

BAB II

TINJAUAN PUSTAKA

II.1.Konsep Dasar Sistem

Konsep dasar sistem akan menguraikan beberapa pengertian sistem, karakteristik sistem, pengertian dan komponen sistem informasi

II.1.1. Pengertian Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kennet Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas memimpin suatu organisasi adalah terlalu memusatkan perhatian pada salah satu komponen saja dari sistem organisasi.

Konsep lain yang terkandung di dalam defenisi tentang sistem adalah konsep sinergi. Konsep ini mengandaikan bahwa di dalam suatu sistem, *output* dari suatu organisasi diharapkan lebih besar dari pada *output* individual atau *output* masing-masing bagian. Kegiatan bersama dari bagian yang terpisah, tetapi saling berhubungan secara bersama-sama akan menghasilkan efek total yang lebih besar dari pada jumlah bagian secara individu dan terpisah. Ini berarti bahwa 2 ditambah 2 tidak sama dengan 4, tetapi memungkinkan sama dengan 5 atau lebih. Karena itu, sistem organisasi mengutamakan pekerjaan-pekerjaan di dalam tim.

Sebuah sistem terdiri atas bagian-bagian atau komponen yang terpadu untuk satu tujuan. Model dasar dari bentuk sistem ini adalah adanya masukan, pengolahan, dan keluaran. Akan tetapi, sistem ini dapat dikembangkan hingga menyetakan media penyimpanan. Sistem dapat terbuka dan tertutup akan tetapi sistem informasi biasanya adalah sistem terbuka. Artinya, sistem tersebut dapat menerima beberapa masukan dari lingkungan luarnya. (Tata Sutabri; 2012 :10-11)

II.1.2. Karakteristik Sistem

Model umum sebuah sistem adalah input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana sebab sebuah sistem dapat mempunyai beberapa masukan dan keluaran. Selain itu sebuah sistem 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 (*Component*)

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, artinya saling berkerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar, yang disebut “supra sistem”.

2. Batas Sistem (*Boundary*).

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat pisahkan.

3. Lingkungan Luar Sistem (*Environment*).

Bentuk apapun yang ada di luar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut operasi lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut. Lingkungan yang menguntungkan merupakan bagi sistem tersebut. Dengan demikian, lingkungan luar tersebut harus dijaga dan dipelihara. Lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem lainnya disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian dapat terjadi suatu integrasi sistem untuk membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Contoh di dalam suatu sistem unit komputer. “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan “data” adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem yang lain. Contoh, sistem informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambil keputusan atau hal-hal lain yang menjadi *input* bagi subsistem lain

7. Pengolah Sistem (*Proses*).

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Contoh, sistem akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat *deterministic*. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan. (Tata Sutabri; 2012 :20-21).

II.2. Pengertian Informasi

Sistem informasi manajemen berhubungan dengan informasi, tetapi apakah informasi itu? Informasi adalah data yang telah diklasifikasikan diolah atau diinterpretasi untuk digunakan dalam proses pengambil keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah dari bentuk tak berguna menjadi berguna bagi penerimanya. (Tata Sutabri; 2012 : 29).

II.3. Pengertian 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 suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. (Tata Sutabri; 2012 :46).

II.3.1. Komponen Dan Jenis Sistem Informasi.

Sistem informasi terdiri dari komponen-komponen yang disebut blok bangunan (*building block*), yang terdiri dari blok masukan, blok model, blok keluaran, blok teknologi, blok basis data dan blok kendali. Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lain membentuk satu kesatuan untuk mencapai sasaran.

1. Blok Masukan (*Input Block*).

Input mewakili data yang masuk ke dalam sistem informasi. *Input* di sini termasuk metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematik yang akan memainipulasi data *input* dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan “*tool box*” dalam sistem informasi. Teknologi digunakan untuk menerima *input*, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi sistem terdiri dari 3 (tiga) bagian utama yaitu teknisi teknologi (*brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data (*database*) merupakan kumpulan data yang saling berkaitan dan berhubungan satu dengan lainnya, tersimpan di perangkat keras komputer

dan menggunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi menggunakan perangkat lunak paket yang disebut DBMS (*Database Management System*).

6. Blok Kendali (*Control Block*)

Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperature, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, ketidak efisienan, sabotase, dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah atau bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi (Tata Sutabri; 2012 :47-48).

II.4. Sistem Informasi Akuntansi

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Misalnya, salah satu input dari sistem informasi akuntansi pada sebuah toko baju, seperti contoh sebelumnya, adalah transaksi penjualan. Kita memperoleh transaksi penjualann dengan mencatat penjualan tersebut kedalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening dan memposting transaksi ke dalam jurnal.

Kemudian, secara periodik sistem informasi akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari neraca dan laporan laba rugi. (Anastasia Diana, dkk; 2011 : 4-5)

II.4.1. Biaya

Defenisi umum biaya adalah nilai moneter barang dan jasa yang dikeluarkan untuk mendapatkan manfaat sekarang atau masa depan. Oleh karena itu merefleksikan arus keluar sumber-sumber, seperti kas, atau komitmen keuangan untuk membayar di masa depan, seperti barang datang, arus kas tersebut mendatangkan manfaat-manfaat seperti bahan baku atau mesin yang dapat digunakan untuk membuat produk yang dapat dijual untuk menghasilkan suatu manfaat kas. (Anthony A. Atkinson, dkk; 2009 : 33).

II.4.2. CPO

Direktorat Jendral Perkebunan, Departemen Pertanian Republik Indonesia (2008) menyebutkan bahwa industri CPO menghasilkan produk strategis sebagai bahan dasar minyak goreng dan sebagai bahan bakar alternative (*bio-fuel*). Data menunjukkan bahwa total jumlah produksi CPO Indonesia terus mengalami peningkatan selama periode 1990-an hingga 2000-an. Dari total produksi CPO, 75 % dialokasikan untuk komoditas ekspor, sedangkan sisanya digunakan untuk memenuhi kebutuhan dalam negeri. CPO merupakan komoditas minyak nabati yang paling besar dalam proporsi perdagangan dunia, yang mencakup 40% dari perdagangan dunia. Produksi minyak kelapa sawit mayoritas di produksi di Asia Tenggara, yaitu lebih dari 84 % produksi dunia. Sedangkan Indonesia mampu

menyediakan hampir 30% produksi dunia. (Kuncoro Harto Widodo, dkk; 2011 : 97)

II.5. Basis Data (*Database*)

Defenisi basisdata menurut McLeod, dkk, (2001) adalah kumpulan seluruh sumber daya berbasis komputer milik organisasi. Sistem manajemen basisdata adalah aplikasi perangkat lunak yang menyimpan struktur basisdata, hubungan antardata dalam basisdata, serta berbagai formulir dan laporan yang berkaitan dengan basisdata. Basisdata yang dikendalikan oleh sistem manajemen basisdata adalah satu set catatan data yang berhubungan dan saling menjelaskan.

Basisdata warisan (*legacy database*) adalah basisdata yang sedang digunakan oleh sebuah perusahaan. Istilah warisan menyatakan bahwa basisdata telah dipakai selama beberapa tahun dan basisdata yang ada tidak sesuai dengan teknologi masa kini. Ketika sebuah perusahaan telah menentukan untuk merancang sebuah basisdata, basisdata yang ada dianggap sebagai basisdata warisan. (Janner Simarmata, dkk; 2010 : 2)

II.5.1. Aplikasi sistem Basisdata

Basisdata digunakan secara luas. Berikut adalah beberapa contoh aplikasi :

- **Perbankan** : untuk informasi pelanggan, rekening, peminjaman, dan transaksi perbankan.
- **Penerbangan** : untuk informasi pemesanan dan jadwal. Penerbangan adalah salah satu yang pertama yang menggunakan basisdata, yaitu

terminal di seluruh dunia mengakses pusat sistem basisdata melalui jaringan telepon dan jaringan data lainnya.

- **Universitas** : untuk informasi mahasiswa, pendaftaran kuliah, dan nilai.
- **Transaksi kartu kredit** : untuk membeli dengan kartu kredit dan membuat tagihan bulanan.
- **Telekomunikasi** : untuk mencatat semua panggilan, membuat tagihan bulanan, mencatat saldo dari kartu prabayar, dan menyimpan informasi tentang jaringan komunikasi.
- **Keuangan** : untuk menyimpan informasi tentang perusahaan serta menjual dan membeli alat keuangan seperti saham dan obligasi.
- **Penjualan** : untuk informasi pelanggan, produk, dan pembelian.
- **Perusahaan** : untuk mengelola manajemen rantai pasokan dan melacak produksi barang-barang di pabrik, inventaris barang-barang digudang atau toko, dan pesanan barang-barang.
- **Sumber daya manusia** : untuk informasi tentang pegawai, gaji, pajak, dan pembuatan cek pembayaran. (Janner Simarmata, dkk; 2010 : 3).

II.5.2. 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.

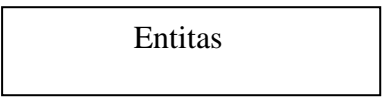
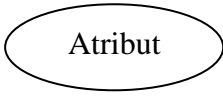
Entitas digambarkan dalam basis data dengan kumpulan atribut. Misalnya atribut nim, nama, alamat, dan kota bisa menggambarkan data mahasiswa tertentu dalam suatu universitas. Atribut-atribut membentuk entitas mahasiswa. Demikian pula, atribut kodeMK, namaMK, dan SKS mendeskripsikan mata kuliah.

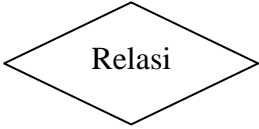

Atribut NIM digunakan sebagai untuk mengidentifikasi mahasiswa secara unik karena dimungkinkan terdapat dua mahasiswa dengan nama, alamat, dan kota yang sama. Pengenal unik harus diberikan pada masing-masing mahasiswa.

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 set*), sedangkan kumpulan semua relasi bertipe sama disebut dengan kumpulan relasi (*relationship set*).

Struktur logis (skema database) dapat ditunjukkan secara grafis dengan diagram ER yang dibentuk dari komponen-komponen berikut dilihat pada tabel II.1.

Tabel II.1. Komponen-Komponen Diagram ER

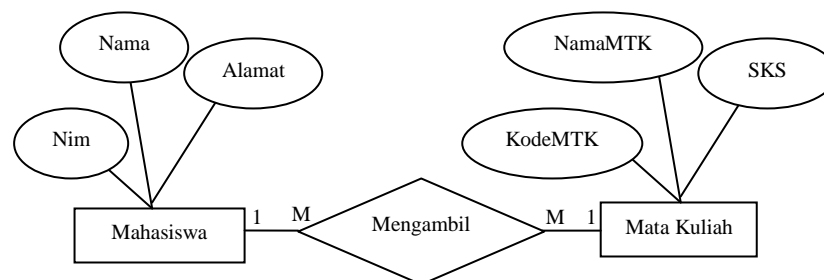
	Persegi Panjang mewakili kumpulan entitas
	Elips Mewakili Atribut

 Relasi	Belah Ketupat Mewakili Relasi
	Garis Menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

Sumber : Janner Simarmata, dkk (2010 :60)

Masing-masing komponen diberi nama entitas atau relasi yang diwakilinya.

Sebagai ilustrasinya bayangkan anda mengambil bagian sistem basis data universitas yang terdiri dari mahasiswa dan mata kuliah. gambar II.2. menunjukkan diagram ER dari contoh. Diagram menunjukkan bahwa ada dua kumpulan entitas yaitu mahasiswa dan mata kuliah dan bahwa relasi mengambil contoh mahasiswa dan mata kuliah, berikut dilihat pada gambar II.1. (Janner Simarmata; 2010 : 59-60)



Gambar II.1. Diagram ER

Sumber : Janner Simarmata, dkk (2010 : 60)

II.5.3. 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 (*www. utexas. edu*).

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.

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

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 transitif 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

daopat digambarkan dengan menunjukkan ketergantungan fungsional pada pemasok2, yaitu :

Pemasok2. p# \longrightarrow Pemasok2.status

Pemasok2. p# \longrightarrow Pemasok2.kota

Pemasok2. kota \longrightarrow Pemasok2.status

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 : 79 - 86)

II.6. Unified Modeling Language (*UML*)

UML singkatan dari Unified Modeling Language yang berarti bahasa pemodelan standar. (Chonoles; 2003 : bab 1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harus diikuti. Bagaimana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya harus mengikuti standar yang ada. UML bukan hanya sekedar diagram, tetapi juga menceritakan konteksnya. Ketika pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana sistem mengatasi error yang terjadi? Bagaimana keamanan terhadap sistem yang kita buat? Dan sebagainya dapat dijawab dengan UML. (Prabowo Pudjo Widodo, Herlawati; 2011 : 6).

II.6.1. Diagram-Diagram *UML*

Beberapa literatur menyebutkan bahwa *UML* menyediakan Sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa yang digabung, misalnya diagram komunikasi, diagram urutan, dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

1. Diagram Kelas. Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi, serta relasi-relasi diagram. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.
2. Diagram Paket (*Package Diagram*) Bersifat Statis. Diagram ini memperlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.
3. Diagram *Use Case* Bersifat Statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. Diagram Interaksi Dan *Sequence* (Urutan). Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*) bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi *UML* yang menekankan organisasi *structural* dari objek-objek yang menerima serta mengirim pesan.
6. Diagram *Statechart* (*Statechart Diagram*) Bersifat Dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*State*), transisi kejadian serta aktifitas. Diagram ini terutama penting

untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutam penting pada pemodelan sistem-sistem yang reaktif.

7. Diagram Aktivitas (*Activity Diagram*) Bersifat Dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan member tekanan pada aliran kendali antar objek.
8. Diagram Komponen (*Component Diagram*) Bersifat Statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan diagram kelas dimana komponen dipetakan kedalam satu atau lebih kelas-kelas. Antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram Deployment (*Deployment Diagram*) Bersifat Statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run time*). Memuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *Deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*). (Prabowo Pudjo Widodo, Herlawati; 2011 : 10-12).

II.7. Bahasa Pemrograman *Microsoft Visual Studio 2008*

Microsoft Visual Studio 2008 merupakan kelanjutan dari *Microsoft Visual Studio* sebelumnya, yaitu *Visual Studio .Net 2003* yang diproduksi oleh Microsoft. Pada bulan Februari 2002 *Microsoft* memproduksi teknologi. *Net Framework* versi 1.0, teknologi. *Net* ini didasarkan atas susunan berupa *Net Framework*, sehingga setiap produk baru yang terkait dengan teknologi. *Net* akan selalu berkembang mengikuti perkembangan. *Net Frameworknya*. Pada perkembangannya nantinya mungkin untuk membuat program dengan teknologi. *Net* memungkinkan para pengembang perangkat lunak akan dapat menggunakan lintas sistem operasi, yaitu dapat dikembangkan di sistem operasi windows juga dapat dijalankan pada sistem operasi lain, misalkan pada sistem operasi *Linux*, seperti yang telah dilakukan pada pemrograman *Java* oleh *Sun Microsystems*. Pada saat ini perusahaan-perusahaan sudah banyak mengupdate aplikasi lama yang dibuat *Microsoft Visual Basic 6.0* ke teknologi. *Net* karena kelebihan-kelebihan yang ditawarkan, terutama memungkinkan pengembang perangkat lunak secara cepat mampu membuat program *robust*, serta berbasiskan integrasi ke internet yang dikenal dengan *XML Web Service* (Ketut Darmayuda ; 2009 : 1)

II.8. MYSQL

Mysql adalah salah satu software sistem manajemen *database* (DBMS) *structured Query Language (SQL)* yang bersifat *open source*. *SQL* adalah bahasa standart untuk mengakses *database* dan didefenisikan dengan standart *ANSI/ISO SQL*. *MYSQL* dikembangkan, disebarluaskan, dan didukung oleh *MYSQL AB*.

MYSQL AB adalah perusahaan komersial yang didirikan oleh pengembang *MYSQL*. *MYSQL* merupakan aplikasi *Relational Database Management System* (RDBMS) yang digunakan untuk aplikasi *client server* atau sistem *embedded*.

Mysql mempunyai beberapa sifat yang menjadikannya sebagai salah satu software database yang banyak digunakan oleh pemakai diseluruh dunia. Sifat-sifat yang dimiliki oleh *MYSQL* antara lain :

- a. *Mysql* merupakan DBMS (*Database Management System*)
- b. *Database* adalah kumpulan data yang terstruktur. Data dapat berupa daftar belanja, kumpulan gambar, atau yang lebih luas yaitu informasi jaringan perusahaan. Agar dapat menambah, mengakses, dan memproses data tersimpan pada sebuah komputer database, kita membutuhkan sistem manajemen *database* (DBMS) seperti *MYSQL Server*. Sejak komputer sangat baik menangani sejumlah besar data, *sistem manajemen database* (DBMS) memainkan peran utama dalam perhitungan baik sebagai peralatan yang berdiri sendiri maupun bagian aplikasi.
- c. *Mysql* merupakan RDBMS (*Relational Database Management System*).
- d. *Database relational* menyimpan data pada table-tabel yang terpisah, bukan menyimpan data dalam ruang penyimpanan yang besar. Hal ini menambah kecepatan *fleksibilitas*.
- e. *Mysql* merupakan *software open source*.
- f. *Open source* berarti setiap orang dapat menggunakan dan mengubah software yang bersangkutan. Setiap orang dapat mendownload *software MYSQL* dari internet dan menggunakan tanpa membayar. Bahkan jika

mengkehendaknya anda bisa mempelajari kode sumber dan mengubah sesuai yang anda butuhkan. Software MySQL menggunakan GNU/GPL (General Public License). (Wahana Komputer; 2010 : 26).

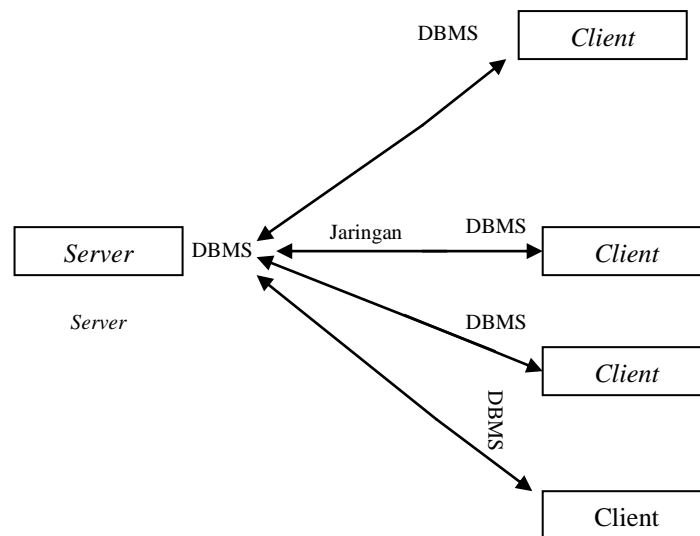
II.8.1. *Client Server*

Client server adalah satu model komunikasi 2 komputer atau lebih yang berfungsi melakukan pembagian tugas. *Client* bertugas untuk melakukan input, update, dan menampilkan data sebuah *database*. Sementara *server* bertugas untuk menyediakan pelayanan untuk melakukan manajemen yaitu : menyimpan dan mengolah *database*. (Wahana Komputer; 2010 : 5).

II.8.2. Arsitekur *Client Server*

1. Arsitektur StandAlone (1-Tier)

Model pertama aplikasi pemograman *database client server* adalah standalone atau 1 *tier* (1 -tingkat) adalah sebuah komputer yang mengakses sebuah *database* dari komponen sendiri. Dengan kata lain, aplikasi antarmuka *user* dan aplikasi DBMS terdapat pada komputer yang sama, berikut dilihat pada gambar II.2.



Gambar II.2. Arsitektur Stand Alone (1-Tier)

Sumber : (Wahana Komputer; 2010 : 6)

Adapun karakteristik arsitektur 1 tier sebagai berikut :

- a. Beban jaringan menjadi tinggi karena yang diminta adalah file database secara keseluruhan pada komputer *server client* melalui jaringan.
- b. Setiap komputer pada jaringan harus mempunyai DBMS tersendiri untuk menyimpan hasil salinan dari *server* sehingga mengurangi sumber daya yang dimiliki oleh komputer *client* terutama *memory*.
- c. Komputer *client* harus mempunyai kemampuan proses yang tinggi untuk mendapatkan waktu respon yang baik saat komputer *server* mengirimkan *file* yang diminta.
- d. Arsitektur *1-tier* cocok untuk bisnis kecil yang hanya membutuhkan data sebuah komputer untuk memproses dan menyimpan data sekaligus, tetapi kurang tepat diterapkan pada model jaringan. (Wahana Komputer; 2010 : 7).

2. Arsitekur 2-Tier

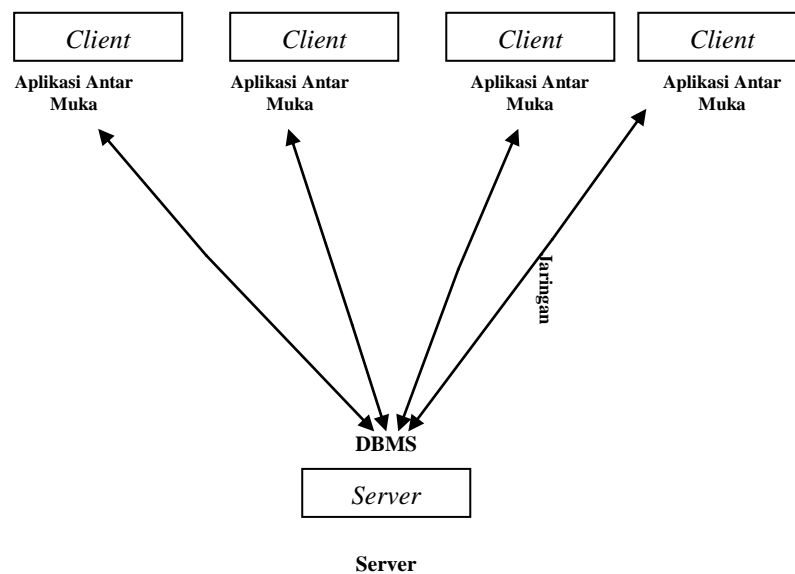
Model kedua sebuah pemrograman *database* adalah model 2-tier. Arsitektur pada model demikian membagi tugas antara komputer *client* server. Komputer *client* bertugas menyediakan antar muka untuk *user*, permintaan (*request data*) ke DBMS Server, serta pemrosesan data (mencakup logika penyajian data, logika pemrosesan data, dan logika atau bisnis) komputer *client* hanya mengirimkan sebuah *statement* untuk menambah (*insert*) data, mengubah (*update*), menghapus (*delete*), dan yang terakhir meminta (*select*) data untuk ditampilkan melalui antarmuka yang dibuat oleh *programmer*.

Karakteristik arsitektur 2-tier adalah :

- a. 2-tier terjadi pada jaringan dan melakukan pemodelan programan *database* dalam 2 tingkat. Tingkat pertama adalah *client* dan tingkat kedua adalah *server*.
- b. Tingkat pertama komputer *client* sebagai penyedia aplikasi antarmuka untuk mengolah *database*, baik menampilkan data kedalam *user interface*, menambah, menghapus data, maupun logika bisnis (*bussines logic*)
- c. Tingkat kedua adalah *server* yang menyediakan aplikasi untuk mengelola *database* serta menyediakan pula *query stored procedure*, dan *triggers*, yang dapat dipanggil *client* untuk mengolah data.
- d. Komputer *client* hanya mengirimkan sebuah *statement sql* untuk meminta data ke *server*.

- e. *Server* hanya memberikan data yang diminta melalui *statement* bersangkutan.
- f. Komputer *server* dituntut untuk memiliki kemampuan pemrosesan yang tinggi karena harus melayani permintaan banyak komputer *client* yang mengakses satu atau lebih DBMS.
- g. Beban jaringan menjadi ringan karena data yang berjalan pada jaringan hanya data yang diminta oleh *client*.
- h. Otentifikasi pemakai, pemeriksaan integritas, dan pemeliharaan kamus data dilakukan pada sisi *server*.
- i. Sederhana dan mudah untuk diterapkan, khususnya pada bisnis kecil yang hanya terdapat pada satu gedung

Berikut Gambar Arsitekur 2-Tier dilihat pada gambar II.3. (Wahana Komputer; 2010 : 7-8).



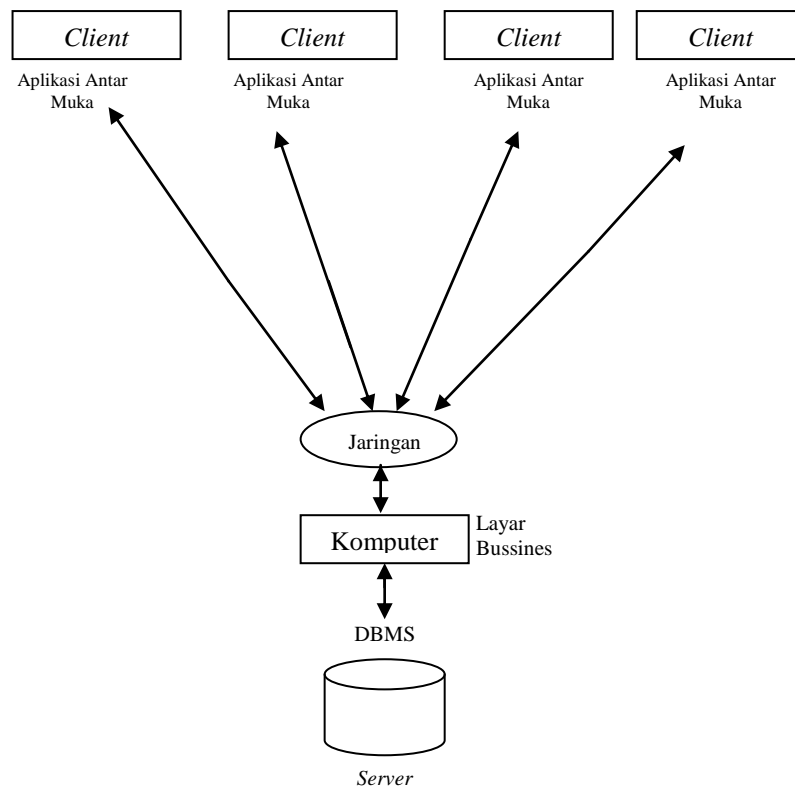
Gambar II.3. Arsitekur 2-Tier

Sumber : (Wahana Komputer; 2008 : 8)

3. Arsitekur N-Tier

Arsitektur *n-tier* berarti membagi komponen menjadi *n* entitas yaitu 1 tier *client* dan *n-1 tier server*. Seperti pada model sebelumnya *client* bertugas menyediakan antarmuka aplikasi, sedangkan bertugas menyediakan data. Pada model *n-tier* (sebagai contoh adalah 3-tier), server dibagi 2 menjadi, yaitu satu *server* yang dipakai sebagai *bussines object (middle tier)* dan satu *server* yang hanya menyimpan *database (server tier)*.

Secara nyata model *3-tier* adalah pada jaringan internetyang hanya memanfaatkan *database*. Internet lapisan pertama adalah komputer *client* yang menampilkan halaman *web*, tempat konten atau data halaman *web* berasal dari *database*. Lapisan kedua adalah *web* atau *HTTP server* yang menterjemahkan *script server side (PHP, JSP, ASP, dan lainnya)* dari komputer *client* untuk meminta data pada *database*. Kemudian lapisan ketiga adalah komputer *database server* yang menyediakan *database* yang diminta oleh *web* atau *HTTP server*, berikut gambar Arsitekur N-Tier dilihat pada gambar II.4. (Wahana Komputer; 2010 : 9).



Gambar II.4. Arsitekur N-Tier

Sumber : (Wahana Komputer; 2010 : 9)