BAB II

TINJAUAN PUSTAKA

2.1 Ruang Lingkup Objek Penelitian

Objek penelitian yang sedang diteliti adalah Sekolah Menegah Atas, yaitu bertempat di SMA Negeri 4 Cimahi.

2.1.1 Sejarah SMA Negeri 4 Cimahi

SMA Negeri 4 Cimahi merupakan sekolah pemekaran dari SMA Negeri 2 Cimahi, yang pada tahun 1992 berdiri sendiri di lokasi Kelurahan Leuwigajah Jl. Kihapit Barat No. 323. Dari tahun ke tahun SMA Negeri 4 Cimahi berbenah diri, dimulai dari peningkatan mutu pendidikan, baik dari segi akademik maupun prestasi non akademik siswa maupun keprofesionalitas tenaga pengajar, serta pengadaan sarana dan prasarana yang dapat menunjang mutu pendidikan di SMA Negeri 4 Cimahi.

Adapun prestasi akademik yang diperoleh oleh SMA Negeri 4 Cimahi dari segi akademik tingkat kelulusan Ujian Nasional 99% lulus, banyaknya lulusan (alumni) yang melanjutkan ke jenjang lebih tinggi diantaranya banyak yang diterima di Perguruan Tinggi Negeri melalui jalur PMDK dan SPMB (UNPAD, IPB, UPI, ITB, STAN maupun Perguruan Tinggi yang ada di Indonesia. Salah satu lulusan yang diterima di ITB jalur PMDK mendapatkan bea siswa dari Pemkot Cimahi. Selain PTN dan PTS lulusan SMA Negeri 4 Cimahi banyak melanjutkan ke militer maupun kepolisian. Selain prestasi akademik, prestasi non akademik pun ditoreh oleh para siswa dari jalur ekstrakurikuler. Tidak

ketinggalan tenaga pengajar dari SMA Negeri 4 terpilih menjadi guru teladan sebagai perwakilan dari Kota Cimahi.

Selain akademik, sarana dan prasarana pun lebih ditingkatkan lagi dengan pembangunan jumlah rombongan belajar (kelas), pengadaan Lab. Komputer (2 lokal), Lab. Bahasa, Lab. Biologi dan kimia, Lab. Fisika, Sangar seni, R. Aula, R. Multimedia1 dan multimedia 2 dan sarana lain yang menunjang mutu pendidikan. Untuk lebih meningkatkan mutu pendidikan, SMA Negeri 4 Cimahi pada tahun 2007 (sekarang) merintis Sekolah Kategori Mandiri (SKM) yaitu sekolah yang menerapkan dan mengembangkan kompetensi / kualitas lulusan (bukan kuantitas lagi yang diterapkan). Sehingga pada tahun 2009 SMA Negeri 4 Cimahi sudah menerapkan SKM yang sesuai dengan Standar Nasional Pendidikan.

2.1.2 Visi dan Misi

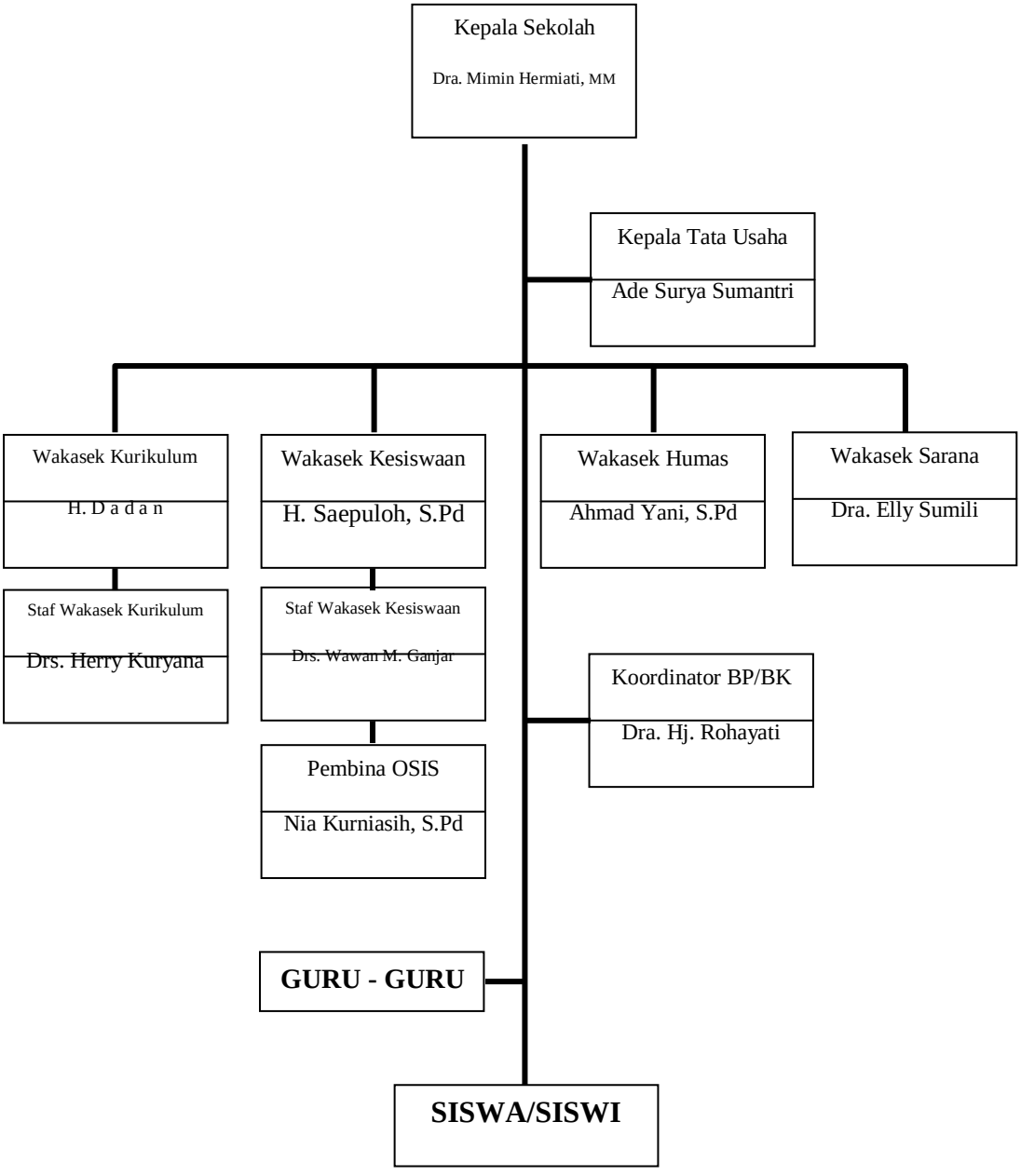
Visi SMA Negeri 4 Cimahi adalah Visi merupakan imajinasi moral yang dijadikan dasar atau rujukan dalam menentukan tujuan atau keadaan masa depan yang secara khusus diharapkan oleh sekolah. Adapun visi yang diemban SMA Negeri 4 Cimahi adalah:**" Menuju Pendidikan Bermutu untuk Mewujudkan Insan Indonesia yang Taqwa, Cerdas dan Kompetitif "**

Misi SMA Negeri 4 Cimahi :

a. Membina insan bertaqwa dan berakhlak mulia
b. Meningkatkan kualitas SDM (Guru, Pegawai dan Siswa)
c. Meningkatkan pelayanan Pendidikan bagi masyarakat
d. Meningkatkan Sistem Pembelajaran berbasis TIK

2.1.3 Struktur Organisasi SMA Negeri 4 Cimahi

Struktur organisasi pada SMA Negeri 4 Cimahi telihat pada gambar 2.1dibawah ini:



Gambar 2. 1 Struktur Organisasi

2.2 Landasan Teori

2.2.1 Definisi sistem Informasi

Deskripsi mengenai pengertian sistem Informasi menurut beberapa ahli [7]:

1. Sistem informasi adalah kombinasi antar prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi [Alter (1992)].
2. Sistem informasi adalah kumpulan perangkat keras dan perangkat lunak yang dirancang untuk mentransformasikan data ke dalam bentuk informasi yang berguna [Bodnar dan Hopwood (1993)].
3. Sistem informasi adalah suatu sistem buatan manusia yang secara umum terdiri atas sekumpulan komponen berbasis komputer dan manual yang dibuat untuk menghimpun, menyimpan, dan mengelola data serta menyediakan informasi keluaran kepada para pemakai. [Gelinas, Oram, dan Wiggins (1990)].
4. Sistem informasi adalah sebuah rangkaian prosedur formal dimana data dikelompokkan, diproses menjadi informasi, dan didistribusikan kepada pemakai. [Hall (2001)].
5. Sebuah sistem informasi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik. [Turban, McLean, dan Wetherbe (1995)] .
6. Sistem informasi adalah kerangka kerja yang mengkoordinasikan sumberdaya (manusia, komputer) untuk mengubah masukan menjadi (informasi), guna mencapai sasaran perusahaan. [Wilkinson (1992)].

2.2.2 Konsep Dasar Informasi

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat tertentu. Informasi dapat didefinisikan sebagai [3] :

"Data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya".

1. Kualitas Informasi

Istilah kualitas informasi *(quality of information)* terkadang juga dipakai untuk menyatakan informasi yang baik. Kualitas informasi sering kali diukur berdasarkan :

1. Relevansi

   Relevan berarti bahwa informasi benar-benar memberikan manfaat bagi pemakai

2. Ketepatan waktu

   Ketepatan waktu berarti informasi tersebut tidak usang / kadaluarsa ketika sampai ke penerima, sehingga masih ada waktu untuk menggunakan informasi tersebut sebagai bahan pengambilan keputusan.

3. Keakurasian

   Akurasi berarti menyatakan derajat kebenaran terhadap informasi dan menentukan kehandalan atau reliabilitas informasi.

2. Nilai Informasi

Nilai informasi *(value of information)* ditentukan oleh dua hal, yaitu manfaat dan biaya untuk mendapatkannya. Suatu informasi dianggap bernilai kalau manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu system informasi umumnya digunakan untuk beberapa kegunaan. Sebagian informasi tidak dapat ditaksir keuntungannya dengan suatu nilai tetapi dapat ditaksir dari nilai keefektifannya.

3. Teknologi Informasi

Information Teknologi (IT) adalah sebuah terminologi kontemporer yang mendeskripsikan kombinasi antara teknologi komputer (*hardware* dan *software*) dengan teknologi komunikasi (data, *image*, dan jaringan suara).

Peranan Teknologi Informasi :

1. Teknologi informasi menggantikan peran manusia. Dalam hal ini, teknologi informasi melakukan otomasi terhadap suatu tugas atau proses.
2. Teknologi memperkuat peran manusia, yakni dengan menyajikan informasi terhadap suatu tugas atau proses.

Teknologi informasi berperan dalam restrukturisasi terhadap peran manusia. Dalam hal ini, teknologi berperan dalam melakukan perubahan-perubahan terhadap sekumpulan tugas atau proses.

2.2.3 SPK (Sistem Pendukung Keputusan)

Pengambilan keputusan merupakan proses pemilihan alternative tindakan untuk mencapai tujuan atau sasaran tertentu. Pengambilan keputusan dilakukan

dengan pendekatan sistematis terhadap permasalahan melalui proses pengumpulan data menjadi informasi serta ditambah dengan faktor – faktor yang perlu dipertimbangkan dalam pengambilan keputusan.

Menurut Keen dan Scoot Morton [11]:

“ Sistem Pendukung Keputusan merupakan penggabungan sumber – sumber kecerdasan individu dengan kemampuan komponen untuk memperbaiki kualitas keputusan. Sistem Pendukung Keputusan juga merupakan sistem informasi berbasis komputer untuk manajemen pengambilan keputusan yang menangani masalah – masalah semi struktur “.

Dengan pengertian diatas dapat dijelaskan bahwa sistem pendukung keputusan bukan merupakan alat pengambilan keputusan, melainkan merupakan sistem yang membantu pengambil keputusan dengan melengkapi mereka dengan informasi dari data yang telah diolah dengan relevan dan diperlukan untuk membuat keputusan tentang suatu masalah dengan lebih cepat dan akurat. Sehingga sistem ini tidak dimaksudkan untuk menggantikan pengambilan keputusan dalam proses pembuatan keputusan.

Alter (2002) mendefinisikan System pendukung keputusan atau *Decision Support Systems* (DSS) adalah sistem informasi interkatif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada siuasi yang semiterstruktur dan situasi yang tidak terstruktur di mana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. Konsep DSS dikemukaakan pertama kali oleh Scoot-Morton

pada tahun 1971 (Turban, McLean, dan Wetherbe, 1999). Beliau mendefinisikan cikal bakal DSS tersebut sebagai [8]:

"*system berbasis computer yang interaktif, yang membantu pengambil keputusan dengan menggunakan data dan model untuk memecahkan persoalan-persoalan tak terstruktur"*

2.2.3.1 Karakteristik SPK (Sistem Pendukung Keputusan)

Dari pengertian Sistem Pendukung Keputusan maka dapat ditentukan karakteristik antara lain [10]:

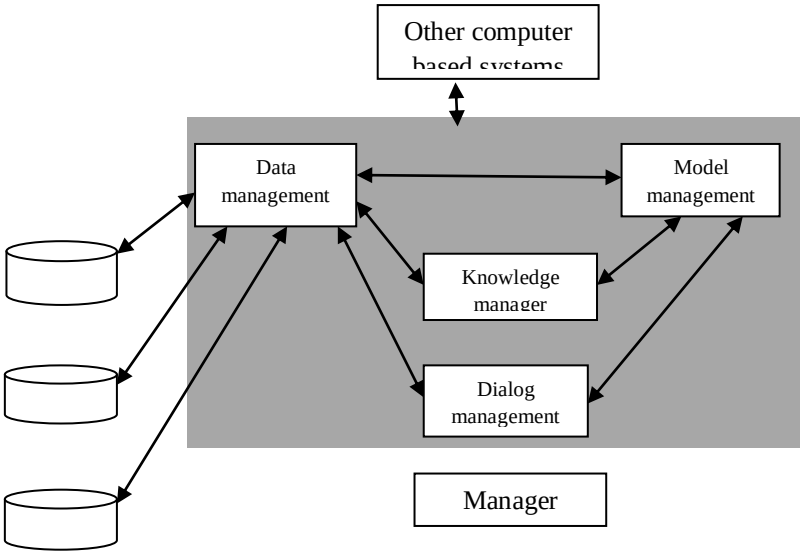
1. Mendukung proses pengambilan keputusan, menitik beratkan pada management by perception
2. Adanya interface manusia / mesin dimana manusia (user) tetap memegang control proses pengambilan keputusan
3. Mendukung pengambilan keputusan untuk membahas masalah terstruktur, semi terstruktur dan tak struktur
4. Memiliki kapasitas dialog untuk memperoleh informasi sesuai dengan kebutuhan
5. Memiliki subsistem – subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai kesatuan item
6. Membutuhkan struktur data komprehensif yang dapat melayani kebutuhan informasi seluruh tingkatan manajemen

2.2.3.2 Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan memiliki beberapa komponen yang terdiri dari sebagai berikut [10]:

1. Data *Management* termasuk *database* yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System* (DBMS).
2. Model *Management* melibatkan model financial, statistical, *management science* atau berbagai model kuantitatif lainnya, sehingga dapatmemberikan ke sistem suatu kemampuan analitis dan manajemen *software* yang diperlukan.
3. *Communication* (dialog subsistem), user dapat berkomunikasi dan memberikan perintah pada DSS (*Decision Support System*) melalui subsistem ini, ini berarti menyediakan antarmuka.
4. *Knowledge Management*, subsistem optional ini dapat mendukung subsistem lain ataubertindak sebagai komponen yang berdiri sendiri.

Model konseptual SPK (Sistem Pendukung Keputusan) dapat dilihat pada gambar 2.2 dibawah ini:



Gambar 2. 2 Model Konseptual SPK

2.2.3.3 Keuntungan Sistem Pendukung Keputusan (SPK)

Dengan berbagai karakter khusus yang dimiliki Sistem Pendukung Keputusan, SPK (Sistem Pendukung Keputusan) dapat memberikan berbagai manfaat dan keuntungan. Manfaat yang dapat diambil dari SPK adalah [11]:

1. SPK memperluas kemampuan pengambil keputusan dalam memproses data / informasi bagi pemakainya..
2. SPK membantu pengambil keputusan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
3. SPK dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.
4. Walaupun suatu SPK, mungkin saja tidak mampu memecahkan masalah yang dihadapi oleh pengambil keputusan, namun ia dapat menjadi stimulan bagi pengambil keputusan dalam memahami persoalannya, karena mampu menyajikan berbagai alternatif pemecahan.

2.2.4 Metode *Multi-Factor Evaluation Process* (MFEP)

Multi Factor Evaluation Process (MFEP) adalah metode kuantitatif yang menggunakan. ‘weighting system’1. Dalam pengambilan keputusan multi faktor, pengambil keputusan secara subyektif dan intuitif menimbang berbagai faktor yang mempunyai pengaruh penting terhadap alternatif pilihan mereka. Untuk keputusan yang berpengaruh secara strategis, lebih dianjurkan menggunakan sebuah pendekatan kuantitatif seperti MFEP. Dalam MFEP pertama-tama seluruh

kriteria yang menjadi faktor penting dalam melakukan pertimbangan diberikan pembobotan (weighting) yang sesuai. Langkah yang sama juga dilakukan terhadap alternatif-alternatif yang akan dipilih, yang kemudian dapat dievaluasi berkaitan dengan faktor–faktor pertimbangan tersebut. Metode MFEP menentukan bahwa alternatif dengan nilai tertinggi adalah solusi terbaik berdasarkan kriteria yang telah dipilih.[9]

2.2.4.1. Konsep Dasar Penggunaan Metode MFEP

Dibawah ini merupakan langkah-langkah proses perhitungan menggunakan metode MFEP, yaitu:

a. Menentukan faktor dan bobot faktor dimana total pembobotan harus sama dengan 1 (∑ pembobotan = 1), yaitu *factor weight*.
b. Mengisikan nilai untuk setiap faktor yang mempengaruhi dalam pengambilan keputusan dari data-data yang akan diproses, nilai yang dimasukkan dalam proses pengambilan keputusan merupakan nilai objektif, yaitu sudah pasti yaitu *factor evaluation* yang nilaianya antara 0 - 1.
c. Proses perhitungan *weight evaluation* yang merupakan proses perhitungan bobot antara *factor weight* dan *factor evaluation* dengan serta penjumlahan seluruh hasil *weight evaluations* untuk memperoleh total hasil evaluasi.

Penggunaan model MFEP dapat direalisasikan dengan contoh berikut:

Steve Marcel, seorang lulusan sarjana bidang bisnis mencari beberapa lowongan pekerjaan. Setelah mendiskusikan gambaran pekerjaan yang akan

dikerjakannya dengan penasehat didiknya dan departemen direktur pusat penempatan pegawai, steve mendapatkan bahwa dari tiga faktor yang terpenting baginya yaitu gaji, peluang karir yang lebih baik, dan lokasi tempat kerja. Steve sudah memutuskan bahwa peluang jenjang karir merupakan faktor yang terpenting baginya. Faktor tersebut diberinya nilai skala 0.6. steve menempatkan gaji diurutan berikutnya dengan nilai skala 0.3. Terakhir, steve memberikan nilai skala 0.1untuk tempat kerja.

Seperti masalah pada model MFEP yang lain, nilai skala jika dijumlahkan harus sama dengan satu (tabel 2.1).

Tabel 2. 1 Nilai Bobot Untuk Faktor

Faktor	Importance (Weight)
Salary	0,6
Career advancement	0,3
Location	0,1

Pada saat itu, steve merasa yakin bahwa ia diterima di perusahaan AA, perusahaan EDS,Ltd, dan perusahaan PW,Inc. Untuk setiap perusahaan, steve menghitung rata-rata variasi faktor dari nilai skala 0 sampai 1. Untuk perusahaan AA, steve memberikan faktor gaji dengan nilai skala 0.4. Peluang jenjang karir dengan nilai skala 0.9 dan lokasi tempat kerja dengan nilai skala 0.6 Untuk perusahaan EDS,Ltd, steve memberikan faktor gaji dengan nilai skala 0.8, peluang jenjang karir dengan nilai skala 0.7 dan lokasi tempat kerja dengan nilai skala 0.8. Untuk perusahaan PW.Inc, steve memberikan nilai faktor gaji dengan

nilai skala 0.9, peluang jenjang karir dengan nilai skala 0.6 dan lokasi tempat kerja dengan nilai skala 0.9. hasilnya dapat dilihat pada tabel 2.2.

Tabel 2. 2 Tabel Nilai Faktor dari Setiap Data Uji

Faktor	AA.CO	EDS.LTD	PW.INC
Salary	0.7	0.8	0.9
Career advancement	0.9	0.7	0.5
Location	0.6	0.8	0.9

Dari informasi yang diperoleh, steve dapat menghitung total bobot evaluasi dari setiap kriteria pekerjaan. Setiap perusahaan menghasilkan nilai evaluasi dari tiga faktor dan bobot faktor dikalikan dengan nilai evaluasi dan dijumlahkan untuk memperoleh total hasil evaluasi.

Tabel 2. 3 Tabel Nilai Evaluation Perusahaan AA

Factor name	Factor weight		Factor evaluation		Weight ed evaluation
Salary	0.3	X	0.7	=	0,21
Career advancement	0.6	X	0.9	=	0.54
Location	0.1	X	0.6	=	0.06
Total	1				**0.81**

Tabel 2. 4 Tabel Nilai Evaluasi Perusahaan EDS.Ltd

Faktor	Bobot Faktor		Evaluasi Faktor		Bobot Evaluasi

Salary	0.3	X	0.8	=	0,24
Career advancement	0,6	X	0.7	=	0.42
Location	0,1	X	0.8	=	0.08
Total	1				**0.74**

Tabel 2. 5 Tabel Nilai Evaluasi Perusahaan PW.Inc

Faktor	**Bobot Faktor**		**Evaluasi Faktor**		**Bobot Evaluasi**
Salary	0,3	X	0.9	=	0,27
Career advancement	0,6	X	0.6	=	0.36
Location	0,1	X	0.9	=	0.09
Total	1				**0.72**

Dari setiap perusahaan, seperti yang dapat dilihat pada tabel 2.3, perusahaan AA memperoleh total bobot evaluasi 0.8. Analisis yang sama dilakukan juga untuk perusahaan EDS,Ltd dan perusahaan PW.Inc pada tabel 2.4 dan tabel 2.5. Sesuai dengan yang dapat dilihat dari hasil analisis, perusahaan AA memperoleh total bobot faktor yang paling tinggi, setelahnya adalah perusahaan EDS.Ltd yang memperoleh total bobot evaluasi 0.74. Dengan menggunakan *Multifactor Evaluation Processi*, steve mengambil keputusan untuk bekerja

diperusahaan AA karena perusahaan tersebut memiliki nilai bobot faktor tertinggi dari yang lainnya.

2.2.5 Pengertian *Client Server*

Semakin berkembangnya teknologi jaringan komputer, sekarang ini terdapat kecenderungan sebuah sistem yang menggunakan jaringan untuk saling berhubungan. Dalam jaringan tersebut, biasanya terdapat sebuah komputer yang disebut server, dan beberapa komputer yang disebut client, dalam pendistribusian suatu aplikasi.

Komponen client sering disebut juga sebagai front-end, sementara komponen server disebut sebagai back-end. Komponen client dari aplikasi dijalankan dalam sebuah workstation dan menerima masukan data dari pengguna. Komponen client tersebut akan menyiapkan data yang dimasukkan oleh pengguna dengan menggunakan teknologi pemrosesan tertentu dan mengirimkannya kepada komponen server yang dijalankan di atas mesin server, umumnya dalam bentuk request terhadap beberapa layanan yang dimiliki oleh server. Komponen server

akan menerima request dari client, dan langsung memprosesnya dan mengembalikan hasil pemrosesan tersebut kepada client. Client pun menerima informasi hasil pemrosesan data yang dilakukan server dan menampilkannya kepada pengguna, dengan menggunakan aplikasi yang berinteraksi dengan pengguna.

Atau dengan kata lain, server adalah komputer yang dapat memberikan service ke server, sedangkan client adalah komputer yang mengakses beberapa service yang ada di client. Ketika client membutuhkan suatu service yang ada di

server, dia akan mengirim request kepada server lewat jaringan. Jika request tersebut dapat dilaksanakan, maka server akan mengirim balasan berupa service yang dibutuhkan untuk saling berhubungan menggunakan Socket (sebuah endpoint untuk komunikasi didalam jaringan) [5].

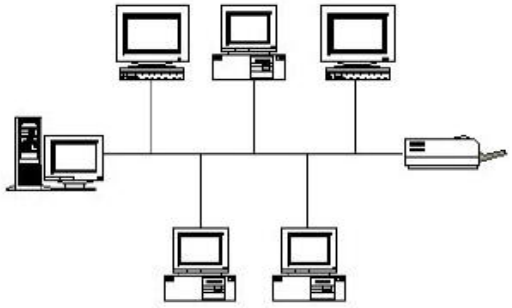
1. Kelebihan model hubungan *Client Server* :
   a. Terpusat (Sumber daya dan keamanan data dikontrol melalui *server*).
   b. Skalabilitas.
   c. Fleksibel.
   d. Teknologi baru dengan mudah terintegrasi kedalam sistem.
   e. Keseluruhan komponen (*client/network/server*) dapat bekerja bersama.
2. Kekurangan model hubungan Client Server :
   a. Mahal.
   b. Membutuhkan investasi untuk *dedicated file server*.
   c. Perbaikan (Jaringan besar membutuhkan seorang staf untuk mengatur agar sistem berjalan secara efisien).
   d. Berketergantungan.
   e. Ketika *server* jatuh, mengakibatkan keseluruhan operasi pada network akan jatuh pula.

Topologi Fisik

Berikut ini akan dijelaskan mengenai topologi fisik yang digunakan di dalam jaringan lokal diantaranya :

1. *Linear Bus* (Garis Lurus)

Topologi *Linear Bus* (Garis lurus) terdiri dari satu jalur kabel utama dimana pada masing-masing ujungnya diberikan sebuah *terminator*. Semua nodes pada jaringan (*file server*, *workstation*, dan perangkat lainnya) terkoneksi sebuah kabel utama (backbone). Jaringan-jaringan Ethernet dan Local Talk menggunakan topologi linear ini.

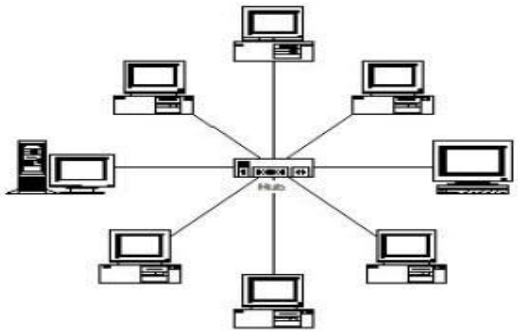


Gambar 2. 3 Topologi *Linear Bus*

2. *Star* (Bintang)

Topologi model ini dirancang, yang mana setiap nodes (*file server*, *workstation,* dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*.

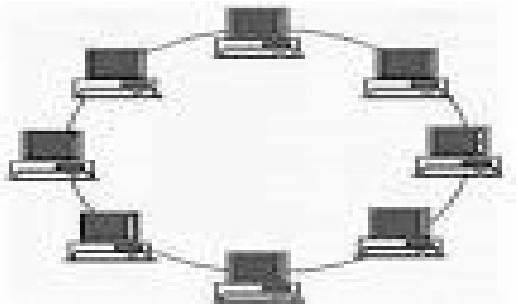
Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. *Concentrator* akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagai repeter (penguat aliran data). Konfigurasi pada jaringan model ini menggunakan kabel Twisted Pair, dan dapat diguanakan pula kabel *coaxial* atau kabel *fiber optic*.



Gambar 2. 4 Topologi *Star*

3. *Ring* (Cincin)

Topologi *Ring* (Cincin) menggunakan teknik konfigurasi yang sama dengan topologi *star* tetapi pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring*.

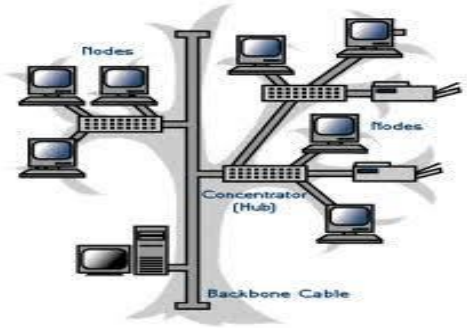


Gambar 2. 5 Topologi *Ring*

4. *Tree* (Pohon)

Topologi model ini merupakan perpaduan antara topologi *Linear Bus* dan *Star*, yang mana terdiri dari kelompok-kelompok dari *workstation* dengan konfigurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *Linear Bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang

telah ada, dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2. 6 Topologi *Tree*

2.2.6 Basis Data

Basis Data terdiri dari kata Basis dan Data. Basis dapat diartikan gudang atau tempat bersarang dan Data yang berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Jadi dapat disimpulkan bahwa basis data (*database)* adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada system yang memakai pendekatan berbasis berkas.[1]

2.2.7 DBMS (Database Management System)

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak system yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang

praktis dan efisien. DBMS dapat digunakan untuk mengakomodasikan berbagai macam pemakai yang memiliki kebutuhan akses yang berbeda-beda.

Komponen-komponen yang menyusun lingkungan DBMS terdiri atas [1] :

1. Perangkat keras

   Perangkat keras digunakan untuk menjalankan DBMS beserta aplikasi-aplikasinya. Perangkat keras berupa computer dan peripheral pendukungnya. Computer dapat berupa PC, minicomputer, *mainframe*, dll.

2. Perangkat lunak

   Komponen perangkat lunak mencakup DBMS itu sendiri, program aplikasi, serta perangkat lunak computer dan jaringan. Program aplikasi dapat dibangun dengan menggunkan bahasa pemrograman seperti C++, Pascal, Delphi, atau Visual BASIC.

3. Data

   Bagi sisi pemakai, komponen terpenting dalam DBMS adalah data karena dari data inilah pemakai dapat memperoleh informasi yang sesuai dengan kebutuhan masing-masing.

4. Prosedur

   Prosedur adalah petunjuk tertulis yang berisi cara merancang hingga menggunkan basis data. Dalam hal ini prosedur terdiri dari:

a. Cara masuk ke DBMS (*login*).
b. Cara memakai fasilitas-fasilitas tertentu dalam DBMS maupun cara menggunkan aplikasi.
c. Cara mengaktifkan dan menghentikan DBMS.

5. *User* (Pengguna)

Komponen pengguna dapat dibagi menjadi tiga kelompok, antara lain:

a. Pemakai terakhir (*end-user*) adalah orang yang mengoperasikan program aplikasi yang dibuat oleh pemrogram aplikasi.
b. Pemrogram aplikasi adalah orang yang membuat program aplikasi yang melibatkan basis data.
c. Administrator basis data adalah orang yang bertanggung jawab terhadap manajemen basis data.

2.2.8 ERD (Enttity Relationship Diagram)

Setelah semua memory esensial ditransformasikan menjadi *object data store*, maka dapat disusun relasi antara satu *object data store* dengan yang lain. *Entity Relationship* (E-R) *Diagram* dibuat untuk memperlihatkan relasi itu. *Entity Relationship Diagram* adalah ilustrasi grafis objek-objek (atau *entities)* dan atribut, serta relasi antara keduanya. Diagram E-R diperkenalkan oleh Peter Chen dan sifatnya independen terhadap teknologi *database* yang digunakan.

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam **basis data** berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi**.** ERD untuk memodelkan struktur data dan hubungan antar data, untuk

menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu [2]:

a. ***Enttity***

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999: 30). Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. **Atribut**

Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar *atribut* diwakili oleh simbol elips.

c. **Hubungan / Relasi**

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu **basis data** yaitu (Abdul Kadir, 2002: 48) :

1). ***Satu ke satu (One to one)***

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2). ***Satu ke banyak (One to many)***

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada

entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3). ***Banyak ke banyak (Many to many)***

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B.

2.2.9 DFD (*Data Flow Diagram*)

Data Flow Diagram adalah suatu *network* yang menggambarkan suatu sistem automat/komputerisasi, manual atau gabungan dari keduanya dalam susunan berbentuk komponen sistem yang saling berhubungan sesuai dengan aturan terterntu. Model analisis harus dapat mencapai tiga sasaran utama, yaitu menggambarkan apa yang dibutuhkan oleh pelanggan, membangun dasar bagi pembuatan disain perangkat lunak, membatasi serangkaian persyaratan yang dpat divalidasi begitu perangkat lunak dibangun. DFD Merupakan salah satu *tools* penting yang harus dikuasai oleh seorang analis sistem. DFD dipopulerkan oleh Tom DeMarco (1978) dan Gane & Sarson (1979), dengan menggunakan metoda analisis sistem terstruktur (*Strustured System Analysis Method*). DFD dapat dipakai untuk mempresentasikan sistem secara otomatis maupun manual.

Diagram aliran data didefinisikan sebagai berikut [7]:

"Model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil".

Salah satu keuntungan menggunakan diagram aliran data adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan.

a. Penggambaran DFD

Tidak ada aturan baku untuk menggambarkan DFD, tapi dari berbagai referensi yang ada secara garis besar penggambaran DFD adalah sebagai berikut:

1. Buat diagram konteks

   Diagram ini adalah diagram level tertinggi dari DFD yang menggambarkan hubungan sistem dengan lingkungan luarnya. Dengan cara :

   a. Tentukan nama sistemnya.
   b. Tentukan batasan sistemnya.
   c. Tentukan terminator apa saja yang ada dalam sistem.
   d. Tentukan apa yang diterima/diberikan terminator dari/pada sistem.
   e. Gambarkan diagram konteks.

2. Buat diagram level *zero* atau level nol, diagram ini adalah dekomposisi dari diagram konteks. Dengan cara :

   a. Tentukan proses utama yang ada pada sistem.
   b. Tentukan apa yang diberikan/diterima masing-masing proses pada/dari sistem sambil memperhatikan konsep keseimbangan (alur data yang keluar/masuk dari suatu level harus sama dengan alur data yang masuk/keluar pada level berikutnya).

c. Apabila diperlukan, munculkan data *store* (*master*) sebagai sumber maupun tujuan alur data.
d. Gambarkan diagram level *zero* atau level nol.
e. Hindari perpotongan arus data.
f. Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses).

3. Buat diagram level satu, diagram ini merupakan dekomposisi dari diagram level *zero*. Dengan cara :
   a. Tentukan proses yang lebih kecil (sub-proses) dari proses utama yang ada di level *zero*.
   b. Tentukan apa yang diberikan/diterima masing-masing sub-proses pada/dari sistem dan perhatikan konsep keseimbangan.
   c. Apabila diperlukan, munculkan data *store* (transaksi) sebagai sumber maupun tujuan alur data.
   d. Gambarkan DFD level Satu
      1. Hindari perpotongan arus data.
      2. Beri nomor pada masing-masing sub-proses yang menunjukkan dekomposisi dari proses sebelumnya. Contoh : 1.1, 1.2, 2.1
4. DFD level dua, tiga, ..

Diagram ini merupakan dekomposisi dari level sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program. Aturan yang digunakan sama dengan level satu.

b. Elemen dasar dari data *flow diagram*

1. Entitas Luar (*External Entity*)

   Sesuatu yang berada diluar sistem, tetapi memberikan data kedalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. *External Entity* tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi *external entity*.

2. Arus Data (*Data Flow*)

   Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atas arus data yang mengalir.

3. Proses (*Process*)

   Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluar.

4. Simpanan Data (*Data Store*)

   Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data *store* dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke simpanan data (*database*).

2.2.10 Kamus data (*Data Dictionary*)

Salah satu komponen kunci dalam sistem manajemen *database* adalah *file* khusus yang disebut kamus data (*data dictionary*). Kamus data didefinisikan sebagai berikut [7]:

"Katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi".

Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir berisi informasi tentang struktur *database*. Untuk setiap elemen data yang disimpan dalam *database* seperti nomor pokok pegawai, diuraikan secara lengkap mulai dari nama, tempat penyimpanan, program komputer yang berhubungan dan lain-lain.

2.2.11 MySQL

SQL adalah sebuah konsep pengoprasian *database*, terutama untuk pemilihan/seleksi dan pemasukan data, yang memungkinkan pengoprasian data dikerjakan dengan mudah secara otomatis. Salah satu aplikasi *database* yang menggunakan bahasa SQL adalah MySQL. MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public Lisence*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source*/komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama SQL dalam *database* sejak lama. MySQL memiliki beberapa kelebihan diantaranya [4]:

1. MySQL sebagai *Data Management System* (DBMS) dan *Relation Databese Management System* (RDBMS).
2. MySQL adalah *software database* yang *OpenSource*, artinya program ini bersifat *free* atau bebas digunakan oleh siapa saja tanpa harus membeli dan membayar lisensi kepada pembuatnya.
3. MySQL merupakan *database server*, jadi dengan menggunakan *database* ini dapat menghubungkan ke media *internet* sehingga dapat diakses jarak jauh.
4. MySQL merupakan sebuah *database* client. Selain menjadi *server* yang melayani permintaan, MySQL juga dapat melakukan *query* yang mengakses *database* pada *server*, jadi MySQL dapat juga berperan sebagai *Client*.
5. MySQL mampu menerima *query* yang bertumpuk dalam satu permintaan atau yang disebut *Multi-Threading*.
6. MySQL merupakan sebuah *database* yang mampu menyimpan data berkapasitas sangat besar hingga berukuran *Gigabyte* sekalipun.
7. MySQL didukung oleh driver ODBC, artinya *database* MySQL dapat diakses menggunakan aplikasi apa saja termasuk berupa visual seperti Delphi maupun Visual Basic.
8. MySQL adalah *database* menggunakan enkripsi *password*. Jadi *database* ini cukup aman karena memiliki *password* untuk mengaksesnya.
9. MySQL dapat menciptakan lebih dari 16 kunci per tabel dan dalam satu kunci memungkinkan berisi belasan *field*. Serta MySQL mendukung *field* yang dijadikan sebagai kunci *primer* dan kunci unik atau (*Unique*).

10. MySQL memiliki kecepatan dalam pembuatan tabel maupun peng-*update*-an tabel.

SQL (*Structure Query Language*) dibagi menjadi 2 bentuk *Query*, yaitu :

1. DDL (*Data Definition Language*)

DDL adalah sebuah metode *query* SQL yang berguna untuk mendefinisikan data pada sebuah *database*, adapun *query* yang dimiliki adalah :

a. *CREATE* : untuk melakukan pembuatan tabel dan *database*

b. *DROP* : untuk melakukan penghapusan tabel maupun *database*

c. *ALTER* : untuk melakukan pengubahan struktur tabel yang telah dibuat, baik menambah *field* (*add*), mengganti mana *field* (*change*) ataupun menamakannya kembali (*rename*), serta menghapus (*drop*).

2. DML (*Data Manipulation Language*)

DML adalah sebuah metode *query* yang dapat digunakan apabila DDL telah terjadi, sehingga fungsi dari *query* ini adalah untuk melakukan pemanipulasian *database* yang telah ada atau telah dibuat sebelumnya. Adapun *query* yang termasuk didalamnya adalah :

a. *INSERT* : untuk melakukan penginputan/pemasukan data pada tabel *database*.

b. *UPDATE* : untuk melakukan perubahan/peremajaan terhadap data yang ada pada tabel.

c. *DELETE* : untuk melakukan penghapusan data pada tabel. Penghapusan ini dapat dilakukan secara sekaligus (seluruh isi tabel) maupun hanya beberapa *recordset*.

Perintah-Perintah SQL

1. Menambah Data (*Insert*)

   Perintah SQL yang digunakan untuk menambah data ke *database* adalah perintah *INSERT*. Perintah ini mempunyai bentuk umum sebagai berikut:

   INSERT [INTO] nama_tabel [(daftar *field*/kolom)] VALUES (daftar_data)

2. Memilih/Mengakses Data (*Select*)

   Perintah SQL yang digunakan untuk memilih data adalah perintah *SELECT*. Bentuk umum perintah ini adalah sebagai berikut:

   SELECT daftar_select FROM daftar_tabel

   [WHERE kondisi_pencarian]

   [GROUP BY daftar_group_by]

   [HAVING kondisi_pencarian]

   [ORDER BY daftar_order[ASC │ DESC]]

3. Mengedit/Mengubah Data (*Update*)

   Perintah SQL yang digunakan untuk mengubah data adalah perintah *UPDATE SET*. Perintah ini mempunyai bentuk umum sebagai berikut:

   UPDATE nama_table SET *field*1=databaru1 [,data2=databaru2]

   [WHERE kondisi_update]

4. Menghapus Data (*Delete*)

Perintah SQL yang digunakan untuk menghapus data adalah perintah *DELETE FROM*. Perintah ini mempunyai bentuk umum sebagai berikut:

DELETE FROM tabel_hapus WHERE kondisi_hapus

2.2.12 Borland Delphi Versi 7.0

Delphi adalah suatu program berbasis bahasa Pascal yang berjalan dalam lingkungan *Windows*. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman *Object Oriented Programming* (OOP).

Borland Delphi atau yang biasa disebut Delphi saja, merupakan sarana aplikasi visual. Delphi merupakan generasi penerus dari Turbo Pascal, Turbo Pascal yang diluncurkan pada tahun 1983 dirancang untuk dijalankan pada sistem operasi DOS (yang merupakan sistem operasi yang paling banyak digunakan pada saat itu). Sedangkan Delphi yang diluncurkan pertama kali tahun 1995 dirancang untuk beroperasi dibawah sistem operasi *Windows*.

Kelebihan Borland Delphi 7.0 yaitu telah dilengkapi dengan sejumlah komponen yang tergolong sebagai *dbExpress*, yang memungkinkan koneksi ke MySQL ataupun Oracle dilakukan dengan mudah, sehingga Delphi dapat digunakan sebagai aplikasi *front-end* yang berhubungan dengan *database server* [4].