

BAB II

LANDASAN TEORI

II.1. Sistem Informasi

II.1.1. Sistem

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. tampak secara fisik, tetapi juga hal-hal yang mungkin bersifat abstrak atau konseptual, seperti misi, pekerjaan, kegiatan, kelompok informal, dan lain sebagainya. Unsur-unsur yang mewakili suatu sistem secara umum adalah masukan (*input*), pengolahan (*processing*) dan keluaran (*output*). (Tata Sutabri ; 2012 : 2)

Gordon B. Davis dalam bukunya menyatakan, bahwa sistem bisa berupa abstrak atau fisis. Sistem yang abstrak adalah susunan yang teratur dari gagasan-gagasan atau konsepsi yang saling bergantung. Misalnya, sistem teologi adalah susunan yang teratur dari gagasan-gagasan tentang Tuhan, Manusia dan lain sebagainya. Sedangkan sistem yang bersifat fisis adalah serangkaian unsur yang bekerjasama untuk mencapai suatu tujuan.

Sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut :

1. Komponen sistem(*Components*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem atau element-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli berapa pun kecilnya, selalu mengandung komponen atau subsistem.

2. Batasan sistem(*Boundary*)

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

3. Lingkungan luar sistem(*Environment*)

Lingkungan luar dari suatu sistem apapun diluar batas dari sistem yang mempengaruhi operasi dari sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem(*Interface*)

Penghubung merupakan media penghubung antara satu subsistem dengan subsistem lainnya, melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya.

5. Masukan sistem(*Input*)

Masukan sistem adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya

sistem tersebut dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran sistem(*Output*)

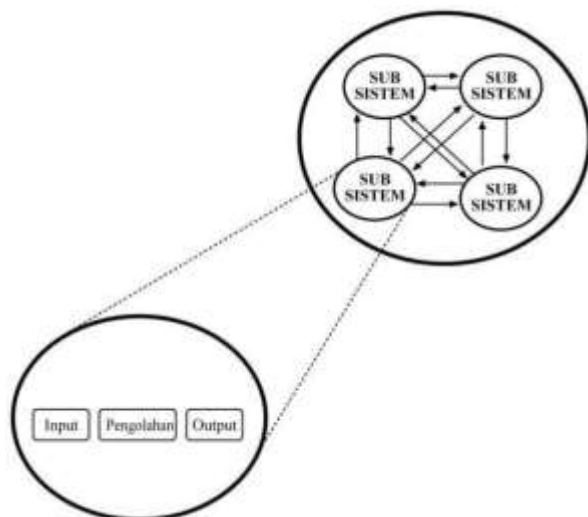
Keluaran sistem adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah sistem(*Proses*)

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

8. Sasaran sistem(*Objective*)

Suatu sistem mempunyai tujuan atau sasaran. Kalau suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak ada gunanya. Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem, suatu sistem dikatakan berhasil jika mengenai sasaran dan tujuan. (Tata Sutabri ; 2012 : 13-14)



Gambar II.1. Karakteristik Sistem

(Sumber : Tata Sutabri ; 2012 : 14)

II.1.2. Informasi

Informasi adalah data yang telah diklasifikasikan atau diolah atau diinterpretasikan untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi akan mengolah data menjadi informasi atau mengolah data dari bentuk tak berguna menjadi berguna bagi yang menerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan maka informasi tidak diperlukan. (Tata Sutabri ; 2012 : 22)

Sumber dari informasi adalah data. Data merupakan kenyataan yang menggambarkan suatu kejadian serta merupakan suatu kesatuan yang nyata, dan merupakan bentuk yang masih mentah sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi.

Fungsi utama informasi adalah menambah pengetahuan atau mengurangi ketidakpastian pemakai informasi. Informasi yang disampaikan kepada pemakai kepada pemakai mungkin merupakan hasil data yang dimasukkan kedalam pengolahan. Informasi juga berfungsi untuk memberikan dasar guna melakukan seleksi. Informasi tidak mengarah kepada yang harus dilakukan, tetapi mengurangi keanekaragaman dan ketidakpastian sehingga dapat dihasilkan keputusan yang baik. Fungsi informasi yang penting lainnya adalah memberikan standar-standar, aturan-aturan ukuran dan aturan-aturan keputusan untuk penentuan dan penyebaran tanda-tanda kesalahan dan umpan balik guna mencapai tujuan kontrol.

Informasi dapat berasal dari pengamatan, percakapan dengan orang lain, rapat-rapat panitia, dari majalah, dari media surat kabar atau laporan pemerintah

dan dari sistem informasi itu sendiri. Informasi diperoleh dari data yang ada, kemudian data tersebut diolah dengan menggunakan suatu proses tertentu.

Data diolah melalui suatu model informasi. Si penerima akan menerima informasi tersebut untuk membuat suatu keputusan dan melakukan tindakan yang akan mengakibatkan munculnya sejumlah data lagi. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya sehingga membentuk sebuah Siklus Informasi. (Tata Sutabri; 2012 : 24)



Gambar II.2. Siklus Informasi

(Sumber : Tata Sutabri ; 2012 : 26)

Kualitas dari suatu informasi tergantung dari 3 hal, yaitu informasi harus akurat (*accurate*), tepat waktu (*timelines*) dan relevan (*relevance*).

a. Akurat (*accurate*)

Informasi harus bebas dari kesalahan dan tidak bias atau menyesatkan.

Akurat juga berarti bahwa informasi harus jelas mencerminkan maksudnya.

Informasi harus akurat karena dari sumber informasi sampe ke penerima informasi mungkin banyak mengalami gangguan (noise) yang dapat mengubah atau merusak informasi tersebut.

b. Tepat waktu (*timelines*)

Informasi yang sampai kepada si penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi, karena informasi merupakan landasan di dalam pengambilan keputusan. Bila pengambilan keputusan terlambat maka dapat berakibat fatal bagi organisasi. Dewasa ini informasi dinilai mahal karena harus cepat 22 dikirim dan didapat sehingga memerlukan teknologi mutakhir untuk mendapatkannya, mengolah, dan mengirimkannya.

c. Relevan (*relevance*)

Informasi tersebut memiliki manfaat untuk pemakainnya. Relevansi informasi utuk setiap orang berbeda. Menyampaikan informasi tentang penyebab kerusakan mesin produksi kepada akuntan perusahaan tentu kurang relevan. Akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi disampaikan untuk ahli teknik merupakan informasi yang kurang relevan, tetapi akan sangat relevan untuk seorang akuntan perusahaan. (Tata Sutabri ; 2012 : 33)

II.1.3. Sistem Informasi

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi 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)

Berikut ini adalah defenisi sistem informasi menurut beberapa ahli:

Tabel II.1. Defenisi Sistem Informasi

Sumber	Defenisi
Alter (1992)	Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi
Bodnar Dan Hopwood (1993)	Sistem informasi adalah sekumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data kedalam bentuk informasi yang berguna
Gelinas, Oram Dan Wiggins (1990)	Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas kumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun menyimpan, dan mengelola data ke serta menyediakan informasi keluaran pada para pemakai
Hall (2001)	Sistem informasi adalah sebuah rangkaian prosedur

	formal dimana data dikelompokkan, diproses menjadi informasi dan didistribusikan kepada pemakai
Turban , Mclean dan Wetherbe (1999)	Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis dan menyebarkan informasi untuk tujuan spesifik
Wilkinson (1992)	Sebuah sistem informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran perusahaan

(Sumber: Abdul Kadir:2012:9)

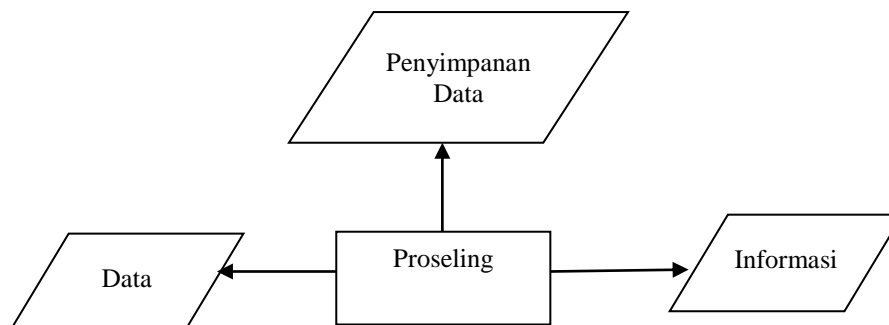
Berikut ini adalah contoh sistem informasi :

1. sistem reservasi pesawat terbang digunakan dalam biro perjalanan untuk melayani pemesanan/ pembelian tiket online.
2. sistem untuk menangani penjualan kredit kendaraan bermotor, sehingga dapat di gunakan untuk memantau piutang para pelanggan, dan sistem seperti ini sangat mempermudah petugas dalam melayani pembayaran para pelanggan.
3. sistem biometrik dapat mencegah orang yang tidak berwenang memasuki fasilitas-fasilitas rahasia atau mengakses informasi yang bersifat rahasia, dengan cara menganalisis sidik jari atau retina mata.
4. sistem point of sale/ POS yang diterapkan pada kebanyakan pasar swalayan dengan dukungan pembaca Barcode ditujukan mempercepat layanan

kepada pelanggan dan memungkinkan persediaan barang bias diketahui oleh petugas dengan cepat. (Abdul Kadir, 2012:2)

II.2. Data

Mengenai pengertian data, lebih jelas apa yang didefinisikan oleh Drs. Jhon J. Longkutoy dalam bukunya “Pengenalan Komputer” sebagai berikut : isitilah data adalah suatu istilah majemuk yang berarti fakta atau bagian dari fakta yang mengandung arti yang dihubungkan dengan kenyataan simbol-simbol, gambar-gambar, angka-angka, huruf-huruf, atau simbol-simbol yang menunjukan suatu ide, objek, kondisi, atau situasi dan lain-lain (Tata Sutabri, 2012:16).



Gambar II.3. Pemrosesan Data

(Sumber : Tata Sutabri, 2012:16)

II.2.1. Pengolahan Data

Data merupakan bahan mentah untuk diolah, yang hasilnya kemudian menjadi informasi. Dengan kata lain, data yang telah diperoleh harus diukur dan dinilai baik buruknya, berguna atau tidak dalam hubungannya dengan tujuan yang akan dicapai. Pengolahan data terdiri dari kegiatan-kegiatan penyimpanan data dan penanganan data. Untuk lebih jelasnya akan diuraikan dibawah ini (Tata Sutabri, 2012:21).

II.3. Rekam Medis

Dalam pelayanan kedokteran/kesehatan, terutama yang dilakukan para dokter di rumah sakit maupun praktek pribadi, peranan rekam medis sangat penting dan sangat melekat dengan kegiatan pelayanan. Sehingga ada ungkapan bahwa rekam medis adalah orang ke tiga pada saat dokter menerima pasien. Sepintas hal itu dapat dipahami, karena catatan demikian akan berguna untuk merekam keadaan pasien, hasil pemeriksaan serta tindakan pengobatan yang diberikan.

Menurut *PERMENKES No.749a/Menkes/XII/1989* tentang rekam medis, disebut pengertian Rekam Medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain kepada pasien pada sarana pelayanan kesehatan.

Menurut *PERMENKES No.269/MENKES/PER/III/2008* yang dimaksud rekam medis adalah berkas yang berisi catatan dan dokumen antara lain identitas pasien, hasil pemeriksaan, pengobatan yang telah diberikan, serta tindakan dan

pelayanan lain yang telah diberikan, serta tindakan dan pelayanan lain yang telah diberikan kepada pasien. Catatan merupakan tulisan-tulisan yang dibuat oleh dokter mengenai tindakan-tindakan kepada pasien dalam rangka pelayanan kesehatan.

Data pasien rawat yang dimasukkan dalam *medical record* atau rekam medis antara lain :

1. Identitas pasien
2. Tanggal dan waktu pemeriksaan
3. Anamnesis (Keluhan dan Riwayat Penyakit)
4. Hasil pemeriksaan fisik dan penunjang medis
5. Diagnosis
6. Rencana penatalaksanaan
7. Pengobatan atau tindakan
8. Pelayanan lain yang telah diberikan kepada pasien
9. Persetujuan tindakan bila perlu

Tujuan rekam medis dibuat untuk menunjang tercapainya tertib administrasi dalam rangka peningkatan pelayanan kesehatan. Sedangkan rekam medis memiliki 5 manfaat, yaitu :

1. Sebagai dasar pemeliharaan kesehatan dan pengobatan pasien.
2. Sebagai bahan pembuktian dalam perkara hukum.
3. Bahan untuk kepentingan penelitian.
4. Sebagai dasar pembayaran biaya pelayanan kesehatan.

5. Sebagai bahan untuk menyiapkan statistik kesehatan. (Jurnal Sistem Informasi Rekam Medis pada Klinik Mitra Medika Semarang ; Kristiawan ; 5)

II.4. Basis Data

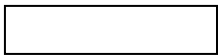


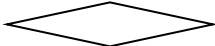
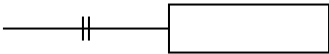
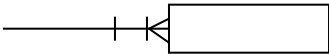

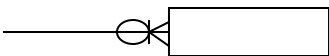
Basis Data adalah kumpulan informasi yang terorganisasi dan disajikan untuk tujuan khusus. Basis data terkomputerisasi dapat di-*update*, file bisa terorganisasi, dan informasi dapat dibaca, dicari dengan cepat, dan di-*retrieve* menggunakan komputer.

Sistem Manajemen Basis Data (DBMS) adalah suatu sistem perangkat lunak kompleks yang mengatur permintaan dan penyimpanan data ke dan dari *disk*. DBMS menyediakan keamanan (*security*), privasi (*privacy*), integritas (*integrity*), dan kontrol konkurensi (*concurrency controls*). Contoh perangkat lunak DBMS adalah Ms. Access, Oracle, Ms. SQL Server, Sybase, DB2, INGRES dan MySQL. (Janner Simarmata ; 2007 : 13-14)

II.5. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah bagian yang menunjukkan hubungan antara entity yang ada dalam sistem. Simbol-simbol yang digunakan dapat dilihat dari tabel II.2. (Jurnal Momentum Vol 17 No.2 : 2015: Hal : 70).

Tabel II.2. Simbol Yang Digunakan Pada *Entity Relationship Diagram (ERD)*

SIMBOL	KETERANGAN
	<i>Entity</i>
	Atribut Dan <i>Entity</i>
	Atribut Dan <i>Entity</i> Dengan <i>Key</i> (Kunci)
	Relasi Atau Aktifitas Antar <i>Entity</i>
	Hubungan Satu Dan Pasti
	Hubungan Banyak Dan Pasti
	Hubungan Satu Tapi Tidak Pasti
	Hubungan Banyak Tapi Tidak Pasti

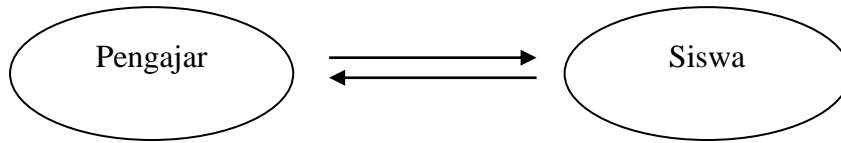
(Sumber : Jurnal Momentum Vol 17 No.2 : 2015: Hal : 70)

II.5.1. Teknik *Entity Relationship*

Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan atribut yang ada di dalam 1 atau 2 file, yaitu sebagai berikut :

a. *One to one relationship 2 file*

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Seperti pada pelajaran privat dimana satu guru mengajar satu siswa dan siswa hanya diajari oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.

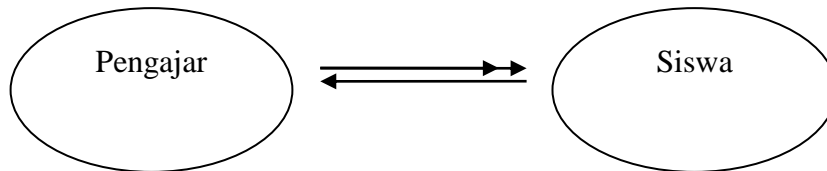


Gambar II.4. One to one relationship 2 file

(Sumber : Tata Sutabri ; 2004 : 209)

b. One to many relationship 2 file

Hubungan antara file pertama dengan file kedua adalah salah satu berbanding banyak atau dapat pula dibalik, banyak lawan satu. Seperti pada sistem pengajaran di sekolah dasar, di mana satu guru mengajar banyak siswa dan siswa hanya diajari oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan table relasi antarkeduanya diwakili dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.



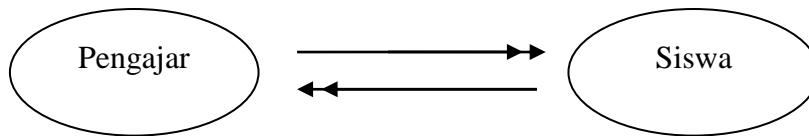
Gambar II.5. One to many relationship 2 file

(Sumber : Tata Sutabri ; 2004 : 209)

c. Many to many relationship 2 file

Hubungan anatara file pertama dengan file kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran di perguruan tinggi, dimana suatu dosen mengajar banyak banyak mahasiswa dan mahasiswa diajari oleh banyak dosen. Hubungan tersebut dapat digambarkan dengan tanda lingkaran

untuk menunjukkan tabel relasi antar keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

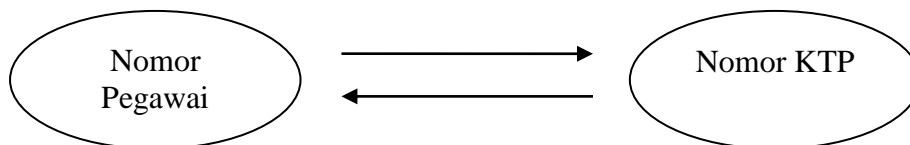


Gambar II.6. Many to many relationship 2 file

(Sumber : Tata Sutabri ; 2004 : 210)

d. Relasi one to one 2 atribut dalam 1 file.

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan satu. Misalnya atribut nomor pegawai yang unik dan atribut satu lawan satu. Satu nomor pegawai hanya satu nomor KTP, tidak ada yang berganda.

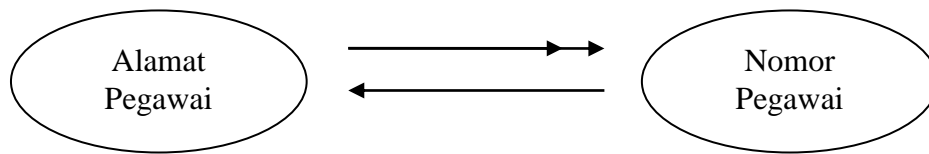


Gambar II.7. Relasi one to one 2 atribut dalam 1 file

(Sumber : Tata Sutabri ; 2004 : 210)

e. Relasi many to one 2 attribute dalam 1 file.

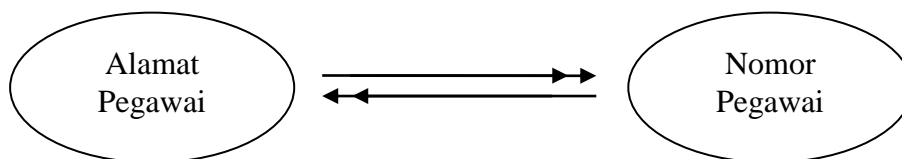
Hubungan antara satu atribut dengan atribut lainnya dalam 1 file yang sama mempunyai hubungan satu lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut, maka hubungan antara atribut alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan banyak pegawai.



Gambar II.8. Relasi many to one 2 attribute dalam 1 file
 (Sumber : Tata Sutabri ; 2004 : 210)

f. Relasi many to many 2 attribute dalam 1 file

Hubungan antara satu dengan atribut yang lain dalam 1 file yang sama mempunyai hubungan banyak lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut dan beberapa pegawai punya nama yang sama yang mempunyai alamat berbeda. Maka hubungan antara atribut alamat pegawai dengan nama pegawai adalah beberapa alamat menunjukkan nama pegawai yang sama dan sebaliknya.



Gambar II.9. Relasi many to many 2 attribute dalam 1 file
 (Sumber : Tata Sutabri ; 2004 : 211)

II.6. Normalisasi

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

1. Bentuk Normal Pertama (1 NF)

Contoh yang kita gunakan di sini adalah sebuah perancangan yang mendapatkan barang dari sejumlah pemasok. Masing-masing pemasok berada pada satu kota. Sebuah kota dapat mempunyai lebih dari satu pemasok dan masing-masing kota mempunyai kode status tersendiri. Masing-masing pemasok bisa menyediakan banyak barang. Tabel relasionalnya dapat dituliskan sebagai berikut :

PEMASOK (P#, Status, Kota, b#, qty) di mana

p# : kode pemasok (kunci utama)

status : kode status kota

Kota : nama kota

b# : barang yang dipasok

qty : jumlah barang yang dipasok.

Sebuah tabel relasional secara defenisi selalu berada dalam bentuk normal pertama. Semua nilai pada kolom-kolomnya adalah atomi. Ini berarti kolom-kolom tidak mempunyai nilai berulang. Tabel II.2. menunjukkan tabel pemasok dalam 1 NF.

Tabel II.3. Normalisasi Pertama Pemasok

P#	Status	Kota	B#	Qty
P1	20	Yogyakarta	B1	300
P1	20	Yogyakarta	B2	200
P1	20	Yogyakarta	B3	400
P1	20	Yogyakarta	B4	200

P1	20	Yogyakarta	B5	100
P1	20	Yogyakarta	B6	100
P2	10	Medan	B1	300
P2	10	Medan	B2	400
P3	10	Medan	B2	200
P4	20	Yogyakarta	B2	200
P4	20	Yogyakarta	B4	300
P4	20	Yogyakarta	B5	400

(Sumber : Janner Simarmata, dkk, 2010).

2. Bentuk Normal Kedua (2 NF).

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1 NF, tetapi tidak pada 2 NF, sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1 NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama. Ini berarti bahwa setiap kolom bukan kunci harus tergantung pada seluruh kolom yang membentuk kunci utama. Tabel pemasok berada pada 1 NF, tetapi tidak pada 2 NF karena status dan kota tergantung secara fungsional hanya pada kolom p# dari kunci gabungan (p#, b#). Ini dapat digambarkan dengan membuat daftar ketergantungan fungsional.

P# → Kota, Status

Kota → Status

(P#, B#) → qty

Proses mengubah tabel 1 NF ke 2 NF adalah :

- a. Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukan.
- c. Pindahkan kolom-kolom yang ditentukan dari tabel asal ke tabel baru penentu akan menjadi kunci utama pada tabel baru
- d. Hapus kolom yang baru dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

Pada contoh, kita memindahkan kolom p#, status, dan kota ke tabel baru yang disebut pemasok2. Kolom p# menjadi kunci utama tabel ini. Tabel II.3. menunjukkan hasilnya.

Tabel II.4. Tabel Bentuk Normal Kedua (2NF).

Pemasok2

Barang

P#	Status	Kota
P1	20	Yogyakarta
P2	10	Medan
P3	10	Medan
P4	20	Yogyakarta
P5	30	Bandung

P#	B#	Qty
P1	B1	300
P1	B2	200
P1	B3	400
P1	B4	200
P1	B5	100
P1	B6	100
P2	B1	300
P2	B2	400
P3	B2	200
P4	B2	200
P4	B4	300
P4	B5	400

(Sumber : Janner Simarmata, dkk, 2010)

3. Bentuk Normal Ketiga (3 NF).

Bentuk normal ketiga mengharuskan semua kolom pada tabel relasional hanya pada kunci utama. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3 NF) jika tabel sudah berada pada 2 NF dan setiap kolom yang bukan kunci tidak tergantung secara transistif pada kunci utamanya. Dengan kata lain, semua atribut bukan kunci tergantung secara fungsional hanya pada kunci utama.

Tabel barang sudah dalam bentuk normal ketiga. Kolom bukan kunci, qty, tergantung sepenuhnya pada kunci utama (p#, b#). Pemasok masih berada pada 2 NF, tetapi belum berada pada 3 NF karena dia mengandung ketergantungan transitif. Ketergantungan transitif terjadi ketika sebuah kolom bukan kunci, yang ditentukan oleh kunci utama, menentukan kolom lainnya. Konsep ketergantungan transitif dapat digambarkan dengan menunjukkan ketergantungan fungsional pada pemasok2, yaitu :

Pemasok2. p# \longrightarrow Pemasok2, status

Pemasok2. p# \longrightarrow Pemasok2, kota

Pemasok2. kota \longrightarrow Pemasok2, status

Perlu dicatat bahwa pemasok2, status ditentukan, baik oleh kunci utama p#, maupun kolom bukan kunci, kota

Proses mengubah tabel menjadi 3 NF adalah :

- a. Tentukan semua penentu selain kunci utama dan kolom yang ditentukannya.
- b. Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom yang ditentukannya.
- c. Pindahkan kolom yang ditentukan dari tabel asal ke tabel baru. Penentu menjadi kunci utama tabel baru.
- d. Hapus kolom yang baru saja dipindahkan dari tabel asal, kecuali penentu yang akan berfungsi sebagai kunci tamu.
- e. Tabel asal bisa diberi nama baru.

Untuk mengubah PEMASOK2 menjadi 3 NF, kita membuat tabel baru yang disebut KOTA_STATUS dan memindahkan kolom kota dan status ke tabel baru. Status dihapus dari tabel diberi nama baru PEMASOK_KOTA. Tabel II.4 menunjukkan hasilnya.

Tabel II.5. Tabel Bentuk Normal Ketiga (3 NF)

PEMASOK_KOTA

P#	Kota
P1	Yogyakarta
P2	Medan
P3	Medan
P4	Yogyakarta
P5	Bandung

KOTA_STATUS

Kota	Status
Yogyakarta	20
Medan	10
Bandung	30
Semarang	40

Sumber : (Janner Simarmata, dkk, 2010).

4. Bentuk Normal Boyce Code (BCNF)

Setelah 3 NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Banyak praktisi berpendapat bahwa menempatkan entitas pada 3 NF sudah cukup karena sangat jarang entitas yang berada pada 3 NF bukan merupakan 4 NF dan 5 NF. Lebih lanjut, mereka berpendapat bahwa keuntungan yang didapat mengubah entitas ke 4 NF dan 5 NF sangat kecil sehingga tidak perlu dikerjakan. Bentuk Normal Boyce- Code (BCNF) adalah versi 3 NF lebih teliti dan berhubungan dengan tabel

relasional yang mempunyai (a) banyak kunci kandidat (b) kunci kandidat gabungan, dan (c) kunci kandidat yang saling tumpang tindih.

BCNF didasarkan pada konsep penentu. Sebuah kolom penentu adalah kolom di mana kolom-kolom lain sepenuhnya tergantung secara fungsional. Sebuah tabel relasional berada pada BCNF jika dan hanya setiap penentu adalah kunci kandidat

5. Bentuk Normal Keempat (4 NF)

Sebuah tabel relasional berada pada bentuk normal keempat (4 NF) jika dia dalam BCNF dan semua ketergantungan multivalued merupakan ketergantungan fungsional.

Bentuk normal keempat (4 NF) didasarkan pada konsep ketergantungan multivalued (MVD). Sebuah ketergantungan multivalued terjadi ketika dalam sebuah tabel relasional yang mengandung setidaknya tiga kolom, satu kolom mempunyai banyak baris bernilai sama, tetapi kolom lain bernilai berbeda.

Defenisi secara formal diberikan oleh CJ. Date, yaitu :

Misalnya, ada sebuah tabel relasional R dengan kolom A, B dan C, Maka $R.A \twoheadrightarrow R.B$ (kolom A menentukan kolom B).

Adalah benar jika dan hanya jika himpunan nilai B yang cocok dengan pasangan nilai A dan nilai C pada R hanya tergantung pada nilai A dan tidak tergantung pada nilai C.

MVD selalu terjadi dalam pasangan, yaitu $R.A \twoheadrightarrow R.B$ dipenuhi jika dan hanya jika $R.A \twoheadrightarrow R.C$ dipenuhi pula.

6. Bentuk Normal Kelima (5 NF).

Sebuah tabel berada pada bentuk normal kelima jika dia tidak dapat mempunyai dekomposisi lossless menjadi sejumlah tabel lebih kecil.

Empat bentuk normal pertama berdasarkan pada konsep ketergantungan fungsional, sedangkan bentuk normal kelima berdasarkan pada konsep ketergantungan gabungan (*join dependence*). Ketergantungan gabungan berarti sebuah tabel, setelah deskomposisi menjadi tiga atau lebih tabel yang lebih kecil, harus dapat digabungkan kembali untuk membentuk tabel asal. Dengan kata lain 5 NF menunjukkan ketika sebuah tabel tidak dapat dideskomposisi lagi (Janner Simarmata, 2012)

II.7. *Unified Modeling Language (UML)*

Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industry perangkat lunak dan pengembangan sistem. (Windu Gata : 2013)


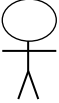
II.7.1. Diagram-Diagram UML


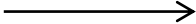
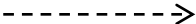
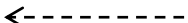
1. Diagram Use Case (*Use Case Diagram*)

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.

Simbol-simbol yang digunakan dalam *use case* diagram, yaitu :

Tabel II.6. Simbol *Use Case Diagram*

Gambar	Keterangan
	Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case.
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan




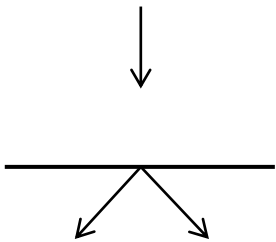
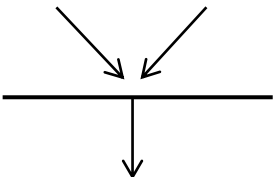
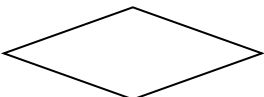

	use case, tetapi tidak memiliki control terhadap use case.
	Asosiasi antara aktor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan use case yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	<i>Include</i> , merupakan di dalam use case lain (<i>required</i>) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Extend</i> , merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

(Sumber : Windu Gata : 2013)

2. Diagram Aktivitas (*Activity Diagram*)

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

Tabel II.7. Simbol Diagram Aktivitas

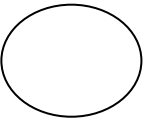
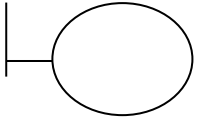
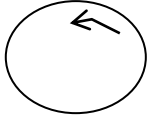

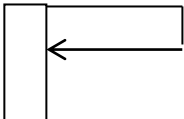
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau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



(Sumber : Windu Gata : 2014)

3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel II.8. Simbol *Sequence Diagram*

Gambar	Keterangan
	<p><i>EntityClass</i>, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.</p>
	<p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.</p>
	<p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.</p>
	<p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p>
	<p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p>

	<p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

(Sumber : Windu Gata : 2013)

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan.

Class diagram secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti.

Tabel II.9. Simbol *Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

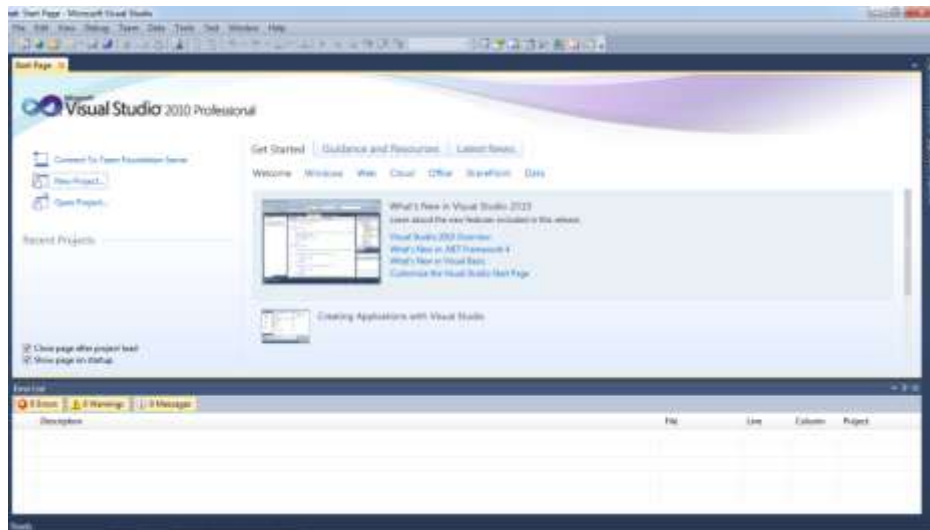
(Sumber : Windu Gata : 2013)

II.8. Aplikasi Yang Digunakan

II.8.1. Bahasa Pemograman *Microsoft Visual Basic*

Visual Basic 2010 adalah Visual Basic yang direkayasa kembali untuk digunakan pada *platform .NET* sehingga aplikasi yang dibuat menggunakan *Visual Basic.NET* dapat berjalan pada sistem komputer apapun dan dapat mengambil data dari *server* dengan tipe data apapun asalkan terinstal *.NET Framework*. (Priyanto Hidayatullah ; 2012 : 5)

Untuk melihat tampilan *visual basic 2010* dapat dilihat pada gambar II.2. sebagai berikut :



Gambar II.10. Tampilan Utama Visual Basic 2010

(Sumber : Priyanto Hidayatullah ; 2012 : 24)

II.8.2. SQL Server

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management Sistem*) 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 objek-objek yang sering digunakan pada aplikasi bisnis, seperti membuat *database*, *table*, *function*, *stored procedure*, *trigger* dan *view*. Selain Object, kita juga menjalankan perintah *SQL (Structured Query Language)* untuk mengambil data.(Elex Media Komputindo ; 2010 : 101)