

BAB II

TINJAUAN PUSTAKA

II.1. Pengertian Sistem

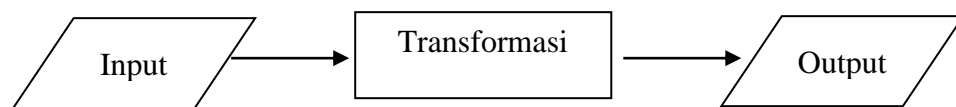
Sistem adalah suatu kesatuan usaha yang terdiri dari bagian – bagian yang berkaitan satu sama lain yang berusaha mencapai suatu tujuan dalam suatu lingkungan kompleks. Pengertian tersebut mencerminkan adanya beberapa bagian dan hubungan antar bagian, ini menunjukkan kompleksitas dari sistem yang meliputi 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. (Prof. Dr. Ir. Marimin, M.Sc; 2011:1)



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

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

II.1.1. Karakteristik Sistem

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. (Hanif Al Fatta; 2010:5)

II.2. Pengertian Sistem Pendukung Keputusan

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 fasilitas kepada manajer dalam bentuk laporan yang terstruktur dan periodik seperti laporan keuangan dan laporan transaksi.

Sebuah penelitian tentang implementasi sistem pendukung keputusan *model-driven decision support system* dipublikasikan dalam jurnal bisnis pada tahun 1970-an. Penelitian ini menjadi pionir bagaimana komputer dan model

dalam sistem pendukung keputusan terus dikembangkan pada tahun 1980-an dan diikuti oleh perkembangan *knowledge-oriented Decision Support System* (DSS). Awal 1990-an, sistem pendukung keputusan dibangun menggunakan teknologi basis data relasional. Sejak dikenalkannya teknologi web sistem pendukung keputusan ini berkembang menjadi *Web-based DSS*. (Magdalena Karismariyanti; 2011:55).

Sistem pendukung keputusan merupakan suatu sistem interaktif yang mendukung keputusan dalam proses pengambilan keputusan melalui alternatif – alternatif yang diperoleh dari hasil pengolahan data, informasi dan rancangan model. Sistem pendukung keputusan (*Decision Support System = DSS*) adalah suatu sistem informasi yang menggunakan model – model keputusan, basis data, dan pemikiran manajer sendiri, proses modeling interaktif dengan komputer untuk mencapai pengambilan keputusan oleh manajer tertentu.

Menurut Keen dan Scoot Morton, sistem pendukung keputusan merupakan penggabungan sumber – sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem pendukung keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah – masalah semi struktur. Menurut Alter, DSS merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem digunakan untuk membantu mengambil keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tidak seorangpun mengetahui secara pasti bagaimana keputusan seharusnya dibuat.

Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan yang melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan. (Asep Abdul Wahid; 2012:2)

II.3. Pengertian Metode TOPSIS

TOPSIS adalah salah satu metode pengambilan keputusan multikriteria (MCDS) yang pertama kali diperkenalkan oleh Yonn dan Hwang pada tahun 1981. Dengan ide dasarnya adalah bahwa alternatif yang dipilih memiliki jarak terdekat dengan solusi ideal dan yang terjauh dari solusi ideal negatif. TOPSIS memperhatikan baik jarak ke solusi ideal positif maupun jarak ke solusi ideal negatif dengan mengambil hubungan kedekatan menuju solusi ideal. Dengan melakukan perbandingan pada keduanya, urutan pilihan dapat ditentukan.

Metode TOPSIS banyak digunakan pada beberapa model MADM (*Multi Attribute Decision Making*) dikarenakan metode ini memiliki beberapa keunggulan, yaitu :

1. Konsepnya sederhana dan mudah dipahami.
2. Komputasinya efisien.
3. Memiliki kemampuan untuk mengukur kinerja relatif dari alternatif – alternatif keputusan dalam bentuk matematis yang sederhana.

Berikut ini adalah matriks D yang memiliki m alternatif dengan n kriteria, dimana X_{ij} adalah pengukuran pilihan dari alternatif ke- i dalam hubungannya dengan kriteria ke- j .

$$D = \begin{pmatrix} X_{11} & X_{12} & X_{13} & \dots & X_{1m} \\ X_{21} & X_{22} & X_{23} & \dots & X_{2m} \\ \vdots & & & & \\ X_{m1} & X_{m2} & X_{m3} & \dots & X_{mn} \end{pmatrix}$$

Langkah –langkah yang dilakukan dalam penyelesaian masalah menggunakan metode TOPSIS adalah sebagai berikut :

1. Normalisasi Matrix Keputusan

Dalam prosedur ini, setiap nilai atribut diubah menjadi nilai yang *comparable*. Setiap normalisasi dari nilai r_{ij} dapat dilakukan dengan perhitungan :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}}, \quad \begin{array}{l} \text{untuk } i = 1,2,3,\dots, m \\ j = 1,2,3,\dots, n \end{array}$$

Sehingga didapat matriks R hasil normalisasi,

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & & & \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{pmatrix}$$

R adalah matriks yang telah di normalisasi, dimana m menyatakan alternatif, n menyatakan kriteria dan r_{ij} adalah normalisasi pengukuran pilihan dari alternatif ke- i dalam hubungannya dengan kriteria ke- j .

2. Pembobotan Pada Matriks Yang Telah Di Normalisasikan

Setelah di normalisasikan, setiap kolom dari matriks R dikalikan dengan bobot – bobot (w_j) yang telah ditentukan oleh pembuat keputusan. Sehingga, *weighted normalized matrix* adalah $V = RW$

$$V = \begin{pmatrix} V_{11} & v_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ \vdots & & & \\ V_{m1} & V_{m2} & \dots & V_{mn} \end{pmatrix} =$$

$$RW = \begin{pmatrix} w_1 r_{11} & w_2 r_{12} & \dots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \dots & w_n r_{2n} \\ \vdots & & & \\ w_1 r_{m1} & w_2 r_{m2} & \dots & w_n r_{mn} \end{pmatrix}$$

Dimana W adalah

$$W = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & w_n \end{pmatrix}$$

3. Menentukan solusi ideal positif dan solusi ideal negatif solusi ideal dinotasikan dengan A^+ dan solusi ideal negatif dinotasikan dengan A^- , sebagai berikut :

$$A^+ = \{(\max v_{ij} | j \in J), (\min v_{ij} | j \in J'), i = 1, 2, 3, \dots, m\}$$

$$= \{v_1^+, v_2^+, \dots, v_n^+\}$$

$$A^- = \{(\min v_{ij} | j \in J), (\max v_{ij} | j \in J'), i = 1, 2, 3, \dots, m\}$$

$$= \{v_1^-, v_2^-, \dots, v_n^-\}$$

Dimana :

$$J = \{j = 1,2,3,\dots, n \text{ dan } j \text{ berhubungan dengan } \textit{benefit criteria}\}$$

$$J' = \{j = 1,2,3,\dots, n \text{ dan } j \text{ berhubungan dengan } \textit{cost criteria}\}$$

Pembangunan A^+ dan A^- adalah untuk mewakili alternatif yang *most preferable* ke solusi ideal dan yang *least preferable* secara berurutan.

4. Menghitung *Separation Measure*

Separation Measure ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal dan solusi ideal negatif, perhitungan matematisnya adalah sebagai berikut :

a. Solusi Ideal Positif

$$S_{i^+} = \sqrt{\sum_{i=1}^n (v_{ij} - v_{j^+})^2}, \text{ untuk } i = 1,2,3, \dots, m$$

b. Solusi Ideal Negatif

$$S_{i^-} = \sqrt{\sum_{i=1}^n (v_{ij} - v_{j^-})^2}, \text{ untuk } i = 1,2,3, \dots, m$$

5. Menghitung Kedekatan Relatif Dengan Solusi Ideal

Kedekatan relatif dari alternatif A_i dekat dengan solusi ideal A^+ direpresentasikan dengan :

$$C_{i^+} = \frac{S_{i^-}}{S_{i^+} + S_{i^-}}, \text{ dimana } 0 < C_{i^+} < 1 \text{ dan } i = 1,2,3, \dots, m$$

Dikatakan alternatif A_i dekat dengan solusi ideal apabila C_{i^+} mendekati 1 jadi, $C_{i^+} = 1$, jika $A_i = A^+$, dan $C_{i^-} = 0$, jika $A_i = A^-$.

6. Mengurutkan Pilihan

Pilihan akan diurutkan berdasarkan pada nilai C_i^+ , sehingga alternatif dengan jarak terpendek dengan solusi ideal adalah alternatif yang terbaik. (Radiant Victor Imbar; 2011:127: 130).

II.4. Entity Relationship Diagram (ERD)

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 :

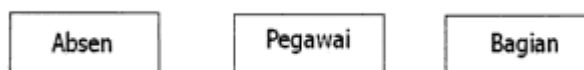
- Atomic, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi. Contoh dari sifat atomik adalah field status dari tabel karyawan yang hanya berisi menikah atau single.

- 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. (Wahana Komputer; 2010:30)

II.4.1. Langkah – langkah Desain Awal ERD

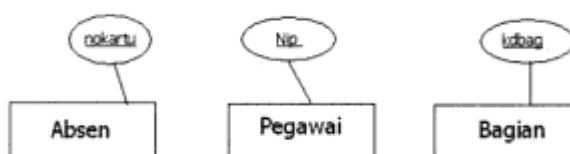
Sebagai contoh, langkah – langkah desain awal ERD sebagai berikut :

1. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang terlibat, seperti gambar berikut :



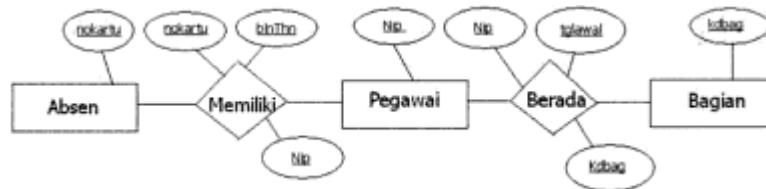
Gambar II.3. Entiti Yang Terlibat
(Sumber : Y. Supardi ; 2010 : 79)

2. Menentukan atribut – atribut key dari masing – masing himpunan entitas, seperti gambar berikut :



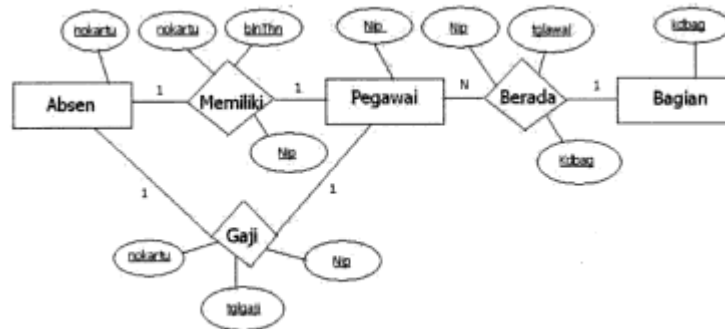
Gambar II.4. Key Masing-Masing Entiti Yang Terlibat
(Sumber : Y. Supardi ; 2010 : 79)

3. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas – entitas yang ada beserta kunci tamu (*foreign key*).



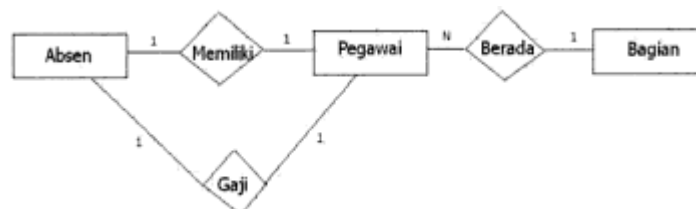
Gambar II.5. Kunci Tamu Entiti Dan Interface (Himpunan Relasi)
(Sumber : Y. Supardi ; 2010 : 79)

4. Menentukan derajat / kardinalitas setiap himpunan.



Gambar II.6. Derajat Kardinalitas
(Sumber : Y. Supardi ; 2010 : 80)

5. Menentukan himpunan entitas dan himpunan relasi dengan atribut – atribut deskriptif (bukan kunci), karena *field* terlalu banyak dapat menggunakan kamus data.



Gambar II.7. Atribut Bukan Kunci
(Sumber : Y. Supardi ; 2010 : 80)

II.5. Normalisasi

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 insert, update, atau delete. Oleh karena itulah diperlukan normalisasi file. Adapun teknik penyusunan normalisasi file adalah dengan menentukan hal – hal sebagai berikut :

- Kunci Tribut

Setiap *file* selalu mempunyai 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 direlasikan. Hubungan antara entiti induk dengan anak adalah (biasanya) hubungan satu ke banyak (*one to many*). (Uus Rusmawan; 2010:92)

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.5.2. Contoh Kasus

Berikut akan diberikan contoh proses perancangan *database* dengan teknik normalisasi.

Dokumen dasar :

Faktur Pembelian Barang				
PT. Ayo Bersama Jl. Senopati 12 yk				
Kode Suplier : G01			Tanggal : 07/10/04	
Nama Suplier : Gober Nustra			Nomor : 004	
Kode	Nama Brg	Qty	Harga	Jml
A01	AC model 1	10	1000000	10000000
A02	AC model 2	5	900000	4500000
			Total Faktur	14500000
Jatuh Tempo 15/10/04				

Faktur Pembelian Barang				
PT. Ayo Bersama Jl. Senopati 12 yk				
Kode Suplier : A03			Tanggal : 09/10/04	
Nama Suplier : Angkasa			Nomor : 006	
Kode	Nama Brg	Qty	Harga	Jml
B01	Kursi	5	100000	500000
A03	Meja	7	250000	3250000
			Total Faktur	3750000
Jatuh Tempo 17/10/04				

Bentuk tidak ternormalisasi :

Kd. Fak	Kode Sup	Nama Sup	Kode Brg	Nama brg	Tgl	...
004	G01	Gobel Nustra	A01, A02	AC Model 1, AC Model 2	07/10/04	...
006	A03	Angkasa	B01, A03	Meja, Kursi	09/10/05	...

...	J.tempo	qty	Hrg	Jml	Tot
...	15/10/04	10,5	1000000, 900000	10000000, 4500000	14500000
...	17/10/04	5,7	100000, 250000	500000, 3250000	3750000

Bentuk normal 1 :

Kd. Fak	Kode Sup	Nama Sup	Kode Brg	Nama brg	...
004	G01	Gobel Nustra	A01	AC Model 1	...
004	G01	Gobel Nustra	A02	AC Model 2	...
006	A03	Angkasa	B01	Meja	
006	A03	Angkasa	A03	Kursi	

...	Tgl	J.tempo	Qty	Hrg
...	07/10/04	15/10/04	10	1000000
...	07/10/04	15/10/04	5	900000
	09/10/05	17/10/04	5	100000
	09/10/05	17/10/04	7	250000

Kelemahan :

- Insert : ingin memasukkan supplier tanpa transaksi
- Delete : menghapus transaksi → supplier terhapus
- Update : mengganti satu nama supplier → lainnya jadi tidak benar
- Redundancy : jumlah ← hasil perhitungan qty * harga

Bentuk 2nd NF :

KdFaktur → Tgl, Jtempo,

KodeSup, NamaSup

KdFaktur, Kodebrg → NamaBrg, Qty, Harga

Bentuk BCNF :

KdFaktur → Tgl, Jtempo, KodeSup

KdSup → NamaSup

KdFaktur, Kodebrg → Qty, Harga

KdBrg → NamaBrg. (Kusrini, M.Kom ; 2010 : 44).

II.6. Pengertian Database

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. (Yuhefizard; 2010:2)

Hubungan data dan informasi dapat digambarkan sebagai berikut :



Gambar II.8. Data dan Informasi.
(Sumber : Yuhefizard ; 2010 : 2)

II.7. Mengetahui Visual Basic

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. (Edy Winarto; 2010:1)

II.8. SQL Server

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. (Cybertron Solution; 2010:101)

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.

- Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

- Diagram Urutan (*Sequence Diagram*)

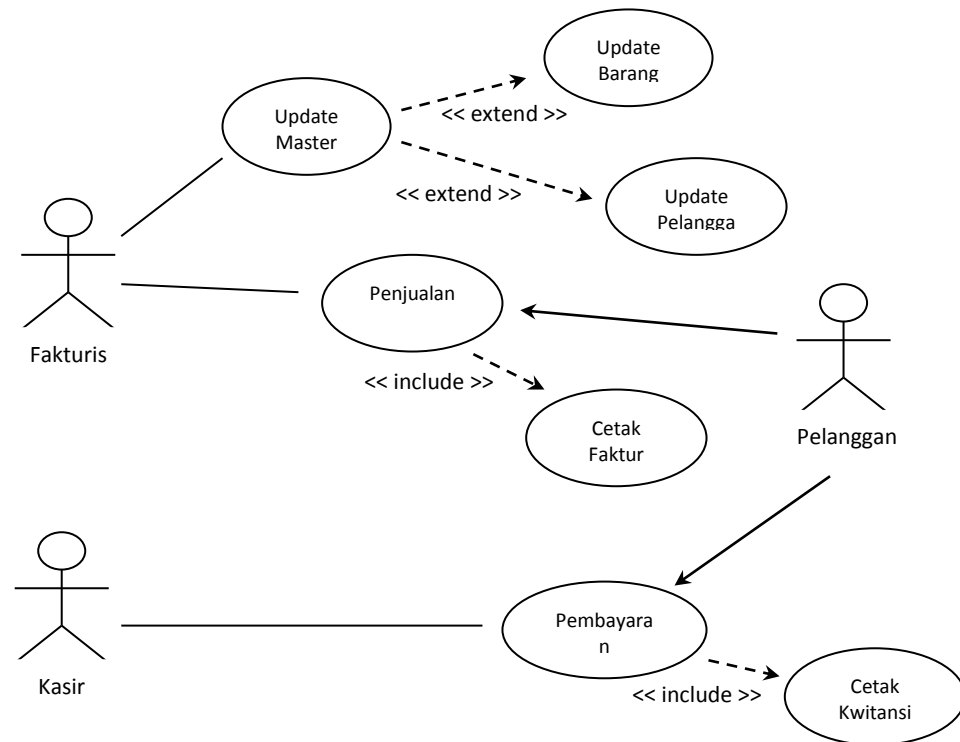
Sequence diagram menggambarkan kelakuan objek pada usecase dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

- *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.

II.9.1. Contoh Diagram *Unified Modeling Language* (UML)

1. Use Case Diagram



Gambar II.9. Use Case Penjualan

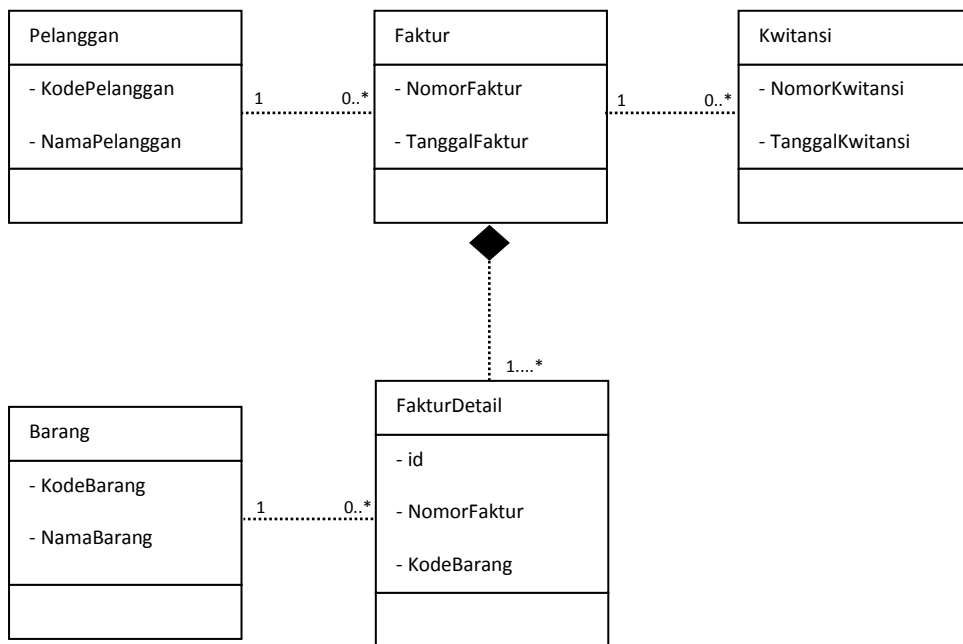
(Sumber : Windu Gata; 2013:13)

Untuk lebih jelasnya mengenai gambar tersebut adalah sebagai berikut :

- Terdapat 3 aktor, yaitu faktoris, pelanggan, dan kasir.
- Faktoris berhubungan secara langsung ke pada case update data, sehingga faktoris dapat melakukan pemuktahiran data barang dan pelanggan.
- Pelanggan (berhubungan tidak langsung) dapat meminta faktoris (yang melakukan operasi komputer) untuk melakukan pemasukan data penjualan sekaligus melakukan pencetakan faktur.

- Berdasarkan faktur yang didapat oleh pelanggan, maka pelanggan melakukan pembayaran ke kasir dan mendapat kwitansi yang dicetak oleh kasir.

2. Class Diagram



Gambar II.10. Diagram Class Penjualan

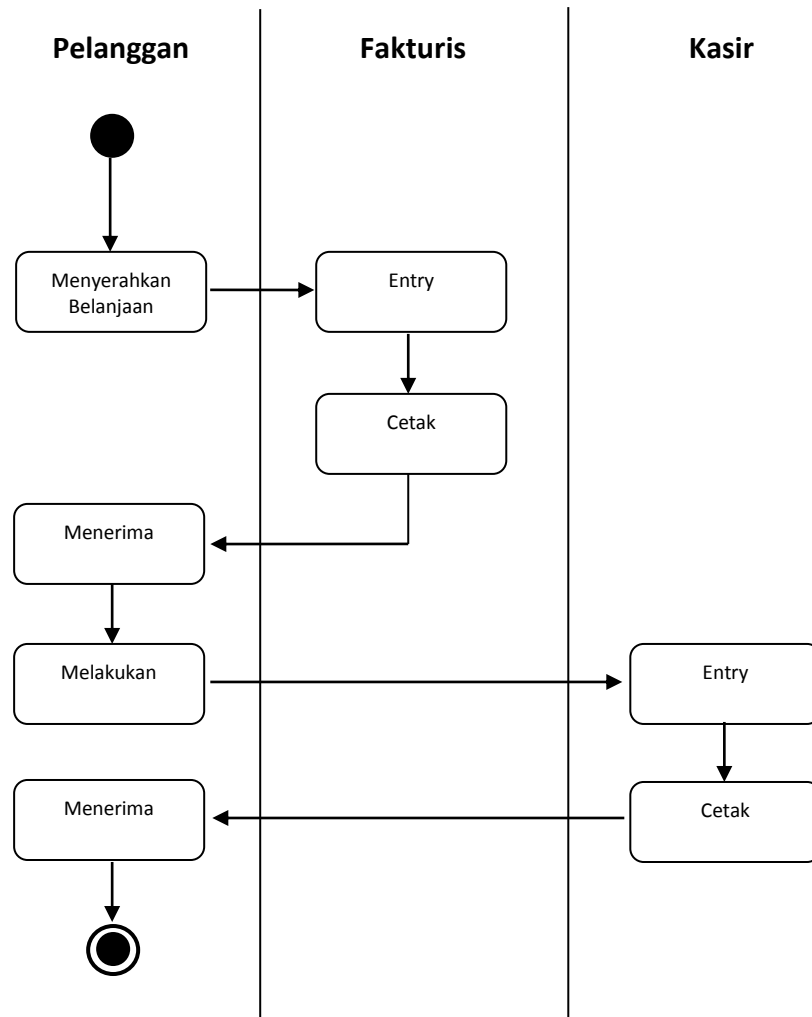
(Sumber : Windu Gata; 2013:14)

Penjelasan gambar tersebut diatas adalah :

- Pelanggan boleh tidak melakukan pembelian atau melakukan pembelian satu kali atau sebanyak mungkin. Sedangkan dalam faktur harus terisi kode pelanggan.
- Faktur tidak akan tercatat dalam kwitansi apabila belum melakukan pembayaran, tetapi akan tercatat pada kwitansi setelah melakukan pembayaran. Sedangkan dalam kwitansi haruslah tercatat NomorFaktur. Id pada FakturDetail merupakan kunci.

- FakturDetail merupakan *agregasi komposit* (faktur memiliki FakturDetail atau FakturDetail tidak akan ada kalau kelas faktur tidak ada), dan
- Dalam faktur *detail* haruslah tertera KodeBarang minimal satu atau lebih.

3. Activity Diagram



Gambar II.11. Diagram Aktivitas
(Sumber : Windu Gata; 2013:15)

Penjelasan gambar *sequence* diagram untuk entry faktur, yaitu :

- Kasir merupakan aktor yang berinteraksi langsung dengan tampilan layar dari *entry* faktur, yang pertamanya dilakukan adalah membuka *form* faktur yang kemudian melakukan *entry* kode pelanggan, dalam tampilan *form entry* tersebut akan dapat secara otomatis tampil kode pelanggan dengan cara membaca didalam entitas pelanggan, sehingga akan terkirim pesan dari kode pelanggan terakhir yang sudah dientri sebelumnya.
- Selanjutnya melakukan pemasukan data barang dalam *form* faktur dengan cara mengirimkan pesan ke kontrol barang dan mencari data barang di entitas barang yang kemudian mengirimkan pesan kembali sehingga tampil data barang yang diinginkan.
- Kemudian dimasukkan jumlah pembelian barang dan dilakukan penghitungan total pembelian barang.
- Setelah semua data *entry* faktur sudah dilakukan maka penyimpanan data faktur dalam entitas faktur dan detail faktur, serta program melakukan pengurangan stok.
- Pencetakan faktur dilakukan dengan menggunakan printer. (Windu Gata; 2013:17)

II.10. Pengertian Produksi

Istilah produksi dipergunakan dalam organisasi yang menghasilkan keluaran atau *output* berupa barang maupun jasa. Secara umum produksi diartikan sebagai suatu kegiatan atau proses yang mentransformasikan masukan (*input*) menjadi keluaran (*output*). Dalam pengertian umum inilah sekarang berkembang

istilah industri, seperti industri manufaktur, industri pengolahan hasil – hasil pertambangan, industri pariwisata, industri jasa keuangan, industri jasa perdangan, dan industri angkutan. Dalam arti sempit, pengertian produksi hanya dimaksudkan sebagai kegiatan yang menghasilkan barang, baik barang jadi atau setengah jadi, barang industri, suku cadang (*sparepart*) maupun komponen – komponen penunjang. Dengan pengertian ini, produksi dimaksudkan sebagai kegiatan pengolahan dalam pabrik. Hasil – hasil produksinya dapat berupa barang konsumsi maupun barang industri. Pengertian produksi dalam ekonomi mengacu pada kegiatan yang berhubungan dengan usaha penciptaan dan penambahan kegunaan atau utilitas suatu barang dan jasa. Penambahan atau penciptaan kegunaan atau utilitas karena bentuk dan tempat ini membutuhkan faktor – faktor produksi. (M. Fuad; 2011:142).

II.11. Pengenalan Kamus Data

Kamus data (*data dictionary*) mencakup definisi – definisi dari data yang disimpan didalam basis data dan dikendalikan oleh sistem manajemen basis data. Figur 6.5 menunjukkan hanya satu tabel dalam basis data jadwal. Struktur basis data yang dimuat dalam kamus data adalah kumpulan dari seluruh definisi *field*, definisi tabel, relasi tabel, dan hal – hal lainnya. Nama *field* data, jenis data (seperti teks atau angka atau tanggal), nilai – nilai yang valid untuk data, dan karakteristik – karakteristik lainnya akan disimpan dalam kamus data. Perubahan – perubahan pada struktur data hanya dilakukan satu kali didalam kamus data, program – program aplikasi yang mempergunakan data tidak akan ikut terpengaruh. (Raymond McLeod; 2010:171).