BAB II

TINJAUAN PUSTAKA

2.1 Administrasi

Administrasi adalah Ilmu yang mempelajari proses kegiatan kerjasama untuk mencapai tujuan yang telah ditentukan. Kegiatan kerjasama itu sendiri merupakan gejala yang sifatnya *universal* dan memerlukan suatu proses pergerakan yang disebut dengan manajemen. Dengan demikian, untuk mencapai tujuannya administrasi perlu membentuk suatu manajemen dalam suatu organisasi sebagai wadah, kerangka, atau struktur untuk menjalin suatu kerjasama yang baik.

2.1.1 Pengertian Administrasi

Secara etimologis administrasi berasal dari bahasa latin yang terdiri dari kata *ad* yang berarti intensif dan *ministraire* yang berarti melayani. Literatur lain menjelaskan bahwa administrasi merupakan terjemahan dari bahasa Inggris yaitu *administration* yang bentuk infinitifnya adalah *to administer.*

Suwarno Handayaningrat dalam bukunya menulis “Kata administrasi berasal dari bahasa Belanda, yaitu *administratie* yang meliputi kegiatan pembukaan ringan, mencatat, menyurat, mengetik, agenda dan sebagainya yang bersifat teknis ketatausahaan“

Sementara The Liang menyatakan bahwa “administrasi adalah segenap rangkaian kegiatan penataan terhadap pekerjaan pokok yang dilakukan oleh sekelompok orang dalam kerja sama mencapai tujuan tertentu”.

Dan William H. Newman dalam bukunya yang berjudul *Administrative Action The Techniques of Organization and Management* mendefinisikan "administrasi sebagai pembimbingan, kepemimpinan dan pengawasan usaha-usaha suatu kelompok orang-orang ke arah pencapaian tujuan bersama"**.**

Berdasarkan uraian diatas, penulis mengambil kesimpulan bahwa administrasi adalah kegiatan penyusunan dan pencatatan data secara sistematis dengan maksud untuk menyediakan informasi serta memudahkan memperolehnya kembali secara keseluruhan.

2.1.2 Administrasi di SMP Negeri 2 Kertasari

Administrasi di SMP Negeri 2 Kertasari meliputi administrasi program pengajaran, administrasi kesiswaan, administrasi kepegawaian, administrasi keuangan dan administrasi perlengkapan atau barang.

1. Administrasi Program Pengajaran

   Administrasi program pengajaran dilakukan dengan tujuan memudahkan Kepala Sekolah dalam menyelenggarakan administrasi sekolah. Kegiatan yang dilakukan meliputi menyusun jadwal pelajaran, program pengajaran, pencatatan nilai dan pelaporan hasil belajar

2. Administrasi Kesiswaan

   Dalam adminstrasi kesiswaan selama satu tahun pelajaran dibagi dalam 3 tahap waktu dan dalam tiap tahapan waktu terdapat beberapa jenis kegiatan, yaitu :

a. Awal Tahun Pelajaran

Penerimaan Siswa Baru

b. Selama Tahun Ajaran

- Penyusun Data Pribadi Siswa
- Keadaan Siswa Awal Tahun
- Kehadiran Siswa

c. Akhir Tahun Pelajaran

- Pelaksanaaan Ujian Nasional
- Kenaikan Kelas

3. Administrasi Kepegawaian

Administrasi kepegawaian menguraikan kegiatan yang berkaitan dengan pengaturan kepegawaian, tugas dan tanggung jawab pengelolaan satuan pendidikan dan peningkatan tata usaha kepegawaian di sekolah.

4. Administrasi Keuangan

Administrasi keuangan bertugas dan bertanggung jawab dalam pengelolaan keuangan sekolah.

5. Administrasi Perlengkapan atau Barang

Administrasi perlengkapan atau barang memiliki tugas dalam perencanaan, pengadaan, penyimpanan dan pemeliharaan semua perlengkapan atau barang yang ada di sekolah.

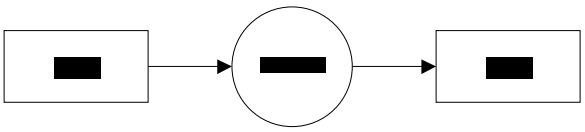
Administrasi program pengajaran, kesiswaan, kepegawaian dibentuk kedalam satu divisi yaitu divisi tata usaha, sedangkan untuk administrasi keuangan dibentuk secara khusus kedalam divisi keuangan.

2.2 Konsep Dasar Sistem

Sistem terdiri dari elemen-elemen yang berhubungan melalui berbagai interaksi untuk mencapai tujuan tertentu. Elemen-elemen tersebut mempunyai sifat yang tidak dapat dipisahkan dan saling mempengaruhi proses secara keseluruhan.

Dalam mendefinisikan suatu sistem, terdapat dua cara pendekatan yang sering digunakan yaitu pendekatan yang menekankan pada prosedur dan pendekatan yang lebih menekankan pada elemen atau komponen. Pendefinisian sistem dengan pendekatan pada prosedur menurut H.M.Jogiyanto “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”. Dan pendekatan yang lebih menekankan pada elemen atau komponennya didefinisikan sebagai berikut “Sistem adalah kumpulan dari elemen-elemen yang berorientasi untuk mencapai suatu tujuan”.

Secara garis besar sistem diartikan sebagai kumpulan elemen-elemen yang berinteraksi untuk mencapai suatu tujuan. Sistem mempunyai karakteristik berupa komponen-komponen, batasan, sistem, lingkungan luar sistem, penghubung sistem, masukan, keluaran, pengelolaan dan sasaran atau tujuan. Model dasar sebuah sistem yang paling sederhana terdiri dari masukan, pengolahan dan keluaran.



Gambar 2.1 Bentuk Umum Sistem

2.3 Konsep Dasar Informasi

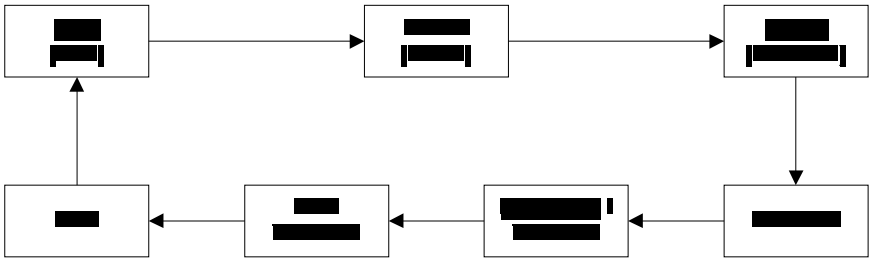
H.M.Jogiyanto dalam bukunya yang berjudul Analisis dan Desain Sistem Informasi menulis “Sistem diartikan sebagai jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran“.

Informasi merupakan hasil pengolahan dari sebuah model, formasi, organisasi ataupun suatu perubahan bentuk dari data yang memiiki nilai tertentu dan bisa digunakan untuk menambah pengetahuan bagi yang menerimanya. Dalam hal ini, data bisa dianggap sebagai obyek dan informasi adalah salah satu subyek yang bermanfaat bagi penerimanya. Informasi juga bisa disebut sebagai hasil pengolahan ataupun pemrosesan data yang mempunyai nilai bagi penerimanya dan dapat digunakan untuk dasar pengambilan keputusan.

2.3.1 Siklus Informasi

Sumber dari informasi adalah data yang merupakan bentuk jamak dari bentuk tunggal item-item. Data merupakan bentuk yang masih mentah sehingga perlu diolah melalui suatu model untuk menghasilkan sebuah informasi yang dapat dijadikan sebagai acuan untuk mengambil keputusan oleh penerima informasi dengan

menghasilkan suatu tindakan yang akan membuat sejumlah data kembali. Data tersebut dijadikan sebagai *input* kemudian diproses lagi melalui suatu model dan seterusnya demikian sehingga membentuk suatu siklus yang disebut siklus informasi seperti pada gambar berikut :



Gambar 2.2 Siklus Informasi

2.3.2 Kualitas Informasi

Kualitas suatu informasi sangat dipengaruhi oleh 3 hal, yaitu :

1. Akurat

   Informasi harus bebas dari kesalahan-kesalahan dan harus jelas dalam menjelaskan maksudnya. Informasi juga harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan yang dapat merubah atau merusak informasi tersebut.

2. Tepat Waktu

   Informasi yang datang kepada penerima tidak boleh terlambat, karena informasi yang sudah usang tidak akan mempunyai nilai lagi. Informasi merupakan landasan didalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa

ini mahalnya nilai informasi disebabkan harus cepatnya informasi tesebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

3. Relevan

   Sebuah informasi harus bermanfaat untuk pemakainya. Relevansi informasi untuk setiap orang berbeda satu sama lain.

2.4 Konsep Dasar Sistem Informasi

Sistem informasi dapat diartikan sebagai suatu susunan dari sumber daya manusia (SDM), proses, komunikasi dan teknologi informasi yang saling berinteraksi untuk mendukung dan meningkatkan operasi harian didalam bisnis seperti mendukung *problem solving*, pengambilan keputusan yang dibutuhkan oleh pihak *management* dan *user*.

Sedangkan menurut H.M.Jogiyanto “Sistem informasi adalah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan”.

Komponen-komponen sistem informasi:

1. Blok Masukan

   Input mewakili data yang masuk kedalam sistem informasi. *Input* ini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, *input* dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data *input* dan data yang tersimpan di dasar data untuk menghasilkan *output* yang diinginkan.

3. Blok Keluaran

   Keluaran merupakan informasi yang berkualitas dan dokumentasi yang berguna utnuk semua tingkatan manajemen serta semua pemakai sistem

4. Blok Teknologi

   Teknologi merupakan *"kotak alat"* dari pekerjaan sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan.

5. Blok Dasar Data

   Dasar data merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali

   Supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa

hal-hal yang merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.5 *Tools* Pengembangan Sistem

Analisis aliran data adalah analisis yang dilakukan untuk mempelajari pemanfaatan data pada setiap aktifitas dengan menampilkan hasil pengamatan kedalam *Data Flow Diagram* (DFD) atau diagram alir data. Diagram alir data adalah satu tampilan grafis yang memunculkan hubungan antara proses dan data beserta kamus data yang menjelaskan rincian data yang digunakan. Analisis berorientasi aliran data menggunakan beberapa teknik yaitu :

2.5.1 Diagram Konteks

Diagram konteks merupakan alat bantu struktur analisis. Pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan. Diagram konteks merupakan kasus khusus dari DFD atau bagian dari DFD yang berfungsi memetakan modul lingkungan yang direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

2.5.2 Diagram Alir Data *(Data Flow Diagram)*

Data Flow Diagram (DFD) adalah suatu model logika data atau proses yang dimuat untuk menggambarkan darimana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dipakai pada data tersebut.

DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur, kelebihan utama pendekatan menggunakan DFD yaitu :

1. Kebebasan dalam menjalankan implementasi teknis sistem.
2. Pemahaman lebih jauh mengenai keterkaitan satu sama lain antara sistem dan subsistem.
3. Mengkomunikasikan pengetahuan sistem yang ada dengan pengguna melalui diagram aliran data.
4. Menganalisis sistem yang diajukan untuk menentukan apakah data-data dan proses yang diperlukan sudah ditetapkan.

DFD terdiri dari *context diagram* dan diagram rinci (*DFD Levelled). Context diagram* berfungsi memetakan model lingkungan yang menggambarkan hubungan antara entitas luar, *input* dan *output* sistem yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. *DFD Levelled* menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dari sudut pandang fungsi.

DFD dibentuk oleh beberapa simbol yang mewakili komponen sistem. Sebagian besar pemodelan dengan aliran data menggunakan empat simbol dasar yang mewakili empat macam komponen sistem yaitu

1. Aliran Data atau *Data Flow*

   Aliran Data menunjukan lintasan perpindahan data didalam sistem yang digambarkan dengan garis yang menghubungkan komponen-komponen

sistem. Arah aliran ditunjukan dengan anak panah dan tiap garis dilengkapi dengan nama data yang mengalir.

2. Proses

   Proses adalah aktivitas pengelolaan data yang akan merubah *input* menjadi *output*. Proses menggambarkan apa yang dilakukan oleh sistem. Setiap proses mempunyai nama proses tunggal dan nomor proses, nama dan proses ini ditulis didalam simbol lingkaran.

3. Entitas atau *External Entity*

   Entitas merupakan kesatuan lingkungan luar sistem, dapat berupa orang, organisasi atau sistem lain yang akan memberikan *input* atau menerima *output* dari sistem.

4. *Data Store*

   Proses-proses dapat memasukan data ke file atau penyimpanan data atau pengambilan data dari file. Tiap file juga harus mempunyai data yang tunggal.

2.5.3 Kamus Data (*Data Dictionary*)

Merupakan alat bantu untuk menjelaskan karakteristik lojik data pada sistem yang memuat nama, deskripsi, alias, isi dan organisasinya. Kamus data juga mengidentifikasi proses yang menggunakan data tertentu untuk mendapatkan informasi yang diperlukan. Selain itu Kamus data juga mempersiapkan dasar identifikasi kebutuhan basis data bagi perancangan sistem.

2.6 Konsep *Database*

Terdapat alasan praktis mengapa perlu mempelajari suatu teori dan konsep, yaitu dapat memberi kerangka kerja untuk memikirkan menyederhanakan persoalan, pemahaman teori dan konsep memberikan dasar ilmu yang kuat, sehingga relatif lebih, mudah untuk mempelajari dan memahami objek yang bersangkutan pada masa-masa yang akan datang.

2.6.1 Pengertian *Database*

Menurut James Martin *Database* adalah suatu kumpulan data terhubung (*Interrelated Data*) yang disimpan secara bersama-sama pada suatu media, tanpa mengatap satu sama lain atau tidak perlu kerangkapan data (*Control Redudancy)* dengan cara tertentu sehingga mudah untuk digunakan atau ditampilkan kembali.

Database merupakan salah satu cabang ilmu komputer yang menangani pengorganisasian data. Sistem *Database* merupakan sistem yang terdiri atas kumpulan file atau tabel yang saling berhubungan dalam sebuah *database* pada sebuah sistem dan sekumpulan program yang memungkinkan beberapa pemakai dan program lainnya untuk mengakses dan memanipulasi file-file atau tabel tersebut.

2.6.2 *Database Management System* (DBMS)

Database Management System (DBMS) merupakan paket program atau *Software* yang dibuat untuk memudahkan dan mengefisienkan tambah, ubah, hapus dan tampil data yang ada dalam sebuah system *database. Software* yang tergolong

kedalam DBMS antara lain, *Microsoft SQL, MySQL, Oracle, MS. Access* dan lain-lain

2.6.3 Abstraksi Data

Abstraksi data merupakan tingkatan-tingkatan pengguna dalam memahami bagaimana data diolah dalam sebuah sistem *database* sehingga menyerupai kondisi yang sebenarnya dihadapi oleh pengguna. DBMS seringkali menyembunyikan detail tentang bagaimana sebuah data disimpan dan diolah dalam sebuah sistem *database*, dengan tujuan untuk memudahkan pengguna dalam menggunakan DBMS tersebut. Karena itu seringkali data yang terlihat oleh pemakai sebelumnya berbeda dengan yang tersimpan secara fisik. Tingkatan atau level dalam abstraksi data :

1. Level Fisik (*Physical Level*)

   Level yang menggambarkan bagaimana data disimpan dalam kondisi sebenarnya. Level ini sangat kompleks karena struktur data dijelaskan secara rinci.

2. Level Konseptual (*Conceptual Level*)

   Level ini menggambarkan data yang disimpan dalam database dan menjelaskan bagaimana hubungan antar datanya secara keseluruhan. Level ini biasa di pakai oleh seorang *Database Administrator* (DBA).

3. Level Pandangan (*View Level*)

   Level yang hanya menggambarkan sebagian saja dari keseluruhan *database* sesuai dengan kebutuhan pengguna. Misalnya : Bagian keuangan

hanya membutuhkan data keuangan, jadi yang digambarkan hanya pandangan terhadap data keuangan saja.

2.6.4 Model *Database*

Model *database* adalah suatu konsep yang terintegrasi dalam menggambarkan hubungan (*relationships*) antar data dan batasan-batasan (*constraint*) dalam suatu sistem *database*. Model data yang paling umum berdasarkan bagaimana hubungan antar *record* dalam *database* (*Record Based Data Models*) ada tiga jenis yaitu :

a. Model Hirarki (*Hierarchical Database Model*)

   Model hierarkis sering disebut dengan model pohon karena bentuknya yang menyerupai pohon. Model ini menggunakan pola hubungan orang tua-anak dimana setiap simpul yang biasa digambarkan sebagai lingkaran atau kotak menyatakan sekumpulan *field*. Simpul yang terhubung ke simpul pada tingkat yang di bawahnya disebut orangtua. Setiap orang tua dapat memiliki satu (hubungan 1:1) atau beberapa anak (hubungan 1:N), tetapi setiap anak hanya memiliki satu orang tua. Simpul-simpul yang dibawahi oleh simpul orang tua disebut anak. Simpul orangtua yang tidak memiliki orang tua disebut akar. Simpul yang tidak memiliki anak disebut daun (*leaf*). Adapun hubungan antara anak dan orangtua disebut cabang.

b. Model Jaringan (*Network Database Model*)

   Model jaringan menyerupai model hirarki. Perbedaannya adalah dimana suatu simpul anak dapat memiliki lebih dari satu orangtua. Oleh karena itu

model ini dapat menyatakan hubungan 1:1 dimana satu orang tua mempunyai satu anak dan 1:N dimana satu orang tua mempunyai banyak anak atau M:N dimana beberapa anak memunyai beberapa orang tua.

c. Model Relasi (*Relational Database Model*)

Model relasi merupakan model *database* yang paling banyak digunakan saat ini, karena paling sederhana dan mudah digunakan serta kemampuannya dalam mengakomodasi berbagai kebutuhan pengelolaan *database*. Sebuah *database* dalam model ini disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*record*) dan kolom (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*), tabel-tabel yang ada di hubungkan (*relationship*) sedemikian rupa menggunakan field-field kunci (k*ey* field) sehingga dapat meminimalkan duplikasi data.

2.6.5 Perancangan Basis Data (*Database)*

Teknik yang digunakan dalam merancang sistem *database* adalah :

1. Normalisasi

Normalisasi merupakan pengelompokkan data elemen menjadi tabel-tabel yang menunjukkan *entity* dan relasinya. Pada proses normalisasi selalu ada pengujian pada beberapa kondisi, apakah ada kesulitan pada saat membaca, menambah, menghapus, mengubah pada suatu *database*. Apabila terdapat kesulitan, maka relasi tersebut dipecahkan pada beberapa tabel lagi atau dengan kata lain perancangan belumlah mendapat *database*

yang optimal. Walaupun jumlah normalisasi bervariasi, dasar normalisasi sebenarnya hanya ada tiga, yaitu bentuk normal pertama, normal kedua, dan normal ketiga. Berikut penjelasan mengenai bentuk normalisasi :

a. Bentuk tidak normal (*unnormalized form*)

   Bentuk ini merupakan kumpulan data yang akan direkam. Bentuk tidak normal tidak harus mengikuti format tertentu, dan sangat mungkin terjadi duplikasi data atau bahkan data tidak lengkap.

b. Bentuk Normal Kesatu (*1 NF/Fist Normal Form*)

   Bentuk ini mempunyai ciri yaitu setiap data dibentuk dalam satu *record* demi satu *record* dan nilai dari *field-field* berupa *atomic value* serta tidak ada atribut yang berulang atau bernilai ganda.

c. Bentuk Normal Kedua (*2NF/Second Normal Form*)

   Aturan normal kedua yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu dan setiap *field* tidak bergantung sepenuhnya kepada kunci primer.

d. Bentuk Normal Ketiga (*3NF/Third Normal Form*)

   Aturan normalisasi ketiga yaitu relasi harus dalam bentuk normal kedua dan tidak boleh ada ketergantungan antara *field-field non*-kunci atau ketergantungan *transitif*.

2. Model Data *Entity Relationship*

Model data *entity relationship* sering dijadikan acuan dalam merancang suatu sistem *database*. Pada Model Data *Entity Relationship* ini, data yang

ada ditransformasikan dengan memanfaatkan sejumlah perangkat konseptual menjadi diagram data, yang sering disebut *Diagram Entity Relationship* (*Diagram E-R*). Ada dua komponen utama pembentuk model data ini, yaitu :

a. *Entity* beserta *attribute*-nya.
b. Relasi dan jenis hubungannya

Setiap *entity* mempunyai atribut atau *field*. *Attribute* adalah ciri khas yang melekat pada suatu *entity*. Keterkaitan antar *entity* digambarkan dengan simbol-simbol sehingga lebih mudah dipahami. Penggambaran hubungan inilah yang disebut dengan *Diagram Entity Relationship* (Diagram E-R). Simbol-simbol yang boleh digunakan adalah :

a. Persegi Panjang, berfungsi untuk menyatakan suatu *entity*.
b. Elips, berfungsi untuk menyatakan *attribute*, apabila diberi garis bawah menandakan bahwa *attribute* tersebut merupakan *attribute* atau *field* kunci.
c. Belah Ketupat, menyatakan jenis relasi.
d. Garis, penghubungan antara relasi dengan *entity* dan antara *entity* dengan *attribute*.

Dalam merancang sistem *database* derajat relasi atau hubungan antar tabel menunjukkan jumlah maksimum record suatu *entity* ber-relasi dengan *record* pada *entity* yang lainnya.

2.6.6 Jenis Hubungan Antar Tabel

Jenis hubungan antar tabel dalam model *database* didefinisikan dengan hubungan satu ke satu (*One to One*), satu ke Banyak (*One to Many*), banyak ke satu (*Many to One*), banyak ke banyak (*Many to Many*).

a. Relasi Satu ke Satu (*One to One*)

   Artinya satu *record* pada *entity* A ber-relasi paling banyak satu record juga pada entity B, begitu juga sebaliknya satu record pada *entity* B, ber-relasi paling banyak satu *record* dengan *entity* A. Dalam diagram E-R, relasi ini disimbolkan dengan angka 1.

b. Relasi Satu ke Banyak (*One to Many)*

   Artinya satu *record* pada *entity* A ber-relasi dengan beberapa *record* pada *entity* B, tapi tidak sebaliknya setiap *record* pada *entity* B ber-relasi paling banyak satu *record* dengan *entity* A. Dalam diagram E-R relasi ini disimbolkan dengan angka 1 untuk menyatakan satu dan huruf M atau N untuk menyatakan banyak.

c. Relasi Banyak ke Satu (*Many to One)*

   Jenis relasi ini adalah kebalikan dari relasi satu ke banyak, dimana setiap *record* pada *entity* A hanya dapat ber-relasi paling banyak 1 *record* pada *entity* B, tapi tidak sebaliknya, satu *record* pada *entity* B dapat ber-relasi dengan beberapa *record* pada *entity* A. Dalam diagram E-R, relasi ini disimbolkan dengan angka 1 untuk menyatakan satu dan huruf M atau N untuk menyatakan banyak.

d. Relasi Banyak ke Banyak (*Many to Many*)

Artinya beberapa *record* pada *entity* A dapat ber-relasi dengan beberapa *record* pada *entity* B, begitu juga sebaliknya beberapa *record* pada entity B dapat ber-relasi dengan beberapa *record* pada *entity* A.. Dalam diagram E-R relasi ini disimbolkan dengan huruf M atau N untuk menyatakan banyak.

2.6.7 Jenis-Jenis Kunci (*Key*)

Setiap *file* terdapat kunci dari *file* berupa satu *field* atau satu *set field* yang dapat mewakili *record*, ada empat macam a*ttribute* kunci antara lain :

1. Kunci Kandidat (*Candidat Key*)

   Sebuah *attribute* atau lebih yang secara unit mengidentifikasi sebuah *record* disebut *candidate key*. *Attribute* ini mempunyai nilai yang unik pada hampir setiap recordnya. Fungsi dari *candidate key* ini adalah sebagai calon *primary key*.

2. Kunci Utama *(Primary Key)*

   Primary key merupakan *candidate key* yang telah dipilih untuk mengidentifikasi setiap *record* secara unik. *Primary key* harus merupakan *field* yang benar-benar unik dan tidak boleh ada nilai NULL.

3. Kunci Alternatif *(Alternate Key)*

   Alternate Key adalah kunci kandidat yang tidak dipakai sebagai *primary key*. Misalnya dalam suatu *entity* terdapat dua *field* yang bisa dijadikan sebagai kunci. Sementara yang boleh dijadikan kunci hanya satu, maka

field yang dipilih disebut *primary key*, sedangkan *field* yang tidak dipilih disebut dengan *alternate key*.

4. *Foreign Key*

   Apabila sebuah *primary key* terhubungan ke *table* atau *entity* lain, maka keberadaan *primary key* pada *entity* tersebut disebut sebagai *foreign key*.

2.7 Sistem *Client-Server*

Konsep Jaringan *client-server* membedakan dengan jelas kedudukan suatu komputer, mana yang dapat memberikan layanan jaringan atau *server* dan mana yang hanya meminta layanan atau *client*. Untuk dapat saling berkomunikasi antara *server* dan *client*, maka digunakan suatu aplikasi jaringan yang dinamakan *client-server program*, dimana *server* menggunakan aplikasi jaringan yang disebut *server program*, sementara *client* menggunakan *client program* untuk dapat berkomunikasi dengan *server program* pada *server*.

Beberapa sifat dari *client-server* diantaranya sebagai berikut :

a. *Server* dan *Client* berada pada posisi serta proses yang berbeda.
b. *Server* dan *Client* dapat dijalankan pada mesin yang sama atau berbeda.
c. Setiap proses dapat menyembunyikan informasi tersendiri.
d. Komunikasi yang dilakukan bisa bersifat protokol *peer to peer*.

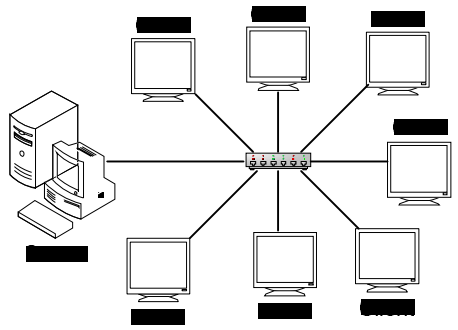
Koneksitas *client ke Server :*

a. *Server* harus terlebih dahulu dijalankan (*running*).

b. *Server* harus menentukan *port* untuk koneksi yang masuk.

c. *Client* harus mengetahui nama *host* dari *server* beserta *port* yang digunakan untuk koneksi.

d. *Client* mengirim pesan ke *server* untuk melakukan suatu pekerjaan (*task*).



Gambar 2.3 Model Jaringan Client Server

2.8 Software Pendukung

2.8.1 Delphi 7.0

Borland Delphi merupakan program aplikasi *database* berbasis *Object Pascal* yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta pemrograman yang terstruktur. Keunggulan lain dari *delphi* adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampailan seperti program aplikasi lain yang berbasis *windows*.

Delphi tidak hanya dapat mengakses *database* seperti *Paradox, xBase, Ms. Access* tetapi dapat juga mengakses *database* lainnya seperti *Oracle, Sybase, Interbase, DB2, MySQL* dan lain-lain.

IDE (*Integrated Development Environment*) adalah sebuah lingkungan yang berisi *tool-tool* yang diperlukan untuk desain dan menjalankan aplikasi. Pada program *delphi* IDE (*Integrated Development Environment*) terdiri atas *Main Window, Component pallete, Toolbar, Form Designer, Code Editor, Code Explorer* dan *Object Treeview.*

2.8.2 MySQL 4.0.21

MySQL adalah sebuah sistem manajemen *database* yang bersifat *open source*. *MySQL* juga merupakan *relational database management system* (RDBMS) yang didistribusikan secara grafis oleh sebuah lisensi GPL *(general public licenci)*, dimana setiap orang bebas menggunakan *MySQL*, tapi tidak boleh dijadikan produk turunan yang bersifat *close source* atau komersial.

MySQL menggunakan bahasa standar SQL *(Structure Query Language)* sebagai bahasa interaktif dalam mengolah data, karena pada dasarnya *MySQL* merupakan turunan salah satu konsep utama dalam database SQL *(structured query language).* SQL adalah sebuah konsep pengoprasian *database* untuk pemilihan dan pemasukkan data yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis. *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainya dalam *query* data, karena dengan *MySQL* kita dapat mengetahui cara kerja *optimizer* ketika melakukan proses perintah-perintah SQL yang dibuat oleh *user*

maupun program aplikasi *database server*. Selain itu, sebagai suatu *database server MySQL* memiliki sebuah konsep *database* modern, dimana *MySQL* memiliki banyak keistimewaan antara lain *portabilitas, opensource, multiuser, performance training, client and tool* dan masih banyak lagi. Kelebihan-kelebihan lain yang dimiliki oleh *MySQL* adalah :

1. Dengan adanya fitur tambahan *Query Chacing* pada *MySQL 4.0*, kinerja *query* secara umum akan naik rata-rata 200% dari kinerja biasanya.
2. Dengan konsep *open source* siapapun dapat berpartisipasi dalam mengembangkan *MySQL* dan hasil penembangan itu dapatdiserahkan kepada umum atau kepada komunitas *Open Source*.
3. *MySQL* telah digunakan untuk mengelola *database* dengan jumlah 50 juta *record*. Bahkan *MySQL* sanggup mengelola 60.000 tabel dengan jumlah baris 5.000.000.000. *MySQL* juga Mendukung penggunaan *index* hingga 32 *index* pertabelnya, dimana setiap *index* terdiri dari 1 hingga 16 kolom kriteria.
4. *MySQL* menerapkan sistem keamanan dan ijin akses tingkat lanjut, termasuk dukungan keamanan dengan pengacakan lapisan data *(SSL transport layer encryption)*. Bahkan pada *MySQL 4.0* dapat diterapkan pembatasan penggunaan sumberdaya *server* berdasarkan penggunaan per-*user*. Adanya tingkatan *user* dan jenis akses yang beragam dapat memberikan batasan akses berdasarkan lokasi pengakses, dan terdapatnya jaminan keamanan pada *MySQL*.

5. *MySQL* mendukung peritah-peritah ANSI SQL 99 dan beberapa perintah *database alternative* lainnya sehingga mempermudah untuk beralih dari dan ke *MySQL*.
6. *MySQL* dapat dijalankan dibeberapa sistem operasi yang berbeda, seperti *Linux, Microsoft Windows, FreeBSD, Sun Solaris, Mac OS, HP-UX, AIX, QNX, Novell NetWare, SCO OpenUnix, SGI Iric, dan Dec OSF.*