

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Informasi**

Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005:11).

Dari pengertian diatas dapat disimpulkan sistem informasi adalah suatu sistem dalam suatu organisasi untuk pengolahan semua kegiatan-kegiatan dalam suatu organisasi dengan tujuan memberikan informasi untuk kebutuhan pihak luar yang membutuhkan.

##### **2.1.1 Definisi Sistem**

Terdapat dua kelompok pendekatan dalam mendefinisikan sistem, yaitu menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan suatu sistem sebagai berikut :

1. Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu (Yasin, 2012:260).
2. Sistem adalah kumpulan atau himpunan dari unsur atau variable-variabel yang saling terorganisasi, saling berinteraksi dan saling bergantung satu sama lain (Fatta, 2007:3).
3. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto, 2014:8).

Pendapat diatas dapat disimpulkan bahwa sistem adalah kumpulan atau sekelompok dari elemen-elemen yang saling terintegrasi untuk mencapai suatu tujuan tertentu. Jadi sistem informasi adalah sekumpulan elemen-elemen atau prosedur yang Saling berinteraksi untuk mengolah data menjadi lebih berguna untuk mencapai tujuan tertentu.

### **2.1.2 Karakteristik Sistem**

Suatu sistem mempunyai karakteristik tertentu, yang menjelaskan bahwa hal tersebut bisa dikatakan sebagai suatu sistem (Verdi Yasin, 2012). Adapun yang termasuk kedalam karakteristik sistem adalah sebagai berikut :

#### **1. Komponen Sistem (*Components System*)**

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama dalam membentuk satu kesatuan. Komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem memiliki sifat-sifat dari sistem untuk menjalankan fungsi tertentu dan mempengaruhi proses secara keseluruhan. Misalnya suatu perusahaan dapat disebut dengan sistem dan industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem.

#### **2. Batasan Sistem (*Boundary*)**

Batasan sistem merupakan daerah yang membatasi antara satu sistem dengan sistem yang lainya atau dengan lingkungan luarnya. Batasan ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup dari sistem tersebut.

#### **3. Lingkungan Luar Sistem (*Environment*)**

Lingkungan luar dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar dapat bersifat menguntungkan dapat juga bersifat merugikan.

#### 4. Penghubung Sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Dengan penghubung setiap subsistem berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

#### 5. Masukkan Sistem (*Input*)

Masukkan dari sistem adalah energi yang dimasukkan kedalam suatu sistem. Masukan dapat berupa perawatan (*maintenance input*) dan masukan sinyal (*Signal Input*). *Maintenance* input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk mendapatkan keluaran.

#### 6. Keluaran Sistem (*Output*)

Keluaran dari pada sistem adalah hasil dari energi yang telah diolah dan diklasifikasi menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat berupa masukan untuk subsistem yang lain atau untuk suprasistem.

#### 7. Pengolahan Sistem (*Process*)

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan mengubah masukan menjadi keluaran.

#### 8. Sasaran Sistem (*Objectives*)

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objectives*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

## 2.2 Pengembangan Sistem

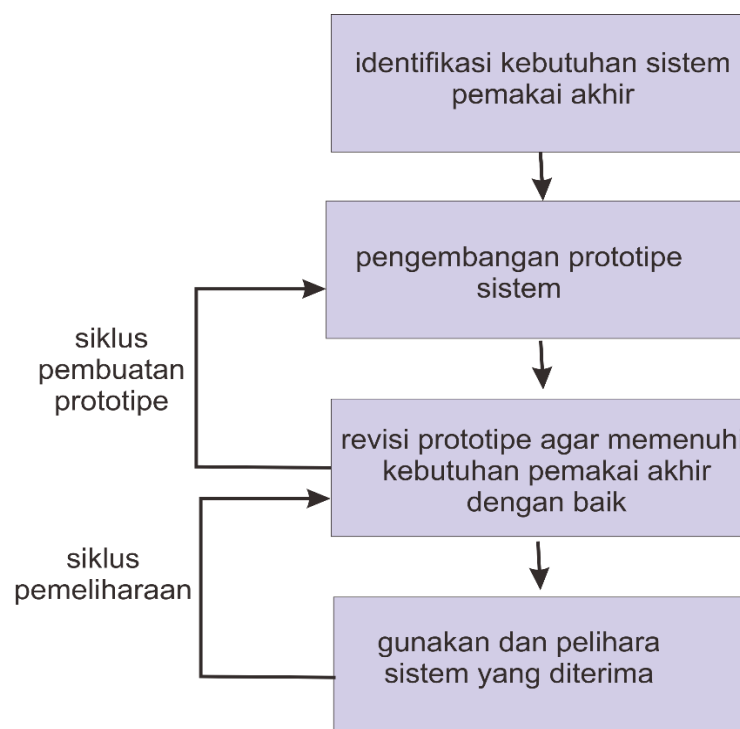
### 2.2.1 Metode Pengembangan Sistem

Metode yang digunakan dalam pengembangan sistem ini menerapkan metode prototipe yaitu Proses pengembangan sistem seringkali menggunakan pendekatan prototipe (*prototyping*). Metode ini sangat baik digunakan untuk menyelesaikan masalah kesalahpahaman antara *user*

dan analisis yang timbul akibat *user* tidak mampu mendefinisikan secara jelas kebutuhannya (Mulyanto, 2009).

*Prototyping* adalah pengembangan yang cepat dan pengujian terhadap model kerja (prototipe) dari aplikasi baru melalui proses interaksi dan berulang-ulang yang biasa digunakan ahli sistem informasi dan ahli bisnis. *Prototyping* disebut juga desain aplikasi cepat (*rapid application design/RAD*) karena menyederhanakan dan mempercepat desain sistem (O'Brien, 2005).

Sebagian *user* kesulitan mengungkapkan keinginannya untuk mendapatkan aplikasi yang sesuai dengan kebutuhannya. Kesulitan ini yang perlu diselesaikan oleh analis dengan memahami kebutuhan *user* dan menerjemahkannya ke dalam bentuk model (prototipe). Model ini selanjutnya diperbaiki secara terus menerus sampai sesuai dengan kebutuhan *user*. Berikut siklus pengembangan perangkat lunak ditunjukkan oleh gambar 2.1.



Gambar 2.1 Sistem Model *Prototyping*

Pada gambar 2.1 dapat dijelaskan mengenai pendekatan alur perangkat lunak secara terurut dimulai dari analisis kebutuhan sistem, desain, pengkodean, serta pengujian. Pada laporan ini, tahap yang sudah dilakukan adalah analisis dan desain, berikut penjelasan masing-masing tahap:

a. Analisis kebutuhan sistem

Peneliti melakukan survei serta analisis pada sistem yang selama ini berjalan pada KPP Pratama Tanjung Karang. Pada tahap ini peneliti dan pemakai saling berdiskusi mengenai sistem yang sudah berjalan dan sistem yang akan dibuat nanti serta melakukan pengumpulan kebutuhan.

b. Desain

Peneliti melakukan perancangan desain pembuatan program perangkat lunak, tampilan sistem yang akan dibuat, serta pemodelan sistem.

### **2.2.2 Konsep OOP (*Object Oriented Programming*)**

OOP merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek dunia nyata. (Rossa, 2013).

Terdapat beberapa konsep mengenai OOP yaitu:

1. Kelas Abstrak (*Class Abstraksi*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti. Suatu kelas dapat diturunkan kepada kelas lainnya, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

## 2. Enkapsulasi (*Encapsulation*)

Enkapsulasi merupakan pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya. Proses enkapsulasi memudahkan kita untuk menggunakan sebuah objek dari suatu kelas karena kita tidak perlu mengetahui segala hal secara rinci.

## 3. Pewarisan (*Inheritance*)

Pewarisan adalah mekanisme yang memungkinkan suatu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

## 4. Polimorfisme (*Polymorphism*)

Polimorfisme adalah kemampuan suatu objek untuk digunakan di banyaktujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

### **2.2.3 UML (*Unified Modeling Language* )**

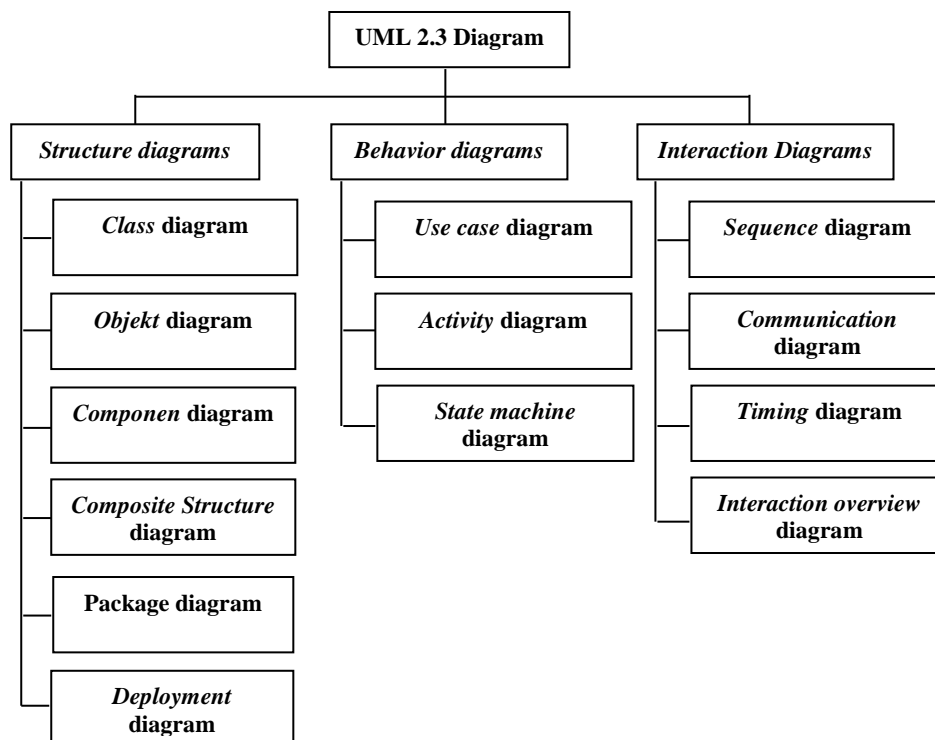
*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia dindustri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (Rosa,2013).

### **2.2.4 Diagram UML**

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Pada UML 2.3 terdiri dari beberapa macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari masing masing pembagian kategori tersebut.

- a. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

- b. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- c. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada sistem.



Gambar 2.2 Diagram UML

(Rosa, 2013)

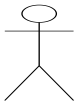


Adapun penjelasan diagram yang digunakan dalam perancangan sistem dari masing masing kategori yang telah diuraikan antara lain :

### 1. *Use Case Diagram*

*Use case Diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* merupakan konstruksi untuk

mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. (Rosa,2013).

Tabel 2.1 Simbol *Use Case*

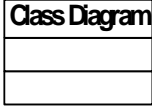
No.	Notasi	Keterangan	Simbol
1.	<i>Actor</i>	Pengguna system atau yang berinteraksi langsung dengan sistem,bias manusia, aplikasi, ataupun objek lain	
2.	<i>Use case</i>	Digambarkan dengan lingkaran elips dengan nama <i>use case</i> nya tertulis ditengah lingkaran	
3.	<i>Association</i>	Digambarkan dengan sebuah garis yang berfungsi menghubungkan <i>actor</i> dengan <i>use case</i> .	

## 2. *Class Diagrams*

*Class Diagram* adalah deskripsi kelompok obyek-obyek dengan *property*. *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Sebuah sistem biasanya mempunyai beberapa *class diagram*. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu sistem (Rosa, 2013).



Tabel 2.2 Simbol *Class Diagram*



No.	Notasi	Keterangan	Simbol
1.	<i>Object</i>	Blok-blok pembangun program. Bagian atas <i>class</i> menunjukkan namadari <i>class</i> , bagian tengah mengindikasikan atribut dari <i>class</i> , dan bagian bawah mendefinisikan <i>method</i> dari sebuah <i>class</i>	
2.	<i>Activation</i>	Menunjukkan <i>relationship</i> atau hubungan antar <i>class</i>	—————
3.	<i>Lifeline</i>	Menunjukkan ketergantungan antara satu <i>class</i> dengan <i>class</i> yang lain	----->
4.	<i>Message</i>	Menunjukkan <i>inheritance</i> dari satu <i>class</i> ke beberapa <i>class</i>	—————>

### 3. *Sequence Diagram*

*Sequence Diagram* digunakan untuk menggambarkan perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek pada sebuah scenario dan sesuatu yang terjadi pada titik tertentu dalam eksekusi system. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang

terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek. (Rosa, 2013).

Tabel 2.3 Simbol *Sequence Diagram*

No.	Notasi	Keterangan	Simbol
1.	<i>Object</i>	Object adalah instance dari sebuah class yang dituliskan tersusun secara horizontal diikuti lifeline	
2.	<i>Activation</i>	Indikasi dari sebuah objek yang melakukan suatu aksi	
3.	<i>Lifeline</i>	Indikasi keberadaan sebuah objek dalam basis waktu	
4.	<i>Message</i>	Indikasi untuk komunikasi antar object	
5.	<i>Self-Message</i>	Komunikasi kembali kedalam object itu sendiri	

### 2.3 Visual Basic .NET

Visual Basic merupakan salah satu bahasa pemrograman yang berorientasi objek (*Object Orientasi Programming*) yang bersifat klasik karena telah banyak dipakai untuk pembuatan aplikasi-aplikasi oleh programmer-programmer diberbagai dunia (Wahana Komputer, 2013 : Kata Pengantar).

Visual basic 2012 merupakan salah satu paket teknologi bahasa pemrograman yang dikeluarkan oleh Microsoft. Bahasa pemrograman visual basic

digunakan untuk membuat aplikasi windows yang berbasis *Graphical User Interface* (GUI) (Wahana Komputer, 2013:2).

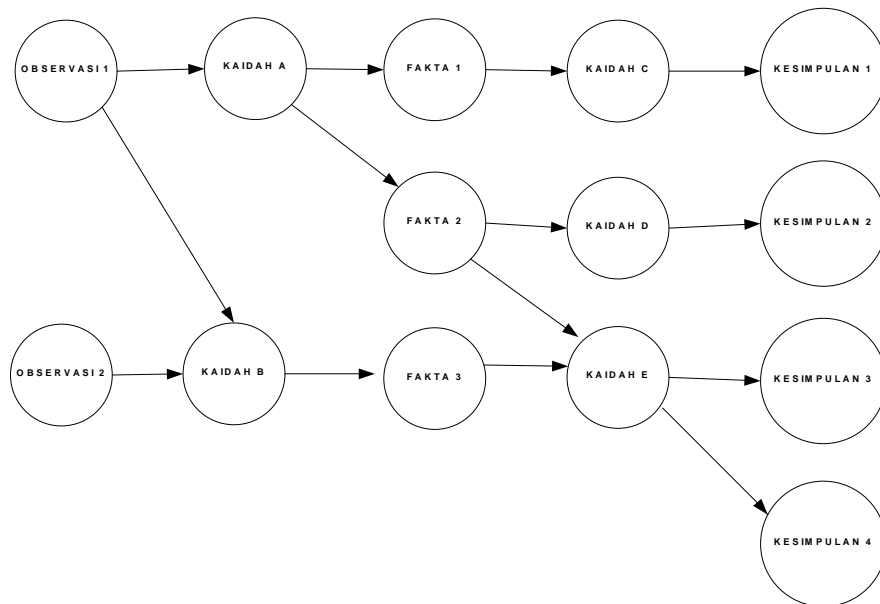
#### **2.4 SQL Server 2012**

SQL server 2012 merupakan database yang merupakan bawaan dari visual studio 2012 yang digunakan dalam pengelolaan data project dalam visual studio 2012 sehingga memudahkan pengguna dalam melakukan *maintenance* database secara langsung karena sudah menjadi satu dengan visual studio 2012 (wahana komputer, 2013).

#### **2.5 Forward Chaining**

Wiwi Verina (2015) Jurusan Teknik Informatika Universitas Potensi Utama dengan judul Penerapan Metode Forward Chaining untuk Mendeteksi Penyakit THT, Forward Chaining adalah teknik pencarian yang dimulai dengan fakta yang diketahui, kemudian mencocokkan fakta-fakta tersebut dengan bagian IF dari rules IF-THEN. Bila ada fakta yang cocok dengan bagian IF, maka rule tersebut dieksekusi. Bila sebuah rule dieksekusi, maka sebuah fakta baru (bagian THEN) ditambahkan ke dalam database. Setiap kali pencocokan, dimulai dari rule teratas. Setiap rule hanya boleh dieksekusi sekali saja. Proses pencocokan berhenti bila tidak ada lagi rule yang bisa dieksekusi. Metode pencarian yang digunakan adalah *Depth-First Search*(DFS), *Breadth-First Search*(BFS) atau *Best First Search*. pendekatan dalam pelacakan dimulai dari informasi masukan dan selanjutnya mencoba menggambarkan kesimpulan, pelacakan kedepan mencari fakta yang sesuai dengan bagian IF dari aturan IF-THEN. Dengan metode forward chaining dari pendekatan dan aturan yang telah dihasilkan dapat ditinjau oleh para ahli untuk diperbaiki atau dimodifikasi untuk memperoleh hasil yang lebih baik. Metode Forward Chaining adalah metode pencarian atau teknik pelacakan ke depan yang dimulai dengan informasi yang ada dan penggabungan rule untuk menghasilkan suatu kesimpulan atau tujuan. Pelacakan maju ini sangat baik jika bekerja dengan permasalahan yang dimulai dengan rekaman informasi

awal dan ingin dicapai penyelesaian akhir, karena seluruh proses akan dikerjakan secara berurutan maju. Dalam metode ini, data digunakan untuk menentukan aturan mana yang akan dijalankan, kemudian aturan tersebut dijalankan. Mungkin proses menambahkan data ke memori kerja. Proses diulang sampai ditemukan suatu hasil. Metode inferensi runut maju cocok digunakan untuk menangani masalah pengendalian (controlling) dan peramalan (prognosis) (Giarattano dan Riley, 1994). Untuk lebih jelasnya dapat kita lihat alur dari metode *Forward Chaining* pada Gambar 2.6



Gambar 2.6. Kaidah Forward Chaining