BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kennet Boulding, terutama menekan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas memimpin suatu organisasi adalah terlalu memusatkan perhatian pada salah satu komponen saja dari sistem organisasi.

Teori sistem melahirkan konsep-konsep futuristik, antara lain yang terkenal adalah konsep sibernetika (*cybernetics*). Konsep atau dibidang kajian ilmiah ini berkaitan dengan upaya menerapkan berbagai ilmu yaitu ilmu perilaku, fisika, biologi, dan teknik. Oleh karena itu sibernetika biasanya berkaitan dengan usaha-usaha otomasi tugas-tugas yang dilakukan manusia, sehingga melahirkan studi-studi tentang robotika, kecerdasan buatan (*artificial intelegence*). Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*), dan keluaran (*output*).

selain itu, suatu sistem tidak bisa lepas dari lingkungan maka umpan balik (*feed back*) dapat berasal dari lingkungan sistem yang dimaksud. Organisasi dipandang sebagai suatu sistem yang tentunya akan memiliki semua unsur ini (Tata Sutabri, 2012)

II.1.1. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen Sistem (*Component*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem *(Boundary).*

   Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem *(Environment).*

   Bentuk apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut operasi lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan yang menguntungkan merupakan bagi sistem tersebut.

4. Penghubung Sistem *(Interface)*

Media yang menghubungkan sistem dengan subsistem lainnya disebut penghubung sistem atau *interface*. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut.

5. Masukan Sistem *(Input)*

Energi yang dimasukan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh di dalam suatu sistem unit komputer. "program" adalah *maintenance* input yang digunakan untuk mengoperasikan komputernya dan "data" adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Yaitu hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambil keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain

7. Pengolah Sistem (*Proses*).

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministik*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem

tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan (Tata Sutabri, 2012).

II.1.2. Informasi

Informasi adalah data yang telah diklasifikasikan diolah atau diinterpretasi untuk digunakan dalam proses pengambil keputusan. Sistem pengolahan informasi megolah data menjadi informasi atau tepatnya mengolah dari bentuk tak berguna menjadi berguna bagi penerimanya.

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam pengambil keputusan (Tata Sutabri, 2012).

II.1.3. Sistem Informasi

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kennet Boulding, terutama menekan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas memimpin suatu organisasi adalah terlalu memusatkan perhatian pada salah satu komponen saja dari sistem organisasi.

Sesungguhnya, yang dimaksud sistem informasi tidak harus melibatkan komputer. Sistem informasi yang menggunakan komputer biasa disebut sistem informasi berbasis komputer (*Computer Base Information System* atau CBIS). Dalam praktik, istilah sistem informasi lebih sering dipakai tanpa embel-embel

berbasis komputer walaupun dalam kenyataannya komputer merupakan bagian yang penting.

Menurut Alter (1992), sistem informasi merupakan kombinasi antar prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Menurut Hall (2001), sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai. (Abdul Kadir, 2014 : 8-9).

II.2. Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah sistem berbasis model yang terdiri dari prosedur-prosedur dalam pemrosesan data dan pertimbangan untuk membantu manajer dalam mengambil keputusan. Agar berhasil mencapai tujuannya, maka sistem tersebut harus sederhana, mudah dikontrol, mudah beradaptasi, lengkap pada hal-hal penting dan mudah berkomunikasi dengan sistem tersebut.

II.2.1. Karakteristik Sistem Pendukung Keputusan

Sistem pendukung keputusan mempunyai karekteristik sebagai berikut :

1. Mendukung pengambilan keputusan secara cepat dan tepat.
2. Menggunakan model matematis yang sesuai.

   Model tersebut merupakan salah satu cara dalam ilmu manajemen yang digunakan untuk memecahkan masalah dengan memakai notasi dan

persamaan matematika yang kemudian direpresentasikan menjadi sebuah sistem.

3. Adanya *interface* manusia dan mesin dimana manusia yang mengontrol.
4. Mempunyai kemampuan dialog.

II.2.2 Komponen SPK

Dari sudut pandang sebagai suatu sistem yang terpadu, sistem pendukung keputusan memiliki beberapa komponen pendukung yaitu sebagai berikut :

1. Manajemen Data

Manajemen data memasukkan satu *database* yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut *DBMS* (*Database Management Sistem*). Manajemen data dapat diinterkoneksikan dengan data *warehouse* perusahaan, suatu repisitori untuk data perusahaan yang relevan untuk mengambil keputusan.

2. Manajemen Model

Manajemen model merupakan paket perangkat lunak yang memasukkan berbagai macam model, diantaranya adalah model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kemampuan analitik dan manajemen perangkat lunak yang tepat.

3. Antar Muka

Antar muka penguna memungkinkan pengguna berkomunikasi dan memerintahkan SPK. *Web Browser* memberikan struktur antarmuka pengguna grafis yang familier dan konsisten.

II.2.3. Fase-fase Pengambilan Keputusan

Terdapat 4 fase dalam pembangunan *Decision Support System*, yaitu :

1. *Intelligence*

Pada *intelligence phase*, masalah diidentifikasi, ditentukan tujuan dan sasarannya, penyebabnya, dan besarnya. Masalah dijabarkan secara lebih rinci dan dikategorikan apakah termasuk *programmed* atau *non-programmed*.

2. *Design*

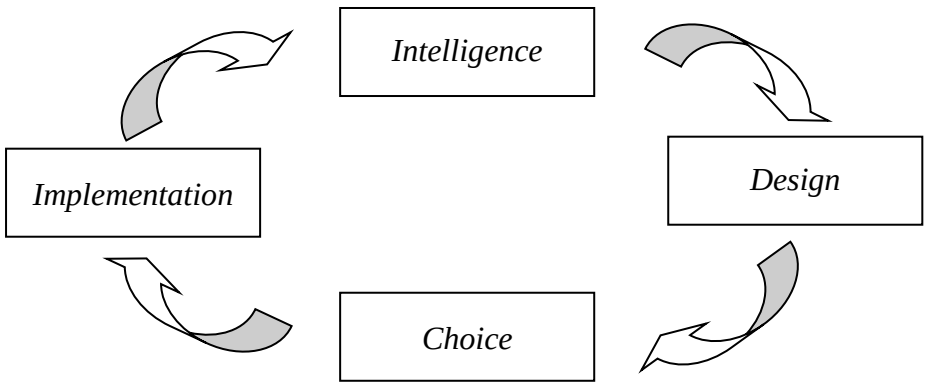
Pada *design phase*, dikembangkan tindakan alternatif, menganalisis solusi yang potensial, membuat model, membuat uji kelayakan, dan memvalidasi hasilnya.

3. *Choice*

Pada *choice phase,* menjelaskan pendeketan solusi yang dapat diterima dan memilih alternatif keputusan yang terbaik.

4. *Implementation.*

Pada *implementation phase,* solusi pada *choice phase* diimplementasikan.



Gambar II.1. Langkah langkah Pengambilan Keputusan

II.2.4. Manfaat Sistem Pendukung Keputusan

SPK dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah :

1. SPK memperluas kemampuan pengambil keputusan dalam memproses data / informasi bagi pemakainya.
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama barbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu SPK mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun dia dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya,karena mampu menyajikan berbagai alternatif pemecahan.

Dalam pengambilan keputusan, keputusan dibedakan menjadi 3 jenis, yaitu terstruktur, semi terstruktur, dan tidak terstruktur. Keputusan terstruktur diambil apabila permasalahan yang terjadi rutin dan selalu berulang. Keputusan semi terstruktur diambil apabila didalamnya terdapat beberapa keputusan terstruktur. Sedangkan keputusan tidak terstruktur diambil apabila tidak ada standar atau *rule* yang bisa digunakan (Turban *et al*, 2011).

II.3. Metode Analytical Hierarchy Process (AHP)

Metode Analytical Hierrchy Process (AHP) dekembangkan oleh Prof. Thomas Lorie Saaty dari Wharton Business School di awal tahun 1970, yang

digunakan untuk mencari rangking atau urutan prioritas dari berbagai alternatif dalam pemecahan suatu permasalahan.

Pada dasarnya AHP adalah suatu teori umum tentang pengukuran yang digunakan untuk menemukan skala rasio, baik dari perbandingan berpasangan yang diskrit maupun kontinu. Perbandingan-perbandingan ini dapat diambil dari ukuran aktual atau skala dasar yang mencerminkan kekuatan perasaan dan preferensi relatif. Metode ini adalah sebuah kerangka untuk mengambil keputusan dengan efektif atas persoalan dengan menyederhanakan dan mempercepat proses pengambilan keputusan dengan memecahkan persoalan tersebut kedalam bagianbagiannya, menata bagian atau variabel ini dalam suatu susunan hirarki, member nilai numerik pada pertimbangan subjektif tentang pentingnya tiap variabel dan mensintesis berbagai pertimbangan ini untuk menetapkan variabel yang mana yang memiliki prioritas paling tinggi dan bertindak untuk mempengaruhi hasil pada situasi tersebut.

Analytic Hierarchy Process (AHP) dapat menyederhanakan masalah yang kompleks dan tidak terstruktur, strategik dan dinamik menjadi bagiannya, serta menjadikan variabel dalam suatu hirarki (tingkatan). Masalah yang kompleks dapat diartikan bahwa kriteria dari suatu masalah yang begitu banyak (multikriteria), struktur masalah yang belum jelas, ketidakpastian pendapat dari pengambil keputusan, pengambil keputusan lebih dari satu orang, serta ketidakakuratan data yang tersedia.

Analytic Hierarchy Process (AHP) mempunyai landasan aksiomatik yang terdiri dari :

1. *Resiprocal Comparison*, yang mengandung arti bahwa matriks perbandingan berpasangan yang terbentuk harus bersifat berkebalikan.Misalnya, jika A adalah k kali lebih penting dari pada B maka B adalah 1/k kali lebih penting dari A.
2. *Homogenity*, yaitu mengandung arti kesamaan dalam melakukan perbandingan. Misalnya, tidak dimungkinkan membandingkan jeruk dengan bola tenis dalam hal rasa, akan tetapi lebih relevan jika membandingkan dalam hal berat.
3. *Dependence*, yang berarti setiap level mempunyai kaitan (complete hierarchy) walaupun mungkin saja terjadi hubungan yang tidak sempurna (incomplete hierarchy).
4. *Expectation*, yang berarti menonjolkon penilaian yang bersifat ekspektasi dan preferensi dari pengambilan keputusan. Penilaian dapat merupakan data kuantitatif maupun yang bersifat kualitatif.

Metode AHP ini membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria, pihak yang berkepentingan, hasil dan dengan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas. Metode ini juga menggabungkan kekuatan dari perasaan dan logika yang bersangkutan pada berbagai persoalan, lalu mensintesis berbagai pertimbangan yang beragam menjadi hasil yang cocok dengan perkiraan kita secara intuitif sebagaimana yang dipresentasikan pada pertimbangan yang telah dibuat. (Prasetyo, 2010 : 304).

II.3.1. Prinsip-Prinsip Dasar Analytic Hierarchy Process (AHP)

Dalam menyelesaikan persoalan dengan metode *Analytic Hierarchy Process* (AHP) ada beberapa prinsip dasar yang harus dipahami antara lain :

1. *Decomposition*

   Pengertian *decomposition* adalah memecahkan atau membagi problema yang utuh menjadi unsur unsurnya ke bentuk hirarki proses pengambilan keputusan, dimana setiap unsur atau elemen saling berhubungan. Untuk mendapatkan hasil yang akurat, pemecahan dilakukan terhadap unsur unsur sampai tidak mungkin dilakukan pemecahan lebih lanjut, sehingga didapatkan beberapa tingkatan dari persoalan yang hendak dipecahkan. Struktur hirarki keputusan tersebut dapat dikategorikan sebagai *complete* dan *incomplete*.

2. *Comparative Judgement*

   Comparative Judgement dilakukan dengan penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkatan di atasnya. Penilaian ini merupakan inti dari AHP karena akan berpengaruh terhadap urutan prioritas dari elemen elemennya. Hasil dari penilaian ini lebih mudah disajikan dalam bentuk *matrix pairwise comparisons* yaitu matriks perbandingan berpasangan memuat tingkat preferensi beberapa alternatif untuk tiap kriteria. Skala preferensi yang digunakan yaitu skala 1 yang menunjukkan tingkat yang paling rendah (*equal importance*) sampai dengan skala 9 yang menunjukkan tingkatan yang paling tinggi (*extreme importance*).

3. *Synthesis of Priority*

Synthesis of Priority dilakukan dengan menggunakan *eigen vector method* untuk mendapatkan bobot relatif bagi unsur unsur pengambilan keputusan.

4. *Logical Consistency*

Logical Consistency merupakan karakteristik penting AHP. Hal ini dicapai dengan mengagresikan seluruh eigen vektor yang diperoleh dari berbagai tingkatan hirarki dan selanjutnya diperoleh suatu vektor *composite* tertimbang yang menghasilkan urutan pengambilan keputusan. (Prasetyo, 2010 : 321).

II.3.2. Konsep Dasar AHP

Konsep dasar AHP adalah penggunaan matriks *pairwise comparison* (atriks perbandingan berpasangan) untuk menghasilkan bobot relative antar kriteria maupun alternative. Suatu kriteria akan dibandingkan dengan kriteria lainnya dalam hal seberapa penting terhadap pencapaian tujuan di atasnya (Saaty, 1986).

Tabel II. 1. Skala Dasar Perbandingan Berpasangan

Tingkat Kepentingan	Definisi	Keterangan
1	Sama Pentingnya	Kedua elemen mempunyai pengaruh yang sama
3	Sedikit lebih penting	Pengalaman dan penilaian sangat memihak satu elemen dibandingkan dengan pasangannya
5	Lebih Penting	Satu elemen sangat disukai dan secara praktis dominasinya sangat nyata, dibandingkan dengan elemen pasangannya.

7	Sangat Penting	Satu elemen terbukti sangat disukai dan secara praktis dominasinya sangat nyata, dibandingkan dengan elemen pasangannya.
9	Mutlak lebih penting	Satu elemen terbukti mutlak lebih disukai dibandingkan dengan pasangannya, pada keyakinan tertinggi.
2,4,6,8	Nilai Tengah	Diberikan bila terdapat keraguan penilaian di antara dua tingkat kepentingan yang berdekatan.

(Sumber : Saaty, 1986)

Penilaian dalam membandingkan antara satu kriteria dengan kriteria yang lain adalah bebas satu sama lain, dan hal ini dapat mengarah pada ketidak konsistensian. Saaty (1990) telah membuktikan bahwa *indeks* konsistensi dari *matrik* ber *ordo n* dapat diperoleh dengan rumus :

$$CI = (\lambda maks-n)/(n-1) \quad \text{..... (1)}$$

Dimana :

CI = Indeks Konsistensi (Consistency Index)

λmaks = Nilai *eigen* terbesar dari matrik berordo n

Nilai *eigen* terbesar didapat dengan menjumlahkan hasil perkalian jumlah kolom dengan *eigen* vector. Batas ketidak konsistensian di ukur dengan menggunakan rasio konsistensi (CR), yakni perbandingan indeks konsistensi (CI) dengan nilai pembangkit random (RI). Nilai ini bergantung pada ordo matrik n.

Rasio konsistensi dapat dirumuskan :

$$CR = CI/RI \quad \text{..... (2)}$$

Bila nilai CR lebih kecil dari 10%, ketidak konsistensian pendapat masih dianggap dapat diterima.

Tabel II.2. Daftar Indeks random konsistensi (RI)

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RI	0,00	0,00	0,58	0,90	1,12	1,24	1,32	1,41	1,45	1,49	1,51	1,48	1,56	1,57	1,59

(Sumber : Saaty, 1986)

II.4. Metode *Simple Additive Weighting* (SAW)

Metode ini merupakan metode yang paling dikenal dan paling banyak digunakan orang dalam menghadapi situasi MCDM (M*ultiple Criteria Decision Making*). Metode ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk sebuah alternatif diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut)dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi yang artinya telah melewati proses normalisasi sebelumnya.

Langkah langkah penyelesaian menggunakan metode SAW adalah sebagai berikut :

1. Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (Ci), kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.

4. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi. (Kusumadewi, 2006).

Formula untuk melakukan normalisasi tersebut adalah sebagai berikut :

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{Max\ x_{ij}} \\ \dfrac{Min\ x_{ij}}{x_{ij}} \end{cases} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\,..(2.1)$$

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj, dimana i=1,2,...,m dan j=1,2,...,n.

Nilai preferensi untuk setiap alternatif (Vi) diberikan sebagai berikut :

$$V_i = \sum_{j=1}^{n} W_j\, r_j \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\,..(2.2)$$

Dimana :

Vi = Nilai akhir dari alternatif

wj = Bobot yang telah ditentukan

rij = Normalisasi matriks

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif V_i lebih terpilih. Sedangkan untuk kriterianya terbagi dalam dua kategori yaitu untuk bernilai positif dan yang bernilai negatif.

Secara singkat algoritma metode SAW adalah:

1. Memberikan nilai setiap alternatif (*Ai*) pada setiap kriteria (*Cj*) yang sudah ditentukan, di mana nilai tersebut di peroleh berdasarkan nilai crisp; *i*=1,2,…m dan *j*=1,2,…n.
2. Memberikan nilai bobot (*W*) yang juga didapatkan berdasarkan nilai crisp.
3. Melakukan normalisasi matriks dengan cara menghitung nilai rating kinerja ternormalisasi (*rij*) dari alternatif *Ai* pada atribut *Cj* berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan/*benefit* = MAKSIMUM atau atribut biaya/*cost* = MINIMUM). Apabila berupa artibut keuntungan maka nilai crisp (*xij*) dari setiap kolom atribut dibagi dengan nilai crisp MAX (MAX *xij*) dari tiap kolom, sedangkan untuk atribut biaya, nilai crisp MIN (MIN *xij*) dari tiap kolom atribut dibagi dengan nilai crisp (*xij*) setiap kolom.
4. Melakukan proses perankingan dengan cara mengalikan matriks ternormalisasi (*R*) dengan nilai bobot (*W*).
5. Menentukan nilai preferensi untuk setiap alternatif (*Vi*) dengan cara menjumlahkan hasil kali antara matriks ternormalisasi (*R*) dengan nilai bobot (*W*). Nilai *Vi* yang lebih besar mengindikasikan bahwa alternatif *Ai* lebih terpilih.

II.5. *Unified Modeling Language* (UML)

UML singkatan dari *Unified Modelling Langguage* yang berarti bahasa pemodelan standart. (Chonoles, 2003) mengatakan sebagai bahasa, berarti *UML* memiliki sintaks dan *semantic*. Ketika kita membuat model menggunakan konsep *UML* ada aturan –aturan yang harus diikuti. Bagaimana elemen pada model-

model yang kita buat harus berhubungan satu dengan lainnya harus mengikuti standart yang ada. *UML* bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang ada kita buat? Dan sebagainya dapat dijawab dengan *UML*. *UML* diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan,lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorentasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensuport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita

sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudjo Widodo Dan Herlawati, 2011 : 10).

II.5.1. Diagram-*Diagram UML*

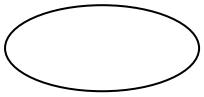
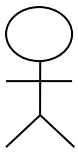
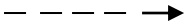
UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Alat bantu yang digunakan dalam perancangan, berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case Diagram*

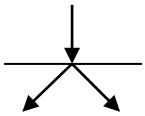
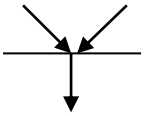
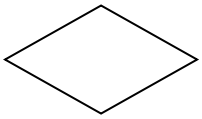
   Use Case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada didalam sistem informasi dan siapa saja yang berhak menggunakan fungsi - fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.3. *Simbol Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama *use case*.
	Aktor adalah abstraction dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan actor, harus ditentukan pembagian tenaga dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan didalam use case lain (*required*) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	Extend, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah system ataau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram*, yaitu :**Tabel II.4. Simbol *Activity Diagram***

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

3. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. Hubungan antar kelas mempunyaai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

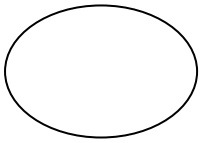
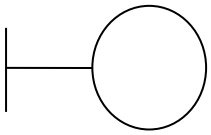
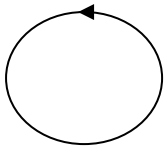
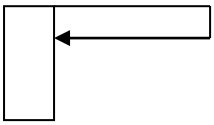
Tabel II.5. *Multiplicity Class Diagram*

Multiplicity	**Penjelasan**
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

4. 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.6. Simbol *Sequence Diagram*

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

II.6. Bahasa Pemograman *Visual Basic*

Pada awalnya bahasa pemrograman Visual Basic merupakan bahasa yang Object-Basetd (komponen-komponen). Namun setelah kehadiran bahasa pemrograman Visual Basic 7 sudah mulai dikenalkan metode pemrograman

berorientasi obyek atau sering disebut OOP (*Object Oriented Programming*), tetapi masih belum sepenuhnya metode tersebut digunakan. Baru setelah Visual Basic nNET merupakan bahasa yang benar-benar berorientasi obyek dengan mendukung empat pilar utama dari OOP, yaitu Abstraction, Inheritance, Polymorphism dan Encapsulation.

Sejarah perkembangan OOP dimulai pada tahun 1966 saat Ole Johan Dhal dan Kristen Nygaard dari universitas Oslo, Norwegia menerbitkan sebuah jurnal kertas kerja dengan judul "SIMULA An Algol Based Sumulation Language". Beberapa kemampuan utama dari pemrograman OOP, antara lain :

1. Pemrograman OOP menekankan pada data daripada prosedur karena data diperlukan sebagai elemen yang penting dan tidak boleh mengalir secara bebas dalam program.
2. Data disembunyikan dari akses program oleh fungsi-fungsi *(function)* ekternal.pat dibagi-bagi ke dalam obyek-obyek yang lebih kecil.
3. Program dapat dibagi-bagi ke dalam obyek-obyek yang lebih kecil.
4. Obyek dapat berkomunikasi satu dengan yang lain melalui function.
5. Data baru dan function dapat dengan mudah ditambahkan pada saat dibutuhkan.
6. Konsep pemrogramannya mengikuti pendekatan buttom up.

Pemrograman berorientasi obyek (OOP) merupakan metode pemrograman di mana pengembangan harus mendefenisikan tipe data dari struktur data dan juga tipe dari operasi yang dapat diaplikasikan ke stuktur data.

Dengan demikian struktur data menjadi obyek yang memiliki data dan fungsi. (Yuswanto, 2007 : 1-2).

II.7. *SQL Server 2008*

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang sangat *powerful* dan telah terbukti kekuiatan nya dalam mengelolah data. Dalam versi terbarunya ini, SQL Server 2008 memiliki banyak fitur yang bisa dihandalkan untuk meningkatkan performa *database*.

SQL Server 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari hari berkaitan dengan database, seperti menulis T-SQL, melakukan beckup dan restore database, melakukan security database terhadap aplikasi, dan sebagainya.

Pada GUI tersebut kita bisa melakukan settingan terhadap SQL Server untuk berkerja lebih *optimal*. Setting juga bisa dilakukan menggunakan script untuk memudahkan developer mengubah Setting Opsions pada SQL Server 2008. (*Jurnal Sigmata*, Vol : 1, No : 2. *2013 : 39*).