BAB II
TINJAUAN PUSTAKA

II.1 Sistem Informasi Akuntansi

II.1.1. Sistem

Sistem merupakan serangkaian bagian yang saling tergantung dan bekerja sama untuk mencapai tujuan tertentu. Suatu sistem pasti tersusun dari sub-sub sistem yang lebih kecil yang saling tergantung dan bekerja samauntuk mencapai tujuan. Menurut Anastasia Diana & Lilis Setiawati (2011:3).

II.1.2. Informasi

Menurut Hanif Al Fatta (2005;9) Informasi adalah suatu alat menanyakan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuan sistem adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan pemeriksaan, pengorganisasian, pengendalian kegiatan operasi sub sistem suatu perusahaan. Data yang telah di olah menjadi bentuk baru atau bentuk lain yang memiliki arti dan manfaat bagi manusia. Dari defenisi tersebut dapat di simpulkan bahwa informasi merupakan hasil pemprosesan data yang di ubah menjadi bentuk baru atau bentuk lain sehingga memiliki arti dan manfaat bagi penggunanya.

II.1.3. Sistem Informasi

Sistem Informasi dapat di defenisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi,

media, prosedur-prosedur dan pengendalian yang di tujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik(Hanif Al Fatta:2005,9).

II.1.4. Akuntansi

Menurut Sumarsono S.R (2009:3) *American Accounting Association* mendefenisikan akuntansi sebagai proses mengindentifikasikan, mengukur, dan melaporkan informasi ekonomi, untuk memungkinkan adanya penilaian dan keputusan yang jelas dan tegas bagi mereka yang menggunakan informasi tersebut. Tujuan utama akutansi adalah menyajikan informasi ekonomi dari suatu kesatuan ekonomi kepada pihak-pihak yang berkepentingan.

Kegiatan akuntansi meliputi:

1. Pengindetifikasian dan pengukuran data yang relevan untuk suatu pengambilan keputusan.
2. Pemprosesan data yang bersangkutan kemudian peloporan informasi yang di hasilkan.
3. Pengkomunikasian Informasi kepada pemakai laporan.

II.1.5. Sistem Informasi Akutansi

Menurut Kusrini & Andi Koniyo (2007 : 10) Sistem Informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari sistem informasi akuntansi adalah :

1. Mendukung operasi sehari-hari.
2. Mendukung pengambilan keputusan manajemen.
3. Memenuhi kewajiban yang berhubungan dengan pertanggung jawaban.

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut:

1. Orang-orang yang mengoperasikan sistem tersebut.
2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis.
4. Software yang dipakai untuk memproses data organisasi.
5. Infrastruktur teknologi informasi.

Didalam organisasi, sistem informasi akuntansi berfungsi untuk :

1. Mengumpulkan dan menyimpan aktivitas yang dilaksanakan disuatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut.
2. Mengubah data menjadi informasi yang berguna bagi manajemen.
3. Menyediakan pengendalian yang memadai.

Sistem informasi akuntansi merupakan pendukung aktivitas organisasi, yang termasuk pendukung aktivitas organisasi adalah:

1. Infrastruktur perusahaan : akuntansi, hukum, administrasi umum.
2. Sumber daya manusia : perekrutan, pengontrolan, pelatihan dan kompensasi kepada pegawai.
3. Teknologi : peningkatan produk dan jasa (penelitian).
4. Pembelian.

Sementara itu aktivitas utamanya adalah:

1. *Inbound Logistics* : Penerimaan, penyimpanan dan distribusi bahan-bahan masukan.
2. Operasi : aktivitas untuk mengubah masukan menjadi barang atau jasa.
3. *Outbound Logistics* : distribusi produk ke pelanggan.
4. Pemasaran dan Penjualan.
5. Pelayanan : Dukungan purna jual dan maintenance.

Dari penjelasan tersebut dapat diambil simpulan bahwa akuntansi adalah proses pengidentifikasian, pengukuran dan melaporkan informasi ekonomi untuk memungkinkan adanya pembuatan pertimbangan dan keputusan bagi yang menggunakan informasi.

Kutipan diatas menjelaskan bahwa sistem informasi akuntansi adalah kumpulan dari sumber-sumber seperti orang dan peralatan yang dirancang untuk mentransformasikan data keuangan dan data lainnya menjadi informasi, dan informasi ini akan dikomunikasikan kepada para pembuat keputusan.

Manfaat Sistem Informasi Akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta/kekayaan perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan.
3. Menghasilkan informasi untuk pihak eksternal
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan)
6. Menghasilkan informasi untuk penyusunan data evaluasi anggaran perusahaan.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian.

II.2. Sekilas Tentang Laporan Keuangan

Menurut Anastasia Diana & Lilis Setiawati (2011 : 252). Laporan keuangan merupakan ringkasan dari semua catatan akuntansi atas seluruh kegiatan bisnis perusahaan. Laporan keuangan tidak akan dapat disusun jika catatan yang dimiliki perusahaan tidak lengap. Laporan keuangan tidak akan memiliki angka yang andal dan dapat dipercaya, jika data-data yang dicatat juga tidak andal. Jadi, dapat dilihat, apabila satu bagian tidak mencatat transaksi dengan baik, maka kekacauan catatan di salah satu bagian tersebut akan mengganggu proses penyusunan laporan keuangan.

Pengadministrasian data keuangan terbukti bukanlah pekerjaan yang mudah. Padahal tanpa pengelolaan data keuangan yang rapi dan pencatatan yang memadai, perusahaan tidak akan mendapatkan informasi yang dibutuhkan.

Menurut Drs. Darsono, MBA,Akt & Ashari, SE,Akt Laporan keuangan adalah hasil dari proses akuntansi yang disebut Siklus Akuntansi. Laporan keuangan menunjukkan posisi sumber daya yang di miliki oleh perusahaan selama satu periode. Selain itu, laporan keuangan juga menunjukkan kinerja keuangan perusahaan yang ditunjukkan dengan kemampuan perusahaan dalam menghasilkan pendapatan dengan sumber daya yang dimiliki oleh perusahaan. Sebagaimana dikemukakan dimuka bahwa akuntansi adalah suatu proses untuk mencatat, menggolongkan, dan meringgkas transaksi ekonomi dan keuangan untuk menghasilkan informasi yang berguna bagi pemakai, dan interpretasi atas hasil proses tersebut.

Proses akuntansi ini disebut istilah siklus akuntansi. Untuk melaksanakan proses akuntansi atau siklus akuntansi tersebut diperlukan beberapa media yang digunakan untuk melaksanakan proses akuntansi sampai dihasilkan laporan. Hasil akhir dari proses akuntansi yaitu kegiatan pelaporan adalah laporan keuangan. Selengkapnya tahap-tahap yang ada dalam proses akuntansi dapat dilihat pada pembahasan berikut.

Aktivitas pencatatan adalah tahap awal dari proses akuntansi yang berupa aktivitas untuk mencatat transaksi dan kejadian ekonomi dan keuangan perusahaan. Proses pencatatan dalam akuntansi ini dilakukan dalam media yang disebut Jurnal. Dalam aktivitas pencatatan, transaksi dicatat dalam suatu catatan dengan bentuk dua kolom yaitu debet dan kredit. Dalam tahap ini, satu kejadian atau transaksiakan dicatat dalam dua kolom yaitu debet dan kredit, dimana jumlah debet harus sama dengan jumlah kredit. Kaidah pencatatan debet dan kredit ini

mengikuti kaidah dalam persamaan akuntansi yang dapat dilihat pada pembahasan berikutnya.

Contoh jurnal adalah transaksi pembelian gedung secara kredit. Pada kejadian ini terjadi penambahan dalam dua pos yaitu akun gedung dan akun hutang. Gedung bertambah karena ada pembelian, sedangkan hutang bertambah karena adanya penambahan hutang sebagai akibat dari pembelian gedung. Pada jurnal transaksi ini akan dicatat gedung dengan nilai pembelian pada kolom debet, sedangkan hutang dengan nilai yang sama dengan harga beli gedung dicatat pada kolom kredit.

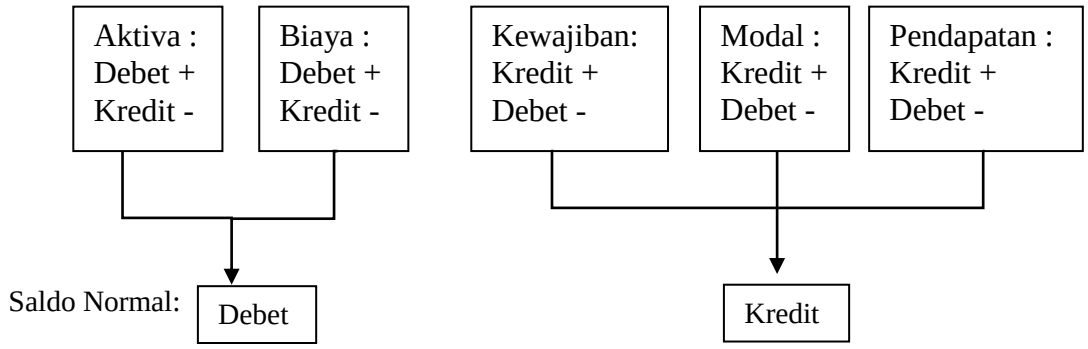
Proses berikutnya dalam siklus akuntansi adalah menggolongkan. Dalam tahap ini transaksi yang mempunyai jenis akun atau rekening yang sama dikelompokkan dalam satu catatan tersendiri yang disebut Buku Besar. Misalnya, transaksi yang mempengaruhi akun kas akan dicatat dalam buku besar kas. Buku besar adalah catatan akuntansi yang berisi transaksi-transaksi yang sejenis baik debet maupun kredit. Pada akhir periode, transaksi yang ada pada buku besar akan dijumlahkan sehingga akan ditemukan hasil akhir yang disebut saldo.

Peringkasan atau penjumlahan dari transaksi-transaksi yang ada dalam buku besar ini menganut kaidah dalam persamaan akuntansi. Kaidah dalam persamaan akuntansi ini adalah :

1. Pada Aktiva, debet adalah penambahan, sedangkan kredit adalah pengurangan.
2. Pada kelompok Kewajiaban, kredit adalah penambahan, sedangakan debet adalah pengurangan.

3. Pada kelompok Modal, kredit adalah penambahan, sedangkan debit adalah pengurangan.
4. Pada kelompok Biaya, debet adalah penambahan, sedangkan kredit adalah pengurangan.
5. Pada kelompok Pendapatan, kredit adalah penambahan sedangakan debet adalah pengurangan.

Istilah debet dan kredit di sini berfungsi untuk menunjukkan sisi kolom dari laporan. Debet akan terletak pada kolom kanan, sedangkan kredit merupakan sisi kolom kiri. Persamaan akuntansi dapat dilihat pada gambar II.1 berikut :



Gambar II.1 : Persamaan Akuntansi

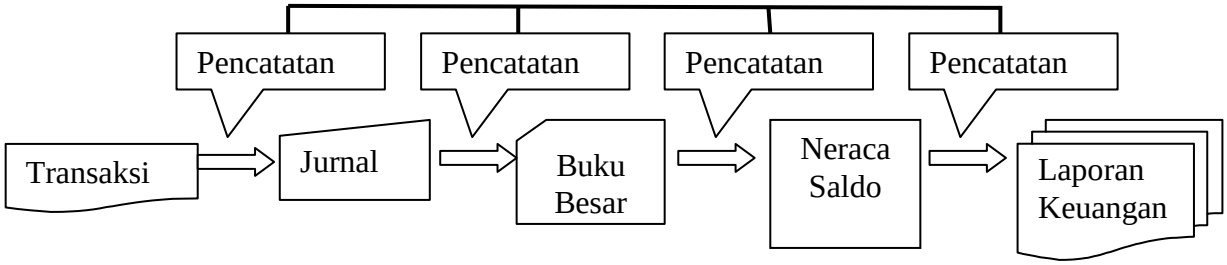
Sumber : Drs. Darsono, MBA.,Akt & Ashari, SE.,Akt (2005)

Tahap berikutnya setelah penggolongan adalah meringkas atau memasukkan saldo-saldo dari buku besar suatu catatan yang disebut neraca lajur. Neraca lajur ini akan berisi saldo atau hasil ringkasan atau penjumlahan dari transaksi-transaksi yang ada pada buku besar. Jika dilihat pada persamaan diatas dapat dilihat bahwa setiap transaksi dan hasil akhir dari transaksi merupakan jumlah debet dan kredit, dimana jumlah debet harus sama dengan jumlah kredit.

Jadi, neraca akan memuat hasil ringkasan (saldo) atau penjumlahan dari pos atau akun-akun di buku besar, baik yang bersaldo debet maupun yang bersaldo kredit.

Tahap berikutnya dari proses akuntansi adalah pelaporan. Pada tahap ini, ringkasan dari transaksi tersebut dikelompokkan dalam format standart laporan akuntansi berupa neraca, laporan laba rugi dan laporan berubahan modal. Neraca adalah ringkasan informasi dari kelompok aktiva, kewajiban dan modal. Jadi informasi yang ada dalam neraca adalah informasi saldo aktiva, kewajiban, dan modal. Laporan berikutnya adalah laporan perubahan modal. Laporan ini berisi informasi tentang perubahan modal pemilik selama satu periode yang dihasilkan dari jumlah debet dan jumlah kredit kelompok modal.

Siklus akuntansi tersebut dapat digambarkan pada gambar II.2 berikut ini :



Gambar II.2 : Siklus Akuntansi

Sumber : Drs. Darsono, MBA.,Akt & Ashari, SE.,Akt (2005)

II.3. Pengenalan Jaringan Komputer

II.3.1. Pengertian Jaringan Komputer

Dengan semakin berkembangnya kebutuhan pengolahan data dan inforamasi, di dalam sebuah perusahaan dibutuhkan beberapa komputer yang digunakan oleh banyak orang yang bekerja dalam sebuah tim. Untuk saling bertukar data dan informasi, maka komputer-komputer yang digunakan akan

terhubung antara satu dengan yang lainnya. Kumpulan komputer yang saling terhubung disebut sebagai jaringan komputer.

Keuntungan yang didapat dengan menggunakan jaringan komputer diantaranya adalah :

a. Dapat mengakses data dikomputer lain dari komputer yang anda gunakan;
b. Data yang digunakan dapat disimpan atau dicopy ke beberapa komputer, sehingga bila salah satu komputer rusak, maka salinan di komputer lain masih dapat anda gunakan;
c. Penggunaan printer, scanner, CD/DVD ROM dan perangkat lainnya dapat digunakan bersama-sama dengan pengguna lain.
d. *Administrator* jaringan dapat mengontrol data-data penting agar dapat diakses oleh pengguna yang berhak saja, sehingga data akan lebih terjamin.
e. Penghematan biaya dapat dilakukan, karena sebuah perangkat dapat digunakan secara bersama-sama.

Keunggulan tipe jaringan *client server* antar lain adalah:

a. Terdapat administrator jaringan yang mengelola sistem keamanan dan administrasi jaringan, sehingga sistem keamanan dan administrasi jaringan akan lebih terkontrol.
b. Komputer *server* difungsikan sebagai pusat data, komputer klien dapat mengakses data yang ada dari komputer klien manapun. Apabila terdapat komputer klien yang rusak, pengguna masih dapat mengakses data dari komputer klien yang lain.

c. Pengaksesan data lebih tinggi karena penediaan dan pengelolaan fasilitas jaringan dilakukan oleh komputer *server*, dan komputer *server* tidak terbebani dengan tugas lain sebagai *workstation*.
d. Pada tipe jaringan *client server*, sistem backup data lebih baik, karena backup data dapt dilakukan terpusat pada komputer *server*. Apabila data pada komputer klien/*workstation* mengalami masalah atau kerusakan masih tersedia backup pada komputer *server*.

Sedangkan kelemahan tipe jaringan *client server* antara lain adalah :

a. Biaya mahal, karena membutuhkan komputer yang memiliki kemampuan tinggi yang difungsikan sebagai komputer *server*.
b. Kelancaran jaringan tergantung pada komputer *server*. Bila komputer *server* mengalami gangguan maka jaringan akan terganggu (Madcom, 2010: 2-4).

Jaringan komputer adalah himpunan “interkoneksi” antara 2 komputer autinomous atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Bila sebuah komputer dapat membuat komputer lainnya restart, shutdown, atau melakukan kontrol lain, maka komputer-komputer tersebut bukan autonomous (tidak melakukan kontrol terhadap komputer lain dengan akses penuh) (Melwin Syafrizal, 2005: 2).

II.3.2. Jaringan

Jaringan adalah sebuah sistem yang terdiri dari atas komputer-komputer yang didesain untuk dapat berbagai sumber daya (printer, CPU), berkomunikasi

dan dapat mengakses informasi. Tujuan dari jaringan komputer adalah agar dapat mencapai tujuan, setiap bagian dari jaringan computer dapat meminta dan memberikan layanan (*service*). Pihak yang meminta/menerima layanan di sebut *client* dan memberikan/mengirim layanan di sebut *server*. Desain ini di sebut dengan sistem *client server*, dan digunakan pada hampir seluruh aplikasi komputer. (Melwin Syafrizal, 2005: 16).

II.3.3. Komputer

Komputer merupakan serangkaian ataupun sekelompok mesin elektronik yang terdiri dari ribuan bahkan jutaan komponen yang dapat saling bekerja sama, serta membentuk sebuah sistem kerja yang rapi dan teliti. Sistem ini kemudian dapat di gunakan untuk melaksanakan serangkaian pekerjaan secara otomatis, berdasarkan urutan intruksi ataupun program yang di berikan kepadanya, (Madcoms, 2010:6).

II.3.4. Tujuan dan Manfaat Jaringan Komputer

Manfaat jaringan komputer bagii *userr* dapat dikelompokkan menjadi dua, yaitu untuk kebutuhan perusahaan dan jaringan untuk umum. Tujuan utama dari terbangunnya sebuah jaringan komputer pada suatu perusahaan antara lain adalah:

a. Resource sharing, yang bertujuan agar seluruh program, peralatan, khususnya data, bisa digunakan oleh setiap orang yang ada pada jaringan tanpa terpengaruh oleh lokasi resource dan pemakai.
b. High *reliability* (keandalan tinggi), yang diperoleh karena tersedianya sumber daya alternatif. Misalnya, semua *file* dapat disalin (*back-up*) ke

semua mesin sehingga bila salah satu mesin mati, maka *file* tetap dapat diakses dari mesin lain yang masih aktif. Selain itu dengan adanya CPU yang banyak maka bila salah satu CPU tidak terpakai, maka CPU lain akan mengambil alih tugasnya, walaupun kinerjanya jadi menurun. Kemampuan melanjutkan pekerjaan saat mendapatkan masalah pada perangkat keras adalah suatu hal yang sangat penting.

c. *Saving money* (menghemat uang), komputer berukuran kecil mempunyai resiko/kinerja yang lebih baik dibanding komputer yang lebih besar. Komputer mainframe kira-kira memiliki kecepatan 10 kali lipat kecepatan komputer pribadi, akan tetapi mainframe 10 kali lebih mahal. Ketidakseimbangan rasio harga/kinerja ini menyebabkan para perancang sistem merasa lebih baik membangun sistem terdiri dari komputer-komputer kecil (PC).

Manfaat jaringan komputer untuk umum adalah jaringan komputer akan memberikan layanan yang berbeda kepada perorangan di rumah-rumah dibandingkan dengan layanan yang diberikan perusahaan. Terdapat tiga hal pokok yang menjadi daya tarik jaringan komputer pada perorangan yaitu:

a. *Access* informasi yang berada di tempat lain (serprti akses berita hari ini, *info e-government, e-commerce atau e-bussiness*), semuanya *uptodate*.
b. Komunikasi orang ke orang (*person to person seperti e-mail, chatting, video conference* dll).
c. Hiburan (seperti nonton acara tv on *line, radio streaming, download film* atau lagu, dll), Melwin Syafrizal, (2005: 14-15).

II.3.5. Jenis-jenis Jaringan Komputer

Berdasarkan jangkauan area atau lokasi, jaringan komputer dibedakan menjadi 3 jenis yaitu :

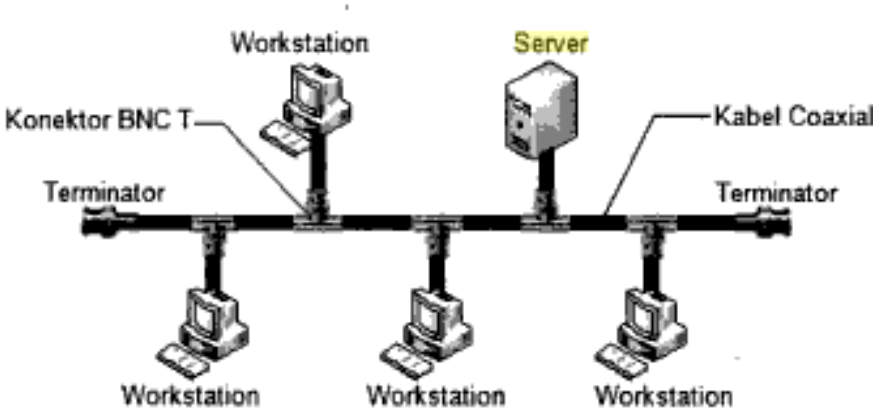
a. *Lokal Area Network* (LAN) merupakan jaringan yang menghubungjkan sejumlah komputer yang ada dalam suatu lokasi dengan area yang terbatas seperti ruang atau gedung. LAN dapat menggunakan media komunikasi seperti kabel dan wireless.
b. *Wide Area Network* (WAN) merupakan jaringan antara LAN satu dengan LAN lain yang dipisahkan oleh lokasi yang cukup jauh. Contoh penggunaan WAN adalah hubungan antara kantor pusat dengan kantor cabang yang ada di daerah-daerah.
c. *Metropolitan Area Network* (MAN) merupakan jaringan yang lebih besar dari jaringan LAN tetapi lebih kecil dari jaringan WAN. Jaringan MAN dan jaringan WAN sama-sama menghubungkan beberapa LAN yang membedakan hanya lingkup area yang berbeda.

II.3.6. Topologi Jaringan Komputer

Topologi jaringan merupakan gambaran pola hubungan antara komponen-komponen jaringan, yang meliputi komputer *server*, komputer klien/*workstation*, *hub/switch*, pengkabelan, dan komponen jaringan yang lain. Terdapat beberapa topolog jaringan yang dapat anda sesuaikan dengan kondisi di lapangan (Madcoms, 2010: 2-4).

a. Topologi BUS

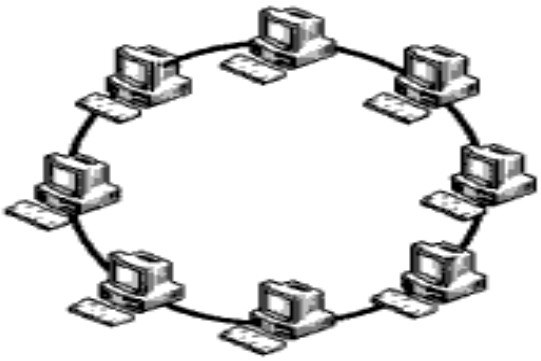
Topologi BUS, merupakan topologi yang menghubungkan beberapa komputer ke sebuah kabel dengan beberapa terminal. Topologi Bus, menggunakan jenis kabel *coaxial* dengan beberapa konektor BNC. Topologi bus menyediakan 1 jalur yang digunakan untuk komunikasi antar perangkat sehingga setiap perangkat harus bergantian dalam menggunakan jalur yang ada. Dalam berkomunikasi antar perangkat, hanya ada 2 perangkat yang dapat saling berkomunikasi. Kecepatan transfer rata-rata data antar perangkat sangat lambat karena harus bergantian dalam menggunakan jalur.



Gambar II. 3 Topologi BUS
(Sumber: Madcoms; 2010: 5)

b. Topologi Ring

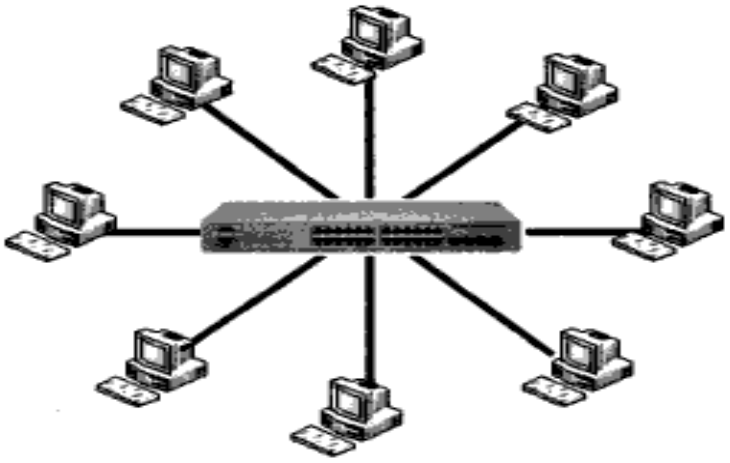
Topologi ring merupakan topologi yang menghubungkan beberapa komputer dengan membentuk lingkaran. Komputer yang terhubung dalam sebuah jaringan akan terkoneksi pada 2 komputer lain.



Gambar II. 4 Topologi Ring
(Sumber: Madcoms; 2010: 5)

c. Topologi Star

Topologi star merupakan topologi yang menghubungkan beberapa komputer dengan menggunakan perangkat yaitu *Hub* atau *Swith*. Perangkat ini berfungsi sebagai pengontrol dari semua komputer yang terhubung dalam jaringan komputer.



Gambar II. 5 Topologi Star
(Sumber: Madcoms; 2010: 5)

II.4. Pengenalan Jaringan *Client Server*

II.4.1. Pengertian Jaringan *Client-Server*

Jaringan *client server* menghubungkan komputer *server* dengan komputer klien/*workstation*. Komputer *server* adalah komputer yang menyediakan fasilitas bagi komputer-komputer klien/*workstation* yang terhubung dalam jaringa. Sedangkan komputer adalah komputer yang menggunakan fasilitas yang disediakan oleh komputer *server*. Komputer *server* pada sebuah jaringan tipe *client server* disebut dedicater *server*, kaerna komputer yang digunakan hanya sebagai penyedia fasilitas untuk komputer klien/*workstation*. Komputer *server* tidak dapat berperan sebagai komputer klien/*workstation* (Madcoms, 2010:3).

II.4.2. Komponen Jaringan *Client-Server*

Client-server adalah suatu bentuk arsitektur, dimana *client* adalah perangkat yang menerima, menampilkan dan menjalankan aplikasi (*software* komputer). Sedangkan *server* adalah perangkat yang menyediakan dan bertindak sebagai pengelola aplikasi, data, dan keamanannya. *Server* biasanya terhubung dengan *client* melalui kabel UTP dan sebuah kartu jaringan (network card). Kartu jaringan ini biasanya berupa kartu PCI atau ISA. *Client-server* adalah suatu model komunikasi dua buah komputer atau lebih yang berfungsi untuk melakukan pembagian tugas. *Client* bertugas untuk melakukan *input*, update, delete, dan menampilkan data dari sebuah *database* . Sementara *server* bertugas menyediakan layanan untuk manajemen data, seperti menyimpan dan mengolah *database*.

Perancangan arsitektur *client-server* harus mempertimbangkan struktur logika aplikasi yang terdiri dari

1. Lapisan presentasi yang berhubungan dengan penyajian informasi ke *userr* dan semua interaksi *userr*.
2. Lapisan pemrosesan aplikasi yang berhubungan dengani mplementasi logika aplikasi.
3. Lapisan manajemen data yang berhubungan dengan operasi *database* .

Aplikasi *client server* merupakan jawaban atas berkembangnya teknologi informasi, dimana di era sekarang ini sebuah perusahaan memiliki banyak departemen dan harus terhubung satu sama lain dalam melakukan akses datanya.

Sistem *client server* mempunyai dua komponen utama yaitu komputer *client* dan komputer *server*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer *client* dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu komputer atau *workstation* yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimiliki. Selain itu *client* komputer yang melibatkan proses-proses *client* yang meminta suatu pengelolaan arus kas data kepada komputer *server* yang menyediakan layanan data tersebut. Sehingga *client* maupun *server* sama-sama melakukan pekerjaan.

Melalui kombinasi *client* (*front-end*) dan *server* (*back-end*), maka kumpulan dari modul-modul tidak dieksekusi dalam memori yang sama namun

terbagi dalam komputer *client-server*. Dengan arsitektur *client/server*, modul yang memanggil menjadi *client* yang meminta suatu layanan data dan modul yang

Hal ini menjadi konfigurasi bagi komputer *client* dan komputer *server* bisa berbeda seperti kapasitas memori, kecepatan prosesor atau alat masukan dan keluaran yang disesuaikan dengan fungsi kerja dari elemen-elemen tersebut. Bagi *server* yang menjalankan tugas pengelolaan suatu *database* digunakan suatu konfigurasi yang khusus menangani tugasnya tersebut dengan sistem operasi yang dikhususkan bagi *server* seperti windows NT *server*, windows 2000 *server*, sedangkan komputer *client* menggunakan konfigurasi yang umum bagi sebuah komputer *desktop* yang terhubung ke jaringan dengan sistem operasi seperti Windows 98, Windows XP, dan lain-lain.

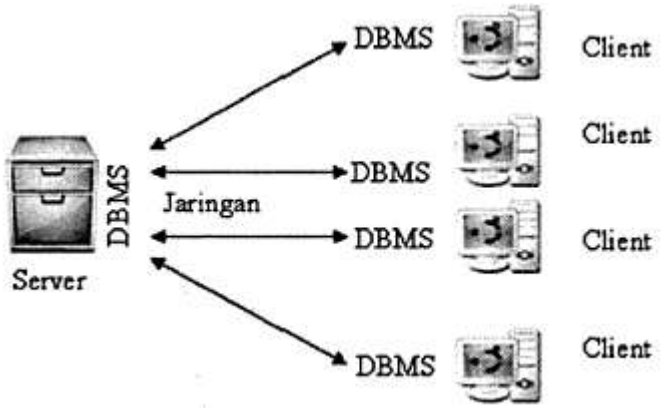
II.4.3. Konsep Jaringan *Client-Server*

II.4.3.1. Konsep One-Tier

Model pertama aplikasi pemrograman *database client*-serrver adalah model standalone atau 1 tier (1-tingkat) adalah sebuah komputer yang mengakses sebuah *database* dari komputer sendiri. Dengan kata lain, aplikasi antar muka *userr* dan aplikasi DBMS terdapat pada komputer yang sama.

Gambar II.6 Model 1-tier Komputer Standalona
(Sumber: Wahana Komputer; 2008, 6)

Arsitektur 1-tier dapat pula terjadi dalam sebuah jaringan *workstation* yang memiliki dua jenis komputer saling berhubungan, yaitu *client* dan *server*. Model pertama aplikasi pemrograman *database* yang menggunakan konsep *client-server* adalah model onetier atau yang dikenal dengan istilah stand alone. Konsep kerjanya adalah sebuah komputer mengakses *database* dari komputer itu sendiri. Dengan model ini Model pertama aplikasi pemrograman *database* yang menggunakan konsep *client-server* adalah model onetier atau yang dikenal dengan istilah stand alone. Konsep kerjanya adalah sebuah komputer mengakses *database* dari komputer itu sendiri. Dengan model ini dapat dipastikan beban traffic menjadi sangat besar, karena *file* yang diminta adalah *file database* secara keseluruhan. Model one-tier ini sangat cocok digunakan untuk bisnis kecil yang hanya membutuhkan sebuah komputer dalam memproses dan menyimpan datanya, tetapi tidak cocok jika diterapkan pada model jaringan. Kemudian, *file* data pada *database* akan diakses dan disalin oleh komputer klien untuk diproses.



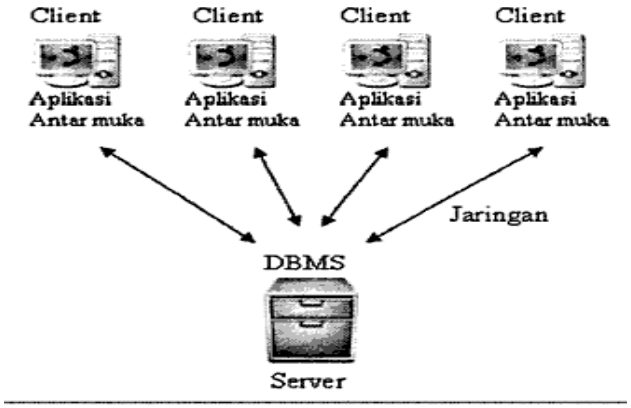
Gambar II. 7 Arsitektur 1-Tier Model Jaringan
(Sumber : Wahana Komputer; 2008, 6)

Adapun karakteristik arsitektur 1-tier sebaga berikut:

a. Beban jaringan menjadi tinggi karena yang diminta adalah *file database* secara keseluruhan pada komputer *server* ke komputer klien 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 memori.
c. Komputer clietn harus mempunyai kemampuan proses yang tinggi untuk mendapatkan waktu *response* yang baik saat komputer *server* mengirimkan *file* yang diminta.
d. Programer bertanggungjawab membuat aplikasi yang dapat menjaga integritas DBMS yang dipakai bersama-sama.
e. Arsitektur 1-tier cocok untuk bisnis kecil yang hanya membutuhkan sebuah komputer untuk merespons dan menyimpan data sekaligus, tetapi kurang tepat diterapkan pada model jaringan.

II.4.3.2. Konsep Two-Tier

Model kedua sebuah prmoograman *database* adalah model 2-tier. Konsep kerjanya adalah adanya pembagian tugas antara komputer *client* dan komputer *server*. Komputer *client* bertugas menyediakan *userr interface* untuk request data ke DBMS *server* serta pemrosesan data yang mencakup logical bussiness. Komputer *client* hanya mengirimkan sebuah statement untuk menambah, mengubah, menghapus dan meminta data untuk ditampilkan melalui *userr interface*. Sementara itu komputer *server* bertugas untuk melakukan penyimpanan, pengelolaan, melayani permintaan akses data dan pemrosesan oleh *client*, sebagaimana yang terlihat pada gambar berikut.



Gambar II. 8 *Arsitektur Client-Server 2-Tier*

(Sumber : Wahana Komputer; 2008, 6)

Karakteristik arsitektur 2-tier adalah:

a. 2-tier terjadi pada jaringan dan melakukan pemodelan pemrograman *database* dalam 2 tingkat. Tingkat pertama adalah *client* dan tingkat kedua adalah *server*.

b. Tingkat pertama komputer *client* sebagai penyedia aplikasi *userr* antarmuka untuk mengolah *database*, baik menampilkan data ke dalam *userr interface*, menambah, mengubah, menghapus data, maupun logina bisnis (*bussiness logic*).
c. Tingkat kedua adalah servcer yang menyediakan aplikasi DBMS untuk mengelola *database* serta menyediakan pula query, stored procedure, dan triggers yangd apat 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* yang bersangkutan
f. Komputer *server* dituntut memiliki kemampuan pemrosesan yang tinggi karena harus melyani permintaan banyak komputer *client* yang mengakses satu atau lebih daru DBMS.
g. Beban jaringan menjadi ringan karena data yang berjalan pada jaringan hanya data yang diminta oleh *client*.
h. Otentikasi 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.

II.4.3.3. Konsep There-Tier

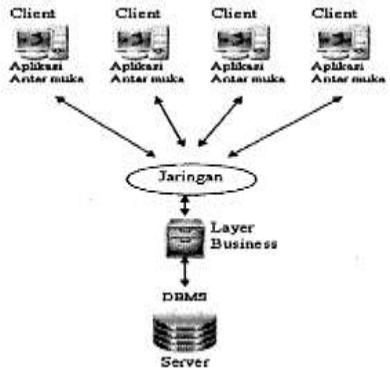
Arsitektur n-Tier berarti membagi komponen menjadi n enititas, yaitu 1 tier *client*, dan n-1 tier *server*. Arsitektur three-tier membagi sistem menjadi tiga tingkatan, yaitu :

1. Lapisan *client* (*presentation tier*), Lapisan ini bertugas untuk menyediakan *userr interface* yang mampu untuk melayani seluruh interaksi *userr*
2. Lapisan aplikasi (*bussiness logic tier*), Lapisan ini bertugas untuk melakukan seluruh pemrosesan dan menampung seluruh business rules. Lapisan inilah yang sesungguhnya bekerja untuk memecahkan masalah dan bertindak sebagai *middleware* antara *presentation layer* dan data *service layer*. *Business logic tier* ini tidak perlu mengetahui detail dari presentation layer maupun data service tier yang digunakan.
3. Lapisan *database* (*data service tier*), Lapisan ini bertugas melakukan manajemen data sebagaimana yang diminta oleh *presentation tier* melalui *business logic tier*. Jika *progammer* ingin melakukan perubahan pada *physical store*, maka hanya lapisan inilah yang dirubah tanpa merubah lapisan lainnya.

Model three-tier ini memiliki kelebihan sebagai berikut :

1. Segala sesuatu mengenai *database* terinstalasikan pada sisi *server*, begitu pula dengan konfigurasinya. Hal ini membuat harga yang harus dibayar lebih kecil.

2. Apabila terjadi kesalahan pada salah satu lapisan tidak akan menyebabkan lapisan lain ikut bermasalah, misalnya *terjadi* crash pada *database* tier maka tidak berpengaruh pada application tier.
3. Perubahan pada salah satu lapisan tidak perlu menginstalasi ulang pada lapisan yang lainnya baik pada sisi *server* ataupun sisi *client*.
4. Skala besar.
5. Keamanan dibelakang *firewall*.
6. Transfer informasi antara web *server* dan *database server* optimal.
7. Komunikasi antar sistem tidak harus didasarkan pada standar internet, tetapi dapat menggunakan *protocol* komunikasi yang lebih cepat dan berada pada tingkat yang lebih rendah.
8. Penggunaan *middleware* mendukung efisiensi *query database* dalam SQL dipakai untuk menangani pengambilan informasi dari *database* .



Gambar II. 9 *Arsitektur Client Server 3-Tier*

(Sumber : Wahana Komputer; 2008, 7)

Karakteristik model 3-tier membagi sistem menjadi 3 lapisan, yaitu lapisan *client*, lapisan *middle tier* (*bussiness logic*) dan lapisan *database server* (DBMS).

a. *Client* bertugas menyediakan antarmuka bagi *userr* untuk menggakses *database*.
b. Lapisan middle tier menyediakan perintah untuk mengelola *database*, seperti stored procedure, rumus untuk mengakses *database*, dan lain-lain.
c. Lapisan *server* DBMS Menyediakan ruang untuk menyimpan *database* yang dapat diakses melalui middle tier.
d. Mudah dalam melakukan perubahan pada *bussines logik*
e. Bussiness logic mudah untuk diterapkan dan diperlihara.
f. Lebih mahal dibandingkan denganm odel 2 tier
g. Memerlukan adaptasi yang luas apabila terjadi perubahan semua sistem
h. Aplikasi *client* dapat mengakses berbagai tipe DBMS berbeda dengan mudah walaupun berbeda platform (Wahana Komputer, 2008, 5-11)

II.5. *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 atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. Sebagai contoh masing-masing mahasiswa adalah entitas dan mata kuliah dapat pula dianggap sebagai entitas.

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

Struktur logis (skema *database*) dapat ditunjukkan secara grafis dengan *diagram* ER yang dibentuk dari komponen-komponen berikut :

Entitas	Persegi panjang mewakili kumpulan entitas
Atribut	Elips mewakili atribut
Relasi	Belah ketupat mewakili relasi
————	Garis menghubungkan atribut dengan kumpulan entitas dan kumpulan entitas dengan relasi

Masing-masing komponen diberi nama entitas atau relasi yang diwakilinya (Janner Simarmata ; 2006 : 59-60).

II.6. Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari sistem informasi. Dengan menggunakan KD, analis sistem dapat didefenisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dengan dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

II.6.1. Isi Kamus Data

Adapun isi dari kamus data adalah sebagai berikut :

1. Nama arus data.
2. Alias.
3. Bentuk data.

4. Arus data.
5. Penjelasan.
6. Periode.
7. Volume.
8. Struktur data. (Jogiyanto ; 2005 : 725-728).

II.7. Normalisasi

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

Tujuan normalisasi adalah membuat kumpulan tabel relasional yang bebas dari data berulang dan dapat dimodifikasi secara benar dan konsisten.

Pada proses normalisasi terhadap tabel pada *database* dapat dilakukan dengan beberapa tahap normalisasi antara lain :

1. Bentuk Normal Pertama (1NF)

   Contoh yang kita gunakan disini adalah sebuah perusahaan yang mendapatkan barang sejumlah pemasok. Meskipun berada pada 1NF, tabel pemasok mengandung data berulang. Sebagai contoh, informasi tentang lokasi pemasok dan status lokasi harus diulang untuk setiap barang yang dipasok.

2. Bentuk Normal Ke Dua (2NF)

Defenisi bentuk normal kedua menyatakan bahwa tabel dengan kunci utama gabungan hanya dapat berada pada 1NF, tetapi tidak pada 2NF. Sebuah tabel relasional berada pada bentuk normal kedua jika dia berada pada 1NF dan setiap kolom bukan kunci yang sepenuhnya tergantung pada kunci utama.

3. Bentuk Normal Ke Tiga (3NF)

   Bentuk normal ketiga ini mengharuskan semua kolom pada tabel relasional tergantung hanya pada kunci utam. Secara defenisi, sebuah tabel berada pada bentuk normal ketiga (3NF) jika tabel sudah berada pada 2NF dan setiap kolom yang bukan kunci tidak tergantung secara transitif pada kunci utamanya. (Janner Simarmata ; 2006 : 77-82).

II.8. Basis Data (*Database*)

II.8.1. Pengertian Basis Data

Database (bass data) merupakan kumpulan data yang saling berhubungan satu dengan lainnya yang tersimpan di perangkat keras komputer dan diperlukan suatu perangkat lunak untuk memanipulasi basis data tersebut. Buku telepon, katalog *file*m merupakan contoh dari basis data (Junindar, 2008: 19).

Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan dari *field* atau kolom. Struktur *field* yang mernyusun sebuah *database* adalah Data Record dan *Field*.

a. Data adalah satu satuan informasi yang akan diolah, sebelum diolah data dikumpulkan di dalam suatu *file database.*

b. *Record* adalah data yang isinya merupakan satu kesatuan seperti Nama *Userr* dan *Password.* Setap keterangan yang mencakup Nama *Userr* dan *Password* dinamakan satu *record.* Setiap *record* diberi nomor urut yang disebut recird (record number).
c. *Field* adalah sub bagian dari *record,* dari contoh di atas maka terdiri dari 2 *field*, yatu: *Field* Nama *Userr* dan *Password* (Anhar, 2010: 45-46).

II.8.2. Sistem Manajemen Basis Data

Keberhasilan suatu sistem informasi sangat dipengaruhi oleh sistem basis data. Sistem basis data merupakan salah atu elemen penyusun sistem. Apabila sistem basis data ini benar-benar lengkap, akurat dan mudah untuk ditampilkan kembali maka hal itu akan meningkatkan kualitas dari sistem manajemen basis data (Kusrini dan Andri Koniyo, 2007: 139).

Sistem manajemen basis data (*database* management sistem/DBMS) adalah perangkat lunak yang digunakan untuk mengendalikan data, termasuk penyimpanan data, pengambilan data, keamanan data, dan integrasi data. Fungsi utama DBMS adalah untuk menyediakan lingkungan yang nyaman dan efisien untuk digunakan dalam pengambilan dan penyimpanan informasi di basis data (Junindar, 2008: 19).

Untuk mengetahui perbedaan antara basis data dengan sistem basis data, kita harus mengetahui definisi keduanya. Menurut James F. Courtney dan David B. Paradice, sistem basis data mempunyai pengertian sebagai berikut: “Sistem basis data adalah kumpulan basis data dengan para pemakai yang menggunakan

basis data secara bersama-sama, personal-personal yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer yang mendukungnya" (Sutanta, 1996 dalam Kusrini dan Andri Koniyo, 2007: 139).

Pengertian basis data ini diperjelas oleh James Martin (1990), yang mengatakan sebgai berikut: "Bais data adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu suatu kerangkapan data denganc ara-cara tertentu sehingga mudah untuk digunakan dan ditampilkan kembali, dapat digunakan untuk satu atau lebih program aplikasi secara optimal, data dapat disimpan tanpa mengalami ketergantungan pada program yang akan menggunakannya, serta disimpan sedemikian rupa sehingga penambahan, pengambilan dan modifikasi data dapat dilakukan dengan mudah dan terkontrol" (Kusrini dan Andri Koniyo, 2007: 140).

II.8.3. Arstektur Basis Data

Terdapat dua bentuk arsitektur sistem basis data, yaitu sistem terpusat dan sistem *client-server*. Sistem basis data terpusat adalah sistem basis data yang dijalankan pada sistem komputer tunggal dan tidak berinteraksi dengan sistem pada komputer lain. Pengguna terkoneksi ke komputer pusat melalui terminal. Sedangkan sistem basis data *client-server* adalah sistem basis data yang memisahkan program pengguna dengan program basis data di sistem yang berbeda. Pengguna terkoneksi ke pusat data yang disebut *server* sistem melalui

suatu program pengguna (*userr interface*) yang terdapat pada komputer. Sistem tempat program pengguna berada disebut *client* system (Junindar, 2008: 20).

II.8.4. Elemen Basis Data

Adapun elemen-elemen sistem manajemen basis data adalah sebagai berikut:

a. *Database, Database* adalah sekumpulan dari sistem data yang saling berhubungan satu sama lain, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di *hardware* komputer, dan harus menggunakan *software* untuk melakukan manipulasi tertentu.
b. *File, File* adalah kumpulan *record* sejenis yang mempunyai panjang *elemen* dan *atribut* yang sama, namun valuenya berbeda. *Database* dibentuk dari kumpulan *file. Fiel* di dalam pemrosesan aplikasi dapat dikategorikan ke dalam beberapa tipe, diantaranya adalah sebagai berikut:
    a. *File* induk *(master file*)
        a) *File* induk acuan (*reference* master *file)*: *file* induk yang *record*nya *relatif statis*, jarang berubah nilainya. Misalnya *file* daftar gaji, *file* mata pelajaran.
        b) *File* induk dinamik (*dynamic master file*): *file* induk yang nilai *record-record*nya sering berubah atau sering dimuthakirkan (*update*) sebagai hasil dari suatu transaksi. Misalnya *file* induk data barang, yang setiap saat harus di *update* bila terjadi transaksi.

b. *File* transaksi (*transaction file*), *File* ini bisa disebut *input file*, digunakan untuk merekam data hasil transaksi yang terjadi. Misalnya *file* penjualan yang berisi data hasil transaksi penjualan.

c. *File* Laporan (*report file*), *File* ini bisa disebut *output file*, yaitu *file* yang berisi informasi yang akan ditampilkan.

d. *File* sejarah *(history file*), *File* ini bisa disebut *file* arsip (*archival file*), merupakan *file* yang berisi data masa lalu yang sudah tiadk aktif lagi tetapi masih disimpan sebagai arsip.

e. *File* pelindung (*backup file*), *File* ini merupakan salinan dari *file-file* yang masih aktif di dalam *database* pada suatu saat tertentu. *File* ini digunakan sebagai pelindung atau cadangan bila *file database* yang akif mengalami kerusakan atau hilang.

f. *Record, Record* adalah kumpulan elemen yang saling berkaitan yang menginformasikan tentang satu enitas secara lengkap. Satu record mewakili satu data atau informasi.

g. *Field*, *Field* adalah bagian tertentu dari data dalam *record* yang mewakili satu *entitas*, misalnya *file* anggota dapat dilihat dari *field*nya, seperti kode anggota, nama dan lain-lain.

h. Data *Value,* Data *value* adalah data aktual atau informasi yang disampaikan pada setiap data *elemen* atau *field* data, misalnya *field* nama anggota memiliki data *value* Susi, Widi dan sebagainya.

i. *Enity, Enity (entitas)* adalah objek *riil* yang dapat dibedakan satu sama lain dan tidak saling bergantung. Misal, pada bidang sirkulasi, *entitas*nya adalah anggota dan buku.

j. *Query, Query* merupakan perintah yang dirancang untuk memanggil keloompok *record* tertentu dari satu *file* atau lebih untuk melakukan operasi pada *file.*

k. *View, View* adalah data yang terdiri dari sejumlah *record* yang diproses dalam urutan penampilan (Kusrini dan Andri Koniyo, 2007: 141-143).

II.8.5. *Relational Database Management System* (RDBMS)

RDBMS merupakan sekumpulan data yang disimpan sedemikian rupa sehingga mudah diambil informasinya bagi pengguna, dan data tersebut saling berhubungan. RDBMS merupakan paket perangkat lunak yang kompleks yang digunakan untuk memanipulasi *database* (Kusrini dan Andri Koniyo, 2007: 143).

Ada tiga prinsip dalam RDBMS, yaitu:

1. *Data definition*

   Mendefinisikan jenis data yang akan dibuat (dapat berupa angka atau huruf), cara *relasi* data, *validasi* data dan lainnya.

2. *Data manipulation*

   Data yang telah dibuat dan didefinisikan akan dikenai beberapa perlakuan, seperti penyaringan, peng-*query*-an dan lain sebagainya.

3. *Data control*

Bagian ini berkenaan dengan cara mengendalikan data, seprti siapa saja yang bisa melihat isi data, bagaimana data bisa digunakan oleh banyak *userr*, dan sebagainya.

Semua operasi *input* dan *output* yang berhubungan dengan *database* harus menggunakan *DBMS*. Bila pemakai akan *mengakses database*, *DBMS* akan menyediakan penghubung (*interface*) antara pemakai dengan *database* (Kusrini dan Andri Koniyo, 2007: 143).

II.9. Pengenalan Aplikasi *Database MySQL*

II.9.1. Pengertian MySQL

MySQL (MyStructure Query Language) adalah salah satu *Database* Managemen Sistem (DBMS) dari sekian banyak DBMS seperti *Oracle, MySQL, Postrahre SQL* dan lainnya. *MySQL* berfungsi untuk mengolah *database* menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis (Anhar, 2010: 45).

MySQL (My Structure Query Language) adalah sebuah perangkat lunak sistem manajemen *basis data SQL (Database Management System*) atau DBMS dari sekian banyak *DBMS*, seperti *Oracle, MS SQL, Porstagre SQL*, dan lain-lain. *MySQL* merupakan *DBMS* yang *multithread, multi userr* yang bersifat gratis dibawah *lisensi GNU General Public Licence (GPL).* Tidak seperti *Apache* yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki penulisnya masing-masing (Anhar, 2010: 21).

Salah satu *database* untuk *server* adalah *MySQL,* jenis *database* ini sangat populer dan digunakan pada banyak *website* di *internet* sebagai bank data. *MySQL* menggunakan *SQL* dan bersifat *free (open source*). *MySQ*L dapat berjalan diberbagai *platform* seperti *Linux, Windows* dan lain sebagainya (Mei Lenawati, 248).

II.9.2. Konsep Pemrograman *MySQL*

Sebuah *website* yang dinamis membutuhkan tempat penyimpanan data agar pengunjung dapat memberikan komentar, saran, dan masukan atas website yang dibuat. Tempat penyimpanan data berupa informasi dalam sebuah tabel disebut dengan *database*. Program yang digunakan untuk mengolah dan mengelola *database* adalah *MySQL* yang memiliki sekumpulan prosedur dan struktur sedemikian rupa sehingga mempermudah dalam menyimpan, mengatur, dan menampilkan data (Anhar, 2010: 45).

II.9.3. Kelebihan Aplikasi *Database MySQL*

Beberapa kelebihan *MySQL,* antara lain:

1. *MySQL* dapat berjalan dengan stabil pada berbagai sistem operasi, seperti *Windows, Linux, FreeBSD, Mac OS X Server, Solaris*, dan masih banyak lagi.
2. Bersifat *Open Souece, MySQL* didistribusikan secara *open source* (gratis), dibawah *lisensi GNU General Public Licence (GPL).*

3. Bersifat *Multiuserr, MySQL* dapat digunakan oleh beberapa *userr* dalam waktu yang bersamaan tanpa mengalami masalah.
4. *MySQL* memiliki kecepatan yang baik dalam menangani *query* (perintah *SQL*). Dengan kata lain, dapat memproses lebih banyak *SQL* per satuan waktu.
5. Dari segi *security* atau keamanan data, *MySQL* memiliki beberapa lapisan *security*, seperti *level subnet mask*, nama *host*, dan izin akses *userr* dengan sistem perizinan yang mendetail serta *password* yang *terenskripsi.*
6. Selain *MySQL* bersifat *fleksibel* dengan berbagai pemrograman, *MySQL* juga memiliki *interface* (antar muka) terhadap berbagai *aplikasi* dan bahasa pemrograman dengan menggunakan fungsi *API (Application Programing Interface).*
7. Dukungan banyak komunitas, biasanya bergabung dalam sebuah *forum* untuk saling berdiskusi membagi informasi tentang *MySQL*. Misalnya, di forum *http://forums.mysql.com (*Anhar, 2010: 22).

II.10. Pengenalan Visual *Basic* Net

Visual Basic .Net merupakan bahasa pemrograman terbaru keluaran *Microsoft* yang merupakan kelanjutan dari *Visual Basic* 6.0. Seperti halnya pada *Visual Basic* 6.0, aplikasi yang dapat dikembangkan oleh *Visual Basic .Net* antara lain adalah *aplikasi database. Aplikasi* yang dihasilkan oleh *Visual Basic Net* akan berjalan di lingkungan *GUI (Graphical Userr Interface*), dimana beberapa

program *modern* telah berjalan di lingkungan tersebut (Eko Priyo Utomo, 2006:11).

II.11. Pengenalan UML (*Unified Modeling Language*)

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

UML diaplikasikan untuk maksud tertentu, bisanya diguanakan untuk :

1. Merancang perangkat *lunak;*
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis;
3. Menjabarkan sistem secara rinci untuk *analisa* dan mencari apa yang diperlukan *sistem;*
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorienstasi objek. Hal ini disebakan karena UML mneyediakan bahasa pemodelan *visual* yang

memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang *efektif* untuk berbagai *(sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain (Munawar, 2005: 16).

Pada umumnya metode-metode yang ditujukan untuk pembangunan aplikasi berorientasi objek menggunakan UML untuk memodelkan berbagai artefak dari perangkat lunak. UML adalah sekumpulan simbol dan *diagram* untuk memodelkan *software*. Dengan menggunakan UML, desain *software* dapat diwujudkan dalam bentuk simbol dan *diagram*. Desain dalam bentuk simbol dan *diagram*, kemudian dapat diterjemahkan menjadi kode program. Telah tersedia *tools* yang dapat membuat kode program berdasar UML *Class Diagram*. *Implementasi* kode program dari *diagram* UML dapat menggunakan bahasa pemrograman apa saja dengan syarat bahasa pemrograman tersebut mendukung pemrograman berorientasi objek (Farid Aziz, 2005: 116).

II.11.1.*User Case* Diagram

User case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna system. *User case diagram* terdiri dari atas *diagram* untuk *user case* dan *actor*. *Actor* memprestasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan system aplikasi. *Actor* digambarkan sebagai berikut.

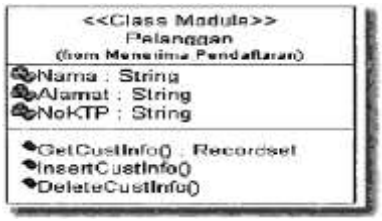
Gambar II. 10 Notasi *Actor*
(Sumber : Julius Hermawan ; 2007: 14)

II.11.2.*Class*

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi yang sama. *Class* digunakan untuk mengimplementasikan *interface*. Claass digunakan untuk mengabstraksikan *elemen-elemen* dari sistem yang sedang dibangun. *Class* bisa memperentasikan baik perangkat lunak maupun perangkat keras, baik konsep maupun benda nyata.

Notasi class berbentuk persegi panjang berisi 3 bagian persegi paling atas untuk nama *class*, persegi panjang paling bawah untuk operasi, dan persegi panjang di tengah untuk *atribut*.

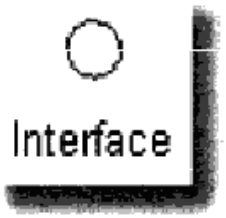
Atribut digunakan untuk menyimpan informasi. Nama *atribut* menggunakan kata benda yang bisa dengan jelas mempresentasikan informasi yang disimpan di dalamnya. Operasi menunjukkan sesuatu yang bisa dilakukan oleh *objek* dan menggunakan kata kerja.



Gambar II. 11 Notasi *Class*
(Sumber : Julius Hermawan; 2007: 14)

II.11.3.*Interface*

Interface merupakan sekumpulan operasi tanpa implementasi dari satu *class*. Implementasi operasi dalam *interface* dijabatkan oleh operasi dalam *class*. Oleh karena itu keberadan *interface* selalu disertai *class* yang mengimplementasikan operasinya. *Interface* ini merupakan salah satu cara mewujudkan prinsip enkapsulasi dalam objek.



Gambar II. 12 Notasi *Interface*
(Sumber : Julius Hermawan; 2007: 15)

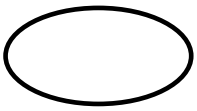
II.11.4.*User Case*

User case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan namun *user user case* hanya menjelaskan yang dilakukan oleh *actor* dan sistem, bukan bagaimana *actor* dan sistem melakukan kegiatan tersebut.

Di dalam *user case* terdapt teks untuk menjelaskan urutan kegiatan yang disebut *user case* specification. *User case specification* terdiri dari :

1. Nama *User Case*: mencantumkan nama dari *user case* yang bersangkutan. Sebaiknya diawali dengan kata kerja untuk menunjukkan aktivitas.
2. *Deskripsi* singkat (*brief description*): menjelaskan secara singkat dalam 1 atau 2 kalimat tentang tujuan dari *user case* ini.

3. Aliran normal *(basic flow*): ini adalah jantung dari *user case*. Menjelaskan *interaksi* antara *actor* dan sistem dalam kondisi normal, yaitu segala sesuatu berjalan dengan lancar, tiada halangan atau hambatan dalam mencapau tujuan dari *user case*.
4. Aliran *alternatif (alternate flow*): merupakan pelengkap dari *basic flow* karena tidak ada yang sempurna dalam setiap kali *user case* berlangsung. Di dalam *alternate flow* ini dijelaskan apa yang akan terjadi bila suatu halangan atau hambatan terjadi sewaktu *user case* berlangsung. Ini terutama berhubungan dengan *error* yang mungkin terjadi, misalnya karena sistem kekurangan data untuk diolah (*usia pegawai belum diinput*), terjadi masalah internal sistem komputer (*folder* terhapus), terjadi masalah eksternal (*printer* belum di *turn-on).*
5. *Special requirement*: berisi kebutuhan lain yang belum tercakup dalam aliran normal dan *alternatif*. Biasanya secara tegas dibedakan bahwa *basic flow* dan *alternate flow* menangani kebutuhan *fungsional* dari *user case*, sementara *special requirement* yang tidak berhubungan dengan kebutuhan *fungsional*, misalnya kecepatan transaksi maksimum berapa cepat dan berapa lama, kapasitas akses yaitu jumlah *userr* yang akan mengakses dalam waktu bersamaan.
6. *Pre-condition*: menjelaskan persyaratan yang harus dipenuhi sebelum *user case* bisa dimulai.
7. *Post-condition*: menjelaskan kondisi yang berubah atau terjadi saat *user case* selesai dieksekusi.



Gambar II. 13 Notasi *User Case*
(Sumber : Julius Hermawan; 2007: 16)

II.11.5.*Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar objek maupun hubungan antar objek. Biasanya *interaction* ini dilengkapi juga dengan teks, bernama operation signature yang tersusun dari nama operasi, parameter yang dikirim dan tipe parameter yang dikembalikan.

Gambar II. 14 Notasi *Interaction*
(Sumber : Julius Hermawan; 2007: 18)

II.11.6.*Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokan *elemen-elemen* darisistem yang sedang dibangun, sehingga bisa dibuat mode yang lebih sederhana. Tujuannya adaah untuk mempermudah pengelihatan (*visibility*) dari model yang sedang dibangun.

Gambar II. 15 Notasi *Package*
(Sumber : Julius Hermawan; 2007: 19)

II.11.7.Note

Note digunakan untuk memberikan keterangan dan komentar tambahan dari suatu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditempelkan ke semua *elemen notasi* yang lain.

Gambar II. 16 Notasi Note
(Sumber : Julius Hermawan; 2007: 19)

II.11.8.Dependency

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu *elemen* memberi pengaruh pada elemen lain. *Elemen* yang ada di bagian tanda panah adalah elemen yang tergantung pada elemen yang ada di bagian tanpa tanda panah.

Gambar II. 17 Notasi Dependency
(Sumber : Julius Hermawan; 2007: 20)

II.11.9.Association

menggambarkan navigasi antar *class (navigation*), berapa banyak objek lain yang bisa berhubungan dengan satu objek (*multiplicity antar class),* dan apakah suatu *class* menjadi bagian dari *class* lainnya (*aggregation*).

Gambar II. 18 Note Association
(Sumber : Julius Hermawan; 2007: 21)

II.11.10. Generalization

Generalization menunjukkan hubungan antara elemen yang lebih umum ke *elemen* yang lebih spesifik. Dengan *generalization*, *class* yang lebih spesifik (*subclass*) akan menurunkan atribut dan operasi dari *class* yang lebih umum (*superclass*), atau ”*subclass* is a super*class*”. Dengan menggunakan notasi *generalization* ini konsep *inheritance* dari prinsip hirarki dimodelkan.

Gambar II. 19 Note Generalization
(Sumber : Julius Hermawan; 2007: 22)

II.11.11. Realization

Realization menunjukkan hubungan bahwa *elemen* yang ada di bagian tanpa panah akan merealisasikan apa yang dinyatakan oleh elemen yang ada dibagian dengan panah. Misalny a*class* merealisasikan *package*, component merealisasikan *class* atau *itnerface*.

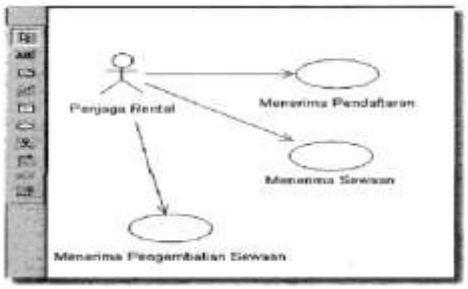
Gambar II. 20 Note Realization
(Sumber : Julius Hermawan; 2007: 22)

II.11.12. *User Case Diagram*

User case diagram (UCD) menjelaskan apa yang akan dilakukan oleh sistem yang akan dibangun dan siapa yang berinteraksi dengan sistem. UCD menjadi dokumen kesepakatan antara *Customer*, *Userr* dan Developer. *Userr* menggunakan dokumen UCD ini untuk memahami sistem dan mengevaluasi bahwa benar yang dilakukan sistem adalah untuk memecahkan masalah yang

userr ajukan atau sedang dihadapi. *Developer* menggunakan dokumen UCD ini sebagai rujukan yang benar dalam mengembangkan sistem.

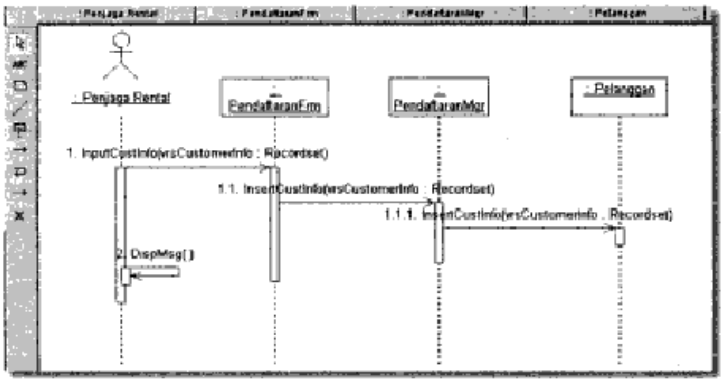
User Case Diagram pada umumnya tersusun dari elemen *actor*, *user case*, dependency, generalization, dan association. UCD ini memberikan gambaran statis dari sistem yang sedang dibangun dan merupakan artifak dari proses analisis.



Gambar II. 21 Note *User Case Diagram*
(Sumber : Julius Hermawan; 2007: 23)

II.11.13. *Sequence Diagram*

Squence *diagram* menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *user case*. Interaksi yang terjadi antar clas, operasi apa saja yang terlibat, urutan antar operasi, dan intormasi yang diperlukan oleh masing-masing operasi. Pembuatan squence *diagram* merupakan aktivitas yang paling kritikal dari proses diasin karena artifak nilah yang menjadi pedoman dalam proses pemrograman nantinya dan berisi aliran kontrol dari program. Oleh karena itu berharga untuk meluangkan waktu lebih lama di pembuatan *sequence diagram* ini untuk menghasilkan squence *diagram* yang terdisain dengan baik.

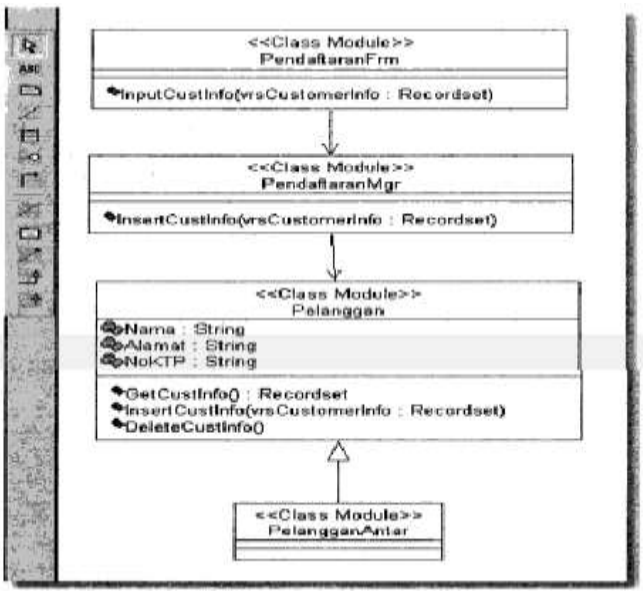


Gambar II. 22 Note *Sequence Diagram*
(Sumber : Julius Hermawan; 2007: 24)

II.11.14. *Class Diagram*

Sama seperti halnya *class*, maka *class diagram* merupakan *diagram* yang selalu ada di pemodelan sistem berorientasi objek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai tujuan.

Clas *diagram* umumnya tersusun dari elemen *Class*, *Interface*, Dependency, Generalization dan Association. Relasi dependency menunjukkan bagaimana ketergantungan terjadi antar *class* yang ada. Relasi generalization menunjukkan bagaimana suatu *class* menjadi *superclass* dari *class* lainnya dan *class* yang lain tesebut menjadi *subclass* dari *class* tersebut (Julius Hermawan, 2007: 14-28).



Gambar II. 23 Note *Class Diagram*
(Sumber : Julius Hermawan; 2007: 27)