# BAB II
# TINJAUAN PUSTAKA

## 2.1 Tinjauan Perusahaan

### 2.1.1 Sejarah Employment Service Centre – Universitas Negeri Jakarta (*ESC – UNJ)*

ESC UNJ adalah sebuah bursa kerja khusus yang bertujuan untuk menjembatani pertemuan antara para pencari kerja (Jobseekers) dan para pemberi kerja (Company). ESC UNJ mengkoordinir kesempatan dan lowongan pekerjaan bagi lulusan UNJ maupun bagi masyarakat umum. ESC UNJ diresmikan oleh Bapak Dr. Erman Soeparno pada tanggal 9 Desember 2008 yang pada waktu itu menjabat sebagai Menteri Tenaga Kerja dan Transmigrasi RI. Secara struktural dan fungsi, ESC UNJ berada di bawah Rektor UNJ dengan koordinasi Pembantu Rektor IV bidang Kerjasama dan Hubungan Luar Negeri. ESC UNJ berdiri sebagai salah satu hasil MoU antara UNJ dengan Kemennakertrans RI.

Untuk melayani kebutuhan informasi lowongan kerja secara online via internet, ESC UNJ dilengkapi dengan 3 buah komputer yang dapat digunakan oleh para pencari kerja secara gratis. Dengan adanya kemudahan ini sehingga mahasiswa tingkat akhir atau para lulusan yang terdaftar sebagai member ESC UNJ dapat memanfaatkan fasilitas tersebut untuk menelusuri informasi lowongan pekerjaan dari berbagai perusahaan secara langsung melalui internet.

### 2.1.2 Visi

Employment Service Centre menjadi puasat informasi, pelatihan, pengembangan kerja dan Sumber Daya Manusia.

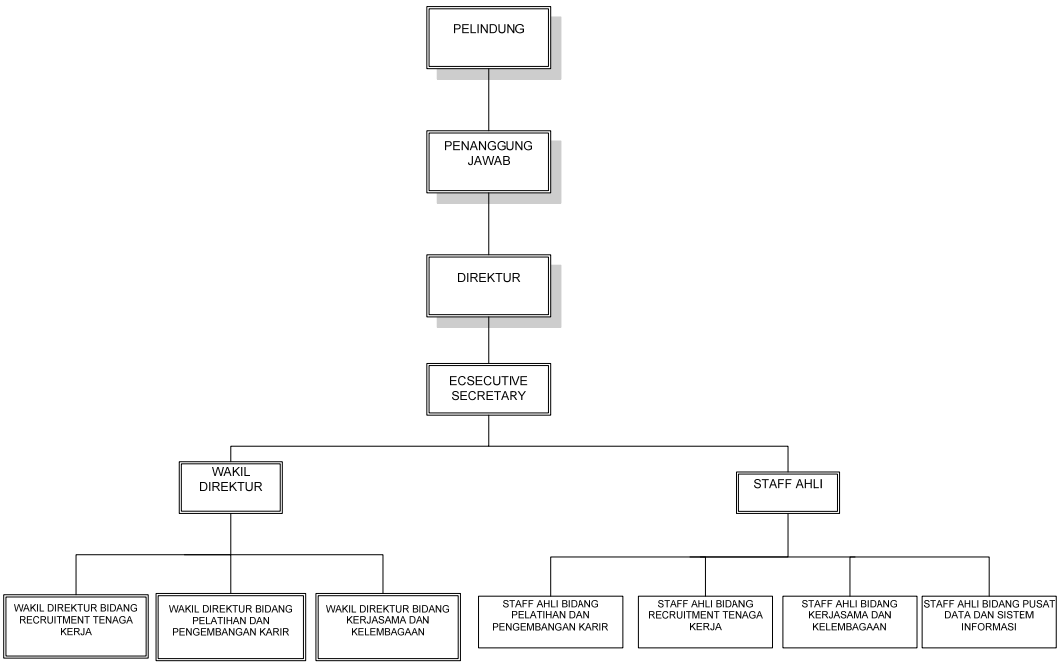
### 2.1.3 Misi

Misi Employment Service Centre yaitu :

a. Membangun jejaringan informasi antara ESC-UNJ dengan dunia usaha dan industry dalam bidang ketenagakerjaan.
b. Membangun jejaringan informasi antara ESC-UNJ dengan asosiasi profesi dalam bidang pelatihan dan pengembangan tenaga kerja.
c. Melaksanakan program pelatihan kerja dengan program CSR dengan dunia usaha dan industry dalam rangka meningkatkan kualitas skill (keahlian) tenaga kerja tingkat operator.
d. Melaksanakan program pembinaan kewirausahaan mahasiswa dan usaha Kecil Menengah.

### 2.1.4 Struktur Organisasi

Didalam suatu perusahaan, agar setiap kegiatan yang dilaksanakan hasilnya selalu baik maka dibutuhkan kerjasama antar individu. Untuk pencapaian tujuan perusahaan, maka dalam setiap perusahaan harus memiliki struktur organisasi yang jelas dan sistematis. Selain itu juga berguna untuk meningkatkan efisiensi kerja dan produktifitas kerja sejalan dengan perkembangan aktifitas usaha. Berikut adalah struktur organisasi di Employment Service Centre :




**Gambar 2.1. Struktur Organisasi Employment Service Centre**

## 2.2 Landasan Teori

### 2.2.1 Sistem Informasi

#### 2.2.1.1 Konsep dasar sistem

Suatu sistem pada dasarnya adalah sekolompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersamasama untuk mencapai tujuan tertentu.Secara sederhana, suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur, komponen, atau variabel yang terorganisir, saling berinteraksi, saling tergantung satu sama lain, dan terpadu. Dari defenisi ini dapat dirinci lebih lanjut pengertian sistem secara umu, yaitu :

1. Setiap sistem terdiri dari unsurunsur
2. Unsurunsur tersebut merupakan bagian terpadu sistem yang bersangkutan.
3. Unsur sistem tersebut bekerja sama untuk mencapai tujuan sistem.
4. Suatu sistem merupakan bagian dari sistem lain yang lebih besar.

#### 2.2.1.2 Konsep dasar informasi

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Informasi merupakan data yang telah diklasifikasikan atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan.

#### 2.2.1.3 Konsep dasar sistem informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

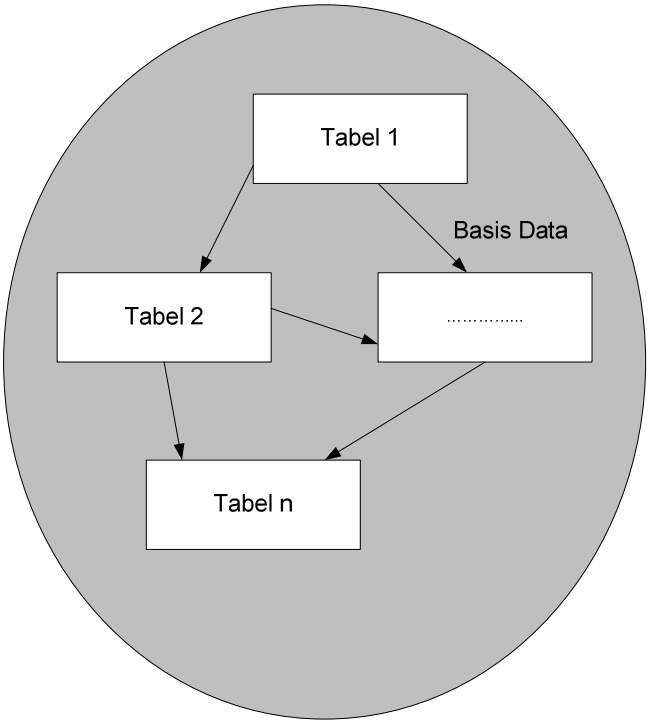
### 2.2.2 Internet

Internet adalah suatu jaringan komputer yang satu dengan yang lain saling terhubung untuk keperluan komunikasi dan informasi. Sebuah komputer dalam satu jaringan internet dapat berada di mana saja atau bahkan di seluruh Indonesia. Sering juga internet diartikan sebagai jaringan komputer di seluruh dunia yang berisikan informasi dan sebagai sarana komunikasi data yang berupa suara, gambar, video dan juga teks. Informasi ini dibuat oleh penyelenggara atau pemilik jaringan komputer atau dibuat pemilik informasi yang menitipkan informasinya kepada penyedia layanan internet.

### 2.2.3 Basis Data

#### 2.2.3.1 Pengertian Basis Data

Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah dioleh atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.




**Gambar 2.2 ilustrasi basis data**

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apa pun bentuknya, entah berupa *file* teks ataupun *database management system* (DBMS).

Kebutuhan basis data dalam sistem informasi meliputi :

1. Memasukkan,menyimpan,dan mengambil data
2. Membuat laporan berdasarkan data yang telah disimpan

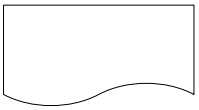
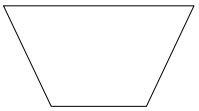
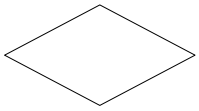
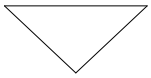
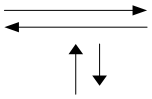
#### 2.2.3.2 Skema Relasi

Proses *relasi* antar artibut merupakan gabungan antar artibut yang mempunyai kunci utama yang sama. sehingga artibut-artibut tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci tersebut pada proses ini elemen-elemen data dikelompokan menjadi satu *field database* beserta *entitas* dan hubungannya.

#### 2.2.3.3 Flow Map

*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 sering disebut juga dengan nama *bubble chart*, *bubble diagram*,model proses, *diagram* alur kerja, atau model fungsi.

Adapun simbol-simbol yang digunakan dalam *flow map* adalah :

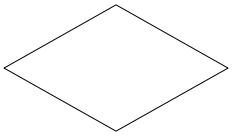
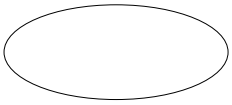
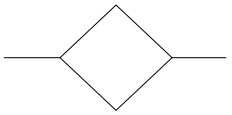
| No. | Symbol | Nama | Keterangan |
|---|---|---|---|
| 1 |  | Proses | Menunjukan proses yang terjadi didalam aliran komputer. |
| 2 |  | Dokumen | Menunjukan arus dari suatu arus. |
| 3 |  | Proses manual | Menunjukan kegiatan proses yang manual. |
| 4 |  | Database | Menunjukan database tempat penyimpanan data-data terkait. |
| 5 |  | Penyelesaian | Untuk menyeleksi kondisi. |
| 6 |  | Simpan | File Non-komputer yang di arsipkan. |
| 7 |  | Garis aliran | Menunjukan arus dari suatu arus. |

**Gambar 2.3 Simbol-simbol *Flowmap***

### 2.2.3.4 ERD ( Entity Relationship Diagram )

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak ERD adalah notasi grafik dari sebuah model *data* atau model jaringan yang menjelaskan tentang *data* yang disimpan (*storage data*) dalam sistem secara acak elemen-elemen diagram hubungan *entity entity* (*entitas*) *relationship* (Hubungan antar *entitas*) *attribute* (Kolom) *cardinality* (Jumlah maksimum baris).

### 2.2.3.5 Simbol-simbol ERD

| No. | Symbol | Keterangan |
|---|---|---|
| 1 |  | Menunjukan suatu objek berupa benda, konsep atau entitas. |
| 2 |  | Menunjukan hubungan logical antara satu file dengan file lainnya. |
| 3 |  | Menunjukan artibut suatu entitas. |
| 4 |  | Menunjukan suatu hubungan. |

**Gambar 2.4 Simbol-simbol ERD**

### 2.2.3.6 Derajat Hubungan Relasi

Derajat hubungan suatu relasi diantaranya :

a. Satu ke satu ( 1-1 )

   Yang berarti setiap *entitas* pada himpunan *entitas* A berhubungan paling banyak sama dengan entitas pada himpunan entitas B.

b. Satu ke banyak ( 1-N )

   Yang berarti setiap *entitas* pada himpunan *entitas* A dapat berhubungan banyak *entitas* pada himpunan *entitas* B. tetapi tidak berlaku sebaliknya.

c. Banyak ke Satu ( N -1 )

   Yang berarti setiap *entitas* pada himpunan *entitas* A dapat berhubungan paling banyak dengan satu *entitas* pada himpunan *entitas* B. tetapi tidak berlaku sebaliknya.

d. Banyak ke banyak ( N-N )

Yang berarti setiap *entitas* pada himpunan *entitas* A dapat berhubungan banyak *entitas* pada himpunan *entitas* B. dan berlaku sebaliknya.

### 2.2.3.7 DFD ( Data Flow Diagram )

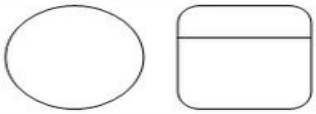
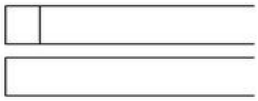
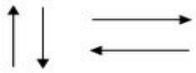
*Data Flow Diagram* (DFD) adalah suatu model logika *data* atau proses yang dibuat untuk menggambarkan dari mana asal *data* dan kemana tujuan *data* yang keluar dari sistem, dimana data tersimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data tersimpan dan proses yang dikenakan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

**1. Keuntungan DFD : Data *Flow* Diagram**

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

**2. Simbol – simbol DFD**

DFD terdiri dari empat simbol yaitu :

| Simbol | Keterangan |
|---|---|
|  | *External Entity*, merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain. |
|  | *Process*, merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan |
|  | *Data Store* (Simpan Data), dapat berupa suatu file atau database pada sistem komputer atau catatan manual |
|  | *Data Flow* (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar |

**Gambar 2.5 Simbol-simbol DFD**

### 2.2.3.8 Tipe DFD

DFD terdiri dari konteks *diagram* dan *diagram rinci (DFD Levelled).* konteks diagram berfungsi memetakan model lingkungan (menggambarkan hubungan antara *entitas* luar, masukan dan keluaran sistem), yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem.

DFD *levelled* menggambarkan sistem sebagai jaringan kerja antara fungsi yang berhubungan satu sama lain dengan aliran dan penyimpanan data, model ini hanya memodelkan sistem dari sudut pandang fungsi.

**1. DFD : konteks Diagram**

Diagram konteks adalah diagram tingkat atas, yaitu diagram yang paling tidak detil dari sebuah sistem informasi yang menggambarkan aliran-aliran data kedalam dan keluar sistem kemudian kedalam dan keluar entitas-entitas eksternal.

**2. DFD *Levelled***

Dalam DFD *levelled* akan terjadi penurunan *level* dimana dalam penurunan *level* yang lebih rendah harus mampu merepresentasikan proses tersebut ke dalam spesifikasi proses yang jelas. DFD *levelled* bisa dimulai dari

DFD *level* 0, kemudian turun ke DFD *level* 1 dan seterusnya. Setiap penurunan hanya dilakukan bila perlu. Aliran data yang masuk dan keluar pada suatu proses di *level* x harus berhubungan dengan aliran data yang masuk dan keluar pada *level* x+1 yang mendefinisikan proses pada *level* x tersebut. Prosses yang tidak dapat diturunkan atau dirinci lagi dikatakan primitif secara fungsional dan disebut sebagai proses primitif.

Secara umum, diagram *level* 1 dibuat dari setiap proses utama pada diagram *level* 0. Menunjukkan semua *internal* proses termasuk proses tunggal pada diagram *level* 0. Menunjukkan semua internal proses termasuk proses tunggal pada diagram *level* 0. Menunjukkan bagaimana informasi mengalir dari dan ke setiap proses tersebut. Jika sebuah *parent* proses didekomposisikan menjadi 3 proses *child* misalnya, maka 3 proses *child* tersebut harus lengkap dan dibangun berdasarkan *parent* prosesnya.

**3. Level 1 Diagrams**

Menunjukkan semua proses yang termasuk pada *single* proses pada diagram *level* 1. Menunjukkan bagaimana sistem informasi mengalir dari dan ke setiap proses tersebut. Diagram *level* 2 boleh tidak dilakukan/dibuat pada setiap proses pada *level* 1. Penomoran secara benar pada setiap proses membantu user untuk memahami dimana proses tersebut terjadi pada keseluruhan sistem.

**4. Level 2 Diagrams**

Membuat konteks diagram. Membuat fragmentasi DFD untuk setiap *case*. Mengorganisasikan *fragmentasi* DFD ke dalam level 0 diagram. Dekomposisi proses pada *level* 0 ke dalam diagram *level* 1 jika diperlukan. Dekomposisi proses *level* 1 ke dalam diagram *level* 2 jika diperlukan. Validasi DFD dengan *uses* untuk kelengkapan dan kebenaran proses dalam sistem.

### 2.2.3.9 Peraturan Pembuatan DFD

Dalam pembuatan DFD, ada beberapa aturan yang harus diperhatikan agar dalam proses penggambarannya tidak terjadi kesalahan. Dalam hal ini saya hanya dapat menyebutkan delapan point saja, diantaranya yaitu :

1. Antar *entity* yang satu dengan yang lain tidak boleh berhubungan dengan anak panah secara langsung atau tidak boleh berhubungan atau ber*relasi*.
2. *Entity* tidak boleh langsung berhubungan dengan penyimpanan data (data *store*).
3. Satu alir data boleh merepresentasikan beberapa struktur data.
4. Untuk alasan kerapian (menghindari aliran data yang bersilangan), entitas eksternal atau data *store* boleh digambar beberapa kali dengan tanda khusus, misalnya diberi nomor.
5. Semua objek harus mempunyai nama.
6. Bentuk anak panah aliran data boleh bervariasi.
7. Aliran data harus selalu diawali dan diakhiri dengan proses atau *entity*.
8. Jumlah proses tidak lebih dari sembilan proses dalam sistem, jika melebihi maka sebaiknya dikelompokkan beberapa proses yang bekerja bersama-sama didalam suatu subsistem..

### 2.2.3.10 Langkah – Langkah Pembuatan DFD

Berikut adalah langkah-langkah dalam pembuatan DFD :

1. Membuat konteks diagram.
2. Membuat *fragmentasi* DFD untuk setiap *case*.
3. Mengorganisasikan *fragmentasi* DFD ke dalam *level* 0 diagram
4. Dekomposisi proses pada *level* 0 ke dalam diagram *level* 1 jika diperlukan, dekomposisi proses *level* 1 ke dalam diagram *level* 2 jika diperlukan, dst.

5. *Validasi* DFD dengan *uses* untuk kelengkapan dan kebenaran proses dalam sistem.
6. Gambarkan satu proses untuk representasikan seluruh sistem (proses 0).
7. Inventarisasi semua *input* dan *output* dari sistem besarannya yang digambarkan sebagai data *flow*.
8. Gambarkan dalam *entitas* eksternal untuk sumber atau tujuan dari data *flow*.

### 2.2.4 MYSQL

MYSQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan

data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query data*. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase *come-first served.*

#### 2.2.4.1 PhpMyAdmin

*PhpMyAdmin* adalah perangkat lunak gratis yang ditulis dalam PHP, dimaksudkan untuk menangani administrasi MySQL melalui *World Wide Web*. phpMyAdmin mendukung berbagai operasi dengan MySQL. Yang paling sering digunakan operasi yang didukung oleh *user interface* (mengelola database, tabel, ladang, hubungan, indeks, pengguna, perizinan, dll), sementara Anda masih memiliki kemampuan untuk secara langsung mengeksekusi pernyataan SQL.

### 2.2.5 WAMP

*Wamp* adalah sebuah aplikasi yang dapat menjadikan komputer kita menjadi sebuah s*erver*. Kegunaan *wamp server* ini untuk membuat jaringan *local* sendiri, dalam artian kita dapat membuat *website* secara *offline* untuk masa coba-coba di komputer sendiri. Jadi fungsi dari wamp *server* itu sendiri merupakan *server website* kita untuk cara memakainya. Mengapa harus menjadi *server* ? Karena dalam hal ini komputer yang akan kita pakai harus memberikan pelayanan untuk pengaksesan *website*, untuk itu komputer kita harus menjadi *server.*

Biasanya para perancang *website* atau *webmaster* jika akan merencanakan (*planing*), kemudian membangun (*building*) dilakukan di komputer *local* atau bisa juga di jaringan *local*, tidak langsung di *host* internet. Oleh karena itu perlu di komputer kita di jadikan *server* sehingga kita seolah-olah sedang meng-*update* di *hostnya* (tempat penyimpanan *file-file* yang diperlukan *website*) internet. Dengan di tempatkannya *file-file* pendukung *website* di komputer kita.

Kita tidak perlu buka masuk internet sehingga hal ini mengurangi presentasi waktu dan biaya, yang jika dilakukan langsung dijaringan internet akan memakan waktu yang lama juga untuk *upload*-nya (*edit file-file website*-nya seperti html, php atau asp).

### 2.2.6 Adobe Dreamweaver CS3

**Adobe Dreamweaver** merupakan program penyunting halaman *website* keluaran Adobe Systems yang dulu dikenal sebagai **Macromedia Dreamweaver** keluaran Macromedia. Program ini banyak digunakan oleh pengembang *website* karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 8. Versi terakhir Dreamweaver keluaran Adobe Systems adalah Adobe Dreamweaver CS5 yang ada dalam Adobe Creative Suite.

### 2.2.7 PHP

#### 2.2.7.1 Definisi PHP

Adalah bahasa pemrograman *script* yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memprogram situs *website* dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Contoh terkenal dari aplikasi PHP adalah phpBB dan MediaWiki (software di belakang Wikipedia). PHP juga dapat dilihat sebagai pilihan lain dari ASP.NET/C#/VB.NET Microsoft, ColdFusion Macromedia, JSP/Java Sun Microsystems, dan CGI/Perl. Contoh aplikasi lain yang lebih kompleks berupa CMS yang dibangun menggunakan PHP adalah Mambo, Joomla!, Postnuke, Xaraya, dan lain-lain.

#### 2.2.7.2 Kelebihan PHP

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.

### 2.2.8 Java Script

#### 2.2.8.1 Definisi Java Script

Java Script adalah bahasa *scripting* yang popular di internet dan dapat bekerja di sebagian besar *browser* popular seperti Internet Explorer (IE), Mozilla Firefox, Netscape dan Opera. Kode Java Script dapat disisipkan dalam halaman *website* menggunakan tag *script*.

Beberapa hal tentang Java Script :

1. Javascript di desain untuk menambah interaktif suatu web.
2. Javascript merupakan sebuah bahasa scripting.
3. Bahasa scripting merupakan bahasa pemrograman yang ringan.
4. Javascript berisi baris kode yang dijalankan di computer (*web browser*)
5. Javascript biasanya disisipkan (*embedded*) dalam halaman HTML.
6. Javascript adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi)
7. Setiap orang dapat menggunakan javascript tanpa membayar lisensi.

Contoh scipt :

```
<html>
<body>
<h1>MyFirstWebPage</h1>
<p id="demo">This is a paragraph.</p>
<script type="text/javascript">
...someJavaScriptcode ...
</script>
```

### 2.2.8.2 Kelebihan Javascript

Kelebihan java script antara lain yaitu :

1. Ukuran file kecil

Script dari Java Script memiliki ukuran yang kecil sehingga ketika *web* yang memiliki Java Script ditampilkan di *browser* maka akses tampilannya akan lebih cepat dibandingkan ketika browser membuka suatu *web* yang memiliki script java. Hal ini juga sangat berkepentingan dengan daya kerja *server*. Semakin kecil *space* suatu *web* yang disimpan dalam suatu *server* maka daya kerja *server* ketika di *browsing* oleh *user* di internet akan tidak terlalu berat, selain itu sifat Java Script *client side* yang tidak perlu lagi di olah oleh *server* ketika *browser* memanggil *web* dari sebuah *server*.

2. Mudah untuk dipelajari

Javascript merupakan bahasa semi pemograman yang merupakan gabungan antara bahasa pemograman java dengan bahasa kode HTML

sehingga disebut bahasa *hybrid*. Walaupun Java Script merupakan turunan dari java namun Java Script tidak memiliki aturan yang serumit java.

3. Terbuka

Javascript tidak terikat oleh hardware maupun software tertentu bahkan system operasi seperti windows maupun unix. Karena ia bersifat terbuka, maka ia dapat dibuat maupun di baca di semua jenis komputer.Ukurannya sangat kecil dibandingkan dengan pemrograman Java.

#### 2.2.8.3 Kekurangan Javascript

Berikut kekurangan javascript :

1. Script tidak terenkripsi

Karena javascript bersifat *client side*, maka script yang kita buat di text editor dan telah dijadikan *web* di *server*, ketika user me*request web* dari *server* tersebut maka sintak javascript akan langsung ditampilkan di *browser*. *User* bisa melihat dan menirunya dari s*ource*nya.

2. Kemampuan terbatas

Walaupun javascript mampu membuat bentuk *website* menjadi interaktif dan dinamis, namun javascript tidak mampu membuat program aplikasi sendiri seperti java.

3. Keterbatasan Objek

Javascript tidak mampu membuat kelas-kelas yang bisa menampung objek-objek tambahan seperti java karena javascript teleh memiliki objek yang builtin pada sturktur bahasanya.

### 2.2.9 CSS (Cascading Style Sheet)

#### 2.2.9.1 Definisi CSS

CSS merupakan singkatan dari *Cascading style sheet*. Kegunaannya adalah untuk mengatur tampilan document HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar.

CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan halaman *website*. Perlu di ingat, CSS hanyalah berupa kumpulan script yang tujuannya bukan untuk menggantikan HTML, melainkan pelengkap agar document HTML bisa tampil lebih cantik dan dinamis. Sejak awal ditemukan CSS pada awal decade 90an, CSS terus dikembangkan dan diserap oleh *web developer*.

Kode CSS bersifat lintas *platform*, yang berarti script ini dapat di baca oleh berbagai macam sistem operasi dan *browser*. Hanya saja *browser* seperti internet Eplorer, seringkali salah mengartikan script CSS yang menyebabkan tidak sempurnanya tampilan document HTML. Script CSS perlu dioptimalkan agar tampil maksimal pada *browser* internet explorer.

Contoh script:

```
body
{
background-color:#d0e4fe;
}
h1
{
color:orange;
text-align:center;
}
p
{
font-family:"Times New Roman";
font-size:20px;

}
```

#### 2.2.9.2 Versi CSS

Untuk saat ini terdapat dua versi CSS, yaitu CSS1 dan CSS2. CSS 1 dikembangkan berpusat pada pemformatan dokumen HTML, sedangkan CSS 2 dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di printer.

Selain itu CSS 2 mendukung penentuan posisi konten, *downloadable*, huruf (*font*), tampilan pada tabel (*table layout*) dan media tipe untuk printer.Kehadiran versi CSS yang kedua diharapkan lebih baik dari versi pertama**.**

### 2.2.10 JQuery

Jquery adalah librari javascript yang memungkinkan kita untuk membuat program *web* pada suatu halaman *web,* tanpa harus secara eksplisit kita menambahkan event atau pun properti pada halaman *web* tersebut. Dengan jquery, suatu halaman *web* yang menjadi aplikasi *web*, jika dilihat *source*nya, akan terlihat seperti dokumen HTML biasa, tidak ada kode javascript yang terlihat langsung. Teknik pemrograman *website* seperti ini disebut sebagai *unobstrusive* javascript programming.

*Jquery* merupakan salah satu *library* yang membuat program *web* di sisi *client*, tidak terlihat sebagai program javascript biasa, yang harus secara eksplisit disisipkan pada dokumen *web*. Pada teknik pemrograman sisi *client* dengan menggunakan javascript biasa, setiap elemen yang akan memiliki event, akan secara eksplisit terlihat ada event yang dilekatkan pada elemen tersebut. Jquery dikembangkan pertama kali oleh John Resig, yang dibuat lebih ramping dari *library* Prototype yang menjadi inspirasi dari *libarary* Jquery ini. Secara pemrograman, jquery memiliki kemiripan seperti prototype. Jquery, merupakan *library* yang sangat ramping, core dari *library* ini dalam keadaan terkompres hanya berukuran sekitar 19KB.

### *2.2.11* Metodologi W*aterfall*

#### *2.2.11.1* Pengertian Metodologi *Waterfall*

Model ini sering juga disebut dengan *classic life cycle*. Dalam metode ini membutuhkan pendekatan sistematis dan sekuensial dalam pengembangan perangkat lunak, dimulai dari tingkat sistem dan kemajuan melalui analisis, desain, coding, testing dan pemeliharaan. pemodelan ini mengikuti beberapa aktivitas berikut :

1. Rekayasa dan pemodelan sistem/informasi (*System/Information Engineering and Modeling*). Tahap ini sering disebut juga dengan *Project Definition.*
2. Analisis kebutuhan perangkat lunak (*Software Requirements Analysis*). Proses pengumpulan kebutuhan diintensifkan ke perangkat lunak. Hasilnya harus didokumentasikan dan di *review* ke pelanggan.
3. Desain (*Design*). Proses desain mengubah kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program.
4. Penulisan Program (Coding). Desain tadi harus diubah menjadi bentuk yang dimengerti mesin (komputer). Maka dilakukan langkah penulisan program.
5. Testing. Setelah kode program selesai dibuat, dan program dapat berjalan, testing dapat dimulai. Testing difokuskan pada logika *internal* dari perangkat lunak, fungsi *eksternal*, dan mencari segala kemungkinan kesalahan.
6. *Support/Maintenance*. Perangkat lunak setelah diberikan pada pelanggan, mungkin dapat ditemui error ketika dijalankan dilingkungan pelanggan. Pemeliharaan ini dapat berpengaruh pada semua langkah yang dilakukan sebelumnya.

#### 2.2.11.2 Kelebihan dari metode WaterFall

Metode ini masih lebih baik digunakan walaupun sudah tergolong kuno, daripada menggunakan pendekatan asal-asalan. Selain itu, metode ini juga masih masuk akal jika kebutuhan sudah diketahui dengan baik.

#### 2.2.11.3 Kekurangan dari metode Waterfall

Berikut beberapa kekurangan dari metodelogi *Waterfall* :

1. Pada kenyataannya, jarang mengikuti urutan sekuensial seperti pada teori. Iterasi sering terjadi menyebabkan masalah baru.
2. Sulit bagi pelanggan untuk menentukan semua kebutuhan secara eksplisit.
3. Pelanggan harus sabar, karena pembuatan perangkat lunak akan dimulai ketika tahap desain sudah selesai. Sedangkan pada tahap sebelum desain bisa memakan waktu yang lama.
4. Kesalahan di tahap awal berakibat sangat fatal pada tahap berikutnya.

### 2.2.12 Desain Message Digest Algorithm 5 (MD5)

**MD-5** merupakan fungsi hash satu arah yang diciptakan oleh Ron Rivest. MD-5 adalah salah satu aplikasi yang digunakan untuk mengetahui bahwa pesan yang dikirim tidak ada perubahan sewaktu berada dijaringan.

Algoritma MD-5 secara garis besar adalah mengambil pesan yang mempunyai panjang variabel diubah menjadi ‘sidik jari’ atau ‘intisari pesan’ yang mempunyai panjang tetap yaitu 128 bit. ‘Sidik jari’ ini tidak dapat dibalik untuk mendapatkan pesan, dengan kata lain tidak ada orang yang dapat melihat pesan dari ‘sidik jari’ MD-5. *Message digest* atau intisari pesan harus mempunyai tiga sifat penting, yaitu:

1. Bila P diketahui, maka MD(P) akan dengan mudah dapat dihitung.
2. Bila MD(P) diketahui, maka tidak mungkin menghitung P.
3. Tidak seorang pun dapat memberi dua pesan yang mempunyai intisari pesan yang sama

Tujuan yang hendak dicapai adalah merencanakan dan merancang suatu aplikasi untuk menganalisa proses keutuhan atau pun perubahan pesan dengan menggunakan Message Digest 5 (MD5) dan juga dapat menganalisa hasil keluaran dari MD5 yang berupa kecepatan dari proses aplikasi yang dibuat.

MD5 merupakan bantahan atas teori yang menyatakan, untuk menghasilkan tanda tangan digital yang baik maka panjang tanda tangan digital harus sama dengan panjang masukannya. Berikut ini adalah contoh tanda tangan *digital* dengan menggunakan algoritma MD5 :

1. md5("B") = 0947f85161b05919d96940f3de14852e
2. md5 ("b") = 92eb5ffee6ae2fec3ad71c777531578f
3. md5 ("a") = 0cc175b9c0f1b6a831c399e269772661
4. md5 ("a.") = 9fbcccf456ef61f9ea007c417297911d
5. md5 ("a ") = 99020cb24bd13238d907c65cc2b57c03
6. md5 ("a ") = d4ac0334c4130de05b4a37a87590ccc4
7. md5 ("a,") = 3ded2184a3e467984dba5788f82cc430

Contoh pertama menunjukkan hasil output karakter “B”. Contoh kedua adalah output karakter “b”. Ternyata dari hasil perbandingan terlihat bahwa walaupun terlihat hampir sama, tetapi jenisnya berbeda maka fungsi MD5 akan mengeluarkan hasil yang tidak identik. Lima contoh terakhir menunjukkan bahwa walaupun huruf yang diinputkan sama, tetapi penambahan karakter atau spasi sebanyak satu atau dua spasi serta perubahan apapun terhadap input akan memberikan output berbeda.

Dari contoh di atas dapat disimpulkan bahwa algoritma MD5 selalu menghasilkan tanda tangan *digital* sepanjang 32 karakter, tanpa tergantung panjang input. Selain itu hasil output tidak akan sama untuk input yang berbeda.

Kelebihan dari MD5 adalah sulit untuk dipecahkan walaupun dengan serangan brute force, tingkat keamanan MD5 adalah salah satu yang terbaik, tidak

bisa diubah kembali menjadi data asli (*irreversible*), hasil keluaran MD5 selalu 32 karakter.

### 2.2.13 Lowongan Kerja

#### 2.2.13.1 Pengertian Lowongan Kerja

Lowongan kerja atau pekerjaan bagi sebagian orang mungkin hanya sebatas peluang untuk kerja di kantor dengan menjadi pegawai baik pegawai negeri sipil (PNS) atau karyawan swasta. Lowongan kerja seperti ini biasanya dimuat di koran, bisa juga di majalah atau tv dan radio. Setelah mengetahui adanya lowongan kerja dan lowongan tersebut cocok dengan yang dicari maka orang akan membuat surat lamaran kerja, mengirimkan surat lamaran kerja tersebut ke perusahaan kemudian nunggu dipanggil untuk wawancara. Lowongan artinya peluang yang kosong alias belum terisi, dalam hal ini peluang untuk bekerja. Kalau didefenisikan seperti ini maka sebenarnya lowongan kerja banyak sekali, tidak terbatas pada pekerjaan kantoran atau kerja di pabrikdan perusahaan swasta lainnya.

Dalam bisnis pun orang bekerja, berjualanpun peluang kerja, bahkan jika ada peluang untuk bekerja di rumah sendiri, misalnya dengan menjalankan bisnis internet, itu juga lowongan kerja. Kalau lowongan kerja dalam arti sempit seperti kerja sebagai karyawan saja seperti diatas, maka persaingan untuk mendapatkannya sangat padat. Bisa jadi lowongan kerja tersebut hanya akan menerima 10 orang saja sedangkan yang melamar bisa ratusan atau bahkan ribuan. Sedangkan lowongan kerja dalam arti yang luas misalnya bekerja sendiri atau menjalankan bisnis member get member, terbuka sangat luas bahkan selalu ditawarkan orang. Namun banyak yang belum mau, sedangkan jadi karyawan belum diterima, sehingga tidak heran banyak sekali orang yang katanya tidak punya pekerjaan atau pengangguran. Pengangguran mulai dari level bawah sampai level sarjana.

### 2.2.14 Situs jejaring sosial

#### 2.2.14.1 Pengertian jejaring sosial

Jejaring sosial adalah struktur sosial yang terdiri dari elemen-elemen individual atau organisasi. Jejaring ini menunjukan jalan dimana mereka berhubungan karena kesamaan sosialitas, mulai dari mereka yang dikenal seharihari sampai dengan keluarga. Istilah ini diperkenalkan oleh profesor J.A. Barnes di tahun 1954.

Jejaring sosial adalah suatu struktur sosial yang dibentuk dari simpulsimpul (yang umumnya adalah individu atau organisasi) yang diikat dengan satu atau lebih tipe relasi spesifik seperti nilai, visi, ide, teman, keturunan dan lainnya.

#### 2.2.14.2 Layanan Jejaring sosial

Banyak layanan jejaring sosial berbasiskan *web* yang menyediakan kumpulan cara yang beragam bagi pengguna untuk dapat berinteraksi seperti *chat, messaging, email,* video, *chat* suara, *share file*, blog, diskusi grup, dan lain-lain. Umumnya jejaring sosial memberikan layanan untuk membuat biodata dirinya.

Pengguna dapat meng-*upload* foto dirinya dan dapat menjadi teman dengan pengguna lainnya. Beberapa jejaring sosial memiliki fitur tambahan seperti pembuatan grup untuk dapat saling sharing didalamnya.

#### 2.2.14.3 Facebook

*Facebook* adalah situs *web* jaringan sosial yang diluncurkan pada 4 Februari 2004 dan didirikan oleh Mark Zuckerberg, awalnya hanya untuk linkungan sendiri dimana digunakan untuk komunikasi antar mahasiswa lulusan Harvard dan mantan murid Ardsley High School.

Dalam dua bulan selanjutnya, keanggotaannya diperluas ke sekolah lain di wilayah Boston (Boston College, Boston University, MIT, Tufts), Rochester, Stanford, NYU, Northwestern, dan semua sekolah yang termasuk dalam Ivy

League. Banyak perguruan tinggi lain yang selanjutnya ditambahkan berturut-turut dalam kurun waktu satu tahun setelah peluncurannya.

Tetapi karena para mahasiswa tersebut mempunyai teman lain dan mengajak teman temannya juga bergabung sehingga anggotanya hampir seluruh mahasiswa sedunia. Selanjutnya dikembangkan pula jaringan untuk sekolah-sekolah tingkat atas dan beberapa perusahaan besar. Sejak 11 September 2006, orang dengan alamat surat email apa pun dapat mendaftar di facebook. Pengguna dapat memilih untuk bergabung dengan satu atau lebih jaringan yang tersedia, seperti berdasarkan sekolah tingkat atas, tempat kerja, atau wilayah geografis.

Hingga Juli 2007, situs ini memiliki jumlah pengguna terdaftar paling besar di antara situs-situs yang berfokus pada sekolah dengan lebih dari 34 juta anggota aktif yang dimilikinya dari seluruh dunia. Dari September 2006 hingga September 2007, peringkatnya naik dari posisi ke-60 ke posisi ke-7 situs paling banyak dikunjungi, dan merupakan situs nomor satu untuk foto di Amerika Serika, mengungguli situs publik lain seperti Flickr, dengan 8,5 juta foto dimuat setiap harinya.

Sekarang Facebook merupakan situs peringkat no satu yang di cari orang indonesia di google *search* dengan kata kunci facebook, *login* facebook, cara membuat facebook, dan semua kata yang ada facebook nya.