

## **BAB II**

### **LANDASAN TEORI**

#### **II.1. Sistem**

##### **II.1.1. Konsep Dasar Sistem**

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu sekelompok yang menekankan pada prosedur dan kelompok yang menekankan pada elemen atau komponennya. Pendekatan yang menekankan pada prosedur dan mendefinisikan sistem sebagai suatu jaringan kerja prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sedangkan pendekatan sistem yang lebih menekankan pada elemen atau komponen mendefinisikan sistem sebagai kumpulan elemen yang berinteraksi untuk mencapai suatu tujuan tertentu, kedua kelompok definisi ini adalah benar dan tidak bertentangan. Yang berbeda adalah cara pendekatannya. (Tata Sutabri : 2012 ; 2).

##### **II.1.2. Pengertian Sistem**

Menurut (Kusrini : 2013 ; 11) Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input* ) sehingga menghasilkan keluaran(*output*).

Menurut (Jogianto : 2013 ; 2) Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini

menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata, seperti tempat, benda dan orang-orang yang betul-betul ada dan terjadi.

### **II.1.3. Data**

Menurut (Kusrini: 2013 : 3) Data merupakan representasi dari fakta atau gambaran mengenai suatu objek atau kejadian, ambil contoh fakta mengenai biodata mahasiswa yang meliputi nama, jenis kelamin, agama, yang dianut dan lain-lain. Contoh lain dari fakta mengenai kejadian / transaksi dalam sebuah perusahaan dagangan adalah seperti transaksi penjualan yang meliputi waktu transaksi, pelaku transaksinya (pelanggan, kasir), barang yang ditransaksikan, serta jumlah dan harganya. Data dinyatakan dengan nilai yang berbentuk angka, deretan karakter, atau simbol.

Menurut (Mulyanto; 2012 : 15) Data didefinisikan sebagai representasi dunia nyata mewakili suatu objek seperti manusia, hewan, peristiwa, konsep, keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Dengan kata lain, data merupakan kenyataan yang menggambarkan suatu kejadian dan kesatuan yang nyata.

### **II.1.4. Informasi**

Menurut (Kusrini : 20013 ; 4) Informasi merupakan hasil olahan data, dimana data tersebut sudah diproses dan diinterpretasikan menjadi sesuatu yang bermakna untuk pengambilan keputusan, informasi juga di diartikan sebagai

himpunan dari data yang relevan dengan satu atau beberapa orang dalam suatu waktu.

Menurut (Jogiyanto; 2012 : 11) Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.

### **II.1.5 Kualitas informasi**

Menurut (Kusrini : 2013 ; 4). Agar bisa menyediakan keluaran yang berguna untuk membantu manager atau para pengambil keputusan, sebuah sistem informasi harus mampu mengumpulkan data dan mentransfortasikan data tersebut kedalam informasi yang memiliki kualitas-kualitas tersebut. Menurut (Jogiyanto ; 2008 : 37) Kualitas informasi berarti informasi harus bebas dari kesalahan-kesalahan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. Informasi harus akurat karena dari sumber informasi ke penerimaan kemungkinan banyak terjadi.

1. Relevan. Berarti informasi tersebut mempunyai mamfaat untuk pemakainya. Relevansi untuk tiap-tiap orang satu dengan yang lainnya berbeda.
2. Akurat. Kecocokan antara informasi dengan kejadian-kejadian atau objek-objek yang diwakilkan. Misalnya, laporan inventaris yang tidak akurat menyebutkan bahwa terdapat 15 unit barang di dalam gudang.
3. Lengkap. Merupakan derajat sampai seberapa jauh informasi menyertakan kejadian-kejadian atau objek-objek yang berhubungan. Misalnya, penjualan selama satu hari yang seharusnya ada 150 transaksi dilaporkan hanya tercatat sebanyak 145 transaksi.

4. Tepat Waktu. Informasi yang tidak tepat waktu akan menjadi informasi yang tidak berguna atau tidak dapat digunakan untuk membantu pengambilan keputusan. Misalnya, informasi jadwal ujian seorang mahasiswa disampaikan setelah kegiatan ujian diselenggarakan. Informasi ini menjadi tidak berguna lagi.
5. Dapat dipahami. Hal tersebut terkait dengan bahasa dan cara penyajian informasi agar pengguna lebih mudah mengambil keputusan.
6. Dapat dibandingkan. Setelah informasi yang memungkinkan seorang pemakai untuk mengidentifikasi persamaan dan perbedaan antara dua objek atau kejadian yang mirip. Misalnya, membandingkan laporan pendapatan antara tahun 2012 dan 2013.

#### **II.1.6. Sitem Informasi**

Menurut (Kusrini : 2013 ;11) Suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organisasi, serta menyediakan laporan-laporan yang diperlukan untuk pihak luar.

Berdasarkan dukungan kepada pemakai, sistem informasi dibagi menjadi:

1. Sistem Pemrosesan Transaksi (*Transaksi Processing System*) atau TPS.
2. Sitem Informasi Manajamen (*Manajemen Information System*) atau MIS.

3. Sistem Otomasi Perkantoran (*Office Automation System/OAS*).
4. Sistem Pendukung Keputusan (*Decision Support System*) atau DSS.
5. Sistem Informasi Eksekutif (*Executive Information System*) atau EIS.
6. Sistem Pendukung Kelompok (*Group Support System*) atau ISS.
7. Sistem Pendukung Cerdas (*Intelligent Support System*) atau GSS.

Menurut (Jogiyanto; 2015 : 18) Sistem informasi merupakan suatu perkumpulan data yang terorganisasi beserta tatacara penggunaanya yang mencangkup lebih jauh dari pada sekedar penyajian. Istilah tersebut menyiratkan suatu maksud yang ingin dicapai dengan jalan memilih dan mengatur data serta menyusun tatacara penggunaanya.

## **II.2. Sistem Pendukung Keputusan**

Menurut (Kusrini ; 2013 ; 15) Sistem Pendukung Keputusan merupakan Sistem Informasi interaktif menyediakan informasi pemodelan dan manipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Sistem Pendukung Keputusan biasanya dibangun untuk mendukung solusi atas masalah atau untuk mengevaluasi suatu peluang. *Decision Support System* yang seperti itu disebut dengan aplikasi (*Decision Support System*). Aplikasi *Decision Support System* digunakan dalam pengambilan keputusan. Aplikasi *Decision Support System* menggunakan CBIS (Computer Based Information

System) yang fleksibel, interaktif, dan dapat diadaptasi yang dikembangkan untuk mendukung solusi atau masalah manajemen spesifik yang tidak terstruktur.

Aplikasi *Decision Support System* menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat digabungkan pengguna yang mudah, dan dapat digabungkan pemikiran pengambilan keputusan.

*Decision Support System* lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analisis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas.

Menurut (Turban : 2013 ; 16) Sistem Pendukung Keputusan (SPK) atau Decision Support System (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat.

### **II.2.1. Karakteristik Sistem Pendukung Keputusan**

Menurut (Kurini : 2013 ; 25) Aplikasi sistem pendukung keputusan bisa berdiri dari beberapa subsistem, yaitu:

1. Subsistem manajemen data

Subsistem manajemen data masukan satu database yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen dan bisa diinterkoneksi dengan data *warehouse*

peusahaan, suatu repository untuk data perusahaan yang relevan dengan pengambilan keputusan.

## 2. Subsistem manajemen modal

Merupakan paket perangkat lunak memasukan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat.

## 3. Subsistem antarmuka pengguna

Pengguna berkomunikasi dengan memeritahakan, sistem pendukung keputusan melalui tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari sistem pendukung keputusan.

Menurut (Turban : 2013 ; 17) Sistem pendukung keputusan pada dasarnya merupakan pengembangan lebih lanjut dari sistem informasi manajemen terkomputerisasi (*computerized management information systems*), yang dirancang sedemikian rupa sehingga bersifat interaktif dengan pemakainya. Sifat interaktif ini diperlukan untuk memudahkan integrasi antara berbagai komponen dalam proses pengambilan keputusan, seperti prosedur, kebijakan, teknik analisis, serta pengalaman dan wawasan manajerial guna membentuk suatu kerangka keputusan yang bersifat fleksibel.

1. Dapat digunakan berulang kali dan bersifat konstan.
2. Terdapat dua komponen utama, yaitu data dan model.
3. Menggunakan baik data eksternal maupun internal.
4. Memiliki kemampuan what-if analysis dan goal seeking analysis

## II.2.2. Metode-Metode Sistem Pendukung Keputusan

### 1. *Analitycal Hierarchy Process* (AHP)

Menurut (Sutanti : 2013 ; 27) *Analitycal Hierarchy Process* (AHP) Adalah metode untuk memecahkan suatu situasi yang kompleks tidak terstruktur kedalam beberapa komponen dalam susunan yang hirarki, dengan memberi nilai subjektif tentang pentingnya setiap variabel secara relatif, dan menetapkan variabel mana yang memiliki prioritas paling tinggi guna mempengaruhi hasil pada situasi tersebut.

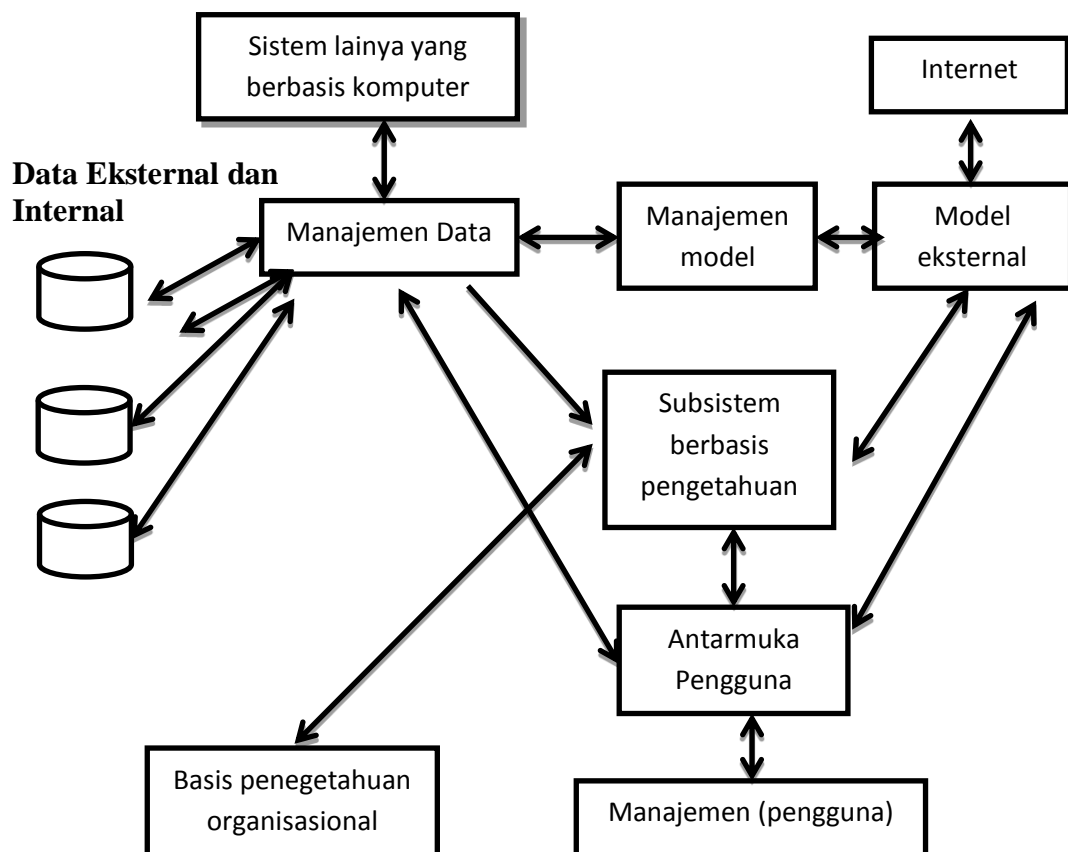
Defenisi metode AHP adalah Proses pengambilan keputusan pada dasarnya adalah memilih suatu alternatif yang terbaik. Seperti melakukan penstrukturan persoalan, penentuan alternatif-alternatif, penetapan nilai kemungkinan untuk variabel aleatori, penatap nilai, persyaratan preferensi terhadap waktu, dan spesifikasi atas resiko. Betapapun melebarnya alternatif yang dapat ditetapkan maupun terperinci penjajagan nilai kemungkinan, keterbatasan yang tetap melingkupi adalah dasar perbandingan berbentuk suatu kriteria yang tunggal.

### 2. *Simple Additive Weighting* (SAW)

Menurut (Hendra Dinata : 2014; 27) Metode Simple Additive Weighting metode yang sering dikenal dengan metode penjumlahan terbobot. Maksud dari penjumlahan terbobot yaitu mencari penjumlahan terbobot dari rating di tiap alternatif pada seluruh atribut/ kriteria. Hasil/ Skor total yang diperoleh untuk sebuah alternatif yaitu dengan menjumlahkan semua hasil perkalian antara rating / yang dibandingkan pada lintas atribut dan bobot setiap atribut. Rating pada

setiap atribut sebelumnya harus sudah melalui proses normalisasi.

Defenisi Metode SAW memerlukan proses normalisasi matriks keputusan  $x$  ke skala yang bisa dibandingkan dengan rating alternatif yang ada. Dalam sebuah penilaian keputusan dari beberapa variable yang mempunyai bobot tidak sama kita harus menentukan terlebih dahulu koefisien dari setiap variable sehingga kita dapat menentukan nilai dari sebuah keputusan tersebut.



**Gambar II.1. Arsitektur DDS**

*(Sumber: Kusrini : 2007 ; 26)*

### II.3. *Metode Profile Matching*

Profile matching adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh calon, bukannya tingkat minimal yang harus dipenuhi dan dilewati. Dalam pencocokan profile, dilakukan identifikasi terhadap kelompok tersebut diukur menggunakan beberapa kriteria penilaian. Jika kalau pelaksana yang baik memperoleh skor yang berbeda dari pelaksana yang buruk atau sebuah karakteristik, maka variabel yang membedakan antara pelaksana-pelaksana yang baik dan buruk telah teridentifikasi, profile ideal dari karyawan yang berhasil bisa dibuat. (Kusrini : 2012 ; 53).

#### II.3.1 *Langkah-Langkah Metode Profile Matching*

Menurut (Nina Sherly : 2013 ; 44-55) adapun langkah-langkah dalam penyelesaian metode *Profile Matching* adalah sebagai berikut:

##### 1. Perhitungan *GAB* Kompetensi

Setelah proses pemilihan kandidat, pada proses ini ialah perhitungan pemetaan gab kompetensi dimana yang dimaksud dengan gab adalah beda antara profil jabatan dengan profile karyawan atau dapat ditunjukkan pada rumus dibawah ini:

$$\mathbf{Gab = Profil Karyawan - Profile Jabatan}$$

##### 2. Perhitungan pemetaan *GAB*

Kompetensi berdasarkan aspek-aspek untuk pengumpulan gab-gab yang terjadi itu sendiri pada setiap aspeknya mempunyai perhitungan yang

berbeda-beda. Setelah didapatkan gab masing-masing karyawan maka tiap profil karyawan diberi bobot nilai dengan patokan tabel nilai *gab* seperti yang dapat dilihat pada Tabel II.2. dibawah ini.

**Tabel II.1. patokan tabel nilai *gab***

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih (kompetensi sesuai dengan yang dibutuhkan)
2	1	4,5	Kompetensi individu kelebihan 1 tingkat/level.
3	-1	4	Kompetensi individu kekurangan 1 tingkat/level.
4	2	3,5	Kompetensi individu kelebihan 2 tingkat/level.
5	-2	3	Kompetensi individu kekurangan 2 tingkat/level.
6	3	2,5	Kompetensi individu kelebihan 3 tingkat/level.
7	-3	2	Kompetensi individu kekurangan 3 tingkat/level.
8	4	1,5	Kompetensi individu kelebihan 4 tingkat/level.
9	-4	1	Kompetensi individu kekurangan 4 tingkat/level.

### 3. Perhitungan dan Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai *gab* untuk setiap kriteria, kemudian tiap aspek kriteria di kelompokkan menjadi 2 (dua) kelompok yaitu kelompok *Core Factor* dan *Secondary Factor*. Untuk perhitungan *Core Factor* dapat dilihat pada rumus dibawah ini:

$$NCF = \frac{\sum NC(k, s, a)}{\sum IC}$$

Keterangan:

NCF : Nilai rata-rata *core factor*

NC(k, s, a) : Jumlah total nilai *core factor*

IC : Jumlah item *core factor*

Sedangkan untuk perhitungan *secondary factor* dapat di lihat pada rumus di bawah ini:

$$NSF = \frac{\sum NS(k,s,a)}{\sum IS}$$

Keterangan:

NSF : Nilai rata-rata *secondary factor*

NS(k, s, a) : Jumlah total nilai *secondary factor*

ISF : Jumlah item *secondary factor*

#### 4. Perhitungan Nilai Total

Dari hasil perhitungan dari tiap aspek diatas kemudian dihitung nilai total berdasarkan presentasi dari *core* dan *secondary factor* yang diperkirakan berpengaruh terhadap kinerja tiap-tiap profil.

Contoh perhitungan dapat dilihat pada rumus dibawah ini:

$$N(k, s, a) = (x)\%NCF(k, s, a)(x)\%NSF(k, s, a)$$

Keterangan:

(k,s,a) : (Kemampuan, Sikap Kerja, Administrasi)

N(k,s,a) : Nilai rata-rata *core factor*

NCF (k,s,a) : Nilai rata-rata *secodary factor*

(x)% : Nilai persen yang diinputkan

#### 5. Perhitungan Penentuan Hasil Akhir Atau Perangkingan

Hasil akhir dari proses ini adalah rangking dari kandidat yang diajukan. Adapun perhitungan tersebut dapat dilihat pada rumus dibawah ini:

$$Ha = (x)\%Nk + (x)\%Ns + (x)\%Na$$

Keterangan:

Ha : Hasil akhir

Nk : Nilai Kemampuan

Ns : Nilai Sikap

Na : Nilai Administrasi

(x)% : Nilai persen yang diinputkan

## II.4. Pengertian Database

Sebuah basis data adalah sebuah kumpulan data yang saling berhubungan secara logis, dan merupakan sebuah penjelasan dari data tersebut, yang didesain untuk menemukan data yang dibutuhkan oleh sebuah perusahaan. Di dalam basis data tersebut. Karena alasan tersebut basis data dapat juga dideskripsikan sebagai

kumpulan data yang saling terintegrasi. Basis data juga merupakan sekumpulan elemen data terintegrasi yang secara logika saling berhubungan. Basis data mengonsolidasikan berbagai catatan yang terlebih dahulu disimpan dalam file-file terpisah kedalam satu gabungan umum elemen data yang menyediakan data untuk banyak aplikasi. Elemen data mendeskripsikan entitas-entitas dan hubungan antara entitas-entitas tersebut (Indrajani : 2015 ; 70).

Database adalah kumpulan data yang saling terkait yang diorganisasi untuk memenuhi kebutuhan dan struktur sebuah organisasi serta bisa digunakan oleh lebih dari satu orang dan lebih dari satu aplikasi. (kusrini : 2013 ; 33).

## **II.5. Normalisasi**

Normalisasi adalah satu teknik dengan pendekatan *bottom-up* yang digunakan untuk membantu mengidentifikasi hubungan. Dimulai dari menguji hubungan, yaitu *functional dependencies* antara atribut. Pengertian lainnya adalah suatu teknik yang menghasilkan sekumpulan hubungan dengan sifat-sifat yang diinginkan dan memenuhi kkebutuhan pada perusahaan (Indrajani : 2015 ; 2).

Adapun bentuk-bentuk normalisasi menurut (Indrajani : 2015 ; 9-10) adalah sebagai berikut:

### 1. Unnormalized Form (UNF)

Merupakan suatu tabel yang berisikan satu atau lebih grup yang berulang. Membuat tabel yang *unnormalized*, yaitu dengan memindahkan data dari sumber informasi.

Contoh nota penjualan yang disimpan ke dalam format tabel dengan baris dan kolom.

### 2. First Normal Form (1NF)

Merupakan sebuah relasi di mana setiap baris dan kolom berisikan satu dan hanya satu nilai.

Proses UNF ke 1NF

- a. Tentukan satu atau sekumpulan atribut sebagai kunci untuk tabel *unnormalized*.
- b. Identifikasikan grup yang berulang dalam tabel *unnormalized* yang berulang untuk kunci atribut.

### 3. Second Normal Form (2NF)

Berdasarkan pada konsep *full functional*, yaitu A dan B merupakan atribut sebuah relasi. B dikatakan *fully dependent* terhadap A jika B *functional dependent* pada A tetapi tidak pada propersubset dari A. 2NF merupakan sebuah relasi dalam 1NF dan setiap atribut *non-primary-key* bersifat *fully functionally dependent* pada *primary key*.

1NF ke 2NF

- a. Identifikasikan *primary key* untuk relasi 1NF
- b. Identifikasikan *functional dependencies* dalam relasi.

c. Jika terdapat *partial dependencies* terhadap *primary key*, maka hapus dengan menempatkan dalam relasi yang baru bersama salinan determinannya.

#### 4. Third Normal Form (3NF)

Berdasarkan pada konsep *transitive dependency*, yaitu suatu kondisi di mana A, B dan C merupakan atribut sebuah relasi, maka A- B dan B-C, maka *transitively dependent* pada A melalui B. 3NF adalah sebuah relasi dalam 1NF dan 2NF, di mana tidak terdapat atribut *non primary key* yang bersifat *transitively dependent* pada *primary key* 2NF ke 3NF

a. Identifikasi *primary key* dalam relasi 2NF.

b. Identifikasi *functional dependences* dalam relasi.

c. Jika terdapat *transitive dependencies* terhadap *primary key* hapus dengan menempatkan dalam relasi yang bersama dengan salinan determinannya.

#### 5. Boyce-code Normal Form (BCNF)

Berdasarkan pada *functional dependencies* yang dimasukkan dalam hitungan seluruh *candidate key* dalam suatu relasi. Bagaimanapun BCNF juga memiliki batasan-batasan tambahan disamakan dengan defenisi umum dari 3NF. Suatu relasi dikatakan BCNF, jika dan hanya jika setiap determinan merupakan *candidate key*. Perbedaan antara 3NF dan BCNF yaitu untuk *functional dependent* A – B, 3NF memungkinkan *dependency* ini dalam suatu relasi jika adalah atribut *primary key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menetapkan dengan jelas bahwa untuk *dependency* ini agar ditetapkan dalam relasi A, maka A harus merupakan *candidate key*. Setiap

relasi dalam BCNF juga merupakan 3NF. Dalam BCNF kesalahan jarang sekali terjadi, Kesalahan dapat terjadi pada relasi yang:

- a. Terdiri atas 2 atau lebih *composite candidate key*.
- b. *Candidate key overlap*, sedikitnya satu atribut.

## II.6. UML (*Unified Modeling Language*)

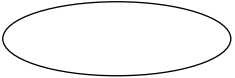
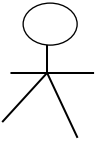


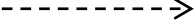

Menurut Windu Gata (2013 : 4) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.

UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

### 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.2. Simbol *Use Case*




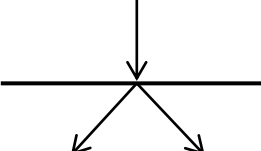
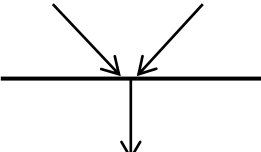
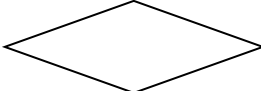

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam <i>use case</i> lain ( <i>required</i> ) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata, 2013 : 4)

## 2. Diagram Aktivitas (*Activity Diagram*)

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :

Tabel II.3. Simbol *Activity Diagram*

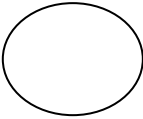
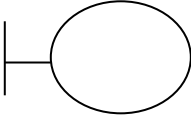
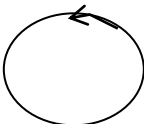

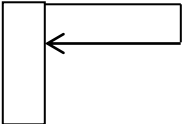


Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

(Sumber : Windu Gata, 2013 : 6)

### 3. Diagram Urutan (*Sequence Diagram*)

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.4. Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata, 2013 : 7)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

*Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

**Tabel II.5. Multiplicity Class Diagram**

<b>Multiplicity</b>	<b>Penjelasan</b>
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Windu Gata, 2013 : 9)

## II.8. Pengertian Visual Basic

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic. Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows. Fitur baru bahasa VB.NET dibandingkan

Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET.

Ibaratnya seperti ikan dan air yang tidak dipisahkan, ini sama halnya pada VB.NET dan .NET Framework. Bahasa VB.NET memerlukan .NET Framework agar dapat dikompilasi dan dijalankan. .NET Framework merupakan framework yang membungkus kompleksitas OS Windows sehingga konsisten API dapat diperoleh dan tidak dipusingkan dengan beragam API tiap OS Windows. Buku ini tidak akan membahas .NET Framework. Pembaca dapat mempelajari buku yang khusus belajar mengenai .NET Framework. Pembaca juga dapat mengunjungi website resminya yaitu <http://www.microsoft.com/net> (Agus Kurniawan ; 2013)

## **II.9. Pengertian SQL Server 2008**

SQL Server 2008 adalah sebuah terobosan baru dari Microsoft dalam bidang *database*. SQL Server adalah DBMS (*Database Management System*) yang dibuat oleh Microsoft untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan Oracle. SQL Server 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL Server 2008 membawa beberapa terobosan dalam bidang pengolahan dan penyimpanan data. Microsoft merilis SQL Server 2008 dalam beberapa versi yang disesuaikan dengan segment-segment pasar yang dituju. Versi-versi tersebut adalah sebagai berikut (Wenny Widya ; 2012 : 3)