BAB II

LANDASAN TEORI

II.1. Sistem

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu. Pendekatan sistem yang merupakan jaringan kerja dari prosedur lebih menekankan urutan-urutan operasi di dalam sistem. Karakteristik sistem terdiri dari :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem.

2. Batasan Sistem (*boundry*)

Batasan merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*environment*)

Lingkungan luar sistem (*environment*) adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkung luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung Sistem (*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lain.

5. Masukan Sistem (*input*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintance input*) dan masukan sinyal (*signa linput*).

6. Keluaran Sistem (*output*)

Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah akan mengubah masukan menjadi keluaran.

8. Sasaran Sistem

Suatu sistem mempunyai tujuan (*goal*) atau sasaran (*objective*). Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya (Jeperson Hutahaean ; 2014 : 2).

II.2. Sistem Pendukung Keputusan

Decision Support Sistem (DSS) atau Sistem Pendukung Keputusan (SPK) secara umum didefinisikan sebagai sebuah sistem yang mampu memberikan kemampuan baik kemampuan pemecahan masalah maupun kemampuan

pengkomunikasian untuk masalah semi-terstruktur. Secara khusus, SPK didefinisikan sebagai sebuah sistem yang mendukung kerja seorang manajer maupun sekelompok manajer dalam memecahkan masalah semi-terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu (Eka Hendra Setyawan ; 2012 : 2).

II.2.1. Karakteristik Sistem Pendukung Keputusan

Karakteristik dan kapabilitas sistem pendukung keputusan antara lain :

1. Dukungan untuk pengambilan keputusan, terutama pada situasi semi terstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi. Masalah-masalah tersebut tidak dapat dipecahkan oleh sistem komputer lain atau oleh metode atau alat kuantitatif standar.
2. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
3. Dukungan untuk individu dan kelompok. Masalah yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain. DSS mendukung im virtual melalui alat-alat Web kolaboratif.
4. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Dukungan disemua fase proses pengambilan keputusan : intelegensi, desain, pilihan, dan implementasi
6. Dukungan diberbagai proses dan gaya pengambilan keputusan.

7. Adaptivitas sepanjang waktu. Pengambil keputusan seharusnya reaktif, dapat menghadapi perubahan kondisi secara cepat, dan dapat mengadaptasi DSS untuk memenuhi perubahan tersebut. DSS bersifat fleksibel dan karena itu pengguna dapat menambahkan, menghapus, menggabungkan, mengubah, atau menyusun kembali elemen-elemen dasar, DSS juga fleksibel dalam hal dapat dimodifikasi untuk memecahkan masalah lain yang sejenis.
8. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timeliness*, kualitas).
9. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah. DSS secara khusus menekankan untuk mendukung pengambil keputusan, bukannya menggantikan.

II.2.2. Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan terdiri dari 4 komponen utama, yaitu :

1. Subsistem manajemen data berfungsi sebagai memasukkan suatu *database* yang berisi data yang relevan untuk situasi dan dikelola oleh perangkat lunak yang disebut sistem manajemen *database* (DBMS).*Knowledge Base* berisi semua fakta, ide, hubungan dan interakasi suatu domain tertentu.
2. Subsistem manajemen basis pengetahuan bertugas untuk mendukung semua subsistem lain atau bertindak sebagai suatu komponen independen. Ia memberikan intelegensi untuk memperbesar pengetahuan pengambil keputusan.

3. Subsistem manajemen model Merupakan paket perangkat lunak yang memasukkan model keuangan statistik, ilmu manajemen atau model kuantitatif lainnya yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat.
4. Subsistem antar muka pengguna (dialog) untuk mengimplementasikan sistem kedalam program aplikasi sehingga pengguna atau pemakai dapat berkomunikasi dengan sistem yang dirancang (Nila Sutanti ; 2013 : 329).

II.2.3. Rancangan dan Jenis Sistem Pendukung Keputusan

Secara structural, sistem pendukung keputusan memiliki empat komponen penting :

1. Sistem Bahasa (LS)
2. Sistem Presentasi (PS)
3. Sistem Pengetahuan (KS)
4. Sistem Masalah – Processing (PPS)

Ini menentukan kemampuan dan perilaku (Bonczek et al. 1980, 1981a, Dos santos dan Holsapple 1989, Holsapple dan Whinstone 1996). Tiga yang pertama sistem representasi. Sebuah sistem bahasa terdiri dari sema pesan DSS dapat menerima. Sebuah sistem presentasi terdiri dari semua pesan DSS dapat memancarkan. Sebuah sistem pengetahuan terdiri dari semua pengetahuan DSS telah disimpan dan dipertahankan.

Dengan sendiri, ketiga jenis sistem dapat melakukan apa – apa, baik secara individu atau bersam – sama. Mereka mati, Mereka hanya mewakili pengetahuan,

baik dalam rasa pesan yang dapat dikirimkan atau representasi yang telah terakumulasi untuk kemungkinan proses selanjutnya.

Tujuan SPK umum :

1. Memberi dukungan untuk proses pengambilan keputusan.
2. Memberikan masukan semua level management.
3. Memberikan dukungan yang baik dependen / independen.
4. Memberikan dukungan langkah – langkah pengambilan keputusan.
5. Memudahkan penggunaan (*User friendly*).

II.2.4. Tiga Tingkatan Sistem Pendukung Keputusan

Dalam menjalankan fungsinya sebagai suatu Sistem pendukung Keputusan (SPK) dibagi menjadi tiga tingkatan. Ketiga tingkatan ini dibagi menurut type komputer, *hardware, software,* dan hirarki penerapan SPK itu sendiri.

Ketiga tingkatan itu adalah :

1. SPK Spesifik

Sistem ini mencakup data Sistem Informasi “aplikasi” namun dengan karakteristik yang berbeda jelas dengan aplikasi pengolah data lain.

SPK Spesifk adalah *hardware/software* yang memungkinkan pembuat keputusan spesifik berinteraksi dengan kumpulan hubungan masalah yang spesifik.

2. SPK Generator

SPK Generator merupakan paket dari kumpulan *hardware/software* yang menyediakan sekumpulan kemampuan untuk membuat SPK Spesifik dengan cepat dan mudah.

3. SPK Tools

SPK Tools merupakan elemen - elemen *hardware/software* yang mana dapat dipergunakan untuk mengembangkan SPK Spesifik maupun SPK Generator. Meskipun SPK Tool ini mampu membuat SPK Spesifik secara langsung, namun mengembangkan SPK Spesifik dengan SPK Generator jauh lebih mudah dan efisien.

II.2.5. Tahapan Proses Pengambilan Keputusan

Tiga tahapan dalam proses pengambilan keputusan :

1. Tahap *Intellgen,* adalah tahap proses pengenalan persoalan melalui penyelidikan lingkungan untuk mengetahui ada atau tidaknya masalah. Kesimpulan dari penyelidikan diperoleh dari pengolahan data dengan metode yang telah ditetapkan sebelumnya atau dengan metode khusus. Aliran informasi bergerak dari tingkatan manajemen terendah menuju tingkatan manajemen tertinggi.
2. Tahap *Design,* merupakan tahap mencari, analisis serta perumusan alternatf tindakan yang akan diambil. Pada tahap design ini, sistem informasi harus mampu membuat keputusan – keputusan.
3. Tahap *Choice,* merupakan tahap memilih suatu tindakan yang paling tepat dari beberapa alternatif yang telah dirumuskan. Langkah selanjutnya adalah pelaksanaan alternatif terpilih. Bila suatu alternatif telah dilaksanakan, fungsi informasi berubah menjadi pengumpul data untuk selanjutnya, merupakan umpan balik. (Rosnaini Ginting ; 2014 : 24) .

II. 3. Analytical Hierarchy Process (AHP)

II.3.1. Pengertian AHP

Analytical Hierarchy Process (AHP) merupakan suatu model pendukung keputusan. AHP menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki. hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis.

Metode AHP adalah suatu model pengambilan keputusan yang komprehensif, karena memperhitungkan hal-hal kualitatif dan kuantitatif sehingga keputusan-keputusan yang diambil bisa lebih obyektif. Metode ini dapat menyelesaikan masalah multikriteria yang kompleks menjadi suatu hirarki. Masalah yang kompleks dapat diartikan bahwa kriteria dari suatu masalah yang begitu banyak (multikriteria), struktur masalah yang belum jelas, ketidak pastian pendapat dari pengambilan keputusan lebih dari satu orang, serta ketidak akuratan data yang tersedia.

Analytical Hierarchy Prosess (AHP) adalah suatu metode yang digunakan dalam menganalisis data untuk mengambil suatu alternatif dari banyak kriteria (*multiple criteria*) yang telah ada sebelumnya. *Analytical Hierarchy Process* (AHP) adalah sebuah hirarki fungsional dengan *input* utamanya persepsi manusia.

Dengan hirarki, suatu masalah kompleks dan tidak terstruktur dipecahkan kedalam beberapa kelompok-kelompoknya. Kemudian kelompok tersebut diatur menjadi suatu bentuk hirarki. AHP memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga mudah dipahami oleh semua pihak yang terlibat dalam pengambilan keputusan.

(Perkasa Putra Nasution ; 2014 : 56).

II.3.2. Prosedur Kegiatan AHP

Pada dasarnya prosedur atau langkah-langkah dalam metode AHP meliputi :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hirarki dari permasalahan yang dihadapi. Penyusunan hirarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menentukan prioritas elemen
    a. Langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.
    b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen yang lainnya.

    Penetapan bobot prioritas diatur seperti pada tabel berikut ini.

Tabel II.1. Skala Penilaian Perbandingan Pasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting dari pada elemen lainnya
5	Elemen yang satu lebih penting dari pada elemen yang lainnya
7	Satu elemen jelas lebih mutlak penting dari pada elemen lainnya
9	Satu elemen mutlak penting dari pada elemen laninnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika aktifitas i mendapat satu angka dibandingkan dengan aktivitas j, maka i memiliki nilai kebalikannya dibandingkan dengan j

(Perkasa Putra Nasution ; 2014 : 56).

3. Sintesis

   Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah :

   a. Menjumlahkan nilai-nilai dari setiap kolom matriks
   b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
   c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4. Mengukur Konsistensi

   Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :

a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua dan seterusnya.
b. Jumlahkan setiap baris
c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
d. Jumlahkan hasil bagi diatas dengan banyaknya elemen yang ada, hasilnya disebut λ maks.
e. Hitung *Consistency Index* (CI) dengan rumus:

$$\mathbf{CI}=\frac{\boldsymbol{\lambda\text{maks}-N}}{\boldsymbol{N-1}}$$

Dimana N = banyaknya elemen

f. Hitung Rasio Konsistensi/*Consistency Ratio* (CR) dengan rumus

$$\boldsymbol{CR}=\frac{\boldsymbol{CI}}{\boldsymbol{Index\ Random\ (IR)}}$$

Dimana CR = *Consistency Ratio*

CI= *Consistency Index*

IR= *Index Random Consistency*

Daftar Indeks Random Konsistensi (IR) bisa dilihat dalam tabel berikut :

Tabel II.2. Daftar Indeks Random Konsistensi

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

(Perkasa Putra Nasution ; 2014 : 56).

g. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 100% maka penilaian data *judgement* harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0.1 maka hasil perhitungan bisa dinyatakan benar.

5. Membuat Matriks Berpasangan Alternatif
   a. Masukkan nilai
   b. Menghitung selisih penilaian kriteria
   c. Konversikan selisih rata-rata ke skala penilaian (Nilai perbandingan antar alternatif)
   d. Jumlahkan nilai setiap kolom

6. Menghitung *eigen vector* alternatif
    a. Nilai setiap kolom dibagi jumlah kolom
    b. - Jumlahkan setiap baris= x
        - *Eigen vector* alternatif= x / banyak alternatif
7. Menghitung prioritas alternatif masing-masing kriteria
    a. Prioritas = *eigen vector* alternatif x *eigen vector* kriteria
    b. Jumlahkan nilai setiap baris

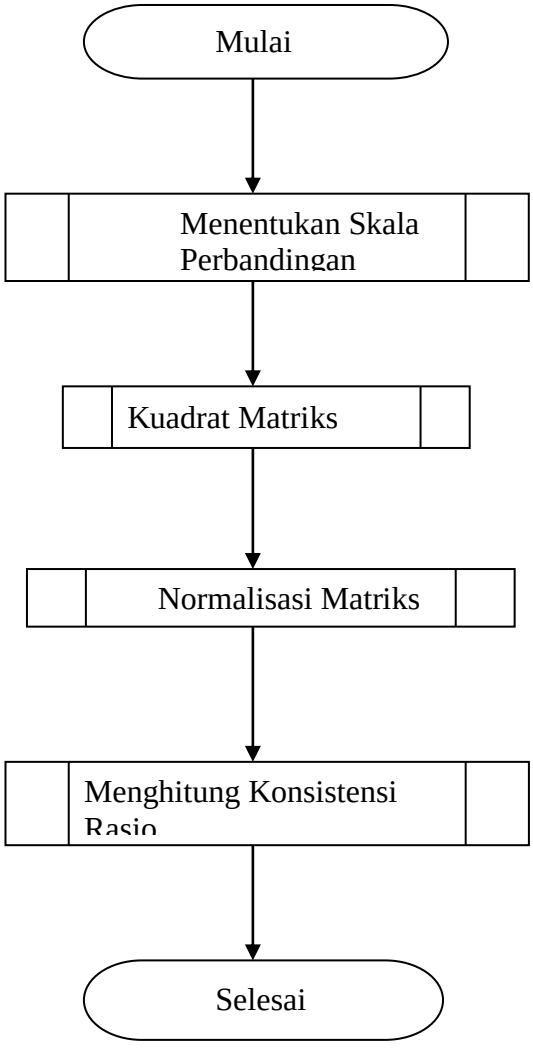
Setelah semua langkah-langkah tersebut dilakukan maka hasilnya adalah perangkingan alternatif.

(Perkasa Putra Nasution ; 2014 : 56)

II.3.3. Flowchart Metode

Proses yang terdapat dalam flowchart AHP adalah proses menentukan skala perbandingan, kuadrat matriks, normalisasi matriks, dan menghitung konsistensi rasio. Flowchart proses AHP dapat dilihat pada gambar II.1.

.



Gambar II.1. Flowchart Metode

(Hafsah : 2011 : 47)

II.3.4. Skala *LIKERT*

Skala *likert* adalah suatu skala psikometrik yang digunakan dalam kuisioner dan merupakan salah satu teknik yang dapat digunakan dalam evaluasi suatu program atau kebijakan perencanaan. Skala *likert* pertama kali dikembangkan oleh Rensis *Likert* pada tahun 1932 dalam mengukur sikap masyarakat. Dengan skala *likert*, variabel yang diukur dijabarkan menjadi indikator variabel. Kemudian ndikator tersebut dijadikan sebagai titik tolak untuk menyususn item-item instrumen yang dapat berupa pertanyaan atau peryataan. Jawaban setiap item instrumen yang mengunakan skala *likert* mempunyai garadasi dari sangat positif sampai sangat negatif, yang dapat berupa kata-kata antara lain : Sangat setuju (SS), setuju (S), netral (N), tidak setuju (TS) dan sangat tidak setuju (STS). Skala *likert* digunakan untuk mengukur sikap. Pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomenal sosial.

Dalam skala *likert*, item ada yang bersifat *favorabel* (baik/positif/tidak mendukung) terhadap masalah yang diteliti, sebaliknya ada pula yang bersifat *unfavorble* (tidak baik/negatif) terhadap masalah yang diteliti. Jumlah item yang positif maupun yang negatif sebaiknya harus seimbang atau sama.

Beberapa bentuk jawaban pertanyaan atau pernyataan yang masuk dalam kategori skala *likert* adalah sebagai berikut :

Alternatif penilaian terhadap item yang positif terhadap masalah penelitian:

Sangat setuju : 5

Setuju : 4

Kurang setuju : 3

Tidak setuju : 2

Sangat tidak setuju : 1

Alternatif penilaian terhadap item yang negatif terhadap masalah penelitian:

Sangat setuju : 1

Setuju : 2

Kurang setuju : 3

Tidak setuju : 4

Sangat tidak setuju : 5

Corak khas dari skala *likert* ialah bahwa makin tinggi skor yang diperoleh oleh seseorang, merupakan indikasi bahwa orang tersebut sikapnya makin positif terhadap objek sikap, demikian sebaliknya.

(Perkasa Putra Nasution ; 2014 : 56).

II.4. Pengertian PHP

PHP (*Hypertext Processor)* adalah bahasa skrip yang dapat ditanamkan atau disisipkan kedalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS.

Pada awalnya PHP merupakan kependekan dari personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdor pada tahun 1995. Pada waktu itu PHP masih bernama *Form Interpreted* (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir dari web.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi sumber

terbukan, maka banyak pemrogram yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini, *interpreter* PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul–modul eksistensi yang meningkatkan kemampuan PHP/FI secara signifikan. (Alan Nur Aditya ; 2011 : 29).

II.5. Pengertian MySQL

MySQL adalah sebuah perangkat lunak system manajemen basis data SQL (bahasa inggris : *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi komersial untuk kasus – kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek – proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing – masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu ornag Finlandia yang mendirikan MySQL AB adalah : David Axmark, Alan Larsson, dan Michael “Monty” Widenius. (Alan Nur Aditya ; 2011 : 61).

II.6. Database

Database dapat diartikan sebagai sebuah kumpulan data yang terdiri atas satu atau lebih tabel yang terintegrasi satu sama lain. Dalam sebuah *database*, seorang pemakai atau *user* dapat mengubah, menambah, menganalisa, bahkan

menghapus data dalam tabel-tabel tersebut. Dalam sebuah *database*, tabel berfungsi untuk menyimpan data yang saling berhubungan, misalnya tabel nama hanya berisi daftar nama-nama. Tabel sendiri memiliki dua bagian penting yaitu kolom (*field*) dan baris (*record*). Dalam komputer, *database* dapat diolah menggunakan *software* pengelola database seperti *Microsoft Access, Microsoft SQL Server, MySQL* (Fajar Rahadian ; 2011 : 4).

II.7. Teknik Normalisasi

Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan, yaitu :

1. Bentuk normal tahap pertama (1" Normal Form)

Contoh yang kita gunakan di sini adalah sebuah perusahaan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri.

2. Bentuk normal tahap kedua (2^{nd} normal form)

Definisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel

relasional berada pada bentuk normal kedua jika dia berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada seluruh kolom yang membentuk kunci utama.

3. Bentuk normal tahap ketiga (3rd normal form)

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utama. Secara definisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya.

4. *Boyce Code Normal Form* (BCNF)

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3NF sudah cukup karena sangat jarang entitas yang berada pada 3NF bukan merupakan 4NF dan 5NF.

5. Bentuk Normal Tahap Keempat dan Kelima

Sebuah tabel relasional berada pada bentuk normal keempat (4NF) jika dia dalam BCNF dan semua ketergantungan multivalue merupakan ketergantungan fungsional. Bentuk normal keempat (4NF) didasarkan pada konsep ketergantungan multivalue (MVD).

Sebuah tabel berada pada bentuk normal kelima (5NF) jika ia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil. Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*) (Janner Simarmata ; 2010 : 76).

II.8. 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.

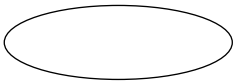
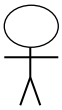
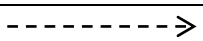
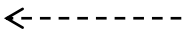
UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem.

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

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*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.3. Simbol *Use Case*

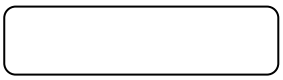
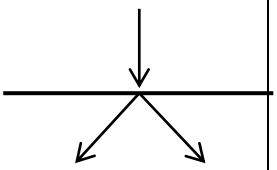
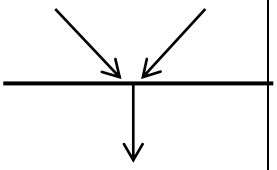
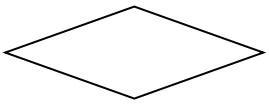
Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukan pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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 *use case*, tetapi tidak memiliki control terhadap *use case*.
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	*Include*, merupakan di dalam *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.

(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.4. Simbol *Activity Diagram*

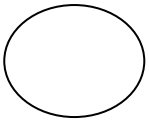
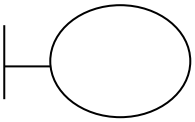
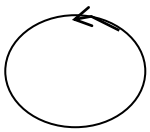
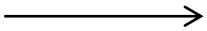
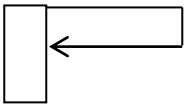
Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*
	*Swimlane*, pembagian *activity* diagram 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.5. 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 aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class,* suatu 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*.

(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 tanggng 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.6. *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

(Sumber : Windu Gata ; 2013 : 9)