

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai tujuan. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukanlah bagian dari sistem. (Abdul Kadir; 2014).

Menurut Asbon Hendra (2012; 157)Sistem merupakan kumpulan unsur atau elemen – elemen yang saling berkaitan / berinteraksi dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisasi, saling berinteraksi, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum pertama kali di uraikan oleh Kenneth Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas untuk memimpin organisasi adalah dia terlalu memusatkan perhatiannya pada salah satu komponen sistem organisasi. Teori sistem mengadakan bahwa unsur pembentuk organisasi itu penting dan harus mendapat perhatian yang utuh supaya manajer dapat bertindak lebih efektif. Yang dimaksud unsur atau komponen pembentuk

organisasi disini bukan hanya bagian-bagian yang tampak secara fisik, tetapi juga hal-hal yang mungkin bersifat abstrak atau konseptual, seperti misi, pekerjaan, kegiatan, kelompok informal, dan lain sebagainya. (Tata Sutabri; 2012; 3)

Dari pengertian sistem di atas dapat disimpulkan bahwa sistem adalah himpunan komponen atau bagian yang saling berkaitan yang bersama – sama untuk mencapai suatu tujuan tertentu.

II.1.1. Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem yaitu:

1. Tujuan

Setiap sistem memiliki tujuan (*goal*) entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem tanpa tujuan, sistem menjadi tidak terarah dan tidak terkendali.

2. Masukan

Masukan (*input*) sistem adalah segala sesuatu yang masuk kedalam sistem dan selanjutnya menjadi bahan untuk diproses. Masukan dapat berupa hal-hal berwujud (tampak secara fisik) maupun yang tidak tampak.

3. Proses

Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan limbah.

4. Keluaran

Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.

5. Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feed back*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah mengatur agar sistem berjalan sesuai tujuan. (Abdul Kadir; 2014; 62-64).

II.1.2. Klasifikasi Sistem

a. Sistem Abstrak (*Abstract System*)

Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Sebagai contoh, sistem teologia yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan manusia.

b. Sistem Fisik (*Phycal System*)

Merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya. Contohnya, sistem komputer, sistem akutansi, sistem produksi, dan lain-lain.

c. Sistem Alamiah (*Natural System*)

Sistem yang terjadi melalui proses alam, dalam artian tidak dibuat oleh manusia, seperti sistem tata surya, sistem galaxy, sistem reproduksi, dan lain-lain.

d. Sistem buatan manusia (*Human Made System*)

Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut *human machine system*, contohnya sistem informasi.

e. Sistem tertentu (*deterministic System*)

Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat didekteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Contohnya sistem komputer.

f. Sistem Tak Tentu (*Probabilistic System*)

Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. Contohnya sistem manusia.

g. Sistem tertutup (*Closed System*)

Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teori, sistem tersebut ada, tetapi kenyataanny tidak ada sistem yang benar-benar tertutup, yang ada hanya ada *relatively closed system*(secara relatif tertutup, tidak benar-benar tertutup).

h. Sistem terbuka(*open system*)

Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomatis, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai

bagian sari sistem yang digunakan dalam masyarakat modern.
(Asbon Hendra; 2012; 60-162).

II.1.3. Sistem Terotomatis

Sistem terotomatis terbagi dalam sejumlah katagori, yaitu :

a. Online system

Sistem online adalah sistem yang menerima langsung *input* pada area dimana *input* tersebut direkam dan menghasilkan output yang dapat berupa hasil komputasi pada area dimana mereka dibutuhkan. Area sendiri dapat dipisah-pisah dalam skala, misalnya ratusan kilometer. Biasanya digunakan bagi resevasi angkutan udara, resevasi kreta api, perbankan, dan lain sebagainya.

b. Real- time system

Sistem real – time adalah mekanisme pengontrolan, perekaman data, dan pemrosesan yang sangat cepat sehingga output yang diterima dalam waktu yang relatif sama. Perbedaan dengan sistem *online* adalah satuan waktu yang digunakan *real-time* biasanya seperseratus atau seperseribu detik sedangkan *online* masih dalam skala detik atau kadang beberapa menit. Perbedaan lainnya, *online* hanya berinteraksi denga pemakai, sedangkan *real-time* berinteraksi langsung dengan pemakai dan lingkungan yang dipetakan.

c. Decision support system and strategic planing system.

Sistem ini memproses transaksi organisasi secara harian atau membantu para manajer mengambil keputusan, mengevaluasi, dan

menganalisa tujuan organisasi. Biasanya digunakan untuk sistem penggajian, sistem pemesanan, sistem akuntansi, dan sistem produksi. Biasanya berbentuk paket statistik, paket pemasaran, dan lain – lain. Sistem ini tidak hanya merekam dan menampilkan data, tetapi juga fungsi matematik, data analisa statistik, dan menampilkan informasi dalam bentuk grafik (tabel, chart) sebagai laporan konvensional.

d. *Knowledge-based system*

Program komputer ini dibuat mendekati kemampuan dan pengetahuan seorang pakar. Umumnya menggunakan perangkat keras dan perangkat lunak khusus seperti *LISP* dan *PROLOG*. (Asbon Hendra; 2012; 162-163).

II.1.4. Metode Sistem

a. *Blackbox Approach*

Suatu sistem dimana input dan output-nya dapat didefinisikan tetapi prosesnya tidak dapat diketahui atau tidak terdefinisi. Metode ini hanya dapat dimengerti oleh pihak dalam (yang menangani) sehingga pihak luar hanya mengetahui masukan dan hasilnya.

Contoh : bagian percetak uang, proses pencernaan.

b. *Analytic System*

Suatu metode yang mencoba untuk melihat hubungan seluruh masalah yang menyelidiki kesistematiskan tujuan dari sistem yang tidak efektif dan evaluasi pilihan dalam bentuk ketidak efektifan dan biaya.

Dalam metode ini, beberapa langkah diberikan seperti dibawah ini :

1. Menentukan identitas dari sistem.
 - a. Sistem apa yang diterapkan.
 - b. Batasanya.
 - c. Apa yang dilaksanakan sistem tersebut.
2. Menentukan tujuan dari sistem.
 - a. Output yang dihasilkan dari isi sistem.
 - b. Fungsi dan tujuan yang diminta untuk mencoba menanggulangi lingkungan.
3. Bagian apa saja yang terdapat dalam sistem dan apa tujuan dari masing – masing bagian tersebut.
 - a. Tujuan masing – masing bagian sistem harus jelas.
 - b. Cara apa yang digunakan subsistem untuk berhubungan dengan subsistem lainnya.
4. Bagaimana bagian – bagian yang ada dalam sistem itu saling berhubungan menjadi satu kesatuan. (Asbon Hendra; 2012; 166 - 167).

II.2. Informasi

Informasi merupakan salah satu sumber daya penting dalam manajemen modern. Banyak keputusan strategis yang bergantung kepada informasi. Sebagaimana diketahui, sumber daya 4M+1I yang mencakup manusia (Sumber Daya Manusia atau SDM), material (termasuk didalamnya energi), mesin, modal, dan informasi merupakan sumber daya vital bagi kelangsungan organisasi bisnis. (Abdul Kadir; 2014; 41).

Informasi merupakan data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi, ada suatu proses transformasi data menjadi suatu informasi = *input – proses – output*. (Asbon Hendra; 2012; 167).

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. (Tata Sutabri; 2012; 22)

II.3. Sistem Informasi

Pengertian Sistem Informasi menurut Alter “Sistem Informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi.

Pengertian Sistem Informasi menurut Gelinas, Oram, dan Wiggins Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelolah data serta menyediakan informasi keluaran kepada para pemakai. (Abdul Kadir; 2014; 9).

“suatu sistem terintegritasi yang mampu menyediakan informasi yang bermanfaat bag penggunanya”. “sebuah sistem terintegritas atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi’. “sekumpulan prosedur manua atau terkomputerisasi yang mengumpulkan / mengambil, mengelolah, menyimpan, dan menyebarkan informasi dalam mendukung pengambilan dan kendali

keputusan”. “sekelompok orang, prosedur, input, output, dan pengelolaannya secara bersama-sama menghasilkan informasi yang akurat, tepat waktu, dan relevan bagi penggunaannya. (Asbon Hendra; 2012; 168-169).

Sistem informasi adalah suatu sistem di dalamnya suatu organisasi yang mempertemukan kebutuhan pengolahan harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan laporan – laporan yang diperlukan oleh pihak luar tertentu. (Tata Sutabri; 2012; 38)

II.4. Data

Data dapat dianalogikan dengan sejumlah blok yang biasa digunakan anak-anak untuk membentuk berbagai struktur sesuai dengan imajinasi mereka. Secara konsep, data adalah deskripsi tentang benda, kejadian, aktifitas, dan transaksi, yang tidak mempunyai makna atau tidak berpengaruh secara langsung kepada pemakai. (Abdul Kadir; 2014; 43).

Data merupakan *raw material* untuk suatu informasi. Perbedaan informasi dan data sangat relatif, tergantung pada nilai gunanya bagi manajemen yang memerlukan. Suatu informasi bagi level manajemen tertentu bisa menjadi data bagi manajemen level di atasnya, atau sebaliknya. (Asbon Hendra; 2012; 167).

II.5. Sistem Pakar

Sistem pakar merupakan sistem yang dirancang untuk dapat menirukan keahlian seorang pakar atau ahli dalam menjawab pertanyaan dan memecahkan suatu masalah. Tujuan utama sistem pakar yaitu untuk

memasyarakatkan atau memindahkan secara efektif pengetahuan dan pengalaman para pakar kepada mereka yang bukan pakar. Pemecahan masalah diberikan pada pemakai melalui dialog dengan mereka. Sistem pakar membantu seseorang yang bukan pakar/ ahli dalam menjawab pertanyaan, menyelesaikan masalah dan mengambil keputusan yang biasanya dilakukan oleh seorang pakar. Sistem pakar yang terdiri dari tiga komponen pendukung yaitu Pangkalan Pengetahuan (*Knowledge Base*) yang berisi fakta-fakta, ide, interaksi suatu domain tertentu, Motor Inferensi (*Inference Engine*) yang bertugas menganalisa pengetahuan dan menarik kesimpulan berdasarkan pangkalan pengetahuan dan Antarmuka Pemakai (*User Interface*) yang berfungsi sebagai media yang melakukan komunikasi dengan pemakai. Pendekatan penyelesaian masalah dalam sistem pakar dalam tes psikologi terdapat banyak metoda yang berbeda untuk merepresentasikan pengetahuan. Beberapa metoda yang terbagi menjadi dua macam, pertama representasi pengetahuan yang bersifat deklaratif, seperti: logika, jaringan semantik, predikat kalkulus, *list*, *frame*, *script* dan kedua representasi pengetahuan yang bersifat prosedural, seperti: prosedur atau subroutine dan kaidah produksi. Sampai saat ini metoda paling banyak dipakai dalam pembuatan sistem pakar adalah jaringan semantik dan kaidah produksi. Mekanisme inferensi adalah rangkaian prosedur yang digunakan untuk menguji pangkalan pengetahuan dengan cara yang sistematis pada saat menjawab pertanyaan, persoalan atau membuat keputusan dalam satu domain yang telah ditentukan. Skema yang digunakan dalam mekanisme ini adalah skema deklaratif yaitu dengan

menggunakan pohon pelacakan yang merupakan bagian dari jaringan semantik dan skema prosedural dengan menggunakan kaidah produksi bentuk IF– THEN. (Prosiding Seminar Nasional Aplikasi Sains & Teknologi (SNAST); Periode III ISSN: 1979-911X ;Yogyakarta; 3 November 2012; 287)

II.6. Metode Fuzzy Logic

Fuzzy logic merupakan suatu metode yang digunakan dalam proses pengambilan keputusan dengan cara memetakan suatu ruang input ke dalam ruang output. Sistem ini diperkenalkan pertama kali oleh Prof. Lotfi Zadeh dari Universitas California, Barkeley tahun 1962, dimana pada saat itu boolean logic hanya mengenal dua keadaan yaitu : ya/tidak, ON/OFF, High/Low atau hanya mempunyai logika 0 dan 1 saja. Sedangkan kondisi nyata di alam ini bukan hanya ya (1, high, on) atau tidak (0, low, off) tetapi seluruh kemungkinan diantara 0 dan 1, sehingga untuk mengenal kondisi ini kita tidak dapat menggunakan boolean logic tetapi dengan menggunakan fuzzy logic.

Proses Sistem Fuzzy , Pada sistem fuzzy terdapat tiga proses sebagai berikut.

1. Fuzzification

Proses ini berfungsi untuk mengubah masukan-masukan berupa nilai analog atau yang nilai kebenarannya bersifat pasti (crisp input) menjadi nilai fuzzy, yang digunakan sebagai fuzzy input. Fuzzy input ini berupa nilai linguistic yang semantiknya ditentukan berdasarkan fungsi keanggotaan tertentu. Jika terdapat suatu nilai analog yang menjadi input pada proses fuzzy

maka input tersebut dimasukkan pada batas scope/domain sehingga didapatkan suatu nilai fungsi keanggotaan. Nilai fungsi keanggotaan inilah yang menentukan proses pengambilan keputusan selanjutnya. Salah satu metode yang digunakan oleh proses ini dalam pengenalan teks adalah feature extraction. Metode ini digunakan untuk mendapatkan karakteristik dari suatu citra dengan melihat bentuk dasar objek pada citra tersebut. Tujuan metode ini adalah untuk melakukan pengukuran terhadap hal-hal yang membedakan pola masukkan sehingga objek pada citra yang satu dan yang lain dapat dibedakan.

2. Rule Evaluation

Proses ini digunakan untuk mencari nilai fuzzy output dari fuzzy input. Jika terdapat suatu nilai fuzzy input dari proses fuzzification nilai tersebut akan dimasukkan ke dalam rule yang telah dibuat untuk mendapatkan nilai fuzzy output. Pada proses inilah suatu sistem dapat dikatakan pintar atau tidak. Jika rule yang dibuat tidak pintar maka sistem yang dikontrol menjadi kacau dan objek yang seharusnya dikenali menjadi tidak dapat dibaca. Dengan rule yang ada diperoleh nilai fuzzy yang digunakan dalam membantu pengambilan keputusan.

3. Defuzzification

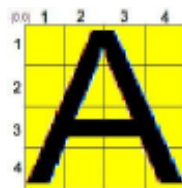
Pada tahap inilah pengambilan keputusan dilakukan. Nilai yang didapatkan berupa nilai crisp, yaitu 0 atau 1. Proses defuzzification melakukan suatu fungsi output yang memproses nilai fuzzy yang berasal dari rule evaluation sehingga keputusan akhir dapat dilakukan. Fungsi output ini dilakukan dengan mencocokkan nilai-nilai yang ada dengan threshold yang

ditentukan. Sebuah input pada proses fuzzy akan diterima sebagai anggota himpunan fuzzy jika memiliki nilai keanggotaan yang melewati batas threshold yang ada. (Jurnal UPN "Veteran" Yogyakarta; 22 Mei 2010; 35-36)

Adapun studi kasus dari metode fuzzy logic :

Pendekatan Fuzzy Logic pada Pengenalan Teks

Pendekatan fuzzy logic merupakan salah satu cara pendekatan yang digunakan dalam pengenalan pola. Pada pendekatan ini digunakan metode pemisahan karakter menjadi daerah-daerah kecil yang disebut box-method. Hal ini terlihat seperti yang terlihat pada gambar berikut.



Gambar : II.1 Box- Metod

(Sumber : Semnas IF : 2010)

Dari box hasil pembagian tersebut akan didapat suatu nilai yang digunakan sebagai nilai input fungsi keanggotaan pada proses fuzzy. Nilai-nilai yang didapatkan berasal dari nilai koordinat pixel terdapat warna hitam (i, j), dimana pada pojok kiri atas masing-masing box memiliki koordinat pixel (0,0). Nilai vector distance masing-masing pixel diperoleh dengan rumus:

$$d_{xb} = \sqrt{i^2 + j^2}$$

Nilai vector distance tersebut kemudian dihitung sebagai normalisasi dari total semua pixel masing-masing box dengan rumus:

$$Y_b = \frac{1}{n} \sum_{k=1}^n x_{kb}$$

Adapun n merupakan jumlah pixel dalam box dan b merupakan nomor box.

Nilai ini digunakan sebagai nilai fungsi keanggotaan pada sistem fuzzy.

Dalam sistem fuzzy, dihitung nilai mean (m) dan variance () menggunakan rumus:

$$m_i = \frac{1}{N_i} \sum_{j=1}^{N_i} f_{ij}$$

$$\sigma_i^2 = \frac{1}{N} \sum_{j=1}^{N_i} (f_{ij} - m_i)^2$$

Dimana N_i adalah jumlah sampel dalam cluster ke- i dan f_{ij} merupakan nilai fungsi keanggotaan dari masing-masing box dengan karakter ke- j . Setelah didapatkan nilai mean dan variance, maka nilai tersebut dapat dibandingkan dengan nilai fungsi keanggotaan masing-masing box yang dicari dengan rumus:

$$\mu_{xi} = e^{-\frac{(x_i - m_i)^2}{\sigma_i^2}} \quad \text{untuk } \sigma_i^2 \geq 1$$

$$\mu_{xi} = e^{-(x_i - m_i)^2 \sigma_i^2} \quad \text{untuk } \sigma_i^2 < 1$$

Selanjutnya dicari nilai μ_{xi} masing-masing karakter dengan mencari nilai rata-rata semua box, dengan rumus:

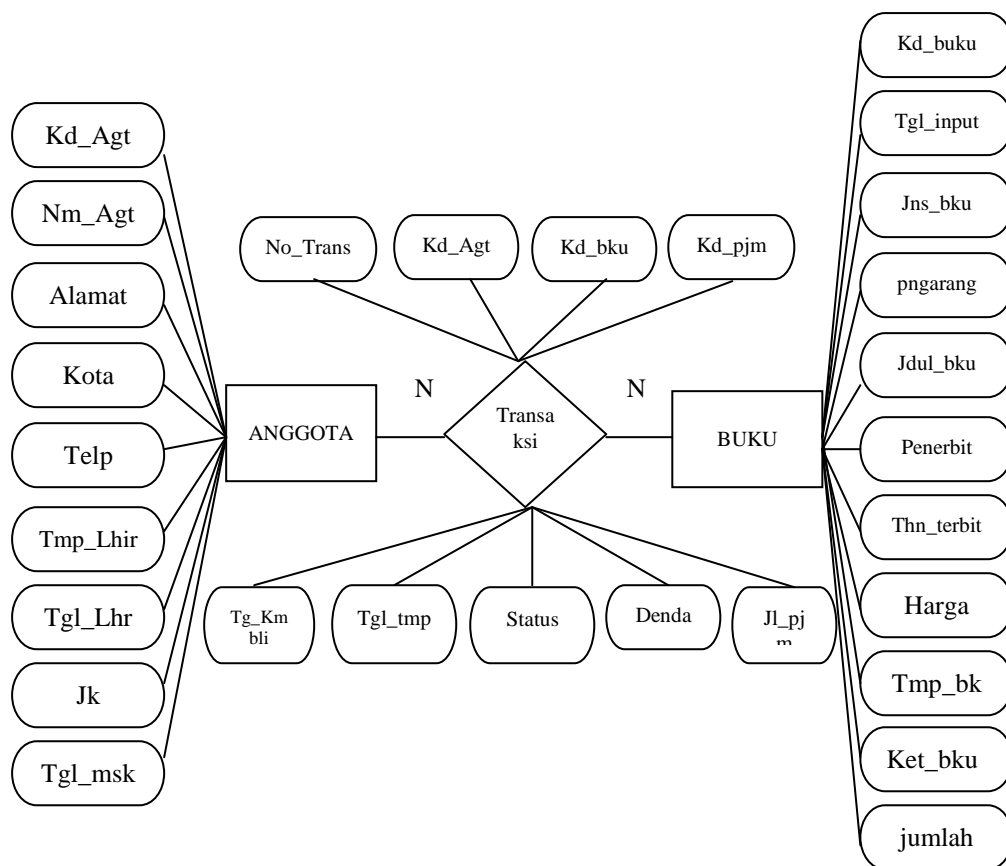
$$\mu_{av}(r) = \frac{1}{c} \sum_{j=1}^c e^{\frac{-(x_j - m(r))^2}{\sigma_j^2(r)}}$$

Adapun c berarti jumlah total semua box dan $r =$ merupakan penomoran karakter-karakter yang ada pada knowledge base. Sebuah karakter dapat dikenali dengan mencari nilai $=av(r)$ yang paling besar atau mendekati angka 1 (satu)

II.7. *Entity Realitionship Diagram (ERD)*

ERD (*Entity Realitionship Diagram*) adalah sebuah diagram yang secara konseptua memetakan hubungan antar penyimpanan pada diagram DFD. ERD ini digunakan untuk melakukan pemodelan terhadap struktur data dan hubunganny. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah *database* yang baik.

Entity dapat berarti sebuah obyek yang daat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau *field*) yang membuatnya dapat dibedakan dari obyek yang lain. Hal ini berarti sebuah entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut. Berikut ini contoh pengambaran entity.



Gambar : II.2 Gambar ERD

Sumber : (Wahana Komputer; 2010; 31)

ERD di atas adalah contoh ERD sistem informasi perpustakaan. Huruf N pada jalur-jalur hubungan antara anggota dan transaksi serta buku ada derajat dari simbol relasi. Ada beberapa derajat yang terjadi, yaitu :

- One to one* , menggambarkan bahwa antara 1 anggota entity A hanya dapat berhubungan dengan 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-1.
- One to many* , menggambarkan bahwa antara 1 anggota entity A hanya dapat memiliki hubungan dengan lebih dari 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.

- c. *Many to many* , menggambarkan bahwa antara satu anggota A dapat memiliki hubungan dengan lebih dari 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol N-N. (Wahana Komputer; 2010; 30-31)

II.8. Normalisasi

Normalisasi adalah proses dimana tabel – tabel pada *database* dites dalam hal kesalingtergantungan di antara *field* – *field* pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu *field* dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel. Setiap tabel hanya boleh memiliki sebuah field kunci yang menjadi ketergantungan dari *field* lainnya dalam tabel tersebut.

Ada beberapa langkah dalam normalisasi tabel yaitu :

1. Dekomposisi (*Decomposition*)

Dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenai dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di dekomposisi. Dekomposisi akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat – syaratnya.

2. Bentuk Tidak Normal

Pada bentuk ini semua data yang ada pada tiap *entity* (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel

inimasih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.

3. Normal *Form* Pertama (*1st Normal Form*)

Pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel – tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.

4. Normal *Form* Kedua (*2nd Normal Form*)

Pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut.

5. Normal *Form* Ketiga (*3rd Normal Form*)

.Setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. (Wahana Komputer, 2010;32-35).

II.9. Basis Data (*Database*)

Database atau basis data adalah sekumpulan data yang memiliki hubungan sara logika dan diatur dengan susunan tertentu serta disimpan dalam media penyimpanan komputer. Data itu sendiri adalah data representasi dari semua fakta yang ada pada dunia nyata yang ada. *Database* sering digunakan untuk melakukan proses terhadap data-data tersebut untuk menghasilkan informasi tertentu. Misalnya dari data nama siswa dan tanggal lahir siswa

anda bisa mendapatkan informasi nama siswa yang berulang tahun pada hari ini. Tentu saja informasi tersebut akan anda dapatkan dari *software* pemrosesan *database* dengan cara anda memberikan perintah dalam bahasa tertentu yaitu *SQL (Structured Query Language)*.

Dalam *database* ada sebutan-sebutan untuk satuan data yaitu :

- a. Karakter, ini adalah satuan data terkecil. Data terdiri atas susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
- b. *Field* , adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama mahasiswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan database, *field* juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi obyek aka sebuah *field* akan memiliki dua properti utama yaitu properti name dan *property type*. Properti name atau nama adalah properti dari *field* yang berisi nama *field* yang mewakili data sejenis yang disimpan. Sedangkan *property type* adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama *field*nya adalah nama siswa maka tipe datanya adalah char, bila nama *field*nya adalah tanggal lahir maka tipe datanya adalah date. *Field* dilihat seperti kolom.
- c. *Record*, adalah kumpulan dari field. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengkombinasikan *field-field* yang ada.

- d. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah *database*, wujud fisik dalam *database* dalam sebuah komputer adalah sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan diatas.
- e. *File*, adalah bentuk fisik dari penyimpanan data file. *Database* berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi.

Sistem *database* adalah sebuah kumpulan dari komponen-komponen *database-database* yang meliputi :

1. *Database*
2. *Database server*
3. Komponen *client software*
4. Aplikasi *database*

Aplikasi *database* adalah sebuah *software* khusus yang di desain dan digunakan oleh user atau pihak lainya seperti penyedia jasa pemrograman atau konsultan. Sedangkan *client server* adalah salah satu komponen yang termasuk dalam sistem *database* yang memungkinkan *software* aplikasi *database* mengakses data secara *remote* pada sebuah *server database*.
(Wahana Komputer, 2010;24-25).

II.10. *Unified Modeling Language (UML)*

Unified Modelling Language (UML) adalah sebuah alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. Juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek.

Sejarah UML sendiri terbagi dalam dua fase ; sebelum dan sesudahnya munculnya UML. Dalam fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990-an namun notasi yang dikembangkan oleh para ahli analisa dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

Fase kedua; dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Management Group (OMG)* maka pengembangan UML dimulai pada akhir tahun 1994 ketika *Grady Booch* dengan metode OOD (*Object-Oriented Design*), *Jim Rumbaugh* dengan metode OMT (*Object Modelling Technique*) mereka ini bekerja pada *Rational Software Corporation* dan *Ivar Jacobson* dengan metode OOSE (*Object-Oriented Software Engineering*) yang bekerja pada perusahaan *Objectory Rational*.

Sebagai pencetus metode-metode tersebut mereka bertiga berinisiatif untuk menciptakan bahasa pemodelan terpadu sehingga pada tahun 1996 mereka berhasil merilis UML versi 0.9 dan 0.91 melalui *Request for Proposal (RFP)* yang dikeluarkan oleh *OMG (Braun, et.al. 2001)*.

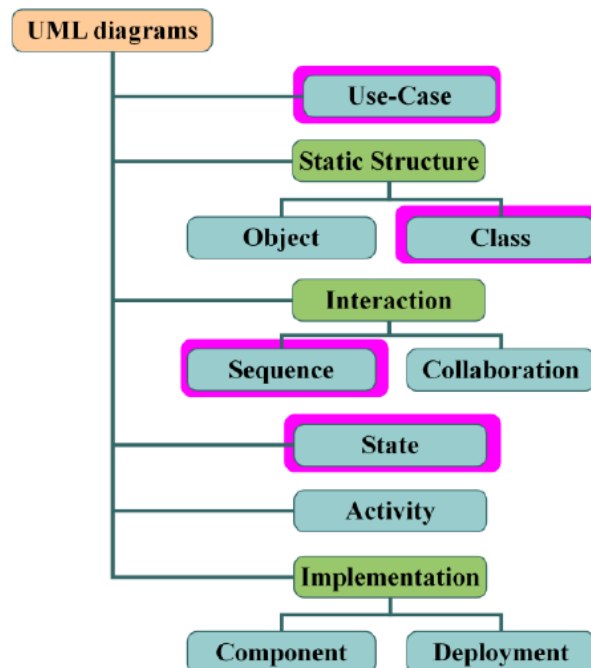
Kemudian pada Januari 1997 IBM, *ObjectTime*, *Platinum Technology*, *Ptech*, *Taskon*, *Reich Technologies* dan *Softeam* juga menanggapi *Request for Proposal (RFP)* yang dikeluarkan oleh *OMG* tersebut dan menyatakan kesediaan untuk bergabung.

Perusahaan-perusahaan ini menyumbangkan ide-ide mereka, dan bersama para mitra menghasilkan UML revisi 1.1. Fokus dari UML versi rilis 1.1 ini adalah untuk meningkatkan kejelasan UML Semantik versi rilis 1.0. Hingga saat ini UML versi terbaru adalah versi 2.0 (<http://www.uml.org/>).

Saat ini sebagian besar para perancang sistem informasi dalam menggambarkan informasi dengan memanfaatkan UML diagram dengan tujuan utama untuk membantu tim proyek berkomunikasi, mengeksplorasi potensi desain, dan memvalidasi desain arsitektur perangkat lunak atau pembuat program.

Secara filosofi UML diilhami oleh konsep yang telah ada yaitu konsep permodelan *Object Oriented* karena konsep ini menganalogikan sistem seperti kehidupan nyata yang didominasi oleh obyek dan digambarkan atau dinotasikan dalam simbol-simbol yang cukup spesifik. (Haviluddin , Jurnal Informatika Muawarman, VOL :6 , No.1 , Februari : 2011, 1-2).

Berikut gambar dari diagram UML



Gambar II.3 Diagram UML

Sumber : (Jurnal Informatika Muawarman, VOL :6 , No.1 , Februari : 2011;2)

Pemodelan adalah gambaran dari realita yang simpel dan dituangkan dalam bentuk pemetaan dengan aturan tertentu. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). *UML* hanya bergungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek. (Rosa A.S, M. Shalahuddin; 2011: 116-118).

Diagram *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori yaitu :

1. Structure Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Yang termasuk dalam structure diagram adalah sebagai berikut:

- a. Class diagram*
- b. Object diagram*
- c. Component diagram*
- d. Composite diagram*
- e. Package diagram*
- f. Deployment diagram*

2. Behavior Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Yang termasuk dalam *behavior* diagram adalah sebagai berikut :

- a. Use case*
- b. Activity diagram*
- c. State machine diagram*

3. Interaction Diagram

Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar sub sistem pada suatu sistem. Yang termasuk dalam *interaction* diagram adalah sebagai berikut :

- a. Sequence diagram*
- b. Communication diagram*

c. *Timing diagram*

d. *Interaction overview diagram* (Rosa A.S, M. Shalahuddin; 2011; 120-121)

II.11. *Use case*

Dalam bukunya Rekayasa Perangkat Lunak Rosa A.S – M. Shalahuddin mengatakan *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan apa saja yang berhak menggunakan fungsi-fungsi itu.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut actor dan *use case*.


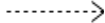

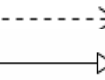

1. Aktor merupakan prang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari actor adalah gambar orang, tapi *actor* belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. (Rosa A.S – M. Shalahuddin; Rekayasa Perangkat Lunak; 130-131)

Untuk menggambarannya dalam *use case* model biasanya digunakan *association relationship* yang memiliki *stereotype include*, *extend* atau

generalization relationship. Hubungan *include* menggambarkan bahwa suatu *use case* seluruhnya meliputi fungsionalitas dari *use case* lainnya.

Hubungan *extend* antar *use case* berarti bahwa satu *use case* merupakan tambahan fungsionalitas dari *use case* yang lain jika kondisi atau syarat tertentu terpenuhi, berikut adalah simbol *use case*.

Tabel II.2 Simbol *Use case*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal <i>frase</i> nama <i>use case</i>
2		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
3		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum dan khusus) antar dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
4		<i>Include / uses</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalkannya <i>use case</i> ini.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Sumber: Rosa A.S, M. Shalahuddin; 2011: 131-132

II.12. *Class Diagram*

Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

Class Diagram adalah menggambarkan struktur statis dari kelas dalam sistem anda dan menggambarkan atribut, operasi dan hubungan antara kelas (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan *Class Diagram* merupakan sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

- Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas.
- Atribut mendeskripsikan properti dengan sebaris teks didalam kontak kelas tersebut.
- Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatukelas

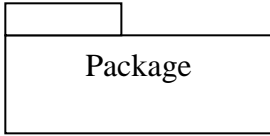
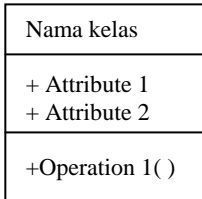


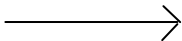
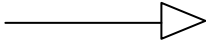
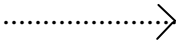
Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan setatis yang terdapat di antara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.

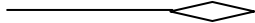
Diagram kelas menggambarkan struktur dan deskripsi class, package dan object beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi dan lain-lain.

Kelas memiliki tiga area pokok :

1. Nama
2. Atribut
3. Operasi

Tabel II.3. Simbol – Simbol *Class Diagram*

Simbol	Deskripsi
Package 	Package merupakan sebuah bungkus dari satu atau lebih kelas
Operasi 	Kelas pada struktur sistem
Antarmuka / interface 	Sama dengan konsep interface dalam pemograman berorientasi object
Asosiasi 	Relasi antara kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah / directed asosiasi 	Relasi antara kelas dengan makna kelas satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antara kelas dengan makna generalisasi-spesialisasi (umum Khusus)
Kebergantungan / defedency 	Relasi antara kelas dengan makna keberuntungan antara kelas

Agregasi 	Relasi antara kelas dengan maknasemua-bagian (whole-part)
---	--

(Sumber :Yuni Sugiarti, S.T.M.Kom, 2013,Hal 58-59)

II.13.. *Sequence Diagram*

Diagram sekuenes menggambarkan kelakuan/perilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuenes maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuenes yang harus digambar adalah sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuenes sehingga semakin banyak use case yang didefinisikan maka diagram sekuenes yang harus dibuat juga semakin banyak (M.shalahuddin 2013, Hal 165).

Sequence Diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram (Mulawarman, Vol 6, 2011).

Menurut para ahli diatas, dapat disimpulkan *Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah sekenario. Diagram

ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *use case*.

II.14. Activity Diagram


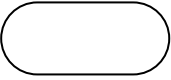



Activity Diagram (diagram aktivitas) merupakan diagram *flowchart* yang disempurnakan. Diagram aktivitas menggambarkan operasi pada suatu Obyek atau proses pada sebuah organisasi. Kelebihan diagram aktivitas dibandingkan dengan diagram *flowchart* adalah adanya dukungan konkurensi (pelaksanaan aktivitas secara bersamaan), pengiriman pesan dan *swimlane* (pelaku/penanggung jawab aktivitas).

Diagram aktivitas sangat berguna ketika ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi.

Diagram aktivitas diawali dengan lingkaran hitam, dan diakhiri dengan lingkaran hitam bertepi putih. Aktivitas digambarkan dengan kotak persegi panjang bersudut lengkung. Setiap aktivitas dihubungkan dengan panah dari awal hingga akhir diagram aktivitas.

Sama halnya dengan diagram *flowchart*, diagram aktivitas pun memiliki simbol yang sama untuk menggambarkan keputusan. Keputusan digambarkan dengan intan, namun di akhir kondisi yang menyertai keputusan diletakkan diluar simbol intan. Diagram aktivitas dapat menggambarkan konkurensi, yaitu satu atau lebih aktivitas yang berjalan secara bersamaan. Konkurensi diawali dengan sebuah garis tebal horizontal yang menjadi tempat keluarnya garis aktivitas. Konkurensi juga diakhiri dengan garis tebal horizontal.

Tabel II.1 Simbol-Simbol Diagram Aktivitas

Simbol	Arti	Keterangan
	Status Awal	Status awal aktivitas sistem sebuah diagram aktivitas memiliki sebuah status awal
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>Decision</i>	Decision, atau pilihan untuk mengambil keputusan.
	Penggabungan / <i>Join</i>	Arah tanda panah alur proses.
	Status Akhir	Titik Akhir atau akhir dari aktivitas.

Sumber: Rosa A.S, M. Shalahuddin; 2011: 134

II.15. Bahasa Pemograman PHP dan HTML

PHP (Hypertext Preprocessor) adalah bahasa pemrograman yang berfungsi untuk membuat website dinamis maupun aplikasi website . PHP bisa berinteraksi dengan database, file dan folder, sehingga membuat PHP bisa menampilkan konten yang dinamis dari sebuah website.

HTML (Hypertext Markup Language) adalah bahasa program yang digunakan untuk menulis format dokumen yang dapat digunakan dalam website. Hypertext markup language (HTML) merupakan bahasa komputer yang digunakan untuk menempatkan teks, gambar, animasi, video dan suara ke halaman web individu yang membentuk situs. (Jurnal Online Institut Teknologi Nasional ; No.01 , Vol.03 ; Januari 2015)

II.16. *Sql Server 2008*

Sql Server 2008 merupakan sebuah terobosan terbaru dari *Microsoft* dalam bidang *database*. *Sql Server* adalah sebuah *DBMS (Database Management System)* yang dibuat oleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahuluannya 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. (Wahana Komputer;2010; 2)