimat aktif atau kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

3. *Relationship Degree*

Relationship degree atau derajat *relationship* adalah jumlah entitas yang berpartisipasi dalam satu *relationship*. Derajat *relationship* yang sering dipakai didalam ERD:

a. *Unary Relationship*

Unary relationship adalah model *relationship* yang terjadi diantara *entity* yang berasal dari *entity set* yang sama. Sering juga disebut sebagai *Recursive Relationship* atau *Reflective Relationship*.

b. *Binary Relationship*

Binary Relationship adalah model *relationship* antara *instance* dari suatu tipe entitas (dua *entity* yang berasal dari *entity* yang sama). *Relationship* ini paling umum digunakan dalam pembuatan model data.

c. *Ternary Relationship*

Ternary Relationship adalah merupakan antara *instance* dari tiga tipe entitas secara serentak.

4. *Attribute Value*

Attribute value atau nilai *atribute* adalah suatu occurrence tertentu dari sebuah *attribute* didalam suatu *entity* atau *relationship*. Ada dua jenis atribut:

- a. Identifier (key) digunakan untuk menentukan suatu *entity* secara unik (primary key).

- b. Descriptor (*nonkey attribute*) digunakan untuk menspesifikasikan karakteristik dari suatu entity yang tidak unik.

5. Kardinalitas (*Cardinality*)

Menurut Simarmata dan Paryudi (2006:63) “Pemetaan kardinalitas menyatakan jumlah entitas di mana entitas lain dapat dihubungkan ke entitas tersebut melalui sebuah himpunan relasi”.

Pemetaan kardinalitas sangat berguna dalam menentukan himpunan relasi biner meskipun pemetaan dapat berperan dalam deskripsi himpunan relasi yang melibatkan lebih dari dua himpunan entitas.

Untuk suatu himpunan relasi biner R antara himpunan entitas A dan B, pemetaan kardinalitas harus salah satu dari berikut:

- a. *One-to-One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B dan sebuah entitas pada B berhubungan dengan paling banyak satu entitas pada A.

- b. *One-to-Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan paling banyak satu entitas pada A.

- c. *Many-to-One*

Sebuah entitas pada A berhubungan dengan paling banyak satu entitas pada B. Sebuah entitas pada B dapat dihubungkan dengan nol atau lebih entitas pada A.

d. *Many-to-Many*

Sebuah entitas pada A berhubungan dengan nol atau lebih entitas pada B dan sebuah entitas pada B dapat dihubungkan nol atau lebih entitas pada A.

E. LRS (Logical Record Structure)

Menurut Riyanto (2005:22) Mendefinisikan bahwa “LRS (logical record structure) adalah representasi dari struktur record-record pada tabel-tabel yang terbentuk dari hasil antar himpunan entitas”.

Logical record structure dibentuk dengan nomor dari tipe record. Logical record structure terdiri dari link-link diantara tipe record. Link ini menunjukkan arah dari satu tipe record lainnya. Banyak link dari LRS yang diberi tanda field-field yang kelihatan pada kedua link tipe record. Penggambaran LRS mulai dengan menggunakan model yang dimengerti. Dua metode yang dapat digunakan, dimulai dengan hubungan kedua model yang dapat dikonversikan ke LRS. Metode yang lain dimulai dengan ER-diagram dan langsung dikonversikan ke LRS.

2.2 Penelitian Terkait

Minarni dan Susanti dalam Jurnal Sistem Informasi Pada Rumah Sakit Umum Daerah (RSUD) Padang merancang sistem persediaan obat yang dapat digunakan oleh bagian gudang barang untuk menginput transaksi persediaan dengan menggunakan database sebagai tempat penyimpanan data yang bertujuan untuk mengurangi duplikasi data (*Data Redudancy*) dan meningkatkan hubungan antar data (*Data Relatability*) dengan hasil pembahasannya adalah sebagai masukan dan bahan pertimbangan bagi bagian gudang obat RSUD Padang dalam mengatasi masalah yang ada, serta sebagai masukan dalam memperbaiki sistem informasi inventory obat yang sudah ada.

Hermansyah dan Rian Jatmiko Winandy dalam Jurnal Monitoring Persediaan Berbasis Web membuat penelitian dengan merancang sistem persediaan berbasis website yang dibuat menggunakan MySQL sebagai

database serta bahasa pemrograman PHP dalam, pembuatannya. Penelitian ini menghasilkan suatu program yang dapat memonitor stok barang secara online dan dapat mengupdate informasi mengenai persediaan barang secara cepat dan tepat mengenai situasi stok tanpa adanya penundaan informasi.