

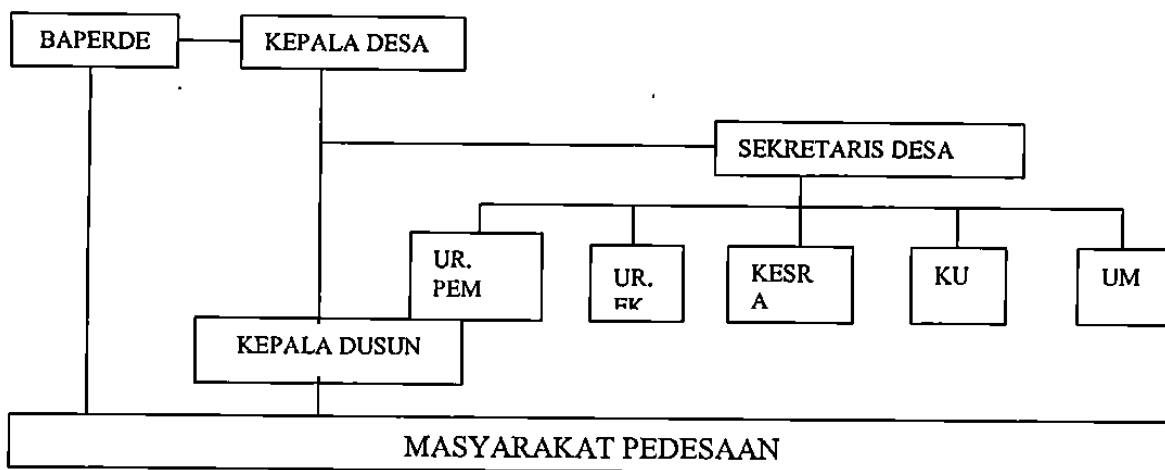
BAB II

TINJAUAN PUSTAKA

A. Definisi Desa

Desa adalah suatu wilayah yang ditempati oleh sejumlah penduduk sebagai kesatuan masyarakat termasuk didalamnya kesatuan masyarakat hukum yang mempunyai organisasi pemerintah terendah dibawah camat dan berhak menyelenggarakan rumah tangganya sendiri dalam ikatan negara kesatuan Republik Indonesia (Republik Indonesia, Undang-undang nomor 5 Tahun 1979 TENTANG PEMERINTAHAN DESA, Lembaran Negara No.56,1979).

Pemerintah telah bertekad untuk menseragamkan struktur pemerintahan desa di Indonesia, mengingat karena berbeda-bedanya struktur pemerintahan tersebut mengakibatkan kesukaran bagi pemerintah dalam rangka usaha memberikan bantuan berupa bimbingan, pembinaan, bantuan material dan finansial. Dibawah ini dikemukakan bagan struktur pemerintahan desa dan struktur pemerintahan kelurahan sesuai dengan undang-undang No.5 Tahun 1979 Tentang pemerintahan Desa.



Gambar 2.1 Struktur Organisasi Desa

Penjelasan :

Kepala Desa mempunyai tugas :

- a. Melaksanakan urusan rumah tangga desa
- b. Menumbuhkan serta mengembangkan semangat gotong-royong masyarakat dalam rangka pelaksanaan pemerintahan desa

Sekretaris Desa Mempunyai tugas :

- a. Membantu Kepala Desa dalam memimpin sekretariat desa
- b. Menjalankan administrasi pemerintahan, administrasi pembangunan dan kemasyarakatan desa serta memberikan pelayanan administratif kepada kepala desa.

Kepala-kepala urusan:

- a. melaksanakan kegiatan-kegiatan urusan sesuai dengan bidang tugasnya masing-masing
- b. Melaksanakan pelayanan administrasi kepala desa

Kepala Dusun (Polisi Desa) :

Melaksanakan kebijaksanaan Kepala Desa di wilayah kerjanya masing-masing (dusun atau blok)

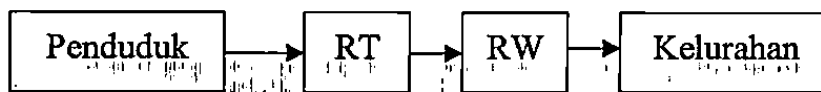
Baperdes :

Merupakan badan perwakilan di tingkat desa, para wakil rakyat atau dapat pula rakyat secara langsung, melakukan musyawarah dan mufakat dengan pemerintahan desa dalam hal kebijaksanaan pemerintahan desa agar tercapai

kecerdasan antara yang memerintah dengan yang diperintah

B. Peraturan pembuatan surat di Desa

1. Skema Pembuatan Surat



Gambar 2.2. Skema pembuatan surat

Skema pembuatan surat-surat dilakukan sesuai dengan tahapan yang telah ditentukan yaitu, apabila ada penduduk yang ingin membuat surat-surat, hal pertama yang dilakukan adalah meminta surat pengantar dari RT dan RW setempat, kemudian menyerahkan surat pengantar kepada petugas di Kelurahan untuk dibuatkan surat sesuai dengan keperluan, baru kemudian surat yang sudah dicetak ditanda tangani oleh Kepala Desa yang menandakan bahwa surat tersebut sudah dapat digunakan.

2. Daftar surat-surat yang dibuat beserta kode surat:

1. Surat Kelahiran : 474.1
2. Surat Kematian : 474.3
3. Surat Pengantar/Keterangan : 472.1
4. Surat Permohonan Keringanan Biaya : 463
5. Surat Keterangan Bepergian : 475.2
6. Surat Keterangan Pindah : 475.1
7. Surat Permohonan Dispensasi : 471.2
8. Surat pernyataan Belum Menikah : 474.2

9. Surat Permohonan Kelakuan Baik : 471.1

3. Aturan penulisan nomor surat dan nomor KTP

1. Aturan penulisan nomor surat : kode /nomor urut/bulan/desa/tahun

2. Aturan penulisan nomor KTP : kode wilayah/tanggal/lahir/kode KK

C. Konsep Dasar Sistem

Terdapat 2 kelompok pendekatan yang digunakan dalam mendefinisikan sistem, yaitu :

1. Lebih menekankan pada prosedur yang digunakan dalam sistem dan mendefinisikan sistem sebagai jaringan prosedur, metode, dan cara kerja yang saling berinteraksi dan dilakukan untuk pencapaian suatu tujuan tertentu.
2. Lebih menekankan pada elemen atau komponen penyusun sistem, mendefinisikan sebagai kumpulan elemen baik abstrak maupun fisik yang saling berinteraksi untuk mencapai tujuan tertentu.

Kedua definisi di atas sama benarnya dan tidak saling bertentangan. Yang berbeda hanyalah cara pendekatan yang dilakukan pada sistem. Karena pada hakekatnya setiap komponen sistem, untuk dapat saling berinteraksi dan untuk dapat mencapai tujuan tertentu harus melakukan sejumlah prosedur, metode, dan cara kerja yang juga saling berinteraksi. Beberapa karakteristik sistem informasi adalah sasaran, sumber daya, jaringan komunikasi, konversi data, masukan data, keluaran informasi, dan pengguna pengguna informasi

1. Sasaran

Setiap sistem berupaya mencapai satu atau lebih sasaran : artinya, sasaran merupakan kekuatan pemotivasi yang mengarahkan suatu sistem.

2. Masukan – Proses – Keluaran

Masukan terdiri dari semua arus berwujud (tangible) yang masuk ke dalam sistem di samping juga dampak tak berwujud (intangibile) terhadap sistem. Keluaran terdiri dari semua arus keluar atau hasil. Dan proses terdiri dari metode yang digunakan untuk mengubah masukan menjadi keluaran. Mekanisme kerja dalam suatu sistem dijelaskan dalam gambar berikut :



Gambar 2.3. Mekanisme Kerja Sistem

Sasaran sistem mempengaruhi dan sering mengendalikan konten masukan menjadi keluaran.

3. Lingkungan

Setiap sistem (barangkali kecuali sistem jagad raya) secara fisik terbatas. Alam yang terletak di luar suatu sistem dinamai lingkungan sistem: Suatu batas sistem memisahkan sistem itu dengan lingkungannya. Walaupun batas batas

sistem tertentu tidak kelihatan dan mungkin sukar ditetapkan secara pasti, setiap sistem pasti dibatasi oleh batas-batas tertentu.

4. Saling Kebergantungan

Setiap sistem mempunyai saling kebergantungan. Selain memiliki subsistem-subsistem yang erat berkaitan, suatu sistem pada dasarnya merupakan bagian integral dari sistem lain yang lebih besar. Hubungan antara subsistem dengan sistem dan dengan supersistem dinamai hirarki sistem.

5. Jaringan Kerja Sistem

Jaringan kerja sistem menggambarkan macam saling kebergantungan yang lain. Jaringan kerja (*network*) terbentuk bilamana sebuah sistem digabungkan dengan sistem lain yang tingkat hirarkinya sama. Sistem-sistem yang membentuk jaringan kerja berinteraksi satu sama lain melalui penghubung (*kopling/coupling*) atau batas bersama (*shared boundaries*) yang dinamakan antarmuka (*interfaces*). Antarmuka ini memungkinkan sumber daya mengalir di antara sistem-sistem yang berinteraksi.

Subsistem-subsistem yang saling bergantung dalam suatu sistem tunggal juga membentuk jaringan kerja, karena mereka saling berhubungan. Sumber daya mengalir di antara subsistem-subsistem, dengan keluaran dari satu subsistem

menjadi masukan bagi subsistem lain yang berantarmuka

Konsep saling kebergantungan sistem ini berguna dalam studi sistem informasi. Konsep ini mengingatkan analis bahwa sebuah sistem atau subsistem tidak boleh dilihat secara terpisah dari sistem atau subsistem lain yang terkait dengannya. Konsep ini juga mengatakan bahwa analis dapat berpindah ke tingkat sistem yang lebih rendah hirarkinya guna menyempitkan cakupan analisis.

6. Kendala

Setiap sistem menghadapi kendala, batasan-batasan intern atau ekstern yang menentukan konfigurasi atau kemampuan sistem. Batas / *boundary* sistem, misalnya, merupakan kendala fisik yang menentukan ukuran dan bentuk sistem. Dalam beberapa keadaan kendala dapat dihilangkan atau dikurangi. Cara yang biasa digunakan untuk mengurangi kendala yang mempengaruhi operasi dikenal dengan dekopling (*decoupling*).

7. Pengendalian Sebagai Konsep Inti Sistem

Pengendalian dapat dianggap sebagai konsepsi inti sistem, karena faktor inilah yang menjiwai ide pokok dari pengembangan sebuah sistem dan sekaligus juga merupakan manifestasi nyata dari tiap sistem. Sistem-sistem dibentuk secara langsung atau tidak, untuk melakukan pengendalian, misal :

Demonstrasi dibentuk untuk menentukan apa yang boleh dilakukan dalam

- Sistem kardiovaskuler bertanggung jawab untuk mengontrol aliran darah dan pendistribusian oksigen dalam tubuh.

Pengendalian bisa berarti menciptakan atau memelihara nilai atau karakteristik suatu variabel agar selalu berada dalam batas yang telah ditentukan. Melakukan pengendalian berarti menyebabkan segala sesuatu berjalan sesuai keinginan atau rencana sehingga suatu tujuan akhir bisa tercapai. Setiap sistem harus mengatur subsistem-subsistemnya agar dapat mencapai sasaran. Pengendalian adalah proses regulasi (pengaturan) yang dilakukan sistem untuk mengoreksi setiap penyimpangan dari rangkaian langkah menuju sasaran yang diinginkan. Pengendalian anggaran, pengendalian mutu, pengendalian kredit, dan pengendalian sediaan, merupakan proses-proses pengendalian yang biasa dilakukan perusahaan.

Pengendalian yang efektif tergantung pada umpan balik. Melalui umpan balik, keluaran sistem diukur terhadap standar untuk menentukan penyimpangan, yang kemudian dikoreksi dengan mengubah masukan atau proses. Pengendalian yang efektif dalam konteks suatu sistem juga ditegaskan dengan kaidah keragaman keperalihan (*law of requisite Variety*). Menurut kaidah ini, sistem yang baik harus memiliki satu atau lebih mekanisme atau variasi pengendalian untuk menanggulangi setiap kemungkinan keadaan yang berbeda-beda.

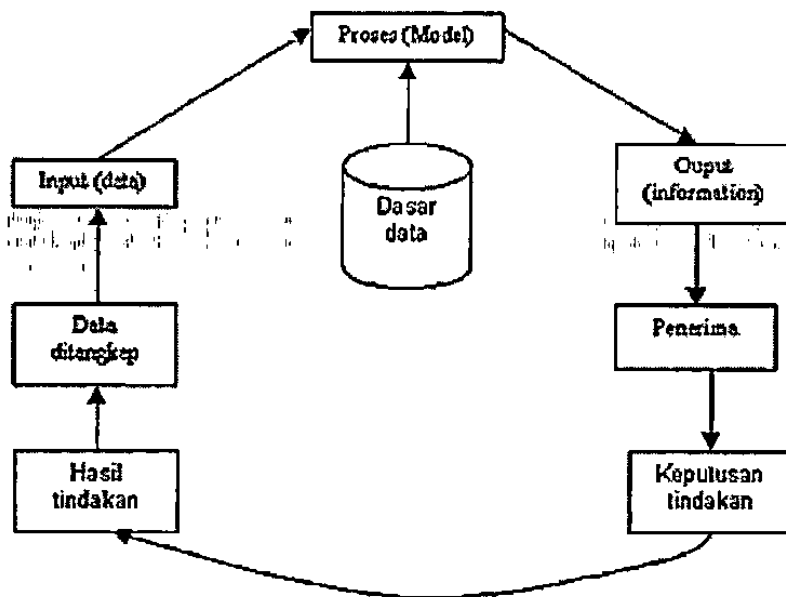
D. Konsep Dasar Informasi

Informasi dibutuhkan oleh manajemen untuk menghindari proses entropi.

Proses entropi adalah proses berakhirnya keberadaan suatu sistem manajemen yang didahului kondisi tanpa pola dan tidak menentu. Informasi adalah hasil pengolahan data yang dapat digunakan sebagai dasar pengambilan keputusan. Data sebagai bahan baku informasi adalah gambaran kejadian yang berwujud karakter, angka, atau simbol tertentu yang memiliki arti.

1. Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat berbicara banyak, sehingga perlu diolah lebih lanjut. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan tindakan lain yang akan membuat sejumlah data kembali. Data yang ditangkap dianggap sebagai input, diproses kembali melalui model dan seterusnya membentuk suatu siklus. Menurut John Burch dan Gary Grudnitski, ("Information Systems Theory and Practice", John Wiley and Sons, New York 1986) siklus ini disebut dengan Siklus Informasi (*Information Cycle*) atau Siklus Pengolahan Data (*Data*



Gambar 2.4. Siklus informasi

2. Kualitas Informasi

Agar informasi dapat mempunyai manfaat dalam proses pengambilan keputusan, informasi harus mempunyai kualitas dan nilai. Kriteria kualitas informasi adalah :

- Akurat : yang berarti informasi harus tidak bias atau menyesatkan dan bebas dari kesalahan.
- Tepat waktu : yang berarti informasi yang sampai kepada penerima tidak boleh terlambat. Mahalnya nilai informasi saat ini adalah karena harus cepatnya informasi tersebut didapatkan, sehingga diperlukan teknologi mutakhir untuk mendapatkan, mengolah, dan mengirimkan.

- Relevan : yang berarti informasi harus mempunyai manfaat bagi pihak

3. Konsep Umpan Balik Informasi

Konsep umpan balik informasi menjelaskan perihal pencarian sasaran dan saling mempengaruhi antar bagian sistem yang mengoreksi dengan sendirinya. Pada dasarnya konsep umpan balik ini berkaitan dengan cara informasi digunakan untuk maksud pengendalian. Pengendalian sebagai konsepsi inti sistem sangat membutuhkan umpan balik informasi. Informasi tentang mekanisme sistem atau input sistem jika perlu, untuk menjaga agar sistem bekerja sesuai dengan rencana pencapaian sistem.

4. Pendekatan Sistem Dalam Pertukaran Informasi

Pendekatan sistem adalah suatu gagasan yang bersumber pada paham sinergistik yang menyatakan bahwa total keluaran suatu organisasi dapat ditingkatkan jika bagian-bagian komponennya dapat diintegrasikan. Penerapan konsep umpan balik informasi, yang juga merupakan pengertian dasar pendekatan sistem, selaras dengan paham sinergistik.

E. Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen dalam pengambilan keputusan. Informasi diperoleh dari sistem informasi (*information systems*) atau disebut juga *processing systems* atau *information processing systems* atau *information-generating systems*. Menurut Robert K. Leitch dan K. Roscoe Davis, ("Accounting Information Systems",

Prentice Hall, New Jersey, 1983), hal. 6, Sistem Informasi adalah :

Suatu 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.

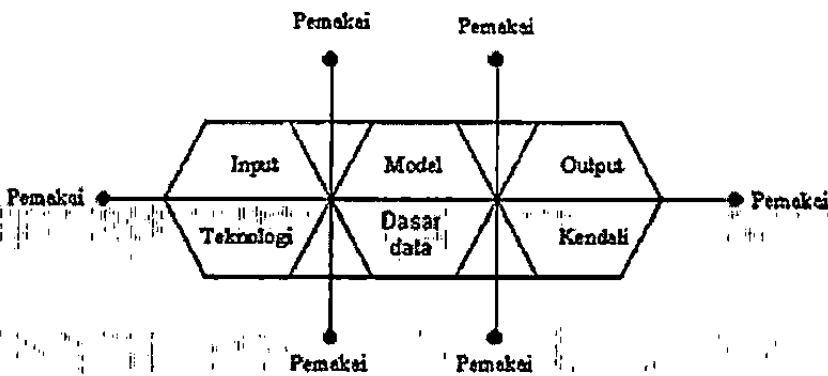
Komponen Sistem Informasi

John Burch dan Gary Grudnitski, ("Information Systems Theory and Practice", John Wiley and Sons, New York, 1986) mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya sebagai blok bangunan (*block building*).

Block building ini kemudian dibagi menjadi :

- 1 Blok masukan (*input block*)
- 2 Blok model (*model block*)
- 3 Blok keluaran (*output block*)
- 4 Blok teknologi (*technology block*)
- 5 Blok basis data (*data base block*)

6 Blok kendali (*control block*)



Gambar 2.5. Blok Bangunan

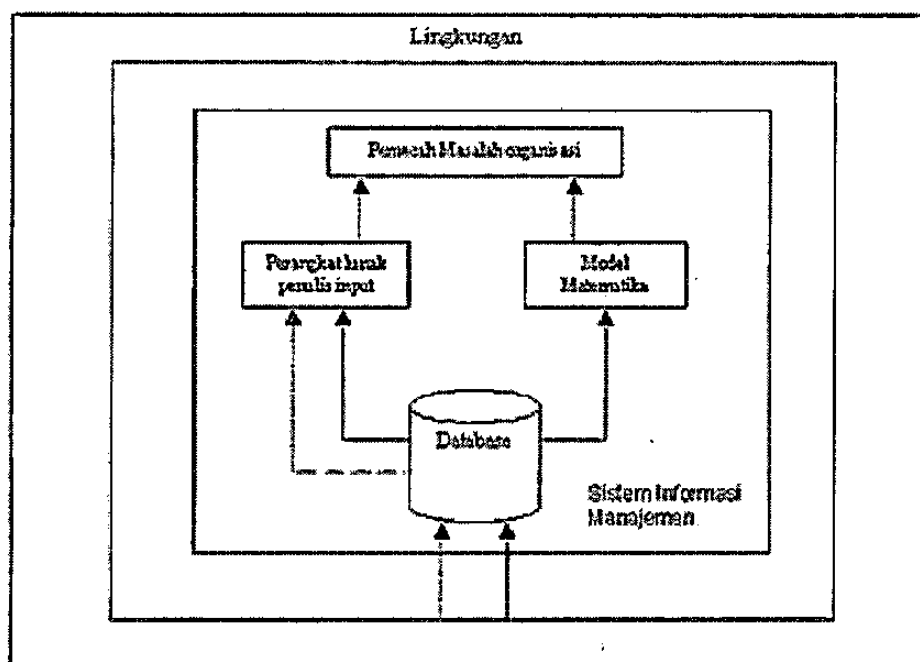
F. Konsep Dasar Sistem Informasi Manajemen

Dalam organisasi bisnis, harus ada hubungan timbal balik dan keterkaitan yang erat antara setiap fungsi manajemen dengan setiap teknik manajemen agar kondisi sinergi bisa tercapai. Fungsi manajemen menjelaskan apa yang dilakukan dan bagaimana mengendalikan sumber daya agar tujuan bisa dicapai dari dalam melakukan hal ini bersandar pada pengetahuan teknik manajemen. Untuk mewujudkan keterkaitan antara setiap fungsi manajemen dengan setiap teknik manajemen, dibutuhkan sistem informasi manajemen yang akan melingkupi seluruh fungsi dan teknik manajemen. Sistem Informasi Manajemen ini bertugas mengumpulkan, menyimpan dan mengolah data untuk akhirnya menyajikan informasi kepada semua tingkatan manajemen berkaitan dengan fungsi manajemen dalam pengelolaan sumber daya. Sistem informasi manajemen bertujuan menunjang proses pengambilan keputusan dalam melaksanakan fungsi manajemen pada berbagai tingkatan manajemen dengan mewujudkan hubungan

timbal balik dan keterkaitan informasi antar bagian organisasi sehingga sinergi organisasi dapat tercapai.

Menurut Raymond McLeod, Jr., ("Sistem Informasi Manajemen: Studi Sistem Informasi Berbasis Komputer", PT. Prenhallindo, Jakarta, ----) hal. 30 menyatakan pengertian Sistem Informasi Manajemen sebagai berikut :

Sistem Informasi Manajemen adalah suatu sistem berbasis komputer yang menyediakan informasi bagi beberapa pemakai dengan kebutuhan sistem serupa Model Sistem Informasi Manajemen menurut Raymond McLeod, Jr. ini dapat digambarkan modelnya sebagai berikut :



Gambar 2.6 Model Sistem Informasi Manajemen Raymond

Peranan Sistem Informasi dalam Proses Manajemen

Peranan sistem informasi dalam kegiatan manajemen adalah menyediakan informasi untuk menunjang proses pengambilan keputusan yang dilakukan manajemen. Informasi yang dipakai untuk membantu pengambilan keputusan dilihat dari asalnya, datang dari luar organisasi (eksternal). Tugas sistem informasi adalah menyediakan informasi yang bersifat internal. Agar informasi yang dihasilkan sistem informasi lebih mengena dan berguna bagi manajemen maka harus dilakukan analisa untuk mengetahui kebutuhan informasi bagi setiap tingkatan manajemen. Untuk maksud ini harus dimengerti dulu mengenai kegiatan dan tipe keputusan yang diambil dalam tiap-tiap tingkatan manajemen.

Sistem Informasi dan Kegiatan Manajemen

Ada 3 tingkatan manajemen, yaitu manajemen tingkat atas, menengah, dan bawah. Masing-masing tingkatan mempunyai tingkatan yang berbeda dan karena itu pengelolaan informasi yang dibutuhkan untuk pengambilan keputusan juga berbeda. Pembagian kegiatan manajemen menurut tingkatannya adalah sebagai berikut :

- Perencanaan strategis, adalah kegiatan manajemen tingkat atas, yang berupa penetapan tujuan dan penentuan strategi serta kebijaksanaan yang dibutuhkan untuk pencapaian tujuan.

- Pengendalian manajemen, adalah kegiatan manajemen tingkat menengah yang dilakukan untuk memastikan bahwa organisasi telah melaksanakan kebijakan yang telah ditetapkan untuk mencapai tujuan organisasi.
- Pengendalian operasional, merupakan kegiatan untuk memastikan bahwa tindakan-tindakan operasional telah dijalankan dengan efisien dan efektif.

Karakteristik informasi yang dibutuhkan oleh setiap tingkatan manajemen dapat dilihat pada tabel.

Tabel 2.1 Karakteristik informasi untuk setiap tingkatan manajemen

Karakteristik Informasi	Pengendalian Operasional	Pengendalian Manajemen	Perencanaan Strategis
Sumber	Internal	<----->	Eksternal
Ruang Lingkup	Sempit	<----->	Sangat Luas
Kedetailan	Sangat detail	<----->	Kurang detail
Waktu	Lampau	<----->	Masa depan
Umur Data	Sangat baru	<----->	Lama
Ketelitian	Tinggi	<----->	Rendah
Frekuensi	Sering	<----->	Jarang

G. Sistem Informasi dan Tipe Keputusan Manajemen

Secara garis besar ada 2 tipe keputusan yang dibuat oleh manajemen, yaitu keputusan yang terstruktur dan keputusan yang tidak terstruktur.

- Keputusan terstruktur, kebanyakan adalah keputusan yang repetitif, rutin, memiliki prosedur dan informasi yang jelas dan karenanya dapat diprogramkan.
- Keputusan tidak terstruktur, adalah keputusan yang tidak memiliki data masukan dan prosedur yang jelas karena jarang dilakukan sehingga tidak perlu dipersiapkan secara khusus, dan karena itu keputusan ini tidak dapat diprogramkan.

H. Sistem Informasi dan Pemakaian Komputer

Sebenarnya sistem informasi telah ada sejak dulu, yang baru pada saat ini adalah komputerasinya. Sebelum ada komputer, teknik sistem informasi manajemen telah ada untuk memberi manajer sistem informasi yang memungkinkan mereka merencanakan serta mengendalikan operasi.

Pada saat ini, ketika segala urusan yang berkaitan pemakaian informasi sudah semakin rumit dan kompleks, maka mutlak diperlukan bantuan komputer. Ada 6 karakteristik komputer sebagai pengolah informasi sebagai alasan tepat dalam pemakaian komputer, yaitu :

- Kecepatan. Dalam distribusi informasi dan pengolahan data, faktor kecepatan sangat dibutuhkan oleh manajemen.
- Kapasitas. Volume data yang sangat besar bisa diolah dengan cepat.
- Repetitif. Dalam pengolahan data yang memiliki prosedur sama, pengolahan dengan komputer akan memakan waktu yang singkat.
- Input yang pasti. Komputer membutuhkan input yang pasti, karenanya intuisi dan pertimbangan bukanlah merupakan atribut mesin.
- Output yang akurat. Hasil yang sangat akurat bisa diperoleh dan hal ini tidak terpengaruh oleh kebosanan dan kelelahan.

I. Pengenalan WEB

1. Sekilas Mengenai Web

World Wide Web (WWW) atau biasa disebut dengan web, merupakan salah satu sumber daya Internet yang berkembang pesat. Informasi web didistribusikan melalui pendekatan hypertext, yang memungkinkan suatu text pendek menjadi acuan untuk membuka dokumen yang lain. Dengan pendekatan hypertext ini seseorang dapat memperoleh informasi dengan meloncat dari suatu dokumen ke dokumen yang lain. Dokumen-dokumen yang dapat diaksespun dapat tersebar di berbagai mesin dan bahkan diberbagai negara.

2. Sejarah Web

Sejarah web dimulai pada bulan maret 1989 ketika Tim Berner-Lee yang bekerja di Laboratorium Fisika Partikel Eropa atau yang dikenal dengan nama CERN (Consei European pour la Recherche Nuclaire) yang berada di Genewa, swiss, mengajukan protokol (suatu tatacara untuk berkomunikasi) system distribusi informasi internet yang digunakan untuk berbagai informasi diantara fisikiawan.

3. Aplikasi Web

Pada awalnya aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut HTML (HyperText Markup Language) dan protocol yang digunakan dinamakan HTTP (Hypertext Transfer Protokol). Pada perkembangan berikutnya , sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML. Pada saat ini, banyak skrip seperti itu antara lain yaitu PHP dan ASP, sedangkan contoh yang berupa objek antara lain adalah applet(java).

Aplikasi web sendiri dapat dibagi menjadi :

- Web Statis
- Web Dinamis

Web statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keperluan untuk memelihara program secara terus

menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model aplikasi web dinamis.

Pengertian web yang dinamis juga terkadang diartikan sebagai halaman yang dilengkapi dengan animasi gambar, selain dapat berinteraksi dengan basis data.

Arsitektur aplikasi web :

Klien berinteraksi dengan web server, secara internal web server tadi akan berkomunikasi dengan middleware dan middleware inilah yang akan berhubungan dengan basis data (data base).

Web server adalah server yang melayani permintaan klien terhadap halaman web. Apache , IIS (Internet Information Server), dan Xitami merupakan contoh perangkat lunak web server.

Middleware adalah perangkat lunak yang bekerja sama dengan web server dan berfungsi menerjemahkan kode-kode tertentu, menjalankan kode-kode tersebut, dan memungkinkan berinteraksi dengan basis data . PHP, ASP, dan Perl adalah beberapa contoh middleware.

Browser atau web browser adalah perangkat lunak di sisi klien yang digunakan untuk mengakses informasi web. Internet explorer, Netscape, dan Mozila merupakan contoh browser.

Dinamis basis pengetahuan dalam web yang berbasis HTML adalah seperti

1. Browser meminta sebuah halaman ke suatu situs web melalui protocol HTTP.
2. Permintaan diterima oleh web server.
3. Web server segera mengirimkan dokumen HTML yang diminta ke klien.
4. Browser pada klien segera menampilkan dokumen yang diterima berdasarkan kode-kode pemformat yang terdapat pada dokumen HTML.

J. Basis Data (*Database*)

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. *Database* merupakan salah satu komponen yang penting dalam sistem informasi, karena merupakan basis dalam menyediakan informasi bagi para pemakai. Penerapan database dalam sistem informasi disebut dengan database system.

Sistem basis data (*database system*) adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam didalam suatu organisasi. Dengan sistem basis data ini tiap-tiap orang atau bagian dapat memandang database dari beberapa sudut pandang yang berbeda. Bagian

kredit dapat memandangnya sebagai data piutang, bagian penjualan dapat memandangnya sebagai data penjualan, bagian personalia dapat memandangnya sebagai data karyawan, bagian gudang dapat memandangnya sebagai data persediaan.

Semuanya terintegrasi dalam sebuah data yang umum. Berbeda dengan sistem pengolahan data tradisional, sumber data ditangani sendiri-sendiri untuk tiap aplikasinya.

K. Langkah Perancangan *Database* Secara Umum

Dalam membuat suatu database diperlukan suatu langkah atau tahapan supaya pengorganisasian file dapat menjadi lebih baik. Langkah utama tersebut adalah

1. Menentukan tipe-tipe file.

Basis data dibentuk dari suatu kumpulan file. File dalam pemrosesan transaksi dapat digolongkan sebagai berikut :

- a. File induk (*Master file*)
- b. File transaksi (*transaction file*)
- c. File laporan (*report file*)
- d. File sejarah (*history file*)
- e. File pelindung (*backup file*)
- f. File kerja (*working file*)

2. Membuat akses dan organisasi file.

Akses file (*file access*) adalah suatu metode yang menunjukkan bagaimana suatu program komputer akan membaca record-record dari suatu file. File dapat diakses dengan dua cara yaitu secara urut (*sequential access*) atau secara langsung (*direct access* atau *random access*). Metode urut dilakukan dengan membaca atau menulis suatu record di file dengan membaca terlebih dahulu mulai dari record pertama, urut sampai dengan record yang diinginkan. Metode akses langsung dilakukan dengan cara langsung membaca record pada posisinya di file tanpa membaca dari record pertama terlebih dahulu.

Organisasi file adalah pengaturan dari suatu record secara logika dalam file dihubungkan satu dengan lainnya. File dapat diorganisasikan secara urut atau secara acak. Walaupun organisasi file dan pengaksesan file dapat dipandang secara terpisah, tetapi biasanya pembahasan mengenai organisasi file menyangkut keduanya, yaitu sebagai berikut :

- a. File urut merupakan file dengan organisasi urut dengan pengaksesan secara urut pula.
- b. File urut berindeks atau sering disebut ISAM (*Indexed Sequential Access Method*) merupakan file dengan organisasi secara urut dengan pengaksesan secara langsung.

- c. File akses langsung atau disebut juga dengan file alamat langsung merupakan file dengan organisasi acak dengan pengaksesan secara langsung.

L. Sistem Manajemen Basis Data

Sistem ini merupakan perangkat lunak yang mengatur proses pengelolaan database. Pengelolaan ini meliputi pembuatan *database*, akses terhadap *database* serta penyimpanan data dalam *database*.

Sedangkan pengertian dari *database* adalah sekumpulan file-file yang paling berhubungan satu sama lain atau beberapa kunci penghubung, tersimpan dalam media penyimpanan diluar memori komputer. Media simpan ini dapat berupa disket, Hardisk.

Database dapat dinyatakan sebagai suatu sistem yang memiliki karakteristik, antara lain :

1. Merupakan suatu kumpulan "*interrelated data*" yang disimpan bersama tanpa mengganggu satu sama lain atau membentuk kerangkapan data.
2. Kumpulan data dalam *database* dapat digunakan oleh sebuah program aplikasi lebih secara optimal.
3. Penambahan data baru, modifikasi dan pengambilan kembali dari data dapat dilakukan dengan mudah dan terkontrol.

Data merupakan suatu sumber yang sangat berguna bagi hampir di semua organisasi. Dengan tersedianya data yang melimpah, maka masalah pengaturan data secara efektif menjadi suatu hal yang sangat penting dalam pengembangan sistem informasi manajemen.

Oleh karena itu, maka tujuan dari diadakannya pengaturan data adalah sebagai berikut :

1. Menyediakan penyimpanan data untuk dapat digunakan oleh organisasi saat sekarang dan masa yang akan datang.
2. Cara pemasukan data sehingga memudahkan tugas operator dan menyangkut pula waktu yang diperlukan oleh pemakai untuk mendapatkan data serta hak-hak yang dimiliki terhadap data yang ditangani.
3. Pengendalian data untuk setiap siklus agar data selalu "*up to date*" dan dapat mencerminkan perubahan spesifik yang terjadi di setiap sistem.
4. Pengamanan data terhadap kemungkinan penambahan, modifikasi, pencurian dan gangguan-gangguan lain.

M. Teknik Perancangan Database

Dalam perancangan database dikenal dua macam cara :

- Teknik Normalisasi

Cara ini dimulai dari dokumen dasar yang sudah ada pada sistem atau sudah dipakai sistem tersebut, data-data pada dokumen dasar tersebut dipisah-pisah menjadi file-file yang tiap field pada file tersebut bergantung penuh pada kunci utama (field kunci) yang biasanya dikenal dengan bentuk normal ketiga.

Kemudian setiap file dalam database tersebut ditentukan hubungannya dengan file-file yang lain dengan cara memasang field tamu pada file-file anak atau file konektor.

· Teknik Entity Relationship

Langkah ini sering digunakan pada perancangan sistem, dimulai dengan pembuatan diagram arus data yang menghasilkan kamus data yang merupakan daftar semua elemen/field yang dibutuhkan dalam sistem tersebut. Dari field-field tersebut dipilih field kunci yang bersifat unik artinya keseluruhan record dapat dicari dari record tersebut, kemudian baru dibuat file-file berdasar kunci record tersebut yang mana elemen/field dalam field tersebut bergantung penuh dengan field kunci tersebut. Setelah membuat tabel baru ditentukan relasi dari tiap tabel tersebut seperti halnya teknik normalisasi.

N. Dasar Pemrograman PHP dan MySQL

1. Hypertext Preprocessor (PHP)

PHP merupakan bahasa pemrograman *web-server-side*, *script* yang

membuat dokumen HTML secara *on the fly*; yaitu dokumen HTML yang

dihasilkan hanya pada saat diakses oleh *user*. Sehingga suatu halaman HTML tidak lagi bersifat statis, namun menjadi bersifat dinamis. Sifat *server-side* ini membuat pengerjaan *script* tersebut dikerjakan di *server* sedangkan yang dikirim kepada *browser* adalah hasil proses dari *script* tersebut sudah berbentuk HTML. Karena PHP merupakan *server-side scripting* maka PHP dapat melakukan pengumpulan *form* data, *generate dynamic page content* atau mengirim dan menerima *cookies*, dan juga banyak fungsi yang lain. Untuk dapat menjalankan fungsi sebagai *server-side scripting* maka dibutuhkan PHP *parser*, *webserver*, dan *webbrowser*. *Webbrowser* perlu dijalankan dan digabungkan dengan instalasi PHP. Untuk mengakses PHP, dapat digunakan *web browser* yang dapat melihat hasil dari *script* PHP.

Keunggulan dari PHP dengan sifat *server-side* adalah:

- Tidak diperlukan adanya kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena *server* yang mengerjakan *script* tersebut. Hasil yang dikirimkan kembali ke *browser* biasanya dalam bentuk *format* HTML, yang berisi teks ataupun gambar sehingga dapat dikenali oleh *browser* apapun.
- Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, contoh : hubungan ke dalam *database*.
- *Script* asli tidak dapat dilihat oleh *browser* sehingga keamanan lebih

PHP adalah sebuah produk *open source*, sehingga *source code* PHP dapat digunakan, diubah. Keunggulan PHP selain sifatnya yang *open source* adalah *multi platform* selain dapat dijalankan pada *platform Linux*, PHP juga dapat dijalankan dengan menggunakan *Apache*, dengan IIS pada Window NT atau PWS pada Windows 98.

PHP dibuat pada tahun 1994 oleh Rasmus Lerdorf. Tetapi kemudian dikembangkan oleh orang lain dan setelah melalui tiga kali karya penulisan akhirnya PHP menjadi bahasa pemrograman *web* seperti sekarang ini. Pada bulan Januari 2001 PHP sudah kurang lebih digunakan lebih dari 5 juta *domain* dan terus berkembang hingga saat ini. Jumlahnya hingga saat ini dapat dilihat pada <http://www.php.net/usage.php>.

2. MySQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) yang sangat cepat dan kokoh. Dengan menggunakan MySQL Server maka data dapat diakses oleh banyak pemakai secara bersamaan sekaligus dapat membatasi akses para pemakai berdasarkan *privilege* (hak akses) yang diberikan. MySQL menggunakan bahasa SQL (*Structured Query Language*) yaitu bahasa standar yang digunakan untuk pemrograman *database*. MySQL dipublikasikan sejak 1996, tetapi sebenarnya dikembangkan sejak 1979. MySQL telah memenangkan penghargaan *Linux Journal Reader's Choice Award* selama tiga tahun.

¹ Dokumentasi PHP, <http://www.php.net/docs.php>

MySQL sekarang tersedia dibawah izin *open source*, tetapi juga ada izin untuk penggunaan secara komersial. Keunggulan dari MySQL adalah:

- Bersifat *open source*.
- Sistem yang digunakan oleh perangkat lunak ini tidak memberatkan kerja dari *server*, karena dapat bekerja di *background*.
- Mempunyai koneksi yang stabil dan kecepatan yang tinggi.

2.1. Structured Query Language (SQL)

Structured Query Language (SQL) adalah bahasa yang digunakan untuk berkomunikasi dengan *database*. Perintah-perintah SQL digunakan untuk melakukan pekerjaan-pekerjaan tertentu pada *database* seperti *update* data atau untuk mengambil data dari *database*. Pada subbab-subbab berikut ini akan dijelaskan mengenai perintah *select*, *insert*, *update* dan *delete* pada SQL.

○ Perintah *Select*

Perintah *select* digunakan untuk mengambil dan menampilkan data dari *database* sesuai dengan syarat-syarat yang ditentukan. Berikut ini adalah format perintah *select* yang umum digunakan:

```
SELECT column1 [, column2, etc]
FROM tablename
[WHERE condition];
```

keterangan: [] = optional

Column menunjukkan nama kolom dari tabel yang ingin diambil. *Column* dapat lebih dari satu atau menggunakan "*" untuk

mengambil semua kolom. *Tablename* menunjukkan nama tabel yang ingin digunakan. Klausa *where* menyaring data berdasarkan kondisi yang dituliskan setelah kata *where*. Operator-operator yang digunakan dalam klausa *where* antara lain dapat dilihat pada tabel berikut:

Tabel 2.2. Operator Untuk Perintah SQL

Lambang	Arti
=	Sama dengan
>	Lebih besar dari
<	Lebih kecil dari
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
<>	Tidak sama dengan
LIKE	(dijelaskan di bawah ini)

Operator *LIKE* dapat menyeleksi data sesuai dengan *string* yang diinginkan. Simbol “%” digunakan untuk mencocokkan karakter yang muncul sebelum atau sesudah karakter yang diinginkan. Contohnya adalah sebagai berikut:

```
SELECT nama_depan, kota
FROM karyawan
WHERE nama_depan LIKE 'Sam%';
```

Perintah SQL di atas memiliki *output* semua nama depan yang diawali dengan “Sam”. Simbol “%” dapat pula diletakkan di

depan atau di belakang. Selain seperti yang dijabarkan di atas, perintah *select* juga memiliki format lain yang lebih lengkap yang mendukung fungsi agregasi, yaitu:

```

SELECT [ALL | DISTINCT] column1[,column2]
FROM table1[,table2]
[WHERE "conditions"]
[GROUP BY "column-list"]
[HAVING "conditions"]
[ORDER BY "column-list" [ASC | DESC] ]

```

Tabel 2.3. Fungsi Agregasi

Nama Fungsi	Kegunaan
<i>SUM()</i>	Untuk menghitung jumlah nilai pada suatu kolom
<i>AVG()</i>	Untuk menghitung nilai rata-rata pada suatu kolom
<i>MAX()</i>	Untuk mencari nilai maksimum pada suatu kolom
<i>MIN()</i>	Untuk mencari nilai minimal pada suatu kolom
<i>COUNT()</i>	Untuk menghitung jumlah baris pada suatu kolom

Keyword *ALL* digunakan untuk mengambil semua data sedangkan *DISTINCT* digunakan untuk mengambil data yang unik dimana baris yang sama akan dijadikan satu baris saja.

- o *Klausula Group By*

Klausula *group by* akan mengumpulkan baris data dari tabel

adalah fungsi agregasi pada satu atau lebih kolom. Untuk lebih jelas dapat dilihat pada contoh berikut:

```
SELECT MAX(gaji), departemen
```

```
FROM karyawan
```

```
GROUP BY departemen;
```

Query ini akan mengambil dan menampilkan gaji maksimum dari karyawan dari tiap departemen dan dikelompokkan berdasarkan nama departemennya.

o Klausula *Having*

Klausula *having* dapat menspesifikasikan kondisi dari baris untuk tiap *group*, dengan kata lain tiap baris akan diambil berdasarkan kondisi yang dispesifikasikan. Jika menggunakan klausula *having* harus diikuti dengan klausula *group by*. Contoh penggunaan dari klausula *having* adalah sebagai berikut:

```
SELECT          departemen,          avg(gaji)
FROM            karyawan
GROUP          BY          departemen
HAVING avg(gaji) > 20000;
```

Pada *query* di atas jika pada *query* tersebut hanya menggunakan *group by* saja maka akan mengambil rata-rata gaji karyawan pada tiap departemen. Tetapi bila ditambahkan dengan klausula *having* maka dapat dispesifikasikan rata-rata gaji yang diambil adalah yang bernilai lebih dari 20000.

- *Klausula Order By*

Klausula *order by* merupakan klausula *optional* yang digunakan jika kita ingin menampilkan hasil dari *query* secara urut, baik urut secara *ascending (ASC)* maupun secara *descending (DESC)*, sesuai dengan kolom yang dispesifikasikan pada *order by*. Contoh penggunaannya jika diinginkan hasil urut data karyawan berdasarkan umurnya secara *ascending* adalah sebagai berikut:

```
SELECT kode_karyawan, nama_depan, umur, gaji
FROM karyawan
WHERE departemen = 'Penjualan'
ORDER BY umur ASC;
```

Bila ingin mengurutkan berdasarkan dua atau lebih kolom, maka harus dipisahkan dengan tanda koma, seperti dijabarkan di bawah ini:

```
SELECT kode_karyawan, nama_depan, umur, gaji
FROM karyawan
WHERE departemen = 'Penjualan'
ORDER BY umur, gaji DESC;
```

Query di atas akan mengurutkan hasil *query* berdasarkan umur, baru kemudian mengurutkan berdasarkan gaji secara *descending*.

- *Join Table*

Dalam kenyataannya, *query* tidak hanya dari satu tabel saja,

melainkan dua atau lebih. Untuk itu diperlukan *join table* untuk

database relational. Dengan *join table* maka hanya dengan satu *query* yang menggunakan perintah *select* dapat menggabungkan data dari dua atau lebih tabel ke dalam satu hasil *query*.

Antar dua tabel yang berelasi pasti terdapat satu kolom yang sama. Kolom inilah yang digunakan untuk menggabungkan dua tabel tersebut. Caranya adalah sebagai berikut:

```
SELECT konsumen.nama_depan, penjualan.no_faktur
FROM           konsumen,           penjualan
WHERE
konsumen.kode_konsumen=penjualan.kode_konsumen;
```

Join table di atas merupakan *inner join* dimana akan menghasilkan semua nama depan dari konsumen (yang diambil dari tabel konsumen) dan nomor faktur penjualan (yang diambil dari tabel penjualan) dari semua baris dimana kode konsumen di tabel konsumen sama dengan kode konsumen di tabel penjualan.

o Perintah *Insert*

Perintah *insert* digunakan untuk memasukkan atau menambah baris data ke dalam sebuah tabel. Format penulisan perintah *insert* adalah sebagai berikut:

```
INSERT INTO tablename (first_column, ...,
last_column)
VALUES (first_value, ..., last value);
```

Parameter setelah *tablename* adalah daftar kolom-kolom yang dipisahkan dengan koma, diikuti dengan kata *values* lalu diikuti lagi

database relational. Dengan join table maka hasil dengan satu query yang menggunakan perintah select dapat menggabungkan data dari dua atau lebih tabel ke dalam satu hasil query.

Antar dua tabel yang berelasi pasti terdapat satu kolom yang sama. Kolom inilah yang digunakan untuk menggabungkan dua tabel tersebut. Caranya adalah sebagai berikut:

```
SELECT konsumen.nama_denda, penjualan.no_faktur
FROM konsumen, penjualan
WHERE
konsumen.kode_konsumen=penjualan.kode_konsumen;
```

Join table di atas merupakan inner join dimana akan menghasilkan semua nama depan dari konsumen (yang diambil dari tabel konsumen) dan nomor faktur penjualan (yang diambil dari tabel penjualan) dari semua baris dimana kode konsumen di tabel konsumen sama dengan kode konsumen di tabel penjualan.

o Perintah insert

Perintah insert digunakan untuk memasukkan atau menambah baris data ke dalam sebuah tabel. Format penulisan perintah insert adalah sebagai berikut:

```
INSERT INTO table_name (first_column, ...
VALUES (first_value, ... last value);
```

Parameter setelah keyword adalah daftar kolom-kolom yang dipisahkan dengan koma. diikuti dengan kata where lalu diikuti lagi

sebelum kata *values*. Untuk nilai yang bernilai *string* harus diapit dengan tanda petik satu sedangkan untuk yang bernilai integer tidak.

Contohnya adalah sebagai berikut:

```
INSERT INTO karyawan (nama_depan, nama_belakang,
umur, kota) VALUES ('Sam', 'Ali', 21,
'Surabaya');
```

Query di atas akan menambahkan baris baru pada tabel karyawan dengan *field* nama_depan bernilai "Sam", *field* nama_belakang bernilai "Ali", *field* umur bernilai 21 dan *field* kota bernilai "Surabaya".

o Perintah *Update*

Perintah *update* digunakan untuk mengubah data yang sudah ada berdasarkan kondisi tertentu yang dijabarkan oleh *keyword where*. Format perintah lengkapnya adalah sebagai berikut:

```
UPDATE tablename
SET columnname =newvalue[,nextcolumn = newvalue2..]
WHERE columnname <OPERATOR> value
[and|or column" <OPERATOR> "value"];
keterangan : [] = optional
```

Contoh penggunaan :

```
UPDATE karyawan
SET umur = 15
```

o Perintah *Delete*

Perintah delete digunakan untuk menghapus baris dari suatu tabel. Formatnya adalah sebagai berikut:

```
DELETE FROM tablename
WHERE columnname <OPERATOR> value [and|or column
<operator> value]
keterangan: [] = optional
```

Contoh penggunaannya adalah seperti di bawah ini:

```
DELETE FROM karyawan
WHERE nama_depan = 'Sam';
```

Contoh di atas akan menghapus baris dari tabel “karyawan” yang field “nama_depan”-nya bernilai “Ali”. Untuk menghapus seluruh baris dari suatu tabel, kosongkan klausa *where*.

2.2.Koneksi PHP dan MySQL

Untuk menghubungkan bahasa pemrograman PHP dengan MySQL dibutuhkan beberapa perintah-perintah khusus, yang terdiri dari:

Pembuatan koneksi antara *server* dari MySQL dengan *web server* tempat menyimpan halaman *web*. Dengan perintah :

```
<? $conn = mysql_connect("nama server
MySQL",login,password); ?>
```

`$conn` adalah variabel bebas yang digunakan untuk menyimpan koneksi antara PHP dan MySQL. Setelah terbentuk koneksi maka selanjutnya dilakukan pemilihan *database* yang akan digunakan dengan perintah :

```
<? $conn = mysql_select_db("nama database"); ?>
```

Barulah kemudian dapat dilakukan perintah-perintah SQL yang lain seperti *select*, *update*, *delete*, *insert*, dan perintah-perintah lainnya.

```
<? $result = mysql_query("perintah query", $conn);  
?>
```

\$result adalah variabel bebas yang digunakan untuk menyimpan hasil dari *query*. Untuk *select query* dilakukan proses pengambilan data dengan perintah:

```
<? $row = mysql_fetch_row($result); ?>
```

Setelah selesai melakukan manipulasi data maka koneksi PHP dan MySQL