

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

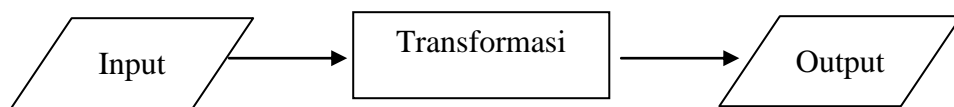
Menurut Prof. Dr. Ir. Marimin, M.Sc; 2011:1. Sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antar bagian, ini menunjukkan kompleksitas dari sistem yang meliputi kerja sama antara bagian yang interdependen satu sama lain. Selain itu, dapat dilihat bahwa sistem berusaha mencapai tujuan. Pencapaian tujuan ini menyebabkan timbulnya dinamika, perubahan yang terus menerus perlu dikembangkan dan dikendalikan. Definisi tersebut menunjukkan bahwa sistem sebagai gugus dari elemen – elemen yang saling berinteraksi secara teratur dalam rangka mencapai tujuan atau subtujuan.

Sifat – sifat dasar suatu sistem, antara lain :

1. Pencapaian tujuan, orientasi pencapaian tujuan akan memberikan sifat dinamis kepada sistem, memberi siri perubahan yang terus – menerus dalam usaha mencapai tujuan.
2. Kesatuan usaha, mencerminkan suatu sifat dasar dari sistem, dimana hasil keseluruhan melebihi dari jumlah bagian – bagiannya atau sering disebut konsep sinergi.
3. Keterbukaan terhadap lingkungan, lingkungan merupakan sumber kesempatan maupun hambatan pengembangan. Keterbukaan terhadap lingkungan membuat penilaian terhadap suatu sistem menjadi relatif atau

yang dinamakan *equifinality* atau pencapaian tujuan suatu sistem tidak mutlak harus dilakukan dengan satu cara terbaik. Tetapi pencapaian tujuan suatu sistem dapat dilakukan melalui berbagai cara sesuai dengan tantangan lingkungan yang dihadapi.

4. Transformasi, merupakan proses perubahan input menjadi output yang dilakukan oleh sistem. Proses transformasi diilustrasikan pada gambar II.1.



Gambar II.1. Proses Transformasi Input Menjadi Output.

(Sumber : Prof. Dr. Ir. Marimin, M.Sc. ; 2011 : 2)

5. Hubungan antara bagian, kaitan antara subsistem inilah yang akan memberikan analisis sistem, suatu dasar pemahaman yang lebih luas.
6. Sistem ada berbagai macam, antara lain sistem terbuka, sistem tertutup, dan sistem dengan umpan balik.
7. Mekanisme pengendalian, mekanisme ini menyangkut sistem umpan balik yang merupakan suatu bagian yang memberi informasi kepada sistem mengenai efek dari perilaku sistem terhadap pencapaian tujuan atau pemecahan persoalan yang dihadapi. Skema proses transformasi sistem dengan mekanisme pengendalian disajikan pada gambar II.2.



Gambar II.2. Skema Proses Transformasi Sistem Dengan Mekanisme Pengendalian.

(Sumber : Prof. Dr. Ir. Marimin, M.Sc. ; 2011 : 3)

II.1.1. Karakteristik Sistem

Menurut Hanif Al Fatta; 2010:5. Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur – unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya :

a. Batasan (*Boundary*)

Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.

b. Lingkungan (*Environment*)

Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.

c. Masukan (*Input*)

Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanupulasi oleh suatu sistem.

d. Keluaran (*Output*)

Sumber daya atau produk (informasi, laporan, dokumen, tampilan layer computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

e. Komponen (*Component*)

Kegiatan – kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*).

Komponen ini bisa merupakan subsistem dari sebuah sistem.

f. Penghubung (*Interface*)

Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.

g. Penyimpanan (*Storage*)

Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga diantara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

II.2. Pengertian Sistem Pendukung Keputusan

Menurut Magdalena Karismariyanti; 2011:55. Sejarah evolusi sistem pendukung keputusan dimulai pada tahun 1965, yang dibutuhkan oleh industri untuk menyimpan data dan menggabungkan ide, orang, sistem dan teknologi. Pada masa itu dimulai pembangunan *mainframe IBM system 360* untuk mendukung terciptanya *Management Information System (MIS)* yang menitikberatkan pada

Dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ;
 $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i)
 diberikan sebagai berikut :

$$V_i = \sum_{J=1}^n w_j r_{ij} \dots\dots\dots (2)$$

V_i = nilai preferensi

W_j = bobot ranking

r_{ij} = rating kinerja ternormalisasi

Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

Langkah – langkah dari metode SAW adalah :

1. Menentukan kriteria – kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (C), 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 peranking yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A) sebagai solusi.

II.4. Entity Relationship Diagram (ERD)

Menurut Wahana Komputer; 2010:30. Pada dasarnya ERD (*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaan ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen – komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database entity memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang dikenal dalam entity yaitu atribut yang berperan sebagai kunci primer dan atribut deskriptif. Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota – anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat – sifat sebagai berikut :

- Atomic, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contoh dari sifat atomik adalah field status dari tabel karyawan yang hanya berisi menikah atau single.
- multivalued, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap entity tertentu. Misalnya adalah field hobi, hobi dari tiap

karyawan mungkin dan hampir pasti lebih dari satu. Misalnya karyawan A memiliki hobi : membaca, nonton TV dan bersepeda.

- Composite, atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik. Contohnya adalah atribut alamat yang dapat dipecah menjadi atribut atomik berupa alamat, kode pos, no telepon, dan kota.

II.5. Normalisasi

Menurut Uus Rusmawan; 2010:92-93. Normalisasi *file* adalah suatu proses pengelompokan elemen data ke dalam tabel yang menyatakan hubungan antara entitas sehingga terwujud suatu bentuk yang memudahkan adanya perubahan dengan dampak terkecil. Proses pada pengolahan data yang mengakibatkan efek samping yang tidak diharapkan sering disebut dengan istilah anomali yang dapat terjadi akibat inserting, updating, atau deleting. Oleh karena itulah diperlukan normalisasi file. Adapun teknik penyusunan normalisasi file adalah dengan menentukan hal – hal sebagai berikut :

- Kunci Tribut

Setiap *file* selalu mempunya kunci berupa satu *set field* yang dapat mewakili *record*. Sebagai contoh, pada tabel barang terdapat *field* kunci berupa kode barang yang mewakili nama barang, harga barang, jumlah barang dan sebagainya.

- Kunci Kandidat (*Candidate Key*)

Kunci kandidat adalah suatu atribut atau set minimal atribut yang mengidentifikasi secara unik suatu kejadian spesifik dari entiti. Jika kunci

kandidat berisi lebih dari satu atribut, maka biasanya disebut sebagai kunci gabungan (*Composite Key*).

- Kunci Primer (*Primary Key*)

Kunci primer adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu spesifik, tetapi juga dapat mewakili setiap kejadian dari suatu entiti. Setiap kunci kandidat mempunyai peluang untuk menjadi kunci primer, tetapi sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap entiti yang ada. Contohnya adalah NIM, sifatnya unik dan tidak mungkin ganda dan mewakili secara menyeluruh terhadap entiti mahasiswa dan setiap mahasiswa selalu memiliki NIM. Selain itu kita harus melihat juga fungsi No KTP, ini dapat digunakan bila sampai suatu saat mahasiswa harus memiliki Kartu Tanda Mahasiswa tetapi NIM belum keluar, maka No KTP dapat digunakan.

- Kunci Alternatif (*Alternate Key*)

Kunci alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci primer. Biasanya kunci ini digunakan sebagai kunci pengurutan data saja, misalnya dalam pembuatan laporan.

- Kunci Tamu (*Foreign Key*)

Kunci tamu adalah satu atribut (atau satu set atribut) yang melengkapi satu hubungan yang menunjukkan ke tabel induknya. Kunci tamu biasanya ditempatkan pada entiti anak dan sama dengan kunci primer induk yang

direlaskan. Hubungan antara entiti induk dengan anak adalah (biasanya) hubungan satu ke banyak (*one to many*).

II.5.1. Bentuk-bentuk Normalisasi

a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

b. Bentuk normal tahap pertama (1st Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

Berikut ini akan dicontohkan normalisasi dari tabel kuliah yang memiliki atribut : kode_kul, nama_kul, sks, semester, waktu, tempat, dan nama_dos.

Tabel kuliah tersebut tidak memenuhi normalisasi pertama, karena terdapat atribut waktu yang tergolong ke dalam atribut bernilai banyak. Agar tabel tersebut dapat memenuhi 1NF, maka solusinya adalah dengan mendekomposisi tabel kuliah menjadi :

- Tabel kuliah (kode_kul, nama_kul, sks, semester, nama_dos).

- Tabel jadwal (kode_kul, waktu, ruang).

c. Bentuk normal tahap kedua (2nd normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

Sebuah tabel dikatakan tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari primary key). Bentuk normal kedua akan dicontohkan berikut.

Misal tabel nilai terdiri dari atribut kode_kul, nim dan nilai. Jika pada tabel nilai. Misalnya kita tambahkan sebuah atribut yang bersifat redundan, yaitu nama_mhs, maka tabel nilai ini dianggap melanggar 2NF.

Primary key pada tabel nilai adalah (kode_kul, nim).

Penambahan atribut baru (nama_mhs) akan menyebabkan adanya ketergantungan fungsional yang baru yaitu $\text{nim} > \text{nama_mhs}$. Karena atribut nama_mhs ini hanya memiliki ketergantungan parsial pada primary key secara utuh (hanya tergantung pada nim, padahal nim hanya bagian dari primary key). Bentuk normal kedua ini dianggap belum memadai karena meninjau sifat ketergantungan atribut terhadap atribut terhadap primary key saja.

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

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah superkey pada tabel tersebut.
- Atau A merupakan bagian dari primary key pada tabel tersebut.

Misalkan pada tabel mahasiswa, atribut alamat_mhs dipecah kedalam alamat_jalan, alamat_kota dan kode_pos. Bentuk ini tidak memenuhi 3NF, karena terdapat ketergantungan fungsional baru yang muncul pada tabel tersebut, yaitu :

alamat_jalan, nama_kota – kode_pos

Dalam hal ini (alamat_jalan, nama_kota) bukan superkey sementara kode_pos juga bukan bagian dari primary key pada tabel mahasiswa. Jika tabel mahasiswa didekomposisi menjadi tabel mahasiswa dan tabel alamat, maka telah memenuhi 3NF. Hal itu dapat dibuktikan dengan memeriksa dua ketergantungan fungsional pada tabel alamat tersebut, yaitu :

alamat_jalan, nama_kota – kode_pos

kode_pos – nama_kota

Ketergantungan fungsional yang pertama tidak melanggar 3NF, karena (alamat_jalan, nama_kota) merupakan superkey (sekali sebagai primary key) dari tabel alamat tersebut. Demikian juga dengan ketergantungan fungsional yang kedua meskipun (kode_pos) bukan merupakan superkey, tetapi nama_kota merupakan bagian dari primary key dari tabel alamat. Karena telah memenuhi 3NF, maka tabel tersebut tidak perlu di-dekomposisi lagi.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1st NF
- Relasi harus bergantung fungsi pada atribut superkey (Kusrini, M.Kom ; 2010 : 41-43).

II.6. Pengertian Database

Menurut Yuhefizard; 2010:2. Banyak sekali definisi tentang database yang diberikan oleh para pakar dibidang ini. Database terdiri dari dua penggalan kata yaitu data dan base, yang artinya berbasiskan pada data. Tetapi secara konseptual, database diartikan sebuah koleksi atau kumpulan data yang saling berhubungan (relation), disusun menurut aturan tertentu secara logis, sehingga menghasilkan informasi. Sebuah informasi yang berdiri sendiri tidaklah dikatakan database.

Contoh : Nomor telepon seorang pelanggan, disimpan dalam banyak tempat apakah itu di file pelanggan, di file alamat dan di lokasi yang lain. Antara file yang satu dengan file yang lainnya tidak saling berhubungan, sehingga apabila salah seorang pelanggan berganti nomor telepon dan anda hanya mengganti di file

pelanggan saja, akibatnya akan terjadi ketidakcocokan data, karena di lokasi yang lain masih tersimpan data telepon yang lama.

Dalam sistem database hal ini tidak boleh dan tidak bisa terjadi, karena antara file yang satu dengan file yang lain saling berhubungan. Jika suatu data yang sama anda ubah, data tersebut di file yang lain akan otomatis berubah juga. Sehingga mampu menjadi informasi yang diinginkan dan dapat dilakukan proses pengambilan, penghapusan, pengeditan, terhadap data secara mudah dan cepat (Efektif, Efisien dan Akurat).

Data adalah fakta, baik berupa sebuah objek, orang dan lain – lain yang dapat dinyatakan dengan suatu nilai tertentu (angka, simbol, karakter tertentu, dan lain – lain). Sedangkan informasi adalah data yang telah diolah sehingga bernilai guna dan dapat dijadikan bahan dalam pengambilan keputusan.

Hubungan data dan informasi dapat digambarkan sebagai berikut :



Gambar II.3. Data dan Informasi.

(Sumber : Yuhefizard ; 2010 : 2)

II.7. Mengenal Visual Basic

Menurut Edy Winarto; 2010:1. Visual Basic dibuat oleh microsoft, merupakan salah satu bahasa pemrograman berorientasi objek yang mudah dipelajari. Selain menawarkan kemudahan, Visual Basic juga cukup andal untuk digunakan dalam pembuatan berbagai aplikasi, terutama aplikasi database. Visual basic merupakan bahasa pemrograman event drive, dimana program akan menunggu sampai ada respons dari user/pemakai program aplikasi yang dapat

berupa kejadian atau event, misalnya ketika user mengklik tombol atau menekan enter. Jika kita membuat aplikasi dengan visual basic maka kita akan mendapatkan file yang menyusun aplikasi tersebut, yaitu :

1. File Project (*.vbp)

File ini merupakan kumpulan dari aplikasi yang kita buat. File project bisa berupa file *.frm, *.dsr atau file lainnya.

2. File Form (*.frm)

File ini merupakan file yang berfungsi untuk menyimpan informasi tentang bentuk form maupun interface yang kita buat.

II.8. SQL Server

Menurut Cybertron Solution; 2010:101. *SQL Server 2008* adalah sebuah RDBMS (*Relational Database Management System*) yang di *develop* oleh *Microsoft*, yang digunakan untuk menyimpan dan mengolah data. Pada *SQL Server 2008*, kita bisa melakukan pengambilan dan modifikasi data yang ada dengan cepat dan efisien. Pada *SQL Server 2008*, kita bisa membuat *object – object* yang sering digunakan pada aplikasi bisnis, seperti membuat *database, table, fuction, stored procedure, trigger* dan *view*. Selain *object*, kita juga menjalankan perintah SQL (*Structured Query Language*) untuk mengambil data.

II.9. 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 industry perangkat lunak dan pengembangan sistem. Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

- *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. (Windu Gata ; 2013 : 4).

- *Diagram Aktivitas (Activity Diagram)*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. (Windu Gata ; 2013 : 6).

- *Diagram Urutan (Sequence Diagram)*

Sequence diagram menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. (Windu Gata ; 2013 : 7).

- *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 tanggungjawab 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. (Windu Gata ; 2013 : 8).