

BAB III

LANDASAN TEORI

3.1 Definisi Pelatihan

Menurut Mathis (2002), pelatihan adalah suatu proses dimana orang-orang mencapai kemampuan tertentu untuk membantu mencapai tujuan organisasi. Oleh karena itu, proses ini terikat dengan berbagai tujuan organisasi, pelatihan dapat dipandang secara sempit maupun luas. Secara terbatas, pelatihan menyediakan para pegawai dengan pengetahuan yang spesifik dan dapat diketahui serta keterampilan yang digunakan dalam pekerjaan mereka saat ini. Terkadang ada batasan yang ditarik antara pelatihan dengan pengembangan, dengan pengembangan yang bersifat lebih luas dalam cakupan serta memfokuskan pada individu untuk mencapai kemampuan baru yang berguna baik bagi pekerjaannya saat ini maupun di masa mendatang.

Menurut Dessler (2009), pelatihan adalah proses mengajarkan karyawan baru atau yang ada sekarang, ketrampilan dasar yang mereka butuhkan untuk menjalankan pekerjaan mereka. Pelatihan merupakan salah satu usaha dalam meningkatkan mutu sumber daya manusia dalam dunia kerja. Karyawan, baik yang baru ataupun yang sudah bekerja perlu mengikuti pelatihan karena adanya tuntutan pekerjaan yang dapat berubah akibat perubahan lingkungan kerja, strategi, dan lain sebagainya.

Menurut Pasal 1 ayat (9) Undang-undang No.13 Tahun 2003 Tentang Ketenagakerjaan, pelatihan kerja adalah keseluruhan kegiatan untuk memberi, memperoleh, meningkatkan, mengembangkan kompetensi kerja, produktivitas,

disiplin, sikap dan etos kerja pada tingkat keterampilan tertentu sesuai dengan jenjang dan kualifikasi jabatan dan pekerjaan.

3.2 Tujuan Pelatihan

Tujuan umum dari pelatihan adalah sebagai berikut :

1. Untuk mengembangkan keahlian, sehingga pekerjaan dapat diselesaikan dengan lebih cepat dan lebih efektif.
2. Untuk mengembangkan pengetahuan, sehingga pekerjaan dapat diselesaikan secara rasional.
3. Untuk mengembangkan sikap, sehingga menimbulkan kemauan kerjasama dengan teman-teman pegawai dan dengan manajemen (pimpinan).

Sedangkan komponen-komponen pelatihan sebagaimana dijelaskan oleh Mangkunegara (2005), terdiri dari :

1. Tujuan dan sasaran pelatihan dan pengembangan harus jelas dan dapat diukur.
2. Para pelatih harus ahlinya yang berkualitas.
3. Materi pelatihan dan pengembangan harus disesuaikan dengan tujuan yang hendak dicapai.
4. Peserta pelatihan dan pengembangan (*trainers*) harus memenuhi persyaratan yang ditentukan.

Dalam pengembangan program pelatihan, agar pelatihan dapat bermanfaat dan mendatangkan keuntungan diperlukan tahapan atau langkah-langkah yang

sistematik. Secara umum ada tiga tahap pada pelatihan yaitu tahap penilaian kebutuhan, tahap pelaksanaan pelatihan dan tahap evaluasi atau dengan istilah lain ada fase perencanaan pelatihan, fase pelaksanaan pelatihan dan fase pasca pelatihan.

Dijelaskan oleh Mangkunegara (2005), bahwa tahapan-tahapan dalam pelatihan dan pengembangan meliputi :

1. Mengidentifikasi kebutuhan pelatihan / *need assesment*.
2. Menetapkan tujuan dan sasaran pelatihan.
3. Menetapkan kriteria keberhasilan dengan alat ukurnya.
4. Menetapkan metode pelatihan.
5. Mengadakan percobaan (*try out*) dan pembenahan.
6. Mengimplementasikan dan mengevaluasi.

3.3 Analisa dan Perancangan Sistem

Analisa sistem dilakukan dengan tujuan untuk mengidentifikasi dan mengevaluasi permasalahan yang terjadi dan kebutuhan yang diharapkan, sehingga dapat diusulkan perbaikannya. Perancangan sistem merupakan penguraian suatu sistem informasi yang utuh ke dalam bagian komputerisasi yang dimaksud, mengidentifikasi dan mengevaluasi permasalahan, menentukan kriteria, menghitung konsistensi terhadap kriteria yang ada, serta mendapatkan hasil atau tujuan dari masalah tersebut serta mengimplementasikan seluruh kebutuhan operasional dalam membangun sebuah aplikasi.

Menurut Kendall (2003), analisa dan perancangan sistem dipergunakan untuk menganalisis, merancang dan mengimplementasikan peningkatan-peningkatan fungsi bisnis yang dapat dicapai melalui penggunaan sistem informasi terkomputerisasi.

Tahap analisis merupakan tahap yang kritis dan sangat penting, karena kesalahan di dalam tahap ini juga akan menyebabkan kesalahan di tahap selanjutnya. Dalam tahap analisis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sistem sebagai berikut:

1. *Identify*, yaitu mengidentifikasi masalah.
2. *Understand*, yaitu memahami kerja dari sistem yang ada.
3. *Analyze*, yaitu menganalisis sistem.
4. *Report*, yaitu membuat laporan hasil analisis.

Setelah tahap analisis sistem selesai dilakukan, maka analis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analis sistem untuk memikirkan bagaimana membentuk sistem tersebut. tahap ini disebut desain sistem.

3.4 Konsep Dasar Sistem Informasi

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya. Data

merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lanjut. (Jogiyanto, 2001)

Untuk memahami apa yang dimaksud dengan sistem informasi, kita perlu mendefinisikan istilah informasi dan sistem. Produk dari sistem informasi adalah informasi yang dihasilkan. Informasi tidak sama dengan data. Data adalah fakta, angka bahkan simbol mentah. Secara bersama-sama mereka merupakan masukan bagi suatu sistem informasi. Sebaliknya, informasi terdiri dari data yang telah ditransformasi dan dibuat lebih bernilai melalui suatu pemrosesan. Idealnya, informasi adalah pengetahuan yang berarti dan berguna untuk mencapai sasaran.

Sistem adalah suatu kerangka kerja terpadu yang mempunyai satu sasaran atau lebih. Sistem ini mengkoordinasi sumber daya yang dibutuhkan untuk mengubah masukan-masukan menjadi keluaran. Sumber daya dapat berupa bahan (material) atau mesin ataupun tenaga kerja, bergantung pada macam sistem yang dibahas. Sistem informasi karenanya adalah suatu kerangka kerja dengan mana sumber daya (manusia dan komputer) dikoordinasikan untuk mengubah masukan (data) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan.





Definisi lain dari sistem informasi adalah sekumpulan *hardware*, *software*, *brainware*, prosedur dan atau aturan yang diorganisasikan secara integral untuk mengolah data menjadi informasi yang bermanfaat guna memecahkan masalah dan pengambilan keputusan. Sistem informasi adalah satu kesatuan data olahan yang terintegrasi dan saling melengkapi yang menghasilkan *output* baik dalam bentuk gambar, suara maupun tulisan.



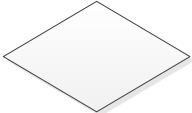

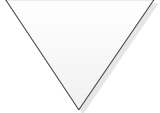
Sistem informasi adalah sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

3.5 Bagan Alir Dokumen

Bagan alir dokumen (*document flowchart*) atau di sebut juga bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukkan dokumen gambaran arus data dengan menggunakan simbol seperti pada tabel 3.1 berikut:

Tabel 3.1 Simbol *Flowchart*

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
1.		Dokumen	Untuk menunjukkan dokumen <i>input</i> dan <i>output</i> baik untuk proses manual, mekanik atau komputer.
2.		Proses Komputerisasi	Menunjukkan kegiatan dari operasi program komputer.
3.		Database	Untuk menyimpan data.
4.		Penghubung	Menunjukkan hubungan di halaman yang sama.

No.	Simbol	Nama Simbol <i>Flowchart</i>	Fungsi
5.		Penghubung Halaman Lain	Menunjukkan hubungan di halaman lain.
6.		Terminator	Menandakan awal/akhir dari suatu sistem.
7.		Decision	Logika keputusan dengan nilai <i>true</i> atau <i>false</i> .
8.		Kegiatan Manual	Untuk menunjukkan pekerjaan yang dilakukan secara manual.
9.		Simpanan <i>Offline</i>	Untuk menunjukkan file non-komputer yang diarsip urutan angka.

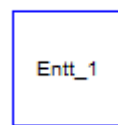
3.6 *Data Flow Diagram (DFD)*

Menurut Kendall (2003), *Data Flow Diagram* menggambarkan pandangan sejauh mungkin mengenai masukan, proses dan keluaran sistem, yang berhubungan dengan masukan, proses, dan keluaran dari model sistem yang dibahas. Serangkaian diagram aliran data berlapis juga bisa digunakan untuk merepresentasikan dan menganalisis prosedur-prosedur mendetail dalam sistem. Prosedur-prosedur tersebut yaitu konseptualisasi bagaimana data-data berpindah di dalam organisasi, proses-proses atau transformasi dimana data-data melalui, dan apa keluarannya. Jadi, melalui suatu teknik analisa data terstruktur yang disebut *Data Flow Diagram*, penganalisis sistem dapat merepresentasi proses-proses data

di dalam organisasi. Menurut Kendall (2003), dalam memetakan *Data Flow Diagram*, terdapat beberapa simbol yang digunakan antara lain:

1. *External entity*

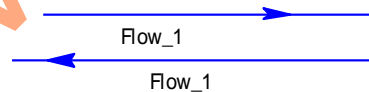
Suatu *external entity* atau entitas merupakan orang, kelompok, departemen, atau sistem lain di luar sistem yang dibuat dapat menerima atau memberikan informasi atau data ke dalam sistem yang dibuat.



Gambar 3.1 Simbol *External Entity*

2. *Data Flow*

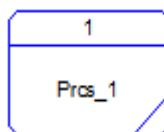
Data Flow atau aliran data disimbolkan dengan data tanda panah. Aliran data menunjukkan arus data atau aliran data yang menghubungkan dua proses atau entitas dengan proses.



Gambar 3.2 Simbol *Data Flow*

3. *Process*

Suatu proses dimana beberapa tindakan atau sekelompok tindakan dijalankan.



Gambar 3.3 Simbol *Process*

4. *Data Store*

Data store adalah simbol yang digunakan untuk melambangkan proses penyimpanan data.



Gambar 3.4 Simbol *Data Store*

3.7 *Entity Relationship Diagram (ERD)*

Entity relationship diagram (ERD) adalah gambaran pada sistem dimana di dalamnya terdapat hubungan antara *entity* beserta relasinya. *Entity* merupakan sesuatu yang ada dan terdefiniskan di dalam suatu organisasi, dapat abstrak dan nyata. Untuk setiap *entity* biasanya mempunyai *attribute* yang merupakan ciri *entity* tersebut. *Attribute* yaitu uraian dari entitas dimana mereka dihubungkan atau dapat dikatakan sebagai *identifier* atau *descriptors* dari entitas.

Entitas digolongkan menjadi *independent* atau *dependent entity*. *Independent entity* adalah apa yang tidak bersandar pada yang lain sebagai identifikasi. Suatu *dependent entity* adalah apa yang bersandar pada yang lain sebagai identifikasi. Selain digolongkan menjadi *independent* atau *dependent entity*, terdapat jenis-jenis entitas khusus yaitu:

1. *Associative Entity*

Associative Entity (juga dikenal sebagai *intersection entity*) adalah entitas yang digunakan oleh rekanan dua entitas atau lebih untuk menyatukan suatu hubungan banyak - ke - banyak (*Many to Many*)

2. *Subtypes Entity*

Subtypes Entity digunakan di dalam hierarki generalisasi (*generalization hierarchies*) untuk menyajikan suatu subset kejadian dari entitas orangtua, yang disebut *supertype*, tetapi yang memiliki atribut atau hubungan yang berlaku hanya untuk *subset*.

Menurut Marlinda (2004), *attribute* sebagai kolom pada sebuah relasi *database* mempunyai macam-macam jenis *attribute* yaitu :

a. *Key Attribute*

Attribute ini merupakan *attribute* yang unik dan tidak dimiliki oleh *attribute* lainnya, misalnya entity mahasiswa yang *attribute*-nya NIM.



Gambar 3.5 *Key Attribute*

b. *Particial key Attribute*

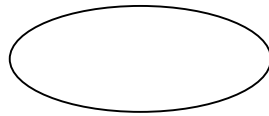
Adalah *Attribute* yang tidak menjadi atau merupakan anggota dari *Key Primer*. Misalnya antara Cabang (toko) dan kode cabang.



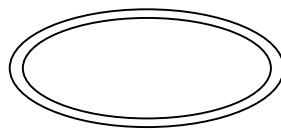
Gambar 3.6 *Particial Key Attribute*

c. *Single Vallue Attribute*

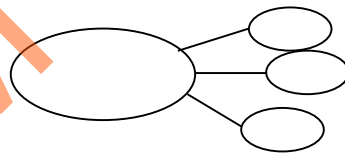
Attribute yang hanya memiliki satu nilai harga, misalnya *entity* mahasiswa dengan *attribute*-nya Umur (Tanggal lahir).

Gambar 3.7 *Single Value Attribute*d. *Multi Value Attribute*

Attribute yang banyak memiliki nilai harga, misalnya *entity* mahasiswa dengan *atribute*-nya pendidikan (SD, SMP, SMA).

Gambar 3.8 *Multi Value Attribute*e. *Composite Attribute*

Attribute yang memiliki dua harga, misalnya nama besar (nama kerja) dan nama kecil (nama asli).

Gambar 3.9 *Composite Attribute*f. *Derived Attribute*

Attribute yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari table *Attribute* atau table lain yang berhubungan.

Gambar 3.10 *Derived Attribute*

Model *Entity - Relationship* (ER) mula-mula diusulkan oleh Peter pada tahun 1976 sebagai cara untuk mempersatukan pandangan basis data jaringan dan relasional. Langkah sederhana dari model ER adalah model data konseptual yang memandang dunia nyata sebagai kesatuan (*entities*) dan hubungan (*relationship*).

Komponen dasar model merupakan diagram *entity-relationship* yang digunakan untuk menyajikan objek data secara *visual*. *Entity Relationship Diagram* mengilustrasikan struktur logis dari basis data yang mempunyai metodologi sebagai berikut:

Tabel 3.2 Ilustrasi pembuatan ERD

Proses	Keterangan
1. Menentukan Entitas	Menentukan peran, kejadian, lokasi, hal nyata, dan konsep dimana pengguna akan menyimpan data.
2. Menentukan Relasi	Tentukan hubungan antara pasangan entitas menggunakan matriks relasi.
3. Gambar ERD Sementara	Entitas digambarkan dengan kotak dan relasi dengan garis yang menghubungkan entitas.
4. Isi Kardinalitas	Tentukan jumlah kejadian dari satu entitas untuk sebuah kejadian pada entitas yang berhubungan.

5. Tentukan Kunci Utama	Tentukan atribut yang mengidentifikasi satu dan hanya satu kejadian pada masing-masing entitas.
6. Gambar ERD berdasar Kunci	Hilangkan relasi <i>Many-to-Many</i> dan masukkan <i>primary</i> dan kunci tamu pada masing-masing entitas.
7. Menentukan Atribut	Tuliskan <i>field-field</i> yang diperlukan oleh sistem.
8. Pemetaan Atribut	Pasangkan atribut dengan satu entitas yang sesuai pada masing-masing atribut.
9. Gambar ERD dengan Atribut	Aturlah ERD dari langkah 6 dengan menambahkan entitas atau relasi yang ditemukan pada langkah 8.
10. Periksa Hasil	Apakah ERD sudah menggambar sistem yang akan dibangun.

Entity Relationship Diagram ini diperlukan agar dapat menggambarkan hubungan antar *entity* dengan jelas, dapat menggambarkan batasan jumlah *entity* dan partisipasi antar *entity*, mudah dimengerti pemakai dan mudah disajikan oleh perancang *database*. Untuk itu, *entity relationship diagram* dibagi menjadi dua jenis model, yaitu:

1. *Conceptual Data model*

Conceptual Data model (CDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara konseptual.

2. *Physical Data Model*

Physical Data Model (PDM) adalah jenis model data yang menggambarkan hubungan antar tabel secara fisikal.

3.8 Program Penunjang

Untuk membuat rancang bangun aplikasi Pendataan Peserta Pelatihan, dibutuhkan beberapa perangkat lunak untuk memudahkan perancangan *design* maupun sistem. Perangkat lunak tersebut antara lain:

3.8.1 *Power Designer*

Power designer merupakan suatu *tool* berupa *software* untuk mendesain sistem dan rancangan *Entity Relationship Diagram* (ERD) yang dikembangkan oleh *Sybase Inc.* Ada dua model data, yaitu: *Entity Relationship Diagram* (ERD) dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

- a. Model ERD atau *Conceptual Data Model* : model yang di buat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas serta hubungan (*relationship*) antara entitas-entitas itu.
- b. Model Relasional atau *Physical Data Model* : model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data

tersebut. Setiap tabel mempunyai sejumlah kolom dimana setiap kolom memiliki nama yang unik.

3.8.2 XAMPP For Windows

Menurut Oktavian (2010), XAMPP merupakan sebuah tool yang menyediakan beberapa paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis. XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan project non-profit yang di kembangkan oleh Apache Friends yang didirikan Kai 'Oswalad' Seidler dan Kay Vogelgesang pada tahun 2002, project mereka ini bertujuan mempromosikan penggunaan Apache web server. XAMPP sendiri menawarkan beberapa paket pengistallan antara lain :

- a) Apache 2.2.6.
- b) MySQL 5.0.45.
- c) PHP 5.2.4 + PHP 4.4.7 + PEAR.
- d) PHP-Switch win32 1.028.
- e) XAMPP Control Version 3.2.1.
- f) XAMPP Security 1.0.
- g) SQLite 2.8.15.
- h) OpenSSL 0.9.8e.

- i) *phpMyAdmin 2.11.1.*
- j) *ADODB 4.95.*
- k) *Mercury Mail Transport System v4.01b.*
- l) *FileZilla FTP Server 0.9.23.*
- m) *Webalizer 2.01-10.*

3.8.3 Notepad++

Menurut Oktavian (2010), *Notepad++* adalah *Code Editor* (software penyunting kode) yang mendukung berbagai bahasa pemrograman seperti HTML, CSS, PHP, XML, *Java*, JSP, *JavaScript*, *Perl Pascal*, dan lain-lain yang dapat bekerja pada *System Operasi Windows*. Kelebihan *Notepad++* jika dibanding *Notepad* bawaan *Windows* adalah memiliki kelengkapan fitur untuk mempermudah pengguna saat mengedit kode termasuk saat mengedit kode HTML dan kode CSS.

Fitur-fitur *Notepad++* :

- a) *Tabbed document interface.*
- b) *Drag-and-drop.*
- c) *Multiple Clipboard (plugin required).*
- d) *Split screen editing and synchronized scrolling.*
- e) *Spell checker (requires Aspell) (Spell checker does not distinguish between text and code).*
- f) *Supports text encoding formats such as Unicode, for international writing systems.*

- g) *Find and replace over multiple documents.*
- h) *File comparison.*
- i) *Zooming.*
- j) *Syntax Highlighting* dan *syntax folding*, yang memudahkan pengguna dengan menampilkan warna yang berbeda dalam penulisan *syntax* agar lebih mudah dikenali.
- k) *Style Configurator*, yang memungkinkan *Software Notepad++* untuk mengatur warna *background*, huruf ukuran dan jenisnya.
- l) *Multi-Document*, yang memungkinkan *Software Notepad++* mengedit dua *file* secara bersamaan.
- m) *Multi-View*, yang memungkinkan *Software Notepad++* membuka banyak *file* dalam satu *window* dengan *tab* yang *user friendly*.
- n) *Search support*, mempermudah pencarian kata kunci pada *syntax* yang rumit dan banyak.

3.9 HTML5

Menurut Lee (2011), HTML5 dapat mengakhiri penggunaan *Flash* untuk banyak aplikasi media, itu yang menyebabkan *JavaScript* bahkan menjadi lebih populer dari sebelumnya. Ada banyak perpustakaan dan plugin yang tersedia untuk meningkatkan dan memperluas HTML5 dan CSS3 untuk menciptakan pengalaman interaktif yang kaya.

Menurut Lee (2011), untungnya HTML5 liberal membantu dari *Cascading Style Sheets*, dan sedikit *JavaScript*, bangkit untuk memenuhi tantangan-tantangan

baru. Inovasi terbaru dalam pengembangan website membuat sebuah zaman keemasan baru untuk penerbit online. Setelah semua, HTML5 merupakan evolusi bukan revolusi .

Menurut Lee (2011), pada saat pengembangan dengan HTML5, maka diambil prinsip dasar dari penamaan semantik (penamaan hal-hal apa yang mereka bukannya penamaan hal-hal bagaimana mereka muncul) ke level baru. Ini adalah faktor kunci yang membuat HTML5 berbeda dari semua pendahulunya. Menurut buku *HTML5 Multimedia Development Cookbook* akan ditemukan pemikiran kembali dan mengoptimalkan banyak kode-penamaan konvensi.

Meskipun HTML5 diusulkan rekomendasi dari *Hypertext Aplikasi Web Teknologi Working Group* (WHATWG) tidak dijadwalkan untuk implementasi penuh sampai tahun 2022, berkat berpikiran maju produsen *browser*, tidak ada alasan pengguna tidak dapat memulai menggunakannya sekarang dan menuai keuntungan dari semantik penamaan yang lebih baik, aksesibilitas yang disempurnakan, dan lebih banyak lagi.

3.10 *Cascading Style Sheets* (CSS)

Menurut Wismakarma (2010), *Cascading Style Sheets* (CSS) adalah salah satu bahasa pemrograman desain *website* (*style sheet language*) yang mengontrol *format* tampilan sebuah halaman *website* yang di tulis dengan menggunakan bahasa penanda (markup language). Biasanya CSS digunakan untuk mendesain sebuah halaman HTML dan XHTML, tetapi sekarang bahasa pemrograman css bisa diaplikasikan untuk segala dokumen XML, termasuk SVG, dan XUL.

CSS dibuat untuk memisahkan konten utama (biasanya dibuat dengan menggunakan bahasa HTML atau sejenisnya) dengan tampilan dokumen yang meliputi *layout*, warna dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada *website*, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dan sebuah tampilan, memungkinkan untuk membagi banyak halaman untuk sebuah *formatting* dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik *tableless* pada *layout* desain *website* (*layout* tanpa tabel).

CSS juga memungkinkan sebuah halaman untuk di tampilkan dalam berbagai style dengan menggunakan metode pembawaan yang berbeda pula, seperti *on screen*, *in-print*, *by voice*, dan lain-lain. Sementara itu, pemilik konten *website* bisa menentukan link yang menghubungkan konten dengan *file* CSS, pembaca bisa menggunakan dengan *style sheet* yang berbeda, mungkin dalam komputer mereka sendiri.

Tujuan utama CSS diciptakan adalah untuk membedakan konten dari dokumen dan tampilan dari dokumen, dengan itu pembuatan ataupun pemograman ulang web akan lebih mudah dilakukan. Hal yang termasuk dalam design web diantaranya adalah warna, ukuran dan *formatting*. Dengan adanya CSS, konten dan desain web akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan-tampilan tertentu dalam suatu halaman *web*, sehingga akan memudahkan dalam membuat halaman *web* yang banyak, sehingga membuat waktu dalam membuat *web* jauh lebih cepat.

3.11 *Hypertext Preprocessor (PHP)*

Mengacu pada pendapat Nixon (2009), PHP adalah *server-side scripting language* yang awalnya dirancang untuk pengembangan *web* untuk menghasilkan halaman *web* yang dinamis. Untuk tujuan ini, kode PHP tertanam ke dokumen sumber HTML dan diterjemahkan oleh sebuah *web server* dengan PHP prosesor modul, yang menghasilkan dokumen halaman *website*. PHP dapat digunakan di sebagian besar *web server* dan sebagai juru mandiri, pada hampir setiap sistem operasi dan *platform* secara gratis.

PHP adalah bahasa *scripting* umum yang terutama cocok untuk pengembangan web sisi server di mana PHP umumnya berjalan pada *web server*. Kode PHP di *file* yang diminta dilaksanakan oleh PHP *runtime*, biasanya untuk membuat halaman konten *website* yang dinamis atau membuat gambar dinamis yang digunakan di *situs website* atau di tempat lain. PHP Dapat pula digunakan untuk baris perintah *scripting* dan digunakan di Aplikasi GUI pada sisi klien. PHP dapat digunakan hampir di semua *web server*, di hampir semua sistem operasi dan *platform*, dan dapat digunakan dengan banyak sistem manajemen *database relational (RDBMS)*. PHP dapat diunduh secara gratis dan PHP Group menyediakan kode sumber lengkap bagi pengguna untuk membangun, menyesuaikan dan mengembangkannya untuk mereka gunakan sendiri.

3.11.1 **Arsitektur PHP**

Menurut pendapat Nixon (2009), *model-view-control* merupakan *design pattern* yang menyediakan aplikasi pengembangan dan pemeliharaan. MVC merupakan suatu istilah untuk memisahkan elemen-elemen atau penulisan *source*

code dari suatu aplikasi agar menjadi lebih rapid dan lebih mudah untuk melakukan perbaikan dan pengembangan.

Ada 3 *layer* utama dalam arsitektur web MVC, yaitu:

1. Model

Model disini berperan sebagai representasi dari data yang terlibat dalam sebuah proses. Setiap kali *method* atau *function* dari suatu aplikasi dibutuhkan untuk melakukan akses ke dalam suatu data, maka *method* atau *function* tersebut tidak langsung berinteraksi dengan sumber data tersebut melainkan harus melalui *model* terlebih dahulu.

2. View

View disini berperan sebagai *presentation layer* atau pengatur *user interface*. Data yang dibutuhkan *user* akan diformat sedemikian rupa agar dapat tampil dan dipresentasikan dengan format tampilan yang memang disesuaikan dengan kebutuhan *user*.

3. Controller

Controller disini berperan sebagai *logic aspect* dari suatu aplikasi atau mengatur *user flow*. *Controller* yang akan menentukan *business process* dari aplikasi yang dibangun. Bagian ini yang akan merespon setiap inputan dari *user* dengan melakukan pemanggilan terhadap *model* dan *view* yang sesuai sehingga permintaan dari *user* tersebut dapat terpenuhi dengan baik.

3.12 *MyStructured Query Language (MySQL)*

Menurut Nugroho (2005), *MyStructured Query Language (MySQL)* adalah salah satu *database server* yang menggunakan bahasa SQL (*Structured Query Language*) adalah bahasa pertanyaan (*query language*) yang distandarisasi untuk menanyakan informasi dari sebuah basisdata (*database*). Versi asalnya dinamakan SEQUEL (*Structure English Query Language*), dirancang oleh peneliti dari pusat penelitian IBM pada tahun 1974 dan 1975. SQL pertama kali diperkenalkan sebagai sistem basisdata komersial pada tahun 1979 oleh *Oracle Corporation*. *MySQL* adalah *Relational Database Management System (RDBMS)* yang sangat cepat dan kuat.

MySQL adalah DBMS yang bersifat *relasional, opensource, berlevel enterprise, multithread*. *MySQL* merupakan bahasa yang memiliki kemampuan cukup baik untuk menunjang kerja *user*, baik yang telah berpengalaman dengan *database* maupun untuk pemula. SQL digunakan untuk mencari informasi (*query*), memanipulasi data (DML) mendefinisikan data (DDL), dan bahasa pengendali dokumentasi. Setiap pengguna basis data memerlukan bahasa pemrograman yang dapat dipakai sesuai dengan fungsi dan tugasnya. Dalam basis data secara umum dikenal dua bahasa, yaitu:

1. *Data Definition Language (DDL)* adalah bahasa yang dipakai untuk menjelaskan objek dari bahasa data. DDL dipakai untuk mendefinisikan kerangka basis data (berorientasi pada tipe pada objek basis data).
2. *Data Manipulation Language (DML)* adalah bahasa yang dipakai untuk memanipulasi objek data dari basis data. DML dipakai untuk operasi terhadap isi basis data.

3.13 Interaksi Manusia dan Komputer

Suatu Aplikasi yang baik tentunya harus mempertimbangkan interaksi antara pengguna dan program yang dibuat. Di sinilah pentingnya penerapan ilmu Interaksi Manusia dan Komputer.

Menurut Santoso (2004), Interaksi Manusia dan Komputer (IMK) adalah sebuah disiplin ilmu yang mempelajari desain, evaluasi, implementasi dari sistem komputer interaktif untuk dipakai oleh manusia, beserta studi tentang faktor-faktor utama dalam lingkungan interaksinya.

Deskripsi lain dari IMK adalah suatu ilmu yang mempelajari perencanaan dan desain tentang cara manusia dan komputer saling bekerja sama, sehingga manusia dapat merasa puas dengan cara yang paling efektif. Dikatakan juga bahwa sebuah desain antar muka yang ideal adalah yang mampu memberikan kepuasan terhadap manusia sebagai pengguna dengan faktor kapabilitas serta keterbatasan yang terdapat dalam sistem.

Pada implementasinya, interaksi manusia dan komputer dipengaruhi berbagai macam faktor antara lain organisasi, lingkungan, kesehatan, pengguna, kenyamanan.