

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Istilah sistem bukanlah hal yang asing bagi khalayak ramai, yang sering kali sistem mengacu pada komputer seperti IBM PC dan Monitor. Sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai tujuan. Jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka elemen tersebut dapat dipastikan bukan bagian dari suatu sistem yang dimaksud. Sebagai contoh, raket dan pemukul bola kasti yang tidak membentuk sebuah sistem, karena tidak sistem permainan permainan olah raga memadukan dua rangkaian peralatan tersebut. (Sumber: Abul Kadir, 2012)

II.1.1. Informasi

Informasi merupakan salah satu sumber daya penting dalam manajemen modern. Banyak keputusan strategis yang bergantung kepada informasi. Sebagaimana diketahui, sumber daya 4M + 1I yang mencakup manusia (sumber daya manusia atau SDM), material, mesin, modal, dan informasi merupakan sumber daya vital bagi kelangsungan organisasi bisnis. (Sumber: Abul Kadir, 2012)

II. I.2. Sistem Informasi

Berikut ini adalah definisi sistem informasi menurut beberapa ahli:

Tabel II.1. Definisi 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: Abul Kadir, 2012).

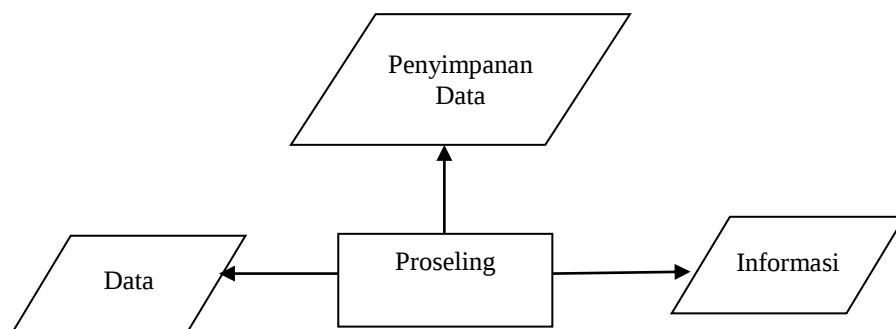
Berikut ini adalah contoh system informasi :

1. system reservasi pesawat terbang digunakan dalam biro perjalanan untuk melayani pemesanan/ pembelian tiket online.
2. system untuk menangani penjualan kredit kendaraan bermotor, sehingga dapat di gunakan untuk memantau piutang para pelanggan, dan system seperti ini sangat mempermudah petugas dalam melayani pembayaran para pelanggan.

3. system biometric 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. system 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. (sumber: Abdul Kadir, 2012)

II.2. Data

Mengenai pengertian data, lebih jelas apa yang didefinisikan oleh Drs. Jhon J. Longkutoy dalam bukunya “Pengenalan Komputer” sebagai berikut : istilah 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).



Gambar II.1. Pemrosesan Data

Sumber : (Tata Sutabri, 2012).

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).

1. Penyimpanan Data (*Data Storage*)

Penyimpanan data meliputi pekerjaan pengumpulan (*filing*), pencarian (*searching*), dan pemeliharaan (*maintenance*). Data disimpan dalam suatu tempat yang lazim dinamakan "*file*". *File* dapat berbentuk map, *ordner*, *disket*, *tape*, *hard disk*, dan lain sebagainya. Sebelum disimpan, suatu data diberi kode menurut jenis kepentingannya. Pengaturan dilakukan sedemikian rupa sehingga mudah mencarinya. Pengkodean memegang peranan penting. Kode yang salah data akan mengakibatkan data masuk ke dalam *file* juga salah, yang selanjutnya akan mengakibatkan kesulitan pencarian data tersebut apabila diperlukan. Jadi *file* diartikan suatu susunan data yang terbentuk dari sejumlah catatan (*record*) yang berhubungan satu sama lain (sejenis) mengenai suatu bidang dalam suatu unit usaha.

Sistem yang umum dalam penyimpanan data (*filing*) ialah berdasarkan lembaga, perorangan, produksi, atau lain-lainya, tergantung dari sifat organisasi yang bersangkutan. Kadang-kadang dijumpai kesulitan apabila menghadapi suatu data dalam bentuk surat misalnya yang menyangkut ketiga klasifikasi tadi.

Metode yang terbaik adalah “referensi silang” (*cross reference*) antara *file* yang satu dengan *file* lain. Untuk memperoleh kemudahan dalam pencarian data (*searching*) di dalam *file*, maka *file* dibagi menjadi 2 (dua) jenis yaitu :

a. *File* Induk (*Master File*)

File induk ini berisi data-data permanen yang biasanya hanya dibentuk satu kali saja dan kemudian digunakan untuk pengolahan data selanjutnya.

Contoh : *file* kepegawaian, *file* gaji.

b. *File* Transaksi (*Detail File*).

File transaksi berisi data-data temporer untuk suatu periode untuk suatu bidang kegiatan atau suatu periode yang dihubungkan dengan suatu bidang kegiatan

Contoh : *file* lembur perminggu, *file* mutasi harian.

Pemeliharaan *file* (*file maintenance*) juga meliputi “peremajaan data” (*data updating*), yaitu kegiatan menambah catatan baru pada suatu data, mengadakan perbaikan, dan lain sebagainya. Misalnya, dalam hubungan dengan *file* kepegawaian, sudah tentu sebuah organisasi, entah itu perusahaan atau jawatan, akan menambah pegawainya. Sementara itu, ada pula pegawai yang pensiun atau berhenti bekerja atau putus hubungan dengan organisasi. Dengan demikian, data mengenai pegawai yang bersangkutan akan dikeluarkan dari *file* tersebut. Tidak jarang pula harus dilakukan perubahan data terhadap data seorang pegawai, misalnya kenaikan pangkat, kenaikan gaji berkala, menikah, pindah alamat, dan lain sebagainya.

2. Penanganan Data (*Data Handling*).

Penanganan data meliputi berbagai kegiatan, seperti pemeriksaan (*verifying*), perbandingan (*comparing*), pemilihan (*sorting*), peringkasan (*extracting*), dan penggunaan (*manipulating*). Pemeriksaan data mencakup pengecekan data yang muncul pada berbagai daftar yang berkaitan atau yang datang dari berbagai sumber, untuk mengetahui berbagai sumber dan perbedaan atau ketidaksesuaian. Pemeriksaan ini dilakukan dengan kegiatan pemeliharaan *file* (*file maintenance*).

Pemilihan atau *sorting* dalam rangka kegiatan penanganan data mencakup pengaturan ke dalam suatu urutan yang teratur, misalnya daftar pegawai menurut pangkatnya, dari pangkat yang tertinggi sampai yang terendah atau daftar pelanggan dengan menyusun namanya menurut abjad dan lain sebagainya. Peringkasan merupakan kegiatan lain dalam penanganan data. Ini mencakup keterangan pilihan, misalnya daftar pegawai yang telah mengabdikan dirinya kepada organisasi/perusahaan lebih dari 10 tahun atau daftar yang memesan beberapa hasil produksi sekaligus dan lain-lain.

Penggunaan data atau informasi “*data manipulation*” merupakan kegiatan untuk menghasilkan informasi. Kegiatan ini meliputi kompilasi tabel-tabel, statistik, ramalan mengenai perkembangan, dan lain sebagainya. Tujuan manipulasi ini adalah menyajikan informasi yang memadai mengenai apa yang terjadi pada waktu lampau guna menunjang manajemen, terutama membantu menyelidiki alternatif kegiatan mendatang. Jadi, hasil pengolahan data itu merupakan data untuk disimpan bagi penggunaan di waktu yang akan datang, yakni informasi

yang akan disampaikan kepada yang memerlukan atau mengambil keputusan mengenai suatu hal (Tata Sutabri, 2005).

II.3. Tender

Tender (Pelelangan) adalah merupakan suatu proses pengajuan penawaran yang dilakukan oleh kontraktor yang akan dilaksanakan dilapangan sesuai dengan dokumen Tender Tujuan Tender : Menyeleksi dan Menetapkan Calon Kontraktor yang akan mengerjakan pekerjaan. Dapat disimpulkan tender adalah tawaran untuk mengajukan harga, memborong pekerjaan, atau menyediakan barang yang diberikan oleh perusahaan swasta besar atau pemerintah kepada perusahaan-perusahaan lain.

Mengikuti tender adalah salah satu cara untuk mendapatkan kontrak bisnis dalam skala besar atau memperluas usaha. Banyak perusahaan yang secara teratur menyelenggarakan tender. Proses tender adalah proses yang penuh persaingan sehingga penting bagi untuk mencantumkan penawaran yang kompetitif di dalam proposal. Mengajukan penawaran melalui tender tidak memberikan jaminan keberhasilan dalam bentuk apapun. Yang penting persiapkanlah dengan matang proposal Anda.

Tender dilaksanakan melalui suatu sistim Pelelangan Umum atau Pelelangan Terbatas bagi para pengusaha yang telah memenuhi syarat-syarat dan standar kualifikasi yang ditentukan sebagai peserta Tender/Pelelangan. Berikut ini diuraikan secara singkat contoh pelaksanaan suatu Tender untuk Barang dan Jasa.

Pihak-pihak yang terlibat dalam suatu Tender Barang & Jasa adalah :

1. Pihak yang membutuhkan Barang dan Jasa;
2. Pihak peserta Tender yang menawarkan Barang dan Jasa.

II.4. *Entity Relationship Diagram (ERD)*

Entity Relationship (ER) data model didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh, masing-masing mahasiswa adalah entitas dan mata kuliah dapat dianggap sebagai entitas.

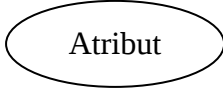
Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya atribut nim, nama, alamat, dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan mata kuliah.

Atribut NIM digunakan sebagai untuk mengidentifikasikan mahasiswa secara unik karena dimungkinkan terdapat dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

Relasi adalah hubungan antara beberapa entitas. Sebagai contoh relasi menghubungkan mahasiswa dengan mata kuliah yang diambilnya. Kumpulan semua entitas bertipe sama disebut kumpulan entitas (*entitas sel*), sedangkan kumpulan semua relasi bertipe sama disebut dengan kumpulan relasi (*relationship sel*).

Struktur logis (skema database) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut pada dilihat pada tabel II.2.

Tabel II.2. Komponen-Komponen Diagram ER

	Persegi Panjang mewakili kumpulan entitas
	Elips Mewakili Atribut
	Belah Ketupat Mewakili Relasi
	Garis Menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

Sumber : (Janner Simarmata, dkk, 2010)

II.4.1. 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 Nornal 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.3. 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# $\longrightarrow$ Kota, Status

Kota $\longrightarrow$ Status

(P#, B#) $\longrightarrow$ qty

Proses mengubah tabel 1 NF ke 2 NF adalah :

- Tentukan sembarang kolom penentu selain kunci gabungan dan kolom-kolom yang ditentukannya.
- Buat dan beri nama tabel baru untuk masing-masing penentu dan kolom-kolom yang ditentukan.
- 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.4. 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 definisi, 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 transistif daopat 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.5 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 C.J. 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.4.2. Basis Data (*Database*)

Secara sederhana database (basis data/ pangkalan data) dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat (Kadir, 2004). Pengertian akses dapat mencakup pemerolehan data maupun pemanipulasian data seperti menambah serta menghapus data. Dengan memanfaatkan komputer, data dapat disimpan dalam media pengingat yang disebut *hard disk*. Dengan menggunakan media ini, keperluan kertas untuk menyimpan data dapat dikurangi. Selain itu, data menjadi lebih cepat untuk diakses terutama jika dikemas dalam bentuk database.

Pengaplikasian database dapat kita lihat dan rasakan dalam keseharian kita. Database ini menjadi penting untuk mengelola data dari berbagai kegiatan. Misalnya, kita bisa menggunakan mesin ATM (anjungan tunai mandiri/ *automatic teller machine*) bank karena bank telah mempunyai database tentang nasabah dan rekening nasabah. Kemudian data tersebut dapat diakses melalui mesin ATM ketika bertransaksi melalui ATM. Pada saat melakukan transaksi, dalam konteks database sebenarnya kita sudah melakukan perubahan (*update*) data pada database di bank.

Ketika kita menyimpan alamat dan nomor telepon di HP, sebenarnya juga telah menggunakan konsep database. Data yang kita simpan di HP juga

mempunyai struktur yang diisi melalui formulir (*form*) yang disediakan. Pengguna dimungkinkan menambahkan nomor HP, nama pemegang, bahkan kemudian dapat ditambah dengan alamat *email*, alamat *web*, nama kantor, dan sebagainya.

Pemahaman tentang database ini dapat didekatkan pada konsep akuntansi. Kita bisa umpamakan bahwa ketika kita melakukan proses akuntansi secara manual, kita menuliskan suatu catatan ke dalam lajur dan kolom buku. Mulai dari jurnal, buku besar, buku pembantu kita memasukkan catatan satu demi satu. Melihat buku akuntansi tersebut, sebenarnya kita sudah melihat konsep database, yang jika dikelola dengan komputer masih diperlukan penyesuaian dalam membentuk kolom-kolomnya. (Mujilan, 2012)

II.4.3. Model Database

Model database yang saat ini banyak digunakan adalah model database relational. Imam (2008) menyebutkan “Model database ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*). Tabel-tabel yang ada dihubungkan (*relationship*) sedemikian rupa menggunakan *field-field* kunci (*key field*) sehingga dapat meminimalkan duplikasi data.”

Model database relational ini dapat kita kenal konsepnya mulai dari yang paling sederhana misalnya dengan penerapan program aplikasi *excel*. Meskipun untuk pengelolaan database secara luas *excel* jarang digunakan dan kurang mencukupi, namun untuk melihat konsep database dan konsep membangunnya program ini dapat dimanfaatkan. *Excel* mempunyai baris yang disebut *raw* dan mempunyai kolom. Kemudian item data merupakan sel atau pertemuan antara

baris dan kolom. Tabel-tabel dapat diumpakan apabila kita menggunakan tabel dalam suatu *sheet* tertentu. Data dari berbagai tabel dapat diambil dari tabel lain menggunakan perintah *look up* yang berdasarkan kode kunci tertentu. Kode kunci tersebut berada pada suatu kolom tertentu, yang dalam konsep database relational disebut sebagai *key field* tadi.

II.4.4. Struktur Database

Untuk memahami konteks database kita perlu memahami istilah dan hal-hal yang terkait dengan database. Dalam berbagai program aplikasi database terdapat kesamaan ataupun sedikit perbedaan di dalamnya. Seseorang yang mempelajari database dengan program aplikasi tertentu harus memperhatikan struktur dan karakteristik sesuai dengan bahasa dalam aplikasi tersebut. Namun demikian, secara umum terdapat karakteristik sebagai berikut:

1. Nama *file*

Nama *file* adalah nama yang digunakan untuk mengidentifikasi adanya data yang disimpan dalam komputer dan digunakan untuk pemanggilan data. *File* yang dikelola akan muncul dalam komputer dengan ekstensi sesuai dengan program aplikasinya. *File* tersebut dapat digunakan untuk menandakan adanya *file* database, ataupun *file table*. Database dan table akan saling terkait, meskipun cara menyimpan dalam komputer akan mengalami sedikit perbedaan pada beberapa aplikasi. Misalnya Ms. Access akan menyimpan dengan *file* yang dapat kita lihat adalah *file* databasenya. Di program *MySQL* nama database ini akan menjadi *folder*. Sementara di *FoxPro* nama database dapat menjadi *file* tersendiri. *Table* cara menyimpannya juga berbeda, dalam Ms. Access mungkin kita tidak melihat

nama *table* secara kasat mata karena akan dikelola di dalam *file* database. Di dalam *MySQL* kita bisa melihat beberapa nama *file* terkait dengan pengelolaan *table*. Dan di dalam *FoxPro* *table* ini dapat menjadi nama *file* terpisah dan dapat dikenali pula sebagai *free table*.

2. Database

Database sebenarnya merupakan nama untuk menampung berbagai *table* di dalamnya. Konsep ini akan sama dalam berbagai program aplikasi. Misalnya kita membangun database akuntansi dengan nama database “*akun_base*”. Di dalam *akun_base* akan diorganisasi berbagai *table* yang terkait dengan kegiatan akuntansi misalnya tabel: rekening, pelanggan, jurnal, buku induk, dan administrator program, dan sebagainya. Setiap data yang masuk tidaklah dicatat dalam database, namun di dalam masing-masing *table* yang sesuai.

3. Table

Table merupakan tempat untuk menyimpan data sesuai dengan kelompok data. Setiap isi *table* mengandung data yang mempunyai karakteristik dalam penggunaannya. Untuk mempermudah pengolahan biasanya pembangun database mengkategorikan *table* sesuai dengan data isinya sebagai berikut :

a. Master table

Master table berisi data tentang hal-hal utama dalam kegiatan database. *Table* ini berisi record yang relatif permanen atau seringkali menjadi acuan ketika mengoperasikan transaksi. Dalam master tabel identitas record menjadi penting dan diusahakan merupakan data atau kode yang bersifat unik. Unik dapat diartikan bahwa tidak ada dalam satu *table* berisi kode yang sama.

Disain kode menjadi penting di sini. Misalnya dalam mendisain nama akun dalam database akuntansi, maka kode akun menjadi sangat penting artinya. Dalam *table* berisi nama barang, maka kode barang menjadi hal penting. Contoh lain dalam database akademik, tabel *master* dapat berupa : mahasiswa, daftar dosen, daftar kurikulum.

b. *Transaction table*

Tabel transaksi digunakan untuk menyimpan data dalam menjalankan suatu kegiatan atau bisnis. Data ini seringkali akan bertambah dalam kesehariannya ketika terjadi transaksi yang sesuai dengannya. Secara lebih mudah dapat dipahami dalam akuntansi seringkali mencatat transaksi dalam jurnal. Terkait hal tersebut, transaksi ini dicatat dalam tabel jurnal. Dalam mencatat transaksi ini, kita harus menyesuaikan kode data tertentu dengan kode yang terdapat dalam *master table*.

c. *Tabulation table*

Tabulasi data dapat digunakan untuk menyimpan data seperti halnya master data namun bersifat sebagai data pembantu ketika menginput formulir baik untuk data master maupun transaksi. Misalnya untuk memetakan keterangan hobi, jenis kelamin, nama golongan, nama level manajemen, dan sebagainya. Dengan konsep penamaan field yang baik mungkin saja table tabulasi ini dapat digunakan untuk memuat berbagai kelompok data. Misalnya fieldnya berupa kode dan keterangan. Contoh kelompok gender dengan L = laki-laki; P = perempuan. Kelompok level dengan M = Manajer, O = operator, S = seller

d. *Temporary table*

Temporary adalah data sementara yang digunakan untuk membantu ketika terjadi proses transaksi. Data ini dapat saja langsung dihapus ketika transaksi selesai terproses. Misalnya digunakan untuk mempermudah perhitungan, penyimpanan data sementara sebelum diproses setuju ke database. Misalnya: ketika terjadi transaksi di depan kasir, data-data pertama akan ditangkap dan dimasukkan dalam file temporary sebelum akhirnya kasir melakukan perintah “ok” yang menandakan data transaksi siap untuk disimpan atau diproses dalam komputer. Ketika masa tunggu ini, data masih dapat diedit, dibatalkan, ataupun ditambah. Sementara ketika sudah masuk ke sistem, edit atau penambahan akan membutuhkan prosedur tertentu. Seandainya dianalogikan dengan sistem akuntansi maka proses edit data yang telah masuk ke sistem dapat digunakan prosedur seperti halnya melakukan jurnal koreksi.

4. *Field*

Field adalah penanda untuk kolom data. Jika dalam *excel* penanda tersebut adalah kolom A, B, dan seterusnya, sementara dalam konsep *table* dalam database maka nama *field* memegang peranan penting. Dalam konsep table dalam database, ketika memanggil dengan nama *field* tertentu maka data-data di dalamnya akan muncul. Pengolahan dapat dilakukan dengan membuat *filter*, misalnya berdasarkan kode tertentu, berdasarkan *record* tertentu. Misalnya kita ingin memanggil record terkait nama karyawan Andi. Dalam database Andi ini diberi ID: 11001. Sehingga kita bisa menggunakan konsep filtrasi untuk memanggil

personalia dengan kode ID 11001. Apabila data ketemu, maka kita dapat menggunakan data berdasarkan *field-field* yang ada, misalnya nama, tempat lahir, tanggal lahir, alamat, dan sebagainya yang mengacu pada ID 11001.

Dalam mengatur setting *field*, biasanya akan terkait hal-hal sebagai berikut :

- a. *Field type* : tipe *field* ini dapat terkait apakah *field* tersebut akan berisi data berupa *key field* (*primary key*, *secondary key*), atau *descriptor*. *Primary key* akan berisi ID atau kode pokok yang akan digunakan dalam mengidentifikasi *record*, sehingga data di dalam *field* tersebut tidak diijinkan untuk memiliki lebih dari satu data yang sama. *Secondary* adalah subset dari *key* utama. Misalnya saja kode mata kuliah dalam satu semester tidak boleh terdapat lebih dari satu pada ID atau NIM yang sama. *Descriptor* adalah *field* berisi data yang akan merupakan satu kesatuan dengan yang lainnya sebagai penjelasan akan adanya *record* atau ID tertentu.
- b. *Data type*: tipe data merupakan jenis data yang dapat dimasukkan dalam *field*. Hal ini dapat dibagi secara umum sebagai karakter/ *text*, numerik, tanggal dan sebagainya.
- c. *Field Size*. Penting untuk memahami ukuran *field* yang akan digunakan dalam menampung data. Dalam pengembangan sistem harus dapat memperkirakan berapa lebar ukuran *field* yang efektif. Apabila terlalu lebar akan terjadi banyak spasi kosong dan berpengaruh pada ukuran *file* yang disimpan. Sementara apabila terlalu sempit akan terdapat data yang tidak tersimpan. Misalnya ketika kita menghitung bahwa nama

menggunakan ukuran 40 karakter sudah memenuhi untuk *field* data kita. Konsekuensinya, apabila terdapat nama di atas 40 karakter maka akan terpotong menjadi 40 karakter. Konsekuensi lain adalah nama tersebut diinput hingga memuat maksimal 40 karakter yaitu dengan mengadakan singkatan nama.

5. *Records*

Records merupakan baris data. Karena satu baris data biasanya mengindikasikan satu kesatuan data tertentu, maka satu *record* ada yang menyebut satu data. Misalnya keterangan mengenai biodata Andi disimpan dalam satu record beridentitas ID 11001, maka untuk menyebutkan satu kesatuan data seputar Andi dalam baris tertentu ada yang menyebutkan sebagai *record* data Andi. Dalam konsep database masing-masing *record* memiliki nomor identitas tersendiri baik itu identitas yang diberikan komputer ataupun yang diinputkan secara manual. Sehingga dalam konteks tertentu dapat digunakan konsep nomor *record*, ID otomatis, ID *primary key*. (Mujilan, 2012)

II.5. *UML (Unified Modelling Language)*

Unified Modelling Language (UML) menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifak dalam UML adalah informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak. Contohnya adalah *source code* yang dihasilkan oleh pemrograman. Yang perlu diperhatikan untuk menjaga konsistensi antar artifak selama proses analisis dan desain adalah bahwa setiap perubahan terjadi pada

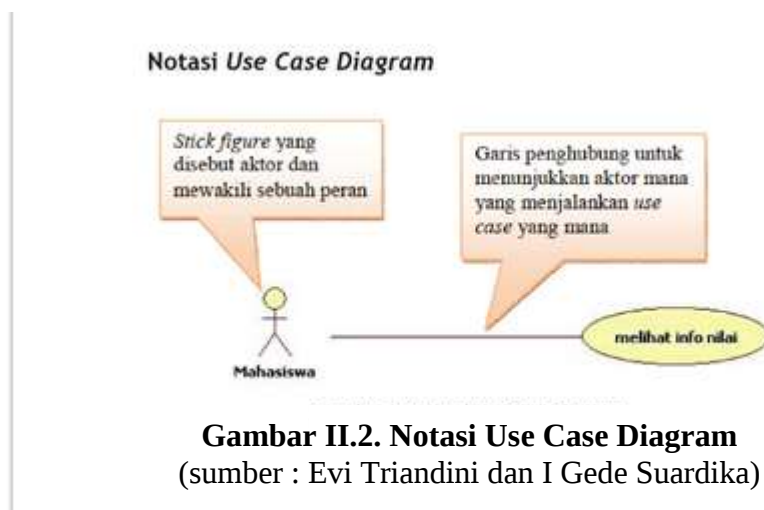
suatu artifak yang harus juga dilakukan pada artifak lainnya. (Sumber: Evi Triandini Dan I gede Suardika, 2012)

II.5.1. Diagram-Diagram UML

1. Diagram Use Case (Use Case Diagram)

Pengertian use case menurut Jhon Satzinger, 2010 dalam buku system analysis and design in a changing world menyatakan bahwa “use case adalah sebuah kegiatan yang dilakukan oleh system, biasanya dalam menanggapi permintaan dari pengguna system. (Sumber: Evi Triandini Dan I gede Suardika, 2012)

Notasi Use Case Diagram dapat dilihat pada gambar berikut ini:



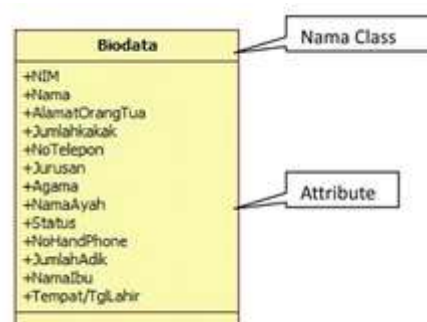
2. Diagram Kelas (Class Diagram)

Menurut John Satzinger, dalam buku system analysis and design in a changing world menyatakan bahwa “dalam uml ada dua jenis clas doiagram yaitu: domain class diagram dan design class diagram

1. Domain class diagram

focus domain class diagram adalah pada sesuatu dalam lingkungan kerja pengguna, bukan pada class perangkat lunak yang nantinya akan dirancang.

Berikut uni adalah gambar dari domain class diagram:

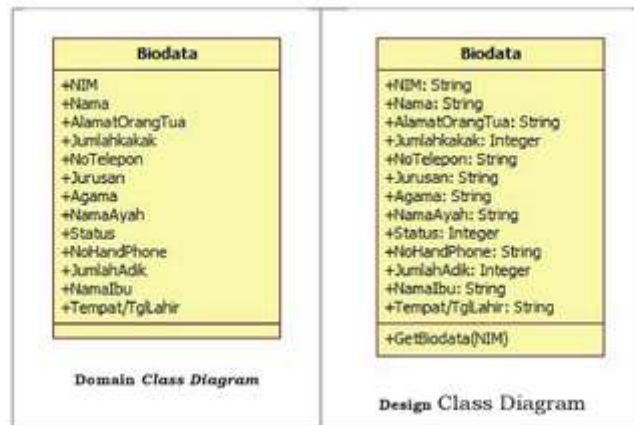


Gambar II.3. domain class diagram
(sumber : Evi Triandini dan I Gede Suardika)

2. Design class diagram

Tujuan utamanya adalah untuk mendokumentasikan dan menggambarkan kelas-kelas dalam pemrograman yang nantinya akan dibangun. Design class diagram menggambarkan kelas berorientasi objek yang dibutuhkan dalam pemrograman, navigasi diantara kelas, atribut names dan properti nya serta method names dan properti nya.

Gambar dari class diagram dapat dilihat pada gambar di bawah ini:

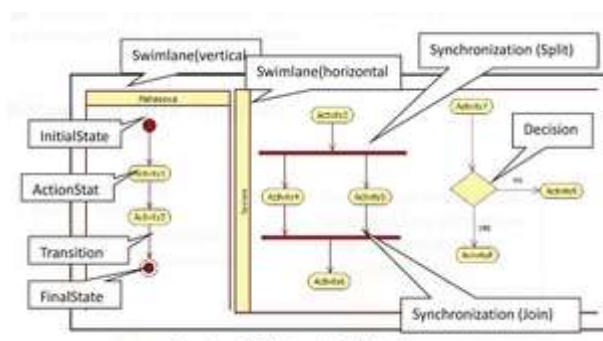


Gambar II.4. calss diagram
(sumber : Evi Triandini dan I Gede Suardika)

3. Diagram Aktivitas (*Activity Diagram*)

Menurut John Satzinger, dalam buku *system analysis and design in a changing world* menyatakan bahwa” activity diagram adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (atau pun sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut.

Notasi activity diagram adalah sebagai berikut:



Gambar II.5. Notasi activity diagram
(sumber : Evi Triandini dan I Gede Suardika)

Penjelasan dari gambar diatas adalah sebagai berikut:

Tabel II.6. Keterangan Notasi activity diagram

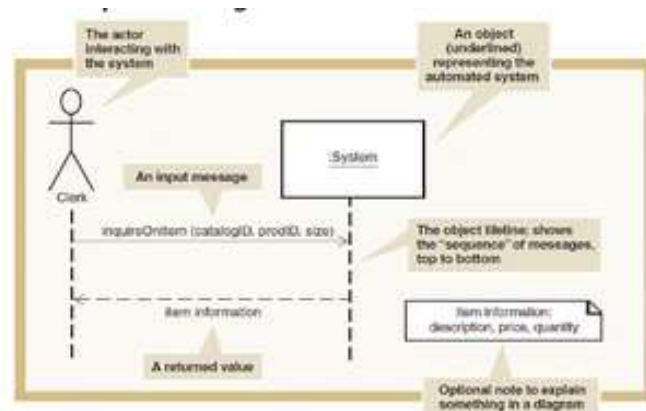
Notasi	Keterangan
Swimlane	Mewakili agen yang melakukan aktivitas karena dalam alur kerja umumnya mempunyai agen yang berbeda dengan yang melakukan langkah yang berbeda dari proses alur kerja.
Initialstate	Awal dari alur kerja
ActionState	Melambangkan aktivitas tersendiri dalam alur kerja
Transition	Melambangkan dari urutan diantara aktivitas
Final State	Akhir dari alur kerja
Sychronization	Membagi alur kerja menjadi beberapa alur yang berbarengan ataupun menggabungkan lagi alur yang berbarengan
Decision	Titik pengambilan keputusan dimana aliran proses tersebut akan mengikuti satu jalur atau jalur lainnya

(sumber : Evi Triandini dan I Gede Suardika)

4. *Sequence Diagram*

Menurut John Satzinger, dalam buku system analysis and design in a changing world menyatakan bahwa” System Sequence Digram (SSD) adalah diagram output serta urutan interaksi antara pengguna dan system untuk sebuah use case.

Berikut ini adalah gambar dari sequence diagram :



Gambar 11.6. sequence diagram
(sumber : Evi Triandini dan I Gede Suardika)

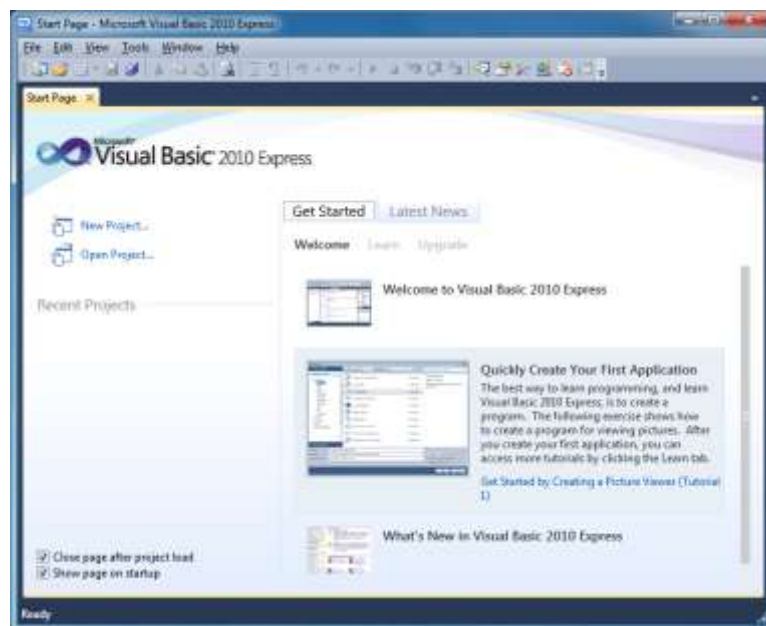
Penjelasan dari gambar diatas adalah sebagai berikut:

- Actor, mewakili seprang aktor (orang atau peran yang berinteraksi dengan sistem)
- Kontak berlabel, system adalah objek yang mwakili keseluruhan sistem yang terotomatisasi
- Garis putus-putus vertikal (lifelines) adalah perpanjangan objek tersebut, baik aktor maupun objek, sepanjang durasi dari sequence diagram
- Anak panah antara lifeline mewakili message yang dikirim atau diterima oleh aktor dari sistem
- Message diberi label untuk menggambarkan maksud message dan input apa pun yang sedang dikirim. Message dipertimbangkan sebagai sebuah inti yang diminta pada tujuan objek, kebanyakan seperti perintah.

II.6. Bahasa Pemrograman *Microsoft Visual Basic 2010*

Visual Basic adalah salah satu bahasa pemrograman berbasis dekstop yang dikeluarkan (diproduksi) oleh perusahaan perangkat lunak komputer terbesar yaitu *Microsoft*. *Visual Basic* merupakan salah satu bahasa pemrograman paling laris dan paling sukses di dunia. Menjadi pilihan berbagai kalangan tentunya *Visual Basic* memiliki berbagai hal yang patut dijadikan alasan, selain bahasa pemrograman yang sangat (paling) mudah dipelajari oleh berbagai kalangan baik awam maupun ahli, *Visual Basic* yang didukung penuh oleh poudsennya (*Microsoft*) selalu dikembangkan dan disesuaikan dengan kebutuhan zaman seperti penyesuaian model pemrograman modern yang berbasis OOP (*Object Oriented Programming*)

Untuk melihat tampilan visual basic 2010 dapat dilihat pada gambar II.7. sebagai berikut :



Gambar II.7. Tampilan Utama Visual Basic 2010

Sumber : (A. M. Hirin, 2011)

II.7. MYSQL

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multi user. *MySQL* memiliki dua bentuk lisensi, yaitu *free software* dan *shareware*. Penulis sendiri dalam menjelaskan buku ini menggunakan *MySQL* yang *free software* karena bebas menggunakan database ini untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensi, yang berada di bawah lisensi GNU/ GPL (*General Public License*), yang dapat anda download pada alamat resminya <http://www.mysql.com>. (Wahana Komputer, 2010)