BAB II
TINJAUAN PUSTAKA

II.1 Profil SMAN 1 Margaasih

SMAN 1 Margaasih, yang berlokasi di Jl. Terusan Taman Kopo Indah III - Mekarrahayu merupakan salah satu sekolah rintisan yang berada di Kabupaten Bandung. SMAN 1 Margaasih dipimpin oleh Bpk. Drs Usep Sutarman MPd. SMAN 1 Margaasih ini sudah berakreditasi "B" (BAIK). Jumlah tenaga pengajar atau guru sebanyak 33 orang guru tetap dan 20 orang guru tidak tetap, sedangkan untuk bagian Tata Usaha (TU) sebanyak 2 orang TU tetap dan 6 orang TU tidak tetap. Bangunan yang dimiliki SMAN 1 Margaasih berjumlah 13 kelas yang di bagi untuk kelas X, XI, dan XII.

II.1.1 Visi dan Misi

Visi SMAN 1 Margaasih : Membentuk Insan yang Kompetitif dan religius yang bertumpu pada lingkungan yang bersih dan sehat.

Misi SMAN 1 Margaasih :

a. Meningkatkan mutu akademis.
b. Meningkatkan mutu non akademis.
c. Membangun dan mengembangkan imajinasi.
d. Mengembangkan dan membimbing keagamaan.
e. Membina dan mengembangkan disiplin dan ketertiban.

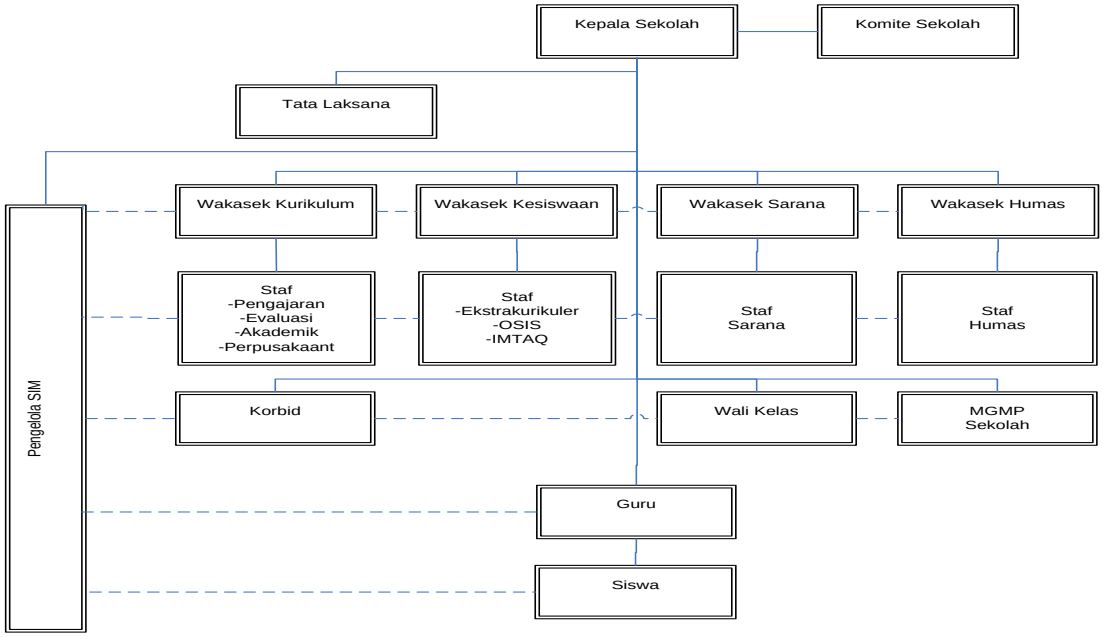
II.1.2 Logo SMAN 1 Margaasih

Gambar II.1 Logo SMAN 1 Margaasih

II.1.3 Struktur Organisasi

Struktur organisasi menunjukan pola hubungan jalur komunikasi diantara fungsi-fungsi, bagian-bagian atau orang-orang yang ada dalam suatu organisasi dapat diketahui bagaimana wewenang dan tanggung jawab mengalir diantara bagian tersebut.

Dibawah ini adalah Bagan Struktur Organisasi SMAN 1 Margaasih, seperti yang ditunjukan pada gambar II.2



Gambar II.2 Struktur Organisasi SMAN 1 MARGAASIH

II.1.4 Deskripsi Pekerjaan

1. Kepala Sekolah
    a. Merencanakan pengembangan penyelenggaraan pendidikan.
    b. Mengorganisasikan seluruh proses pendidikan di sekolah yang meliputi aspek edukatif dan administratif.
    c. Mengetahui perkembangan siswa.
    d. Mengetahui perkembangan guru bidang studi.

2. Wakil Kepala Sekolah Bidang Kurikulum
    a. Menyusun kalender pendidikan.
    b. Menyusun jadwal pelajaran.
    c. Menyusun program semester, satpel, dan renpel.
    d. Mengatur pelaksanaan penilaian.
    e. Menyusun pembagian tugas mengajar.
    f. Menyusun Wali kelas.

3. Wakil Kepala Sekolah Bidang Kesiswaan
    a. Mengatur penerimaan siswa baru.
    b. Mengatur program BP.
    c. Mengatur penasehatan pemilihan program.
    d. Mengatur pengelompokan belajar siswa.
    e. Meneliti kehadiran siswa.
    f. Mengatur keaktifan kegiatan OSIS.
    g. Mengatur mutasi siswa.
    h. Mengatur kegiatan MPLS.

4. Staff Urusan Sarana dan Prasarana
    a. Pengadaan sarana dan prasarana administrasi tata laksana.
    b. Rehabilitasi buku-buku.
    c. Inventarisasi buku-buku siswa

d. Penjualan buku-buku pada siswa

e. Usul perubahan asrama

5. Kepala Perpustakaan dan Staff Perpustakaan

a. Mengelola penyimpanan buku-buku

b. Melakukan pencatatan koleksi bahan pustaka.

c. Melakukan pengelompokan koleksi dengan memberi kode-kode tertentu.

d. Penyusunan koleksi pada rak-rak sehingga mempermudah dalam perpustakaan.

6. Pengelola SIM (Sistem Informasi Manajemen)

Mengelola seluruh data yang ada di sekolah.

7. Tata Laksana

a. Menyiapkan rencana anggaran organisasi.

b. Mengkoordinasikan bagian administrasi.

c. Membantu pimpinan dalam menyiapkan rencana pendidikan.

8. Wali Kelas

Bertanggung jawab atas seluruh siswa dalam satu kelas.

9. Guru mata pelajaran

a. Melakukan proses belajar mengajar.

b. Menyiapkan ringkasan materi pelajaran.

c. Mengetahui perkembangan siswa.

10. Siswa

Melakukan proses belajar mengajar.

II.2 Pengertian Internet

Menurut Lani Sidharta menyebutkan :

"Internet adalah suatu interkoneksi sebuah jaringan komputer yang dapat memberikan layanan informasi secara lengkap. Dan, terbukti internet dilihat sebagai media maya yang dapat menjadi rekan bisnis, politik, sampai hiburan. Semuanya tersaji lengkap dalam media ini" **[2]**.

II.3 *E-Learning*

Salah satu kosa kata yang muncul dan populer bersamaan dengan hadirnya TIK dalam dunia pembelajaran adalah e-*learning*. *E-learning* merupakan kependekan dari elektronik *learning* yang berarti belajar dengan menggunakan media elektronik. Kata elektronik sendiri mengandung pengertian yang spesifik yakni komputer dan jaringan internet, sehingga *e-learning* sering diartikan sebagai proses belajar yang menggunakan komputer dan jaringan internet.

II.3.1 Konsep *E-Learning*

Kemajuan internet mempengaruhi hampir setiap sendi kegiatan operasional di organisasi. Banyak kegiatan perusahaan mulai dilakukan lewat internet dan menyebabkan fenomena penggunaan awalan "e" dan "*online*" di kamus bisnis. *E-commerce*, *e-mail*, *online application, e-procurement*, *online hiring*, *e-CRM*, e-*HRM*, *online auction, e-catalogue,* adalah contoh tren penggunaan internet pada kegiatan yang biasa dilakukan secara manual. Segala kegiatan mutakhir tersebut menjanjikan efektivitas dan efisiensi yang menakjubkan. Fenomena tersebut menyentuh dunia pendidikan dan pelatihan dengan lahirnya *e-learning.*

II.3.2 Apa itu *E-Learning*

E-learning atau elektronik *learning* adalah pembelajaran yang dilaksanakan dengan memanfaatkan fungsi internet dalam kegiatan pembelajaran dengan menjadikan fasilitas elektronik sebagai media pembelajaran [**3**].

Didunia pendidikan dan pelatihan sekarang banyak sekali praktik yang disebut *e-learning*. Sampai saat ini, pemakaian kata *e-learning* sering digunakan semua kegiatan pendidikan yang menggunakan media komputer dan internet. Banyak pula penggunaan terminologi yang memiliki arti hampir sama dengan *e-learning*. *Web-based learning*, *online learning, computer-aided instruction*, dan lain sebagainya adalah terminologi yang sering digunakan untuk menggantikan *e-learning*. Terminologi *e-learning* sendiri dapat mengacu pada semua kegiatan pelatihan yang menggunakan media elektronik atau teknologi informasi.

Karena ada bermacam penggunaan *e-learning* saat ini, maka ada pembagian atau pembedaan *e-learning*. Pada dasarnya, *e-learning* mempunyai dua tipe, yaitu *synchronous* dan *asynchronous.*

a. *Synchronous Training*

Synchronous berarti "pada waktu yang sama". Jadi, *synchronous training* adalah tipe pelatihan, dimana proses sedang mengajar dan siswa sedang belajar. Hal tersebut memungkinkannya interaksi langsung antara guru dan siswa, baik melalui internet maupun intranet. *Synchronous training* mengharuskan guru dan siswa mengakses internet bersamaan. Pengajar memberikan makalah dengan slide presentasi dan peserta *web conference* dapat mendengarkan presentasi melalui hubungan internet. Jadi *synchronous training* sifatnya mirip pelatihan di ruang kelas. Namun, kelasnya bersifat maya (*virtual*) dan peserta tersebar di seluruh dunia dan terhubung melalui internet. Oleh karena itu, *synchronous training* dinamakan *virtual classroom*.

b. *Asynchronous Training*

Asynchronous training berarti "tidak pada waktu yang bersamaan". Jadi, seseorang dapat mengambil pelatihan pada waktu yang berbeda dengan pengajar memberikan pelatihan. Pelatihan ini lebih popular di dunia *e-learning* karena memberikan keuntungan lebih bagi peserta pelatihan karena dapat mengakses

pelatihan kapanpun dan dimanapun. Pelatihan berupa paket pelajaran yang dapat dijalankan di komputer manapun dan tidak melibatkan interaksi dengan pengajar atau pelajar lainnya. Oleh karena itu, pelajar dapat memulai pelajaran dan menyelesaikannya setiap saat. Paket pelajaran berbentuk bacaan dengan animasi, simulasi, permintaan edukatif, maupun latihan atau tes dengan jawabannya. Akan tetapi, ada pelatihan *Asynchronous training* yang terpimpin, dimana pengajar memberikan materi pelajaran lewat internet dan peserta pelatihan mengakses materi pada tugas atau latihan dan peserta mengumpulkan tugas lewat *e-mail*. Peserta dapat berdiskusi atau berkomentar dan bertanya melalui *bulletin board*.

II.3.3 Kelebihan *E-Learning*

Kemajuan penggunaan *e-learning* dimotivasi oleh kelebihannya. Kelebihan yang ditawarkan *e-learning* :

a. Biaya

Dengan adanya *e-learning*, instansi tidak perlu mengeluarkan biaya untuk menyewa pelatih dan ruang kelas serta transportasi peserta pelatihan atau pelatih. Instansi tidak perlu menyediakan makan siang, kopi, maupun peralatan kelas seperti papan tulis, proyektor dan alat tulis.

b. Fleksibilitas Waktu

E-learning membuat siswa dapat menyesuaikan waktu belajar. Siswa dapat menyisipkan waktu belajar setelah istirahat, setelah pulang sekolah, atau kapanpun saat siswa memiliki waktu luang. Siswa mudah mengakses *e-learning* ketika waktu sudah tidak memungkinkan atau ada hal lain yang lebih mendesak, siswa dapat meninggalkan pelajaran di *e-learning* saat itu juga. Banyak program pelajaran *e-learning* memilki fasilitas *bookmark*. Fasilitas tersebut membuat siswa yang kembali mengakses *e-learning* secara otomatis dibawa ke halaman terakhir pelajaran sebelumnya. Oleh karena itu, siswa dengan cepat dan nyaman melanjutkan pelajaran.

c. Fleksibilitas Tempat

Adanya *e-learning* membuat siswa sanantiasa dapat mengakses pelatihan atau untuk mengikuti tambahan pelajaran hanya dengan mengakses *e-learning* yang bisa dilakukan di rumah, warnet, lab. komputer, atau dimanapun selama komputer terhubung dengan internet.

d. Fleksibilitas Kecepatan Pembelajaran

E-learning dapat disesuaikan dengan kecepatan belajar masing-masing siswa. Siswa mengatur sendiri kecepatan pelajaran yang diikuti. Apabila belum mengerti, siswa dapat tetap mempelajari modul tertentu dan mengulanginya nanti. Apabila seorang siswa mengerti dengan cepat, siswa dapat menyelesaikan pelajaran lebih cepat dan mengisi waktu dengan belajar topik lain. Hal ini berbeda sekali dengan pelatihan di kelas karena semua siswa mulai dan berhenti di waktu yang sama.

e. Efektivitas Pengajaran

E-learning yang di desain dengan *instructional design* mutakhir membuat siswa lebih mengerti isi pelajaran. Penyampaian pelajaran *e-learning* dapat berupa simulasi dan kasus-kasus, menggunakan bentuk permainan dan menerapkan teknologi animasi canggih. Bentuk-bentuk pembelajaran tersebut dapat membantu proses pembelajaran dan mempertahankan minat belajar.

f. Kecepatan Distribusi

E-learning dapat cepat menjangkau siswa yang berada diluar sekolah. Tim hanya perlu mempersiapkan bahan latihan secepatnya dan menginstal hasilnya diserver pusat *e-learning*. Jadi, semua komputer yang terhubung ke *server* dapat langsung mengakses.

g. Otomatisasi Proses Administrasi

Learning Management System (LMS) dapat menyimpan dan membuat laporan tentang kegiatan belajar seorang siswa, mulai dari pelajaran yang telah diambil,

tanggal akses, berapa persen pelajaran yang diselesaikan, berapa lama pelajaran yang diikuti, sampai berapa hasil tes akhir yang diambil. Sehingga guru dapat setiap saat mencetak sendiri laporan dengan otomatis untuk memonitor kemajuan belajar siswanya.

h. Keterbatasan *On-Demand*

Karena *e-learning* dapat sewaktu-waktu diakses, banyak yang menganggapnya sebagai ”buku saku” yang membantu pekerjaan setiap saat. Sebagai contoh, bila siswa akan membuat grafik di program *Microsoft Excel* dan tidak tahu caranya, maka siswa dapat langsung memasuki program *e-learning* yang mengajarkan aplikasi *Microsoft Excel* dan langsung kebagian tentang grafik, dengan begitu dalam waktu 15 menit siswa dapat segera mempraktikan pelajaran dan menyelesaikan tugas dengan baik.

II.4 *Text Mining*

Text mining adalah salah satu bidang khusus dari *data mining*. Sesuai dengan buku *The Text Mining Handbook*, *text mining* dapat didefinisikan sebagai suatu proses menggali informasi dimana seorang *user* berinteraksi dengan sekumpulan dokumen menggunakan *tools* analisis yang merupakan komponen-komponen dalam *data mining* yang salah satunya adalah kategorisasi **[4]**. Tujuan dari *text mining* adalah untuk mendapatkan informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan pada *text mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau minimal semi terstruktur. Adapun tugas khusus dari *text mining* antara lain yaitu pengkategorisasian teks *(text categorization)* dan pengelompokan teks *(text clustering)*.

Text mining bisa dianggap subjek riset yang tergolong baru. *Text mining* dapat memberikan solusi dari permasalahan seperti pemrosesan, pengorganisasian atau pengelompokkan dan menganalisa *unstructured text* dalam jumlah besar. Dalam memberikan solusi, *text mining* mengadopsi dan mengembangkan banyak teknik dari

bidang lain, seperti *Data mining*, *Information Retrieval*, Statistik dan *Matematik*, *Machine Learning*, *Linguistic*, *Natural Languange Processing*, dan *Visualization*. Kegiatan riset untuk *text mining* antara lain ekstraksi dan penyimpanan *text*, *preprocessing* akan konten *text*, pengumpulan data statistik dan *indexing* dan analisa konten.

Permasalahan yang dihadapi pada *text mining* sama dengan permasalahan yang terdapat pada data *mining*, yaitu jumlah data yang besar, dimensi yang tinggi, data dan struktur yang terus berubah, dan data *noise*. Perbedaan di antara keduanya adalah pada data yang digunakan. Pada data mining, data yang digunakan adalah *structured data*, sedangkan pada *text mining*, data yang digunakan *text mining* pada umumnya adalah *unstructured data*, atau minimal *semistructured*. Hal ini menyebabkan adanya tantangan tambahan pada *text mining* yaitu struktur text yang *complex* dan tidak lengkap, arti yang tidak jelas dan tidak standar, dan bahasa yang berbeda ditambah translasi yang tidak akurat.

Salah satu elemen kunci dari *text mining* adalah kumpulan dokumen yang berbasis teks. Pada prakteknya, *text mining* ditujukan untuk menemukan pola dari sekumpulan dokumen yang jumlahnya sangat besar dan bisa mencapai jumlah ribuan bahkan sampai jutaan. Koleksi dokumen bisa statis, dimana dokumen tidak berubah, atau dinamis, dimana dokumen selalu di*update* sepanjang waktu.

Algoritma yang digunakan pada *text mining*, biasanya tidak hanya melakukan perhitungan hanya pada dokumen, tetapi juga pada fitur. Empat macam fitur yang sering digunakan:

1. *Character*, merupakan komponan individual, bisa huruf, angka, karakter spesial dan spasi, merupakan blok pembangun pada level paling tinggi pembentuk *semantik feature*, seperti kata, *term* dan *concept*. Pada umumnya, representasi *character-based* ini jarang digunakan pada beberapa teknik pemrosesan teks.
2. *Words.*

3. *Terms* merupakan *single word* dan frasa *multiword* yang terpilih secara langsung dari *corpus*. Representasi *term-based* dari dokumen tersusun dari subset *term* dalam dokumen.
4. *Concept*, merupakan *feature* yang di-*generate* dari sebuah dokumen secara manual, *rule-based*, atau metodologi lain.

II.4.1 Pencarian

Pencarian data sering juga disebut *table look-up* atau *storage and retrival information* adalah suatu proses untuk mengumpulkan sejumlah informasi di dalam pengingat komputer dan kemudian mencari kembali informasi yang di perlukan secepat mungkin **[5]**.

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).

Metode pencarian data dapat dilakukan dengan dua cara yaitu pencarian internal (*internal searching*) dan pencarian eksternal (*external searching*). Pada pencarian internal, semua rekaman yang diketahui berada dalam pengingat komputer sedangkan pada pencarian eksternal, tidak semua rekaman yang diketahui berada dalam pengingat komputer, tetapi ada sejumlah rekaman yang tersimpan dalam penyimpan luar misalnya pita atau cakram magnetis.

Selain itu metode pencarian data juga dapat dikelompokan menjadi pencarian statis (*static searching*) dan pencarian dinamis (*dynamic searching*). Pada pencarian statis, banyak rekaman yang diketahui dianggap tetap, pada pencarian dinamis, banyak rekaman yang diketahui bisa berubah-ubah yang disebabkan oleh penambahan atau penghapusan suatu rekaman.

Ada dua macam teknik pencarian yaitu pencarian sekuensial dan pencarian biner. Perbedaaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial

digunakan apabila data dalam keadaaan acak atau tidak terurut. Sebaliknya pencarian biner digunakan pada data yang sudah dalam keadaan urut.

II.4.1.1 Pencarian Berurutan (*Sequential Searching*)

Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut : data yang ada dibandingkan satu persatu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan. Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali pula.

II.4.1.2 Pencarian Biner (*Binary Searching*)

Salah satu syarat agar pencarian biner dapat dilakukan adalah data sudah dalam keadaan urut. Dengan kata lain, apabila data belum dalam keadaan urut, pencarian biner tidak dapat dilakukan. Dalam kehidupan sehari-hari juga, sebenarnya sering menggunakan pencarian biner. Misalnya saat ingin mencari suatu kata dalam kamus.

Prinsip dari pencarian biner dapat dijelaskan sebagai berikut : mula-mula diambil posisi awal 0 dan posisi akhir = N-1, kemudian dicari posisi data tengah dengan rumus (posisi awal + posisi akhir) / 2. Kemudian dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1. Jika lebih besar, proses dilakukan kembali tetapi posisi awal di anggap dengan posisi tengah +1. Demikian seterusnya sampai data tengah sama dengan data yang di cari.

II.5 Aplikasi *Web*

Pada awalnya aplikasi *Web* dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*). 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 adalah *applet*. Aplikasi *Web* itu dapat dibagi menjadi *web* statis dan *web* dinamis. *Web* statis dibentuk dengan menggunakan HTML saja. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus-menerus untuk mengikuti setiap perubahan yang terjadi. Kelemahan ini diatasi dengan model *Web* dinamis. Dengan menggunakan pendekatan *web* dinamis, dimungkinkan untuk membentuk sistem informasi berbasis *web*. Dari sisi teknologi yang digunakan untuk membentuk *web* dinamis terdapat dua pengelompokan, yaitu teknologi pada sisi *client* dan teknologi pada sisi *server* **[6]**.

Teknologi *Web* pada sisi *client* diimplementasikan dengan mengirimkan kode perluasan HTML atau program tersendiri dan HTML ke *client*. *Client*-lah yang bertanggung jawab dalam melakukan proses terhadap seluruh kode yang diterima. Kelemahan pendekatan seperi ini adalah terdapat kemungkinan bahwa *browser* pada *client* tidak mendukung fitur kode perluasan HTML. Kelebihan teknologi pada sisi *client*, yaitu memungkinkan penampilan yang bersifat dinamis. Contoh teknologi pada sisi *client*, yaitu Kontrol *ActiveX*, *Java Applet*, dan Skrip sisi-*client*. Teknologi *web* pada sisi *server* memungkinkan pemrosesan kode di dalam *server* sehingga kode yang sampai pada pemakai berbeda dengan kode asli pada *server*. Contoh teknologi yang berjalan di *server*, yaitu CGI, ASP, JSP, PHP dan lain sebagainya. Keuntungan penggunaan teknologi pada sisi *server* adalah sebagai berikut:

1 Mengurangi lalu lintas jaringan dengan cara menghindari percakapan bolak-balik antara *client* dan *server*.
2 Mengurangi waktu pemuatan kode, mengingat *client* hanya mengambil kode HTML saja.
3 Mencegah masalah *browser* yang tidak kompatibel.
4 *Client* dapat berinteraksi dengan data yang ada pada *server*.
5 Mencegah *client* mengetahui rahasia kode (mengingat kode yang diberikan ke *client* berbeda dengan kode asli pada *server*).

II.6 Pengertian Data

Menurut Fathansyah menyebutkan bahwa :

"Data merupakan fakta atau bagian dari fakta yang mengandung arti, yang dihubungkan dengan kenyataan, gambar-gambar, kata-kata, angka-angka, huruf atau simbol-simbol menyatakan suatu ide objek kondisi atau situasi dan lain-lain." **[7]**.

II.6.1 Pengertian Informasi

Informasi merupakan kumpulan data yang telah diolah sehingga dapat dijadikan hasil yang dapat dipresentasikan. Menurut Jogiyanto H.M dalam bukunya Pengenalan Komputer berpendapat bahwa:

"Informasi yaitu hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian kejadian *(events)* yang nyata *(fact)* yang digunakan untuk pengambilan keputusan" **[8]**.

Berikut adalah kualitas dari suatu informasi menurut Jogiyanto H.M dalam bukunya Pengenalan Komputer **[8]** :

a. Akurat

   Berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bisa menyesatkan yang berarti juga informasi harus mencerminkan maksudnya dengan jelas.

b. Tepat waktu

   Berarti informasi yang datang pada penerima tidak boleh terlambat, karena informasi yang telah usang tidak akan mempunyai arti lagi.

c. Relevan Berarti informasi tersebut mempunyai manfaat untuk pemakainya dan relevansi informasi tiap-tiap orang berbeda-beda.

II.6.2 Pengertian Sistem

Pengertian dari sistem di sini merupakan pengertian yang diambil dari Jogiyanto H.M dalam bukunya Pengenalan Komputer mengatakan bahwa :

> "Sistem adalah kumpulan dari sub-sub sistem atau komponen-komponen yang saling berinteraksi, saling bekerjasama dalam satu kesatuan kerja untuk mencapai satu tujuan tertentu" **[8]**.

Sedangkan sistem informasi disebutkan :

> "Sistem informasi adalah suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik" **[8]**.

Menurut DeLone dan McLean (1992) kualitas sistem yaitu karakteristik dari informasi yang melekat mengenai sistem itu sendiri **[9]**. Kualitas sistem memerlukan indikator untuk dapat mengukur seberapa besar kualitas dari sistem informasi tersebut. Indikator kualitas sistem diwujudkan dalam seperangkat pertanyaan kualitas sistem yang dapat diukur melalui beberapa indikator meliputi:

1. Kemudahan untuk digunakan (*ease of use*)

   Sistem informasi yang dapat dikatakan sebagai sistem yang berkualitas jika dirancang untuk memenuhi kemudahan dalam penggunaan sistem informasi tersebut. Perhatian dapat diukur berdasarkan pengguna dalam menggunakan sistem informasi tersebut yang hanya memerlukan sedikit waktu untuk mempelajari sistem informasi, hal ini dikarenakan sistem informasi tersebut sederhana, mudah dipahami, dan mudah pengoperasiannya.

2. Kecepatan akses (*response time*)

   Kecepatan akses merupakan salah satu indikator kualitas sistem informasi. Jika sistem informasi memiliki kecepatan akses yang optimal maka layak untuk dikatakan bahwa sistem informasi yang diterapkan memiliki kualitas yang baik. Kecepatan akses akan meningkatkan kepuasan pengguna dalam menggunakan sistem informasi. *Response time* juga dapat dilihat dari kecepatan pengguna dalam menelusur akan informasi yang dibutuhkan.

3. Keandalan sistem (*reliability*)

   Keandalan sistem informasi adalah ketahanan sistem informasi dari kerusakan dan kesalahan. Keandalan sistem informasi ini juga dapat dilihat dari sistem informasi dalam melayani kebutuhan pengguna tanpa adanya masalah yang dapat mengganggu kenyamanan pengguna dalam menggunakan sisten tersebut.

4. Fleksibilitas sistem (*flexibility*)

   Fleksibilitas yang dimaksud adalah kemampuan sistem informasi dalam melakukan perubahan-perubahan yang terkait dengan memenuhi kebutuhan pengguna. Pengguna akan merasa lebih puas menggunakan sistem informasi jika sistem tersebut fleksibel dalam memenuhi kebutuhan pengguna.

5. Keamanan sistem (*security*)

   Keamanan sistem dapat dilihat melalui program yang tidak dapat diubah-ubah oleh pengguna yang tidak bertanggung jawab dan juga program tidak dapat terhapus jika terdapat kesalahan dari pengguna.

II.6.3 *Analysis Modeling*

Model Analisis adalah teknik pertama untuk merepresentasikan sistem. Pemodelan analisis menggunakan kombinasi dari *text* dan diagram untuk merepresentasikan kebutuhan P/L (data, fungsi, dan tingkah). Ada dua tipe pemodelan analisis yang digunakan yaitu diantaranya analisis terstruktur dan analisis berorientasi obyek.

II.6.3.1 *Metode Sequensial Linear* (*Waterfall*)

Metode model *waterfall* mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan **[1]**. Tahap-tahap yang dilalui pada model *waterfall* adalah sebagai berikut :

a. Mendefinisikan kebutuhan (*Requirements definition*)
   Pada tahap ini pengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

b. Desain sistem dan perangkat lunak (*System and software design*)
   Pada tahap ini desain dikerjakan setelah kebutuhan selesai didefinisikan secara lengkap. Desain sistem merupakan perancangan sistem yang dilakukan berdasarkan data-data yang telah dikumpulkan pada tahap sebelumnya.

c. Implementasi dan pengujian perangkat lunak (*Implementaion and unit testing*)
   Implementasi perangkat lunak dilakukan dengan cara mengimplementasikan hasil dari perancangan perangkat lunak ke dalam kode program yang dimengerti oleh bahasa mesin. Pengujian perangkat lunak yaitu memfokuskan pada logika internal dari perangkat lunak, fungsi eksternal, dan mencari segala kemungkinan kesalahan, memeriksa apakah *input* sesuai dengan hasil yang diinginkan setelah proses.

d. Pengujian integrasi dan sistem (*Integration and system testing*)
   Pengujian perangkat lunak yaitu memfokuskan pada logika internal dari perangkat lunak, fungsi eksternal, dan mencari segala kemungkinan kesalahan, memeriksa apakah *input* sesuai dengan hasil yang diinginkan setelah proses.

Pengujian sistem merupakan penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).

e. Operasi dan Pemeliharaan (*Operation and Maintenance*)
   Pemeliharaan yaitu penerapan secara keseluruhan disertai pemeliharaan jika terjadi perubahan struktur baik dari segi *software* maupun *hardware*.

II.6.3.2 Diagram Alir Data (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem. DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

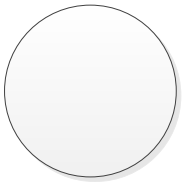
DFD merupkan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut **[10]** :

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2. Penurunan dilakukan apabila memang diperlukan.
3. Tidak semua bagian dari sistem harus ditunjukan dengan jumlah level yang sama.

Simbol-simbol yang digunakan dalam *Data Flow Diagram* adalah sebagai berikut :

1. Proses

Proses adalah simbol pertama data *flow* diagram **[10]**. Proses dilambangkan dengan lingkaran, dimana proses ini menunjukan bagian dari sistem yang mengubah satu atau lebih *input* dan *output*. Nama proses dituliskan dengan satu kata, singkatan atau kalimat sederhana. Lambang sebuah proses dapat dilihat pada gambar II.3 berikut :



Gambar II.3 Simbol Dari Sebuah Proses

2. Aliran Data

Aliran Data digambarkan dengan tandah panah **[10]**. Aliran data juga digunakan untuk menunjukan bagian-bagian informasi dari suatu bagian ke bagian lain. Pembagian nama untuk aliran ini menunjukan sebuah arti untuk sebuah aliran. Untuk kebanyakan sistem yang dibuat, aliran data sebenarnya menggambarkan data yakni angka, huruf, pesan, *floating point*, dan macam-macam informasi lainnya.



Gambar II.4 Simbol Dari Aliran Data

3. Simpanan Data

Simpanan data digunakan sebagai penyimpanan bagi paket-paket data **[10]**. Notasi penyimpanan data digambarkan dengan garis horizontal yang pararel. Simpanan data merupakan simpanan data dari data yang berupa satu *file* atau *database* di sistem komputer ataupun berupa arsip atau catatan manual. Nama dari simpanan data menunjukan nama *file*.

Gambar II.5 Simbol Dari Simpanan Data

4. Terminator

Terminator digambarkan dengan sebuah kotak yang menggambarkan kesatuan luar (*eksternal entity*) yang berhubungan dengan sistem **[10]**. Kesatuan luar merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang. Organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau *output* dari sistem.



Gambar II.6 Simbol Dari Sebuah Terminator

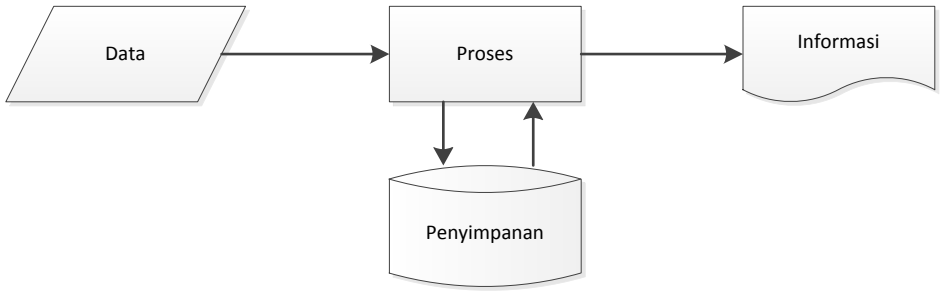
II.6.3.3 Diagram Konteks (DFD Level 0)

Diagram konteks merupakan langkah awal dalam perancangan sistem terstruktur yang merupakan gambaran sistem secara garis besar (umum), dengan menggambarkan aliran-aliran data ke dalam dan ke luar sistem. Diagram konteks merupakan pola pengambaran yang berfungsi untuk memperlihatkan interaksi sistem informasi tersebut dengan lingkungan di mana sistem tersebut ditempatkan. Dalam pengambaran ini, sistem dianggap sebuah objek yang tidak dijelaskan secara rinci karena yang ditekankan adalah interaksi sistem dengan lingkungan yang akan mengaksesnya.

II.6.3.4 Pengolahan Data

Pengolahan data merupakan segala macam pengolahan terhadap data atau kombinasi dari berbagai macam pengolahan data untuk menghasilkan sebuah informasi yang diharapkan. Pengolahan data jika ditinjau dari aplikasinya merupakan satu kesatuan yang saling berhubungan satu sama lain yang terdiri dari perangkat

keras (*hardware*), perangkat lunak (*software*), pemakai atau tenaga pelaksana (*brainware*). Data merupakan bagian yang sangat penting dalam suatu pengolahan data untuk dijadikan sebuah informasi yang berguna. Sebelum data diolah menjadi sebuah informasi, maka harus melalui tahapan-tahapan sebagai berikut :



Gambar II.7 Aliran Pemrosesan Data menjadi Informasi

II.6.4 Basis Data *(Database)*

Pengertian Basis Data menurut Fathansyah adalah Sekumpulan data *persistence* yang saling terkait, menggambarkan suatu organisasi (*Enterprise*) [**7**]. Basis data digunakan karena memiliki keuntungan sebagai berikut:

1. Mengurangi redundansi.
2. Data dapat di-*share* antar aplikasi.
3. Dapat dilakukan standardisasi data.
4. Batasan *security* dapat diterapkan.
5. Mengelola integritas data (akurasinya terjamin).
6. Independensi data (objektif DBS), basis data dapat berkembang tanpa mempengaruhi aplikasi yang telah ada.

Basis Data merupakan komponen terpenting dalam pembangunan Sistem Informasi, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat diekplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Basis Data merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak

terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau diekplorasi secara cepat dan mudah untuk menghasilkan informasi.

Sistem Basis Data terus dikembangkan oleh para ahli agar dapat diperoleh cara pengorganisasian data yang efesien dan efektif. Hal ini diperlukan karena sekarang ini berbagai bidang usaha telah menjadikan sebagai tumpuan manajemen informasi perusahaan. Adapun penerapan sistem basis data ini antara lain untuk pembangunan sistem informasi, penyediaan barang, kepegawaian, akutansi, pemasaran, produksi, reservasi, lanyanan pelanggan yang digunakan dalam perusahaan retail, perbankan, perhotelan dan pariwisata, rumah sakit, institusi pendidikan dan lain sebagainya.

II.6.5 *Database Management System* (DBMS)

Definisi DBMS pada sejumlah literartur sangatlah bervariasi. Secara umum, DBMS diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi, dan memperoleh data atau informasi dengan praktis dan efisien. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam *field* dan menulis kode aplikasi yang spesifik untuk pengaturannya. Diperlukan suatu sistem untuk diintegrasikan data *file* kedalam suatu *file* sehingga bisa melayani berbagai *user* yang berbeda. Perangkat keras dan lunak serta prosedur yang mengelola *database* merupakan suatu *database* manajemen sistem. DBMS memungkinkan untuk membentuk dan meremajakan *file-file,* memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporan-laporan.

Fungsi yang penting dari DBMS adalah sebagai berikut :

1. Menyediakan sistem *access* cepat.
2. Mengurangi kerangkapan data atau *redudancy* data.
3. Memungkinkan adanya *updating* secara bersama.
4. Menyediakan sistem yang memungkinkan dilakukannya pengembangan terhadap *database*.

5. Memberikan perlindungan dari pihak pemakai yang tidak berhak.

Dibandingkan dengan sistem yang berbasis kertas, DBMS memiliki empat keunggulan :

1. Kepraktisan

   Sistem yang berbasis kertas akan menggunakan kertas yang sangat banyak untuk menyimpan informasi, sedangkan DBMS menggunakan media penyimpan sekunder yang berukuran kecil tetapi padat informasi.
2. Kecepatan

   Mesin dapat mengambil atau mengubah data jauh lebih cepat daripada manusia.
3. Mengurangi kejemuan

   Orang cenderung menjadi bosan kalau melakukan tindakan-tindakan berulang yang menggunakan tangan (misalnya harus mengganti suatu informasi).
4. Kekinian

   Informasi yang tersedia pada DBMS akan bersifat mutakhir dan akurat setiap saat.

II.6.6 ERD (*Entity Relationship Diagram*)

ERD adalah suatu pemodelan dari basis data relasional yang didasarkan atas persepsi di dalam dunia nyata, dunia ini senantiasa terdiri dari sekumpulan objek yang saling berhubungan antara satu dengan yang lainnya. Suatu objek disebut *entity* dan hubungan yang dimilikinya disebut *relationship* **[7]**. Suatu *entity* bersifat unik dan memiliki atribut sebagai pembeda dengan *entity* lainnya. Contoh: *entity* siswa, mempunyai atribut nis (nomor induk siswa) yang bersifat unik, nama, alamat dan atribut yang lainnya. Fungsi dari ERD adalah untuk menggambarkan relasi dari dua *file* atau dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi yaitu satu ke satu, satu ke banyak dan banyak ke banyak. Elemen-elemen *Entity Relationship Diagram* adalah sebagai berikut **[7]** :

1. *Entity*

Pada *E-R* diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan. Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama, yaitu: orang, benda, lokasi kejadian (terdapat unsur waktu didalamnya).



Gambar II.8 *Entity*

2. *Relationship*

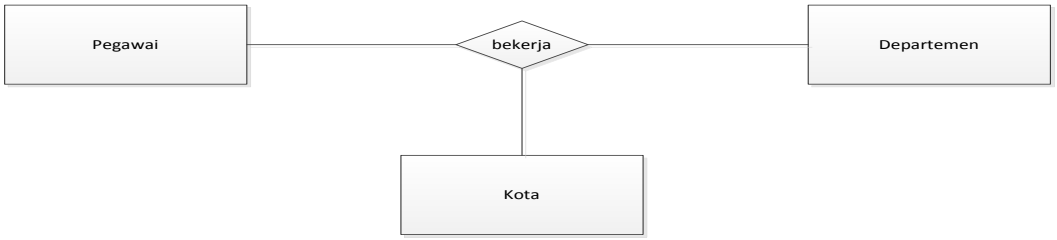
Pada *E-R diagram*, *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau dengan kalimat pasif). Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.



Gambar II.9 *Relationship*

3. *Relationship Degree*

Relationship degree atau derajat relasi adalah jumlah entitas yang berpartisipasi dalam suatu *relationship*.



Gambar II.10 *Relational Degree*

4. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *relationship*. Maksudnya, atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *relationship*, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan *relationship*.



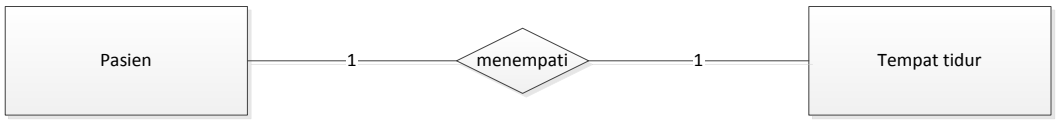
Gambar II.11 Atribut

5. Kardinalitas

Kardinalitas relasi menunjukan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antar entitas, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu :

a. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.



Gambar II.12 Kardinalitas *One to One*

b. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua . Sebaliknya satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.



Gambar II.13 Kardinalitas *One to Many*

c. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.



Gambar II.14 Kardinalitas *Many to Many*

II.7 Perangkat Lunak Pendukung

Berikut ini adalah beberapa perangkat lunak pendukung dalam pengerjaan aplikasi *e-learning* adalah sebagai berikut :

II.7.1 HTML (*Hypertext Markup Languange*)

HTML (*Hypertext Markup Languange*) adalah suatu bahasa yang digunakan untuk menulis halaman *web*. HTML dirancang untuk digunakan tanpa tergantung pada suatu *platform* tertentu (*platform independent*). Dokumen HTML adalah suatu dokumen teks biasa, dan disebut sebagai *markup language* karena mengandung tanda-tanda (*tag*) tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu SGMK (*Standart Generalized Markup Languange*). Sejak awal perkembangan sampai sekarang ini telah tersedia bermacam-macam level (versi) HTML, ada HTML level 1.0, HTML 2.0, HTML 3.0 dan HTML 4.0 **[6]**.

II.7.2 CSS (*Cascading Style Sheet*)

CSS (*Cascading style sheet*) adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa *markup* lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). CSS memungkinkan halaman yang sama untuk menampilkan cara yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh *browser* basis suara atau pembaca layar), dan juga alat pembaca *braille*, halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS **[6]**.

II.7.3 PHP & MySQL

PHP merupakan bahasa pemrograman untuk *script web server-side*. Bahasa pemrograman PHP diciptakan pertama kali oleh Rasmus Lerdorf, seorang pemrogram C yang sangat handal. Semula PHP hanya digunakan untuk mencatat seberapa jumlah pengunjung pada *hompage*-nya. Rasmus adalah salah seorang pendukung *open source*. Karen itulah ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis atau *freeware* pada tahun 1995. Setelah mempelajari YACC dan GNU Bison, Rasmus menambah kemampuan pada PHP 1.0 dan menertbitkan PHP 2.0 sebagai pengembangan dari PHP 1.0 yang telah ada sebelumnya.

PHP 2.0 mampu berhubungan dengan *database* dan dapat diintegrasikan dengan HTML. Pada tahun 1996, PHP telah digunkana oleh banyak *website* di dunia. Sebuah kelompok pengembang *software* yang terdiri dari Rasmus, Zeew Suraski, Andi Gutman, Stig Bakken, Shane Carveo dan Jim Winstead bekerja selama tujuh bulan untuk melakukan penyempurnan terhadap PHP 2.0, akhirnya pada tanggal 6 Juni 1998, PHP 3.0 resmi dikeluarkan ke dunia pemrograman. Pada pertengahan tahun 1999, Zend merilis interpreter PHP baru dan rilis tersebut dikenal dengan PHP 4.0. PHP 4.0 adalah versi PHP yang paling banyak dipakai. Versi ini banyak dipakai sebab versi ini mampu dipakai untuk membangun aplikasi *web* kompleks tetapi tetap

memiliki kecepatan proses dan stabilitas yang tinggi. PHP versi 4.2 telah diterbitkan pada tanggal 22 April 2002 dengan *log* kelompok fungsi, sampai dengan versi 4.3.7 tercatat 125 kelompok fungsi yang dimiliki oleh PHP. Pada Juni 2004, Zend merilis PHP 5.0. Versi ini adalah versi yang paling mutakhir dari PHP **[11]**.

PHP memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa-bahasa sejenisnya, yaitu:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. *Web Server* yang mendukung PHP dapat ditemukan dimana-mana dari mulai IIS sampai dengan apache, dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (linux, unix, windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem.

   Berikut adalah contoh *script* php :

```
<html>
<body>
      <?php
            echo "Hello World";
      ?>
</body>
      </html>
```

MySQL (*My Structure Query Language*) adalah sebuah program pembuat *database* yang bersifat *open source*. MySQL sebenarnya produk yang berjalan pada *platform* Linux. Karena sifatnya *open source*, maka MySQL dapat dijalankan pada semua *platform* baik Windows maupun Linux. Selain itu, MySQL juga merupakan program pengakses *database* yang bersifat jaringan sehingga dapat digunakan untuk

aplikasi *Multi User* (Banyak Pengguna). Saat ini database MySQL telah digunakan hampir oleh semua programer *database*, apalagi dalam pemrograman *web* **[6]**.

Kelebihan lain dari MySQL adalah menggunakan bahasa *Query* standar yang dimiliki SQL (*Structure Query Language*). SQL adalah suatu bahasa permintaan yang terstruktur yang telah distandarkan untuk semua program pengakses *database* seperti Oracle, Posgres SQL, SQL *Server*, dan lain-lain. Sebagai sebuah program penghasil *database*, MySQL tidak dapat berjalan sendiri tanpa adanya sebuah aplikasi lain (*interface*). MySQL dapat didukung oleh hampir semua program aplikasi baik yang *open source* seperti PHP maupun yang tidak, yang ada pada *platform* Windows seperti Visual Basic, Delphi, dan lainnya.

Berikut adalah contoh perintah-perintah dalam MySQL :

Contoh perintah *create table*;

```
CREATE TABLE user
(
       username VARCHAR(30) CONSTRAINT PRIMARY KEY,
       password VARCHAR(20) NOT NULL,
       tanggal_lahir DATETIME
     );
```

Contoh perintah *select*;

```
SELECT * FROM user
```

Contoh perintah *update*;

```
UPDATE Msuser set password="123456" where username="abc"
```

Contoh perintah *insert*;

```
INSERT INTO TEST (NAMA, ALAMAT, PASSWORD)
        VALUES ('test', 'alamat', 'pass');
```

Contoh perintah *delete*;

```
DELETE FROM TEST WHERE NAMA='test';
```

II.7.4 Macromedia Dreamweaver 8

Macromedia Deramweaver 8 adalah HTML editor professional yang berfungsi mendesain, melakukan *coding*, dan mengembangkan *website* yeng paling terkenal di dunia *web*. Macromedia adalah produsen *software website* yang paling

besar di dunia. Selain Dreamweaver, Macromedia juga mengeluarkan produk seperti Macromedia Flash, Fireworks, Coldfusion, Freehand, Director dan sebagainya **[12]**.

Salah satu kelebihan Dreamweaver adalah ruang kerja Dreamweaver beserta *tool* yang tersedia dapat digunakan dengan sangat mudah dan cepat, sehingga Anda bisa membangun suatu *website* dengan cepat dan tanpa harus melakukan *coding*. Selain itu, Dreamweaver juga mempunyai integrasi dengan produk Macromedia lainnya, seperti Flash, Fireworks, Flash sudah sangat terkenal sebagai program untuk membuat animasi yang berbasis *web*. Dengan perkembangan kebutuhan dan teknologi, Flash akhir-akhir juga digunakan untuk membuat animasi untuk video.

Fungsi *coding* Dreamweaver tidak hanya mendukung coding HTML, tetapi juga CSS (*Cascading Style Sheet*), Javascript, Coldfusion, ASP (*Active Server Pages*), JSP (*Java Server Pages*), dan Dreamweaver juga memungkinkan Anda membangun *website* dengan *server* berbahasa CFML (*ColdFusion Markup Language*), ASP.net, ASP, JSP dan PHP.

II.7.5 Wamp *Server* 2.0

Wamp *Server* atau wamp adalah sebuah aplikasi yang dapat menjadikan komputer pemakai menjadi sebuah *server*. Kegunaan wamp ini untuk membuat jaringan lokal sendiri dalam artian pemakai dapat membuat *website* secara *offline* untuk masa coba-coba di komputer sendiri. Kelebihan wamp, yaitu praktis dan tidak perlu banyak melakukan *setting*. Saat ini wamp *server* hanya berjalan pada sistem operasi windows.

II.7.6 *Browser* & *Webserver*

Dalam dunia *web*, perangkat lunak *client*, yaitu *browser web* mempunyai tugas yang sama yaitu menterjemahkan informasi yang diterima oleh *server web* dan menampilkannya pada layer komputer pengguna, oleh karena HTTP memungkinkan *server web* mengirimkan beragam data, seperti teks atau gambar, *browser* harus bisa mengenali berbagai macam data yang akan diterimanya, dan selanjutnya harus tahu

cara untuk menampilkanya dengan benar. Teks ditampilkan sebagai teks dan gambar ditampilkan sebagai gambar. Umumnya *browser web* menerima data dalam bentuk HTML. *File* HTML sebenarnya adalah *file* teks biasa yang selain berisi informasi yang hendak ditampilkan kepada pengguna, juga mempunyai perintah-perintah untuk mengatur tampilan data tersebut. *Browser*-lah yang memiliki kuasa penuh dalam menterjemahkan perintah-perintah tadi. Meskipun sudah dibuat *consensus* untuk menstandarkan format dan elemen-elemen HTML, setiap jenis *browser* bisa menterjemahkan *file* HTML secara berbeda.

Pada awalnya pertama kalinya protokol–protokol dasar *web* dikembangkan yaitu sekitar awal tahun 1990-an, *browser web* pertama yang diperkenalkan adalah mosaic yang dibuat oleh *National Center for Supercomputing Application* (NCSA) di Amerika Serikat. Mosaic dimaksudkan agar menjadi sebuah *interface* grafis yang mudah digunakan, yang demikian diharapkan dapat mempercepat perkembangan dan dukungan umum akan *web*.

Beberapa *server web* memiliki fitur seperti *server side programming*, *security control* dan lain sebagainya. Meskipun beragam macamnya, secara fungsional semua jenis *server web* adalah sama saja, yaitu berfungsi melayani permintaan-permintaan dari *browser web*.

II.8 Skala Pengukuran Sistem

Metode skala pengukuran merupakan kesepakatan yang digunakan sebagai acuan untuk menentukan panjang pendeknya interval yang ada dalam alat ukur, sehingga alat ukur tersebut bila digunakan dalam pengukuran akan menghasilkan data kuantitatif. Metode penghitungan yang digunakan untuk mengukur sikap atau pendapat atau persepsi dari pengisi kuesioner akan dihitung menggunakan metode likert seperti berikut **[10]**.

II.8.1 Skala *Likert*

Skala *Likert* digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dalam penelitian, fenomena sosial ini telah ditetapkan secara spesifik oleh peneliti, yang selanjutnya disebut sebagai variabel penelitian **[13]**.

Dengan skala Likert, maka variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun *item-item* instrumen yang dapat berupa pernyataan atau pertanyaan.

Pilihan jawaban pada kuesioner akan dituliskan dengan menggunakan skala *likert*, yang dimana jawaban dari setiap instrumen memiliki gradasi dari sangat positif sampai sangat negatif. Berikut ini merupakan nilai skala *likert* yang akan digunakan :

1. Skor 5 untuk Sangat Setuju
2. Skor 4 untuk Setuju
3. Skor 3 untuk Cukup Setuju
4. Skor 2 untuk Kurang Setuju
5. Skor 1 untuk Tidak Setuju

II.8.2 Skala Kontinum

Sedangkan untuk menghitung data kuantitatif yang didapatkan dari skala *likert* tersebut, hasil kuesioner akan digambarkan dengan menggunakan metode kontinum dengan rumus :

$$Y=P/Q*100\% \qquad \text{(II.1)}$$

Keterangan : Y : Nilai Persentase
P : Total Skor
Q : Skor Tertinggi

Berikut ini adalah contoh penerapan rumus dari hasil jawaban yang didapat dari 100 responden dengan hasil jawaban sebagai berikut :

25 orang menjawab Sangat Setuju

40 orang menjawab Setuju

5 orang menjawab Cukup Setuju

20 orang menjawab Kurang Setuju

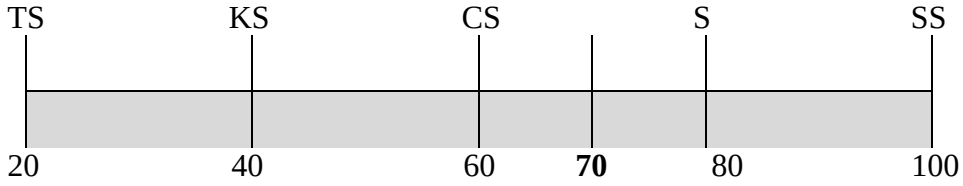
10 orang menjawab Tidak Setuju

Berdasarkan skor yang telah ditetapkan dapat dihitung sebagai berikut :

Jumlah skor untuk 25 orang yang menjawab Sangat Setuju	= 25 * 5	= 125
Jumlah skor untuk 40 orang yang menjawab Setuju	= 40 * 4	= 160
Jumlah skor untuk 5 orang yang menjawab Cukup Setuju	= 5 * 3	= 15
Jumlah skor untuk 20 orang yang menjawab Kurang Setuju	= 20 * 2	= 20
Jumlah skor untuk 10 orang yang menjawab Tidak Setuju	= 10 * 1	= 10
Total		= 350

Skor tertinggi untuk seluruh jawaban = 5 * 100 = 500 (jika seluruh responden menjawab Sangat Setuju). Total skor yang diperoleh = 350. Berdasarkan data yang dihitung maka diperoleh hasil (350/500) * 100% = 70.

Untuk mengetahui interpretasi skor hasil perhitungan dapat dilihat pada gambar berikut :



Gambar II.15 Interpretasi hasil perhitungan

Berdasarkan data yang diperoleh dari 100 responden maka nilai persentase yang didapat adalah 70 dan terletak pada daerah setuju.

II.9 Pengujian *Black-Box*

Menurut Roger S Pressman dalam bukunya Rekayasa Perangkat Lunak :

"Pengujian *Black-Box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian *Black-Box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program" **[1]**.

Pengujian *Black-Box* bukan merupakan alternatif dari teknik *White-Box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*. Pengujian *Black–Box* berusaha menemukan kesalahan dalam kategori sebagai berikut; (1) Fungsi-fungsi yang tidak benar atau hilang (2) Kesalahan *Interface* (3) Kesalahan dalam struktur data atau akses *database* eksternal (4) Kesalahan kinerja (5) Inisialisasi dan kesalahan terminasi. Pengujian *Black-box* cenderung diaplikasikan selama tahap akhir pengujian, karena pengujian *black-box* memperhatikan struktur kontrol maka perhatian berfokus pada *domain* informasi.