

BAB II

LANDASAN TEORI

2.1. SISTEM INFORMASI

Sistem informasi yang digunakan oleh seseorang di hampir setiap bidang kehidupan, memungkinkan untuk memfungsikan tidak hanya organisasi atau perusahaan tetapi juga dalam kehidupan sehari-hari. Dalam literatur tentang sistem informasi, ada banyak konsep dan definisi mengenai substansi dari sistem informasi. Dalam hal ini, kita dapat mengatakan bahwa sistem informasi dibentuk oleh informasi itu sendiri (Paula, 2014).

Sistem informasi adalah kombinasi yang sepenuhnya terorganisir dari peralatan teknis, perangkat lunak, infrastruktur, serta orang-orang yang merencanakan, mengontrol, dan mengoordinasi dalam pengambilan keputusan di perusahaan (Cashman, 2010).

2.1.1. Konsep Perancangan Sistem Informasi

Perancangan sistem dapat didefinisikan sebagai berikut : menurut Robert J. Verzello atau John Reuter III. Perancangan Sistem adalah tahap setelah analisis dari siklus pengembangan sistem sebagai pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi,

menggambarkan bagaimana suatu sistem dibentuk. Menurut John Burth dan Gary Grundnitski Perancangan Sistem didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Menurut George M. Scott perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem setelah instalasi dari sistem akan benar-benar memuat rancang bangun yang telah ditetapkan pada akhir analisis sistem.

2.2. SISTEM PENDUKUNG KEPUTUSAN

Sistem pendukung keputusan menurut Alters Keen merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi semistruktur dan situasi yang tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. (Kusrini, 2011).

Sistem Pendukung Keputusan (SPK) dapat didefinisikan sebagai suatu program komputer yang menyediakan informasi dalam domain aplikasi yang diberikan oleh suatu model analisis keputusan dan akses ke *database*, dimana hal ini ditujukan untuk mendukung pembuat keputusan (*decision maker*) dalam mengambil keputusan secara efektif, termasuk dalam kondisi yang kompleks dan tidak terstruktur. Sistem organisasi

paling tidak mencakup sistem fisik (sistem operasional), sistem manajemen (sistem keputusan), dan sistem informasi (Suyadi, 2011).

“Sistem Pendukung Keputusan merupakan penggabungan sumber - sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah – masalah semi struktur“ (Keen, 2011).

2.2.1. Pengambilan Keputusan

Pengambilan keputusan merupakan proses pemilihan *alternative* tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor – faktor yang perlu dipertimbangkan dalam pengambilan keputusan.

Pengambilan Keputusan terdapat beberapa tahapan, Dibawah ini tahap – tahap yang harus dilalui dalam proses pengambilan keputusan sebagai berikut :

1. Tahap Pemahaman (*Intelligence Phace*)

Pada tahap ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data *input* diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phace*)

Pada tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil. Tahap tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*Choice Phace*)

Pada tahap ini dilakukan pemilihan terhadap berbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan/ dengan memperhatikan kriteria – kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Impelementasi (*Implementation Phace*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan. (Simon, 2012).

2.3. ANALYTICAL HIERARCHY PROCESS (AHP)

Ciri khas sebuah *Decision Support System* digunakan model yang salah satu fungsinya adalah menyederhanakan masalah. AHP yang

dikembangkan oleh Thomas L Saaty merupakan model *hierarchy* fungsional dengan *input* utamanya adalah persepsi manusia.

Dengan adanya hierarki masalah yang kompleks atau tidak terstruktur dipecah dalam sub sub masalah kemudian disusun menjadi suatu bentuk hierarki. AHP mempunyai kemampuan untuk memecah masalah multi kriteria yang berdasar pada perbandingan preferensi dari setiap elemen dalam hierarki (Dalu, 2012).

Adapun langkah-langkah dalam metode AHP adalah sebagai berikut (Dalu, 2012) :

- i. Mendefinisikan masalah dan menentukan solusi yang diinginkan.
- ii. Membuat struktur hierarki, yang diawali dengan tujuan umum, dilanjutkan dengan sub-sub tujuan, kriteria dan kemungkinan alternatif pada tingkatan kriteria yang paling bawah.
- iii. Membuat matriks perbandingan berpasangan yang menggambarkan kontribusi relatif pengaruh setiap elemen terhadap masing-masing tujuan kriteria yang setingkat di atasnya. Perbandingan berdasarkan *judgement* dari pengambil keputusan dengan menilai tingkat kepentingan suatu elemen lainnya.
- iv. Melakukan perbandingan berpasangan sehingga diperoleh *judgment* seluruhnya sebanyak $n \times [(n-1)/4]$ buah, dengan n adalah banyaknya elemen yang dibandingkan.
- v. Menghitung nilai eigen dan mengkaji konsistensinya, jika tidak konsisten maka pengambilan data diulangi.
- vi. Mengulangi langkah 3,4, dan 5 untuk seluruh tingkat hierarki.

- vii. Menghitung *vector eigen* dari setiap matriks perbandingan berpasangan. Nilai *vector eigen* merupakan bobot setiap elemen. Langkah ini untuk mensintesis *judgment* dalam penentuan prioritas elemen-elemen pada tingkat hierarki terendah sampai pencapaian tujuan.
- viii. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10% maka penilaian data *judgement* harus diperbaiki.

2.4. Sistem Pendukung Keputusan Kenaikan Jabatan Metode *Analytical Hierarchy Process*

Merupakan penggabungan sumber-sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan dalam hasil dari proses kenaikan jabatan menggunakan suatu metode yang menggunakan persepsi manusia sebagai input utamanya dengan menyusun hierarki masalah yang kompleks atau tidak terstruktur dipecah dalam sub-sub masalah kemudian disusun menjadi suatu bentuk hierarki. Sehingga memperoleh hasil keputusan terbaik yaitu karyawan yang berhak mendapatkan kenaikan jabatan.

2.5. APLIKASI

Aplikasi adalah program yang digunakan untuk melakukan pekerjaan tertentu, misalkan aplikasi untuk menghitung gaji suatu perusahaan dan aplikasi untuk kasir dalam swalayan. Biasanya aplikasi dibuat oleh perusahaan atas permintaan seseorang/lembaga perusahaan-perusahaan, baik besar maupun kecil yang melayani pembuatan aplikasi untuk keperluan perusahaan, lembaga ataupun perorangan. (Maryono, 2010)

2.6. WEB

Web adalah bagian tertentu dari berbagai dokumen yang saling dihubungkan satu sama lain sehingga terbentuk jejaring web yang saling kait-mengait. Apabila diimplementasikan dalam sebuah jaringan komputer, dokumen yang berada dalam jaringan semacam itu dapat berdiam pada mesin-mesin berbeda membentuk sebuah jaring yang membentuk seluruh jaringan komputer. (Sadiman, 2010)

2.7. UML

Unified Modeling Language (UML) merupakan alat merancang perangkat lunak, sarana komunikasi antara perangkat lunak dengan menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem, mendokumentasikan sistem yang ada, proses-proses dan organisasinya (Herlawati, 2011).

Unified Modeling Language (UML) adalah salah satu bentuk *language* atau bahasa, menurut pencetusnya UML di definisikan sebagai bahasa visual untuk menjelaskan, memberikan spesifikasi, merancang, membuat model, dan mendokumentasikan aspek-aspek dari sebuah sistem. Salah satu cara untuk mengatur diagram UML adalah dengan menggunakan *view*. *View* adalah kumpulan diagram yang menggambarkan aspek yang sama dari proyek. *View* mempunyai 3 pelengkap, yaitu *Static View*, *Dynamic View*, dan *Functional View*.

2.7.1. *Static View*

Static View termasuk diagram yang memberikan gambaran dari unsur-unsur dari sistem tetapi tidak memberitahu bagaimana

elemen akan berperilaku. Hal ini sangat mirip *Blueprint*. *Blueprint* itu komprehensif, tetapi mereka hanya menunjukkan apa yang tetap diam, maka disebut *Static View*. *Static View* dibentuk oleh dua diagram, yaitu *Class Diagram* dan *Object Diagram*.

2.7.2. *Dynamic View*

Pada *Dynamic View* meliputi diagram yang mengungkapkan bagaimana benda berinteraksi dengan satu sama lain dalam respon terhadap lingkungan. Ini termasuk *Sequence Diagram* dan *Collaboration Diagram*, yang kolektif disebut sebagai diagram interaksi. Mereka secara khusus dirancang untuk menjelaskan bagaimana benda berbicara satu sama lain. Ini juga mencakup *Statechart Diagram*, yang menunjukkan bagaimana dan mengapa perubahan objek dari waktu ke waktu dalam menanggapi lingkungan.

2.7.3. *Functional View*




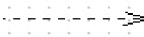
Functional View terbentuk oleh *Use Case Diagram* dan *Activity Diagram*.

2.7.3.1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja

yang berhak menggunakan fungsi-fungsi tersebut. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*. Simbol-simbol yang ada pada diagram *use case* :

Tabel 2.1 Simbol *Use Case* Diagram (Shalahuddin dan Rosa, 2011)

Simbol	Keterangan
	<i>Actor</i> : Sebuah peran yang dimainkan oleh seseorang, sistem, atau perangkat yang memiliki saham dalam keberhasilan operasi dari sistem.
	<i>Use Case</i> : Untuk mengungkapkan tujuan bahwa sistem harus dicapai.
	<i>Association</i> : Mengidentifikasi interaksi antara aktor dan <i>Use Case</i>
	<i>Dependency</i> : Mengidentifikasi hubungan komunikasi antara dua <i>Use Case</i>










2.7.3.2. Activity Diagram

Diagram aktivitas atau *activity* diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

Berikut simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.2 Simbol *Activity* Diagram (Shalahuddin dan Rosa , 2011)

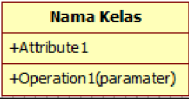





Simbol	Keterangan
	Titik Awal
	Titik Akhir
	<i>Activity</i>
	Pilihan untuk pengambilan keputusan
	<i>Fork</i> : Untuk menunjukkan kekuatan yang dilakukan secara paralel
	<i>Rake</i> : Menunjukkan adanya dekomposisi
	Tanda Waktu
	Tanda Penerimaan
	Aliran Akhir (<i>Flow Final</i>)

2.7.3.3. Class Diagram

Kelas Diagram terdiri dari tiga kompartemen (ruang persegi panjang) yang mengandung informasi yang

berbeda diperlukan untuk menjelaskan sifat-sifat satu jenis objek. Simbol-simbol yang digunakan dalam *class* diagram. Simbol-simbol yang digunakan dalam *class* diagram :

Tabel 2.3 Simbol *Class* Diagram (Rosa, 2011)

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Paket/ <i>package</i> 	Paket/ <i>package</i> merupakan sebuah bungkus dari satu atau lebih kelas (kumpulan kelas)
Asosiasi 	Asosiasi merupakan hubungan antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>
Generalisasi 	Generalisasi merupakan hubungan generalisasi dan spesialisasi (umum khusus) antara dua kelas dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi yang lainnya
Dependency 	Dependency merupakan hubungan antarkelas yang saling bergantung, membutuhkan satu sama lain.
Agregasi 	Agregasi merupakan hubungan antar kelas dimana satu kelas merupakan semua bagian dari kelas-kelas yang lain.






2.7.3.4. *Sequence* Diagram

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. 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 itu.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada *Sequence Diagram*:

Tabel 2.4 Simbol *Sequence Diagram* (Shalahuddin dan Rosa, 2011).

Nama	Simbol	Keterangan
Objek dan kelas		Objects; mewakili peserta
<i>Lifeline</i>		<i>Lifeline</i> merupakan siklus hidup sebuah pesan/ <i>message</i> berdasarkan waktu
<i>Message</i>		Pesan/ <i>message</i> merupakan komunikasi antara objek yang satu dengan objek yang lainnya
<i>Return message</i>		<i>Return message</i> merupakan balasan/hasil yang berisi nilai dari sebuah objek yang meminta (mengirim pesan)
<i>Self message</i>		<i>Self message</i> merupakan pesan dari sebuah objek kepada objek itu sendiri untuk melakukan suatu aksi
<i>Return self message</i>		Balasan/hasil dari <i>self message</i> yang berisi suatu nilai kepada objek itu sendiri

2.8. ENTITY RELATIONSHIP DIAGRAM (ERD)

Entity Relationship Diagram (ERD) merupakan suatu kumpulan *file* yang saling berkaitan satu dengan yang lainnya pada model data *relation* hubungan antar *file* direlasikan dengan kunci relasi (*Relation Key*)

yang merupakan kunci utama di masing-masing *file*. Perancangan *database* yang tepat akan menyebabkan *MySQL*/paket program lainnya akan bekerja dengan optimal. (Fathansyah, 2010)

(a) Entitas set atau tipe – tipe objek

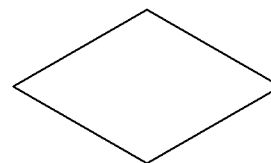
Tipe objek dapat digambarkan dengan simbol persegi panjang. Tipe objek menyertakan komponen objek (benda) yang anggotanya berperan dalam sistem yang dikembangkan dan dapat diidentifikasi secara utuh dalam satu atau banyak atribut.



Gambar 2.1. Simbol Entitas Set

(b) Hubungan *relationship set*

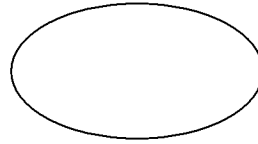
Digambarkan dengan simbol belah ketupat. Hubungan *relationship set* atau relasi menyatakan hubungan atau persatuan antara tipe – tipe objek yang dihubungkan dengan panah – panah.



Gambar 2.2. Simbol Hubungan Relationship Set

(c) Atribut

Atribut secara umum adalah sifat atau karakteristik dan entitas hubungan dengan maksud menjelaskan entitas atau relationship tersebut. Atribut digambarkan sebang simbol elips sebagai berikut :



Gambar 2.3. Simbol Atribut

Dalam ERD terdapat tingkatan hubungan antar entitas yang satu dengan yang lainnya, dilihat dari segi kejadian atau banyak tidaknya hubungan antara entitas tersebut. Hal inilah yang dinamakan sebagai *cardinality* atau *occurance*.

2.9. DATABASE

Merupakan sembarang pengumpulan data, sebuah *file* terdiri dari atas sejumlah record atau tabel. Masing-masing terbentuk dari *field* atau kolom dari tipe tertentu, dan kumpulan operasi yang memudahkan pencarian, penyimpanan, kombinasi ulang, dan aktivitas sejenisnya. (Fathansyah, 2010)

2.10. MySQL

MySQL atau sering dibaca dengan ejaan “*mai es que el*” ini merupakan sebuah *database* yang berbasis *server database*. Kemampuannya dalam menangani RDBMS (*Relational Database Management System*) mengakibatkan *database* ini menjadi database yang sangat populer saat ini.

Database MySQL bersifat *open source* dan mampu menangani data yang sangat besar hingga ukuran GigaByte, dengan kemampuan daya tampung data ini maka *MySQL* sangat cocok digunakan untuk mengcover

data pada perusahaan baik yang kecil sampai perusahaan besar (Madcoms. 2012).

2.11. PHP

PHP adalah salah satu bahasa pemrograman yang berjalan dalam sebuah *web server* dan berfungsi sebagai pengolah data pada sebuah *server*. Dengan menggunakan *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Data yang dikirim oleh pengunjung *website* / komputer client akan diolah dan disimpan dalam *database web server* dan dapat ditampilkan kembali apabila diakses. (Fathansyah, 2010)