BAB II

TINJAUAN PUSTAKA

2.1 Profil Tempat Kerja Praktek

2.1.1 Sekilas Lembaga Penelitian dan Pengabdian Institut Teknologi Bandung (ITB)

Institut Teknologi Bandung yang sejak tahun 2001 berstatus Badan Hukum Milik Negara (ITB-BHMN) merupakan lembaga pendidikan tinggi dan pusat ilmu pengetahuan, teknologi, dan seni (IPTEKS) yang bertumpu pada penelitian dan berorientasi pada pelayanan kepada masyarakat. Pengembangan IPTEKS yang berkesinambungan dilakukan melalui upaya-upaya yang gigih guna meningkatkan kualitas ITB sebagai lembaga pendidikan terdepan dan paling bergengsi di Indonesia.

LPPM (Lembaga Penelitian dan Pengabdian kepada Masyarakat) yang terbentuk pada akhir tahun 2001 merupakan penggabungan Lembaga Penelitian ITB yang berdiri sejak 1959 dengan Lembaga Pengabdian Masyarakat ITB. Lembaga ini merupakan wadah bagi sivitas akademika untuk melaksanakan penelitian dan pengabdian kepada masyarakat.

Kemitraan ITB difungsikan sebagai direktorat pada awal tahun 2002 dan pada tahun 2005 dilebur ke kantor LPPM. Fungsi dari Kemitraan ITB adalah mengembangkan jejaring (*networking*) serta menjadi jembatan antara kepakaran ITB dengan kebutuhan dari luar (institusi pendidikan, pemerintahan, industri, baik di dalam maupun di luar negeri).

LPPM merupakan strategi ITB dalam upaya mengatasi ketidakseimbangan antara kemampuan IPTEKS di pergutuan tinggi dan peluang yang ada di masyarakat, dan sekaligus mewadahi pelaksanaan tugas Tri Dharma yang meliputi pendidikan, penelitian, dan pengabdian kepada masyarakat secara seimbang.

Kerjasama LPPM-ITB di bidang penelitian dan pengabdian kepada masyarakat dalam bentuk pelatihan, pendampingan, dan konsultansi telah terjalin baik dengan berbagai mitra lembaga penelitian, pemerintah, industri, swasta, BUMN, dan lembaga-lembaga kemasyarakatan lainnya dari dalam negeri maupun luar negeri. Berlangsungnya kerjasama tersebut merupakan perwujudan dari tingginya kepercayaan masyarakat sebagai mitra kerja terhadap kemampuan LPPM-ITB. Melalui kesempatan ini disampaikan terima kasih sebesar-besarnya kepada semua mitra kerja karena berkat adanya kerjasama telah terjadi saling memberdayakan menuju kesejahteraan umat manusia melalui pengembangan IPTEKS berdasarkan pada pencapaian peningkatan kecendekiaan bangsa termasuk komunitas ITB yang meliputi kecendekiaan penemuan, integrasi, terapan, dan pengajaran.

2.1.2 Visi dan Misi

1. Visi

   Menjadi Perguruan Tinggi yang unggul, bermartabat, mandiri, dan diakui dunia serta memandu perubahan yang mampu meningkatkan kesejahteraan bangsa Indonesia dan dunia.[9]

2. Misi

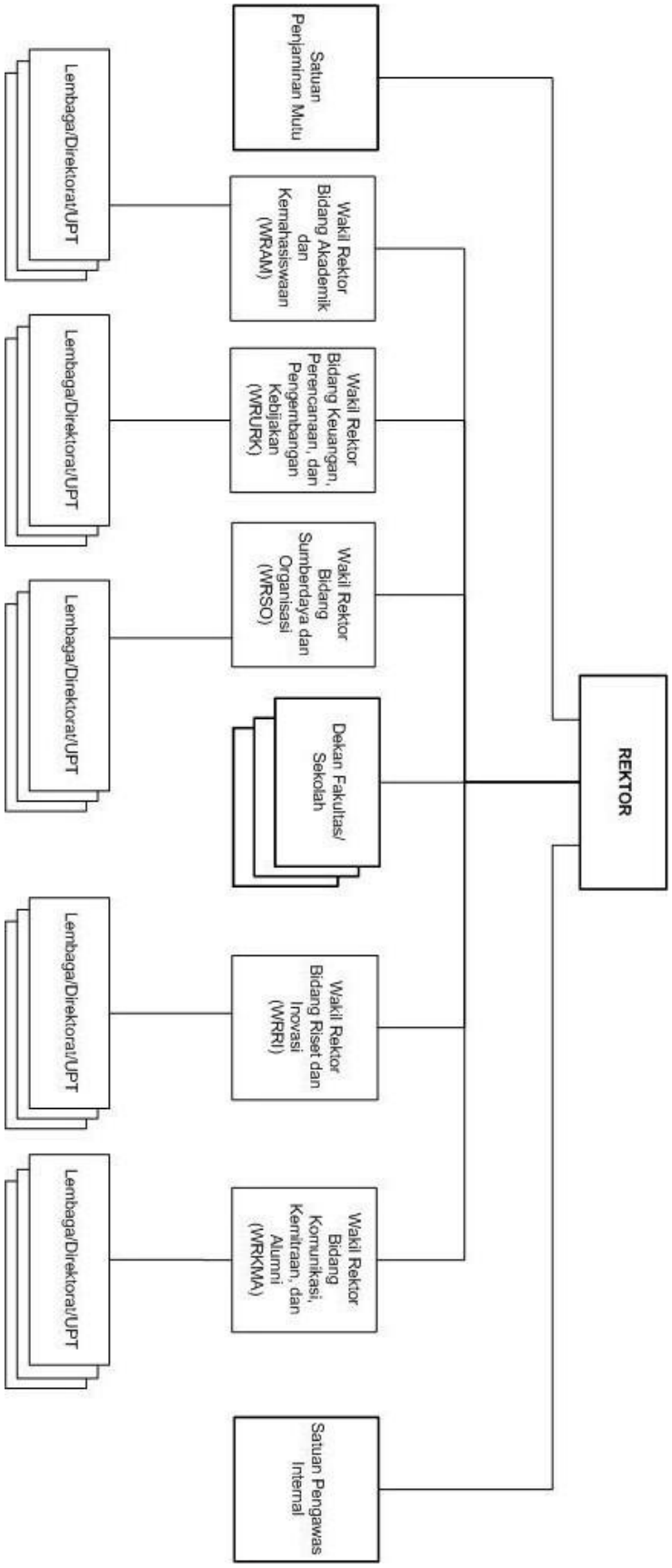
   Menciptakan, berbagi dan menerapkan ilmu pengetahuan, teknologi, seni dan kemanusiaan serta menghasilkan sumber daya insani yang unggul untuk menjadikan Indonesia dan dunia lebih baik.[9]

2.1.3 Logo



Gambar 2.1 Logo Institut Teknologi Bandung (ITB)

2.1.4 Struktur Satuan Organisasi



Gambar 2.2 Struktur Sastuan Organisasi ITB

2.1.5 Jenis Pelayanan

1. Penelitian

   Melakukan penelitian-penelitian guna kepentingan pengembangan keilmuan dan penerapan langsung kepada masyarakat.

2. Pendidikan Dan Pelatihan

   Melakukan pelatihan dan kursus untuk meningkatkan kualitas sumberdaya manusia dalam rangka menumbuhkan motivasi mandiri serta memacu pembangunan masyarakat.

3. Jasa Konsultasi

   Melakukan kegiatan-kegiatan konsultasi yang meliputi; studi, survey dan investigasi, perencanaan dan perancangan/desain, pelaksanaan fisik dan jasa, manajemen dan supervisi proyek.

4. Manajemen Haki

   Memberikan layanan konsultasi dalam pengelolaan asset intelektual; menyelenggarakan pelatihan untuk eksekutif perusahaan dalam bidang pengelolaan asset intelektual; dan memberi layanan yang menyeluruh dalam menangani pengelolaan asset intelektual yang dimiliki oleh suatu organisasi.

5. Kemitraan

   Mengembangkan jejaring (*networking*) serta menjadi jembatan antara kepakaran ITB dengan kebutuhan dari luar (institusi pendidikan, pemerintahan, industri, baik di dalam maupun di luar negeri).

2.1.6 Pusat Rekayasa Industri (ITB)

Pusat Rekayasa Industri Adalah bagian dari Institut Teknologi Bandung (ITB) yang menyediakan Jenis Layanan berupa Penelitian, Pendidikan dan Pelatihan, Jasa Konsultasi, Managemen HAKI dan Kemitraan yang bergerak di bidang Ilmu Rekayasa.

Pusat Rekayasa Industri di Institut Teknologi Bandung (ITB), yang saat ini memiliki anggota staf lebih dari 50 dengan pekerjaan gelar

Doktor di 6 laboratorium yang langka. Mengembangkan lingkungan ilmiah untuk anggota staf dari universitas lain atau lembaga penelitian melalui program Master dan Doktor. Sumber daya manusia yang berkualitas, sebagai keluaran dari Pusat Rekayasa Industri, menjadi agen pembangunan dan akan bertindak sebagai mengalikan faktor dalam pengembangan rekayasa.

Pusat Rekayasa Industri ITB yang didirikan pada tahun 1985 adalah salah satu dari beberapa orang lain antar-universitas pusat penelitian di Indonesia. Itu ditugaskan oleh DIKTI (Direktorat Jenderal Pendidikan Tinggi) untuk streghten pendidikan teknik di Indonesia dengan mendukung pendidikan pascasarjana di ITB.

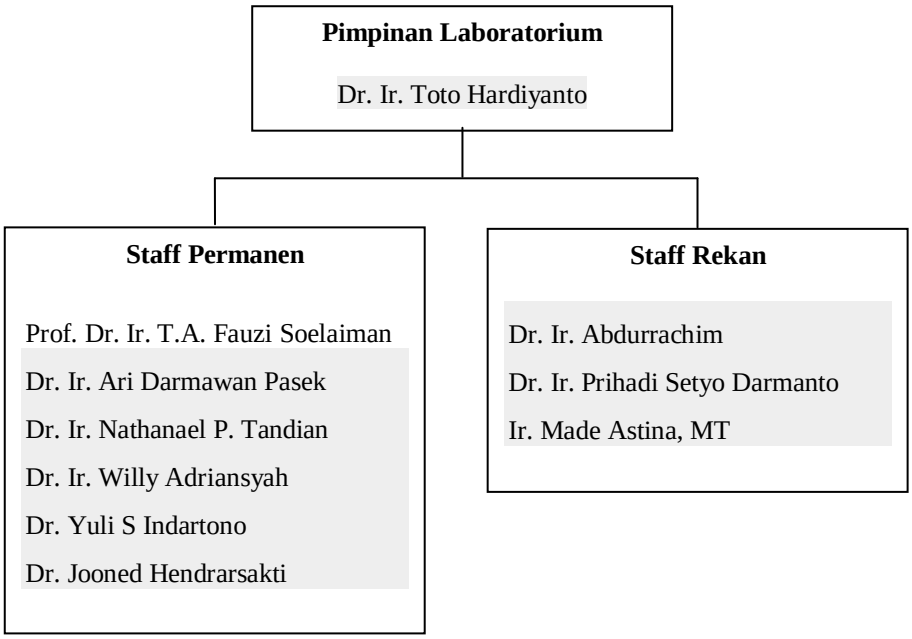
Pusat Rekayasa Industri ITB memiliki 6 laboratorium yang melaksanakan misi di 6 disiplin utama ilmu teknik. Laboratorium tersebut adalah : Mekanika Fluida & Hidrodinamika, Struktur Ringan & Aerodinamika, Termodinamika, Dinamika, Mekanika Struktur, dan Laboratorium Rekayasa Geoteknik.

Gambar 2.3 Logo Pusat Rekayasa Industri ITB

2.6.1.1 Laboratorium Termodinamika

Sebagai salah satu Laboratorium di Universitas Riset Inter Pusat Ilmu Rekayasa (PPAU-IR) ITB, Laboratorium Termodinamika mendasarkan program-programnya pada penilaian ilmiah dan pengembangan teknologi konversi energi termal dan sistem proses termal.

Saat ini, Laboratorium Termodinamika didukung oleh 8 staf permanen dan 3 rekan staf di bidang yang berbeda.



Gambar 2.4 Struktur Organisasi Laboratorium Termodinamika

2.6.1.2 Alamat

Gedung Pusat Antar Universitas (PAU) Lantai 3 ITB

Jl. Ganesha 10 Bandung 40132

Fax : 62-22- 2512971 / 72

E-mail : pri@pri.itb.ac.id

2.2 Landasan Teori

Dalam sub bab ini akan di bahas mengenai teori-teori yang mendukung pembangunan aplikasi simulasi pemantau suhu.

2.2.1 Konsep Sistem

2.2.1.1 Definisi Sistem

Sistem adalah komponen-komponen atau subsistem-subsistem yang saling berinteraksi, dimana masing-masing bagian tersebut dapat bekerja secara sendiri sendiri (independen) atau bersama-sama serta saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut dapat tercapai secara keseluruhan.

Menurut Jogianto Hartono (1999) "Sistem adalah suatu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang berinteraksi untuk mencapai suatu tujuan."

Sistem dapat didefinisikan dengan dua pendekatan yaitu sistem yang menekan pada prosedur dan sistem yang menekan pada elemen atau komponennya.

"Suatu sistem adalah jaringan kerja pada prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu." (Jogiyanto Hartono : 1999).

Sedangkan pendekatan sistem yang menekan pada elemen atau komponen komponen adalah sebagai berikut : "Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu." (Jogiyanto Hartono : 1999).

2.2.1.2 Karakteristik Sistem

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu memiliki komponen, batas, lingkungan luar sistem, penghubung, masukan, keluar*an*, pengolah, sasaran dan tujuan.

1. Komponen Sistem (*System Components*)

   Komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem

baik besar maupun kecil, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu yang lebih besar yang disebut *supra system.*

2. Batas Sistem (*System Boundary*)

   Batas sistem merupakan daerah-daerah yang membatasi antara satu sistem dengan sistem lainnya dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menujukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*System Environment*)

   Lingkungan luar sistem dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem yang dapat bersifat menguntungkan dan dapat pula merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan yang merugikan harus ditahan dan dikendalikan, karena akan mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem *(System Interprest)*

   Penghubung sistem merupakan media penghubung antara satu sub sistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumbersumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung akan terjadi interaksi antar subsistem, sehingga membentuk satu kesatuan.

5. Masukan Sistem *(System Input)*

   Masukan adalah suatu energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Contoh *maintenance input* di dalam sistem komputer adalah program, yang digunakan untuk mengoperasikan

komputer. Sedangkan *signal input* adalah energi yang diproses untuk mendapatkan keluaran. Contoh *signal input* di dalam sistem komputer adalah data, yang dapat diolah menjadi Informasi.

6. Keluaran Sistem *(*System *Output)*

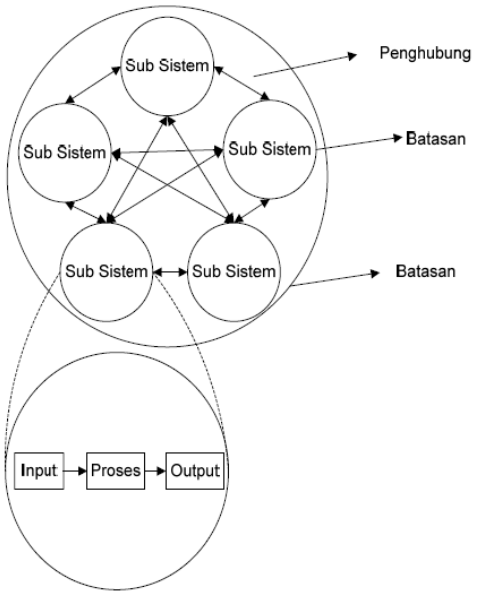
   Keluaran (*Output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolah Sistem (System *Process*)

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya, yang bertugas untuk merubah masukan menjadi keluaran.

8. Sasaran Sistem (*System Objective*)

   Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Suatu operasi sistem akan berguna dan berhasil apabila mencapai sasaran atau tujuannya. Sasaran sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.



Gambar 2.5 Karakteristik Sebuah Sistem

2.2.2 Konsep Data Dan Informasi

2.2.2.1 Definisi Data

Data adalah kumpulan dari fakta-fakta, kejadian-kejadian yang dapat berupa simbol, angka, huruf, dan lain-lain yang berguna bagi suatu pengolahan data (process) atau sebagai masukan (*input*) bagi suatu proses.

2.2.2.2 Definisi Informasi

Informasi merupakan hasil suatu proses. Proses itu terdiri dari kegiatan-kegiatan mulai dari pengumpulan data, menyusun serta menghubungkannya, meringkas, mengambil inti sarinya, dan mengimplementasikannya sesuai dengan presepsi sistem informasi penerima agar menjadi sebuah informasi yang berguna. Informasi adalah sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian *(event)* yang nyata *(fact)* yang digunakan untuk mengambil keputusan. Kualitas suatu informasi tergantung dari hal berikut :

1. Relevan *(relevancy),* Berarti informasi harus memberikan manfaat bagi pemakainya.
2. Akurat *(accuracy),* Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan dan harus jelas mencerminkan maksudnya.
3. Tepat waktu *(timeliness),* Informasi yang dihasilkan / dibutuhkan tidak boleh terlambat *(usang)*.
4. Ekonomis *(economy),* Biaya sistem informasi akan mengalami kenaikan sesuai dengan berjalannya waktu.
5. Efisien *(efficiency),* Adalah berapa banyak produksi meningkat karena tambahan unit sumber daya dalam proses produksinya.
6. Dapat dipercaya *(reliability),* Sebuah indikator penting dari sistem informasi adalah dengan memperhatikan masalah

realibilitasnya.

Nilai Informasi ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Pengukuran nilai informasi biasanya dihubungkan dengan analisis cost effectiveness atau cost benefit.

2.2.2.3 Definisi Sistem Informasi

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Pengertian lain dari sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan dan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan informasi

Sistem informasi dalam sebuah sistem meliputi pemasukan data (*input*) kemudian diolah melalui suatu model dalam pemrosesan data, dan hasil informasi akan ditangkap kembali sebagai suatu input dan seterusnya sehingga membentuk siklus informasi yang dapat diperoleh dari sistem informasi sebagai sistem khusus dalam organisasi untuk mengolah informasi tersebut.

2.2.2.3.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building blok*) yaitu (Bently : 2004):

1. Hardware yaitu suatu perangkat keras komputer yang dapat kita sentuh dan rasakan.
2. Software yaitu suatu perangkat lunak komputer yang

berfungsi untuk mengoprasikan suatu aplikasi.

3. Data yaitu sekumpulan karakter yang diterima sebagai masukan () untuk sistem informasi dan disimpan serta diolah.
4. Prosedur yaitu langkah-langkah penggunaan sistem.
5. Personil untuk pengelolaan operasi (SDM), meliputi :
   a. Clerical personnel (untuk menangani transaksi dan pemrosesan data dan melakukan inquiry = operator).
   b. First level manager : untuk mengelola pemrosesan data didukung dengan perencanaan, penjadwalan, identifikasi situasi out-of-control dan pengambilan keputusan level menengah ke bawah.
   c. Staff specialist : digunakan untuk analisis untuk perencanaan dan pelaporan.
   d. Management : untuk pembuatan laporan berkala, permintaan khsus, analisis khusus, laporan khsusus, pendukung identifikasi masalah dan peluang.

2.2.2.3.2 Tujuan Sistem Informasi

Sistem Informasi memiliki beberapa tujuan, yaitu:

1. Integrasi Sistem
   a. Menghubungkan sistem individu/kelompok
   b. Pengkolektifan data dan penyambungan secara otomatis
   c. Peningkatan koordinasi dan pencapaian sinergi
2. Efisiensi Pengelolaan
   a. Penggunaan basis data dalam upaya kesamaan pengadministrasian data
   b. Pengelolaan data berkaitan dengan karakteristik Informasi
   c. Penggunaan dan pengambilan Informasi
3. Dukungan keputusan untuk manajemen
   a. Melengkapi Informasi guna kebutuhan proses pengambilan kebutuhan

b. Akuisisi Informasi eksternal melalui jaringan komunikasi
c. Ekstraksi dari Informasi internal yang terpadu.

2.2.2.3.3 Manfaat Sistem Informasi

Sistem Informasi Memiliki beberapa manfaat, yaitu:

1. Menghemat tenaga kerja
2. Peningkatan efisiensi
3. Mempercepat proses
4. Perbaikan dokumentasi
5. Pencapaian standar
6. Perbaikan keputusan

2.2.3 Konsep Perancangan Sistem

Penggambaran dan analisis model sistem yang digunakan untuk menggambarkan pembangunan perangkat lunak adalah analisis terstruktur. Analisis terstruktur adalah aktivitas pembangunan model dengan menggunakan notasi yang sesuai dengan prinsip analisis operasional. Analisis terstruktur bukan merupakan metode tunggal yang diaplikasikan secara konsisten oleh semua yang menggunakannya. Pada model ini analisis terhadap perangkat lunak dapat digambarkan dalam bentuk *Flow Map*, Diagram konteks, *Data Flow Diagram* (DFD) dan Kamus Data.

2.2.3.1 *Flowmap*

Flowmap adalah representasi grafik dari sistem informasi, proses-proses, aliran-aliran data logis, masukan, keluaran, dan *file-file* serta entitas-entitas sistem operasi yang berhubungan dengan sistem informasi [4].

2.2.3.2 Digram Konteks

Diagram Konteks adalah bagian dari DFD yang berfungsi memetakan model model lingkungan, yang dipresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem. Diagram ini adalah diagram *level* tertinggi dari DFD. Diagram konteks menyoroti sejumlah karateristik penting sistem, yaitu :

1. Kelompok Pemakai, organisasi atau sistem lain dimana sistem melakukan komunikasi (sebagai terminator).
2. Data Masuk, yaitu data yang diterima sistem dari lingkungan dan harus diproses dengan cara tertentu.
3. Data Keluar, yaitu data yang dihasilkan sistem dan diberikan kedunia luar.
4. Penyimpanan Data (*storage*), yaitu digunakan secara bersama antara sistem dengan terminator. Data ini dapat dibuat oleh sistem dan digunakan oleh lingkungan atau sebaliknya dibuat oleh lingkungan dan digunakan oleh sistem. Hal ini berarti pembuatan simbol *data storage* dalam diagram konteks dibenarkan, dengan syarat simbol tersebut merupakan bagian dari luar sistem.
5. Batasan, yaitu batasan antara sistem dan lingkungan.

2.2.3.3 *Data Flow Diagram* (DFD)

DFD atau singkatan dari *Data Flow* Diagram merupakan representasi grafik dari suatu sistem yang menunjukan proses atau fungsi, aliran data, tempat penyimpan data dan *entitas eksternal*. DFD juga digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan. Dengan menggunakan DFD, rancangan yang akan kita buat akan lebih terarah dan lebih rinci. Sehingga kita tidak akan mengalami kesulitan dalam melakukan perancangan. *Data flow diagram* memiliki empat komponen, antara lain akan dijelaskan dibawah ini [5] :

1. *Terminator (external exitity)*

   Terminator mewakili *entity external* yang berkomunikasi dengan sistem yang sedang dikembangkan. *Terminator* merupakan kesatuan dilingkungan sistem. Biasanya terminator ini dikenal dengan nama entitas (*external*) sumber atau tujuan (*source* dan *sink*). Terminator dapat juga berupa departemen, divisi atau sistem diluar sistem yang berkomunikasi dengan sistem yang dikembangkan.

2. Proses

   Proses sering dikenal dengan nama *Bubble*, fungsi atau informasi. Komponen proses menggambarkan bagian dari sistem yang mentransformasikan *input* ke *output*, atau dapat dikatakan bahwa komponen proses menggambarkan transformasi satu *input* atau lebih menjadi *output.* Setiap proses harus diberikan penjelasan yang lengkap sebagai berikut :

   a. Identifikasi Proses

      Umumnya berupa angka yang menunjukan nomor dari proses atau ditulis pada bagian atas simbol proses.

   b. Nama Proses

      Menunjukan apa yang dikerjakan oleh proses tersebut. Nama proses harus jelas dan lengkap menggambarkan bagian prosesnya nama proses diletakan dibawah identifikasi proses.

   c. Penyimpanan data *(data store)*

      Data store digunakan sebagai sarana untuk mengumpulkan data. *Data store* disimbolkan dengan dua garis horizontal yang paralel dimana tertutup pada salah satu ujungnya atau dua garis horizontal saja. *Data store* ini biasanya berkaitan dengan penyimpanan-pemyimpanan seperti *file* atau *database* yang berkaitan dengan penyimpanan secara komputerisasi, contohnya *file* pita magnetic, *file* disket atau *file harddisk*. *Data store* juga berkaitan dengan pemyimpanan data.

   d. Alur *Data (Data Flow)*

Suatu data *flow* atau alur data dapat dipresentasikan dengan anak panah yang menunjukan arah menuju ke dan keluar dari suatu proses. Alur data ini digunakan untuk menerangkan perpindahan data atau satu paket data atau informasi dari suatu bagian sistem ke bagian lainnya. Selain menunjukan arah, alur data pada model yang dibuat dapat merepresentasikan bit, karakter, pesan, formulir, bilangan *real* dan macam-macam informasi yang berkaitan dengan komputer.

2.2.3.4 *Kamus* Data

Kamus data adalah suatu daftar atau kamus dari seluruh elemen-elemen data yang diperlukan oleh suatu sistem. Kamus data dibuat berdasarkan arus data yang terdapat pada DFD. Arus data dalam DFD bersifat global sehingga hanya dapat ditunjukkan nama arus datanya saja dan keterangan lebih lanjut mengenai struktur arus data dapat dilihat dari kamus data [4].

Secara umum kamus data diklasifikasikan manjadi dua yaitu :

1. Kamus data elementer yaitu daftar tentang semua elemen data yang berhubungan dengan sistem sehingga data yang mengalir dapat didefinisikan dan dapat tersimpan secara lengkap.
2. Kamus Data Komposit, yaitu daftar tentang semua elemen data yang berhubungan dengan system dimana elemen data komponen ini terdiri dari dua elemen data elementer yang saling berkaitan**.**

2.2.4 Konsep Basis Data

2.2.4.1 Definisi Basis Data

Basis data terdiri dari 2 kata, yaitu basis dan data, basis dapat diartikan sebagai maskas atau gudang tempat bersarang atau berkumpul [6]. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli dan lain-lain), barang hewan, peristiwa, konsep keadaan dan sebagainya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau

kombinasinya. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang, seperti :

1. Himpunan kelompok data (*arsip*) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu untuk memenuhi berbagai kebutuhan.
3. Kumpulan file, tabel, atau arsip yang saling berhubungan yang akan disimpan dalam media penyimpanan elektronis.

Basis data dalam lemari arsip sesungguhnya memiliki prinsip kerja dan tujuan yang sama. Prinsip utamanya adalah pengeluaran data dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengembalian data. Perbedaannya hanya terletak pada media penyimpanan yang digunakan. Maka basis data menggunakan media penyimpanan elektronik seperti disk (disket atau *hardisk*). Hal ini karena lemari arsip langsung dikelola oleh manusia sementara basis data dikelola melalui perantara alat atau mesin pintar elektronis (komputer).

Satu hal yang harus diperhatikan, bahwa basis data bukan hanya sekedar tidak menyimpan data secara elektronis (dengan bantuan komputer), artinya tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data. Kita dapat menyimpan dokumen berisi data dalam file teks (dengan program pengolah data), *file spread* sheet dan lain-lain. Tetapi tidak bisa disebut basis data karena didalamnya tidak ada pemilahan dan pengelompokan data sesuai jenis dan fungsi data. Yang sangat ditonjolkan basis data adalah pengaturan, pengelompokan, pengorganisasian data yang akan kita simpan sesuai dengan fungsinya.

2.2.4.2 Operasi Dasar Basis Data

Didalam sebuah disk, basis data dapat diciptakan dan dapat pula ditiadakan. Sementara dalam sebuah basis data, kita dapat menempatkan satu atau lebih file ataupun table. Setiap basis data pada umumnya dibuat untuk mewakili sebuah basis data sebuah semesta data yang spesifik.

Operasi-operasi yang dapat kita lakukan berkenaan dengan basis data dapat meliputi :

1. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus dengan isinya, jika ada).
3. Pembuatan file atau tabel dari suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan file atau tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
5. Penambahan atau pengisian data baru kesebuah file atau tabel disebuah basis data (*insert*) yang identik dengan penambahan lembaran arsip kesebuah map arsip.
6. Penghapusan data dari sebuah table disebuah basis data (*drop)*
7. Pengambilan data dari sebuah file atau tabel (*retrieve atau search*), yang identik dengan pencarian lembaran arsip yang ada disebuah map arsip.
8. Pengubahan data dari sebuah file atau tabel (*update*).

2.2.4.3 *Database* Management *System* (DBMS)

Berikut ini akan dikemukakan definisi dari *Database Management System* (*DBMS*) :

"Manajemen Sistem Basis Data (Database Management System / DBMS) adalah perangkat lunak yang didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar". [1]

"DBMS diartikan sebagai suatu program komputer yang digunakan untuk memasukkan, mengubah, mengapus, memanipulasi dan memperoleh data/informasi dengan praktis dan efisien". [3]

DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, semisal penyimpanan data dalam file dan menulis kode aplikasi yang spesifik untuk pengaturannya.

Penggunaan DBMS untuk suatu aplikasi tergantung pada kemampuan dan dukungan DBMS yang beroperasi secara efisien. Sehingga agar bisa menggunakan DBMS dengan baik, perlu diketahui cara kerja dari DBMS tersebut. Pendekatan yang dilakukan untuk menggunakan DBMS secara baik meliputi : implementasi DBMS dan arsitektur secara mendetail untuk dapat memahami desain dari suatu basis data

2.2.4.4 Pemodelan Data

Pemodelan sistem memainkan peranan yang penting dalam pengembangan sistem. Pemodelan data kadang-kadang disebut pemodelan *database* karena model data kadang-kadang diimplementasikan sebagai sebuah *database*. Pemodelan data dapat digambarkan dengan ERD (*Entity Relationship Diagram*).

2.2.4.4.1 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram adalah suatu model atau teknik pendekatan yang dapat menyatakan atau menggambarkan hubungan *entity* didalam sebuah sistem, dimana hubungan tersebut dinyatakan sebagai *one-to-one*, *one-to-many* dan *many-to-many*. [4]

Entity relationship bertujuan untuk menentukan objek-objek data (*Entity*) dan hubungan (*relationship*) yang ada pada objek-objek tersebut.

Komponen *Entity Relationship Diagram* :

1. *Entity*

Entity adalah suatu yang dapat dibedakan atau diidentifikasikan secara unik, objek disini sifatnya berupa orang, tempat, peristiwa atau konsep yang informasinnya direkam.

2. *Atribut*

Atribut adalah kumpulan dari beberapa elemen data yang membentuk suatu entity. Atribut merupakan ciri sebutan atau karakteristik yang dapat mewakili suatu simbol yang digunakan untuk menggambarkan atribut pada *entity relationship diagram.*

3. *Relationship*

Relationship adalah relasi atau hubungan yang terjadi diantara beberapa *entity*. Simbol yang digambarakan untuk menggambarkan *relationship* pada teknik *entity relationship diagram* adalah berupa belah ketupat.

Setiap *relationship* mempunyai derajat *relationship*, yaitu tingkat atau hubungan yang terdiri dari tiga derajat, yaitu :

a. *Unary* (derajat 1)

   Suatu *relationship* yang dihubungkan dengan suatu *entity* set, dimana penghubungnya ada dua.

b. *Binary* (derajat 2)

   Relationship derajat dua adalah *relationship* yang biasa terjadi yaitu dua *entity* set dihubungkan dengan satu *entity relationship*.

c. *Ternary* (derajat 3)

   Relationship derajat tiga.

Relationship diagram juga mempunyai *Cardinalitas* yaitu batasan pada jumlah *entity* yang terdiri dari:

a. Relasi satu ke satu (*one-to-one*)

   Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa satu berbanding satu.

b. Relasi satu ke banyak (*one-to-many*)

Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak atau banyak berbanding satu.

c. Relasi Banyak ke Banyak (*many-to-many*)
Adalah suatu relasi yang terjadi dengan batasan relasi yang berupa banyak berbanding banyak.

2.2.5 Normalisasi

Normalisasi adalah suatu teknik dalam menstruktur data dalam cara-cara tertentu untuk mengurangai atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam *database*. Normalisasi juga diartikan sebagai proses pengelompokan data elemen menjadi tabel-tabel yang menunjukan entitas dan relasinya.

Konsep-konsep pada normalisasi, antara lain :

1. Kunci atribut (*Key field* atau *Key attribut*) yautu suatu kunci *field* yang mewakili *record* atau *tupple*
2. Kunci kandidat (*Candidate key*) yaitu suatu atribut atau satu set atribut yang mengidentifikasi secara unik suatu *entity.*
3. Kunci primer (*Primary key*) yaitu satu atribut atau satu set atribut yang yang mengidentifikasi secara unik dan mewakili setiap kejadian pada satu *entity*.
4. Kunci alternatif (*Alternate key*) yaitu kunci kandidat yang dipakai sebagai kunci primer.
5. Kunci tamu (Foreign *key*) yaitu suatu atribut atau satu set atribut dan melengkapi hubungan yang menunjukan ke induknya.

Bentuk-bentuk Normalisasi, antara lain :

1. Normal satu (1NF atau *First Normal Form*)
   Relasi berada pada normal kesatu jika semua atribut mempunyai nilai yang bersifat *atomic*.
2. Normal kedua (2NF atau *Second Normal Form*)

Relasi berada pada normal kedua jika relasi tersebut merupakan normal satu dan atribut bukan merupakan kunci tergantung penuh pada kunci primer.

3. Normal ketiga (3NF atau *Three Normal Form*)

   Relasi berada pada normal ketiga jika relasi tersebut merupakan normal kedua dan atribut bukan kunci tidak tergantung secara transitif pada kunci primer.

4. BCNF (Boyce *Cood Normal Form*)

   Relasi berada pada BCNF jika dan hanya jika faktor penentunya adalah kunci kandidat dan relasi tersebut merupakan normal ketiga.

2.2.6 Internet dan *World Wide Web* (www)

2.2.6.1 Definisi Internet

Istilah Internet berasal dari bahasa latin, *inter*, yang berarti ”*antara*”. Secara kata per kata Internet berarti jaringan antara atau penghubung. Memang itulah fungsinya, Internet menghubungkan berbagai jaringan yang tidak saling bergantung pada satu sama lain dengan sedemikian rupa, sehingga mereka dapat berkomunikasi.

Internet adalah jaringan komputer yang saling terhubung ke seluruh dunia tanpa mengenal batas teritorial, hukum dan budaya. Secara fisik dianalogikan sebagai jaring laba-laba (*The Web*) yang menyelimuti bola dunia dan terdiri dari titik-titik (*node*) yang saling berhubungan [6].

Node bisa berupa komputer, jaringan lokal atau peralatan komunikasi, sedangkan garis penghubung antar simpul disebut tulang punggung (*back-bone*) yaitu media *terestrial* (kabel, serat optic, microwave, radio link) maupun satelit. *Node* terdiri dari pusat informasi dan database, peralatan komputer dan perangkat interkoneksi jaringan serta peralatan yang dipakai pengguna untuk mencari, menempatkan dan bertukar informasi di Internet.

Walaupun secara fisik Internet adalah interkoneksi antar jaringan komputer namun secara umum Internet harus dipandang sebagai sumber informasi [6]. Isi Internet adalah informasi, dapat dibayangkan sebagai

suatu database atau perpustakaan multimedia yang sangat besar dan lengkap. Bahkan Internet dipandang sebagai dunia dalam bentuk lain (maya) karena hamper seluruh aspek kehidupan di dunia nyata ada di Internet seperti bisnis, hiburan, olah raga, politik dan lain sebagainya.

Salah satu isu terpenting salah satu isu terpenting di Internet adalah penerapan *standar* komputerisasi terbuka (*open computing standart*). Karena *internetworking* dan Internet mengintegrasikan semua sistem, jenis dan tipe komputer yang ada di dunia, maka harus ada *standar* yang menjamin komputer dapat saling berbicara satu sama lain dalam bahasa yang sama. Menurut Drew Heywood (1966): *standar* bahasa komputer universal telah dikembangkan sejak 1969, terdiri dari serangkaian protocol komunikasi disebut *Transfer Control Protocol (TCP)* yang bertugas mengendalikan transmisi paket data, koreksi kesalahan dan kompresi data dan Internet Protocol (IP) yang bertugas sebagai pengenal (*identifier*) dan pengantar paket data ke alamat yang dituju.

Protokol TCP / IP menyatukan bahasa dan kode berbagai komputer di dunia, sehingga menjadi *standar* utama jaringan komputer. TCP / IP berkembang cepat dan kaya fasilitas karena bersifat terbuka dan bebas digunakan, oleh siapa pun. Oleh karena itu, hingga saat ini jaringan-jaringan tersebut saling terhubung di seluruh belahan dunia.

Seluruh jaringan yang tersebar di setiap penjuru dunia sudah terkondisikan dengan internet. Lalu lintas data Internet ke seluruh Amerika dan penjuru dunia saat ini disebarkan oleh jaringan pusat yang disebut VBNS (*Very High Speed Backbone Network Service*). Adapun seluruh biaya yang dikeluarkan di biayai oleh badal Ilmu Pengetahuan Nasional Amerika Serikat. Dari jaringa itu, komputer-komputer yang terhubung dengan internet akan berkomunikasi dan saling mengirimkan data sehingga internet dapat dinikmati oleh seluruh umat manusia untuk berkomunikasi melalui komuputer baik di rumah maupun di kantor.

Dengan demikian, internet dapat didefinisikan dengan hubungan berbagai jenis komputer dan jaringan di dunia, yang berbeda sistem operasi maupun aplikasinya di mana hubungan tersebut memanfaatkan

kemajuan media komunikasi (telepon dan satelit) yang menggunakan protocol *standar* dalam berkomunikasi yaitu protocol TCP / IP. Internet berfungsi sebagai media komunikasi dan informasi modern.

2.2.6.2 Definisi *World Wide Web* (www)

World Wide Web adalah fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen *web* disebut *web page* dan *link* dalam *web* membuat *user* bisa pindah dari satu *page* ke *page* lain (*hyper text*), baik antar *page* yang disimpan dalam *server* yang sama atau *server* di seluruh dunia. *Pages* diakses dan dibaca melalui *web browser* seperti *netscape navigator* atau *internet explorer.*

World Wide Web sering disingkat WWW atau *web* saja, yaitu sebuah sistem dimana informasi dapat kita nikmati dalam bentuk teks, gambar, suara, dan lain-lain serta ikut dapat diakses oleh perangkat lunak yang disebut *browser*. Informasi di *web* pada umumnya ditulis dalam bentuk HTML. Informasi lainnya disajikan dalam bentuk grafis (dalam format GIF, JPG, PNG), suara (dalam bentuk format AU, WAV), dan objek multimedia lainnya (seperti MIDI, Shockwave, Quicktime Movie, 3D World). WWW dijalankan dalam *server* yang disebut HTTPD.

2.2.6.3 *Universal Resource Locator* (URL)

URL adalah konsep nama file standar yang diperluas dengan jaringannya. Nama file ini tidak hanya menunjukan direktori dan nama filenya saja, tetapi juga nama mesinnya dalam jaringan. URL dapat disediakan (ada atau diakses) dengan berbagai metode dan bisa jadi bukan sekedar file. Karena URL dapat menunjukan queri-queri dokumen dapat disimpan dalam database, hasil dari suatu perintah finger atau achie atau apapun yang berkaitan dengan data hasil proses.

2.2.6.4 *Domain Name Service* (DNS)

DNS digunakan untuk mempermudah penggunaan internet, dengan memetakan IP *address* ke nama host. Agar data nama host dapat

didistribusikan di banyak server, format data yang digunakan harus mencerminkan terdistribusinya data tersebut. Untuk itu, digunakan format tree dengan masing-masing nodenya disebut domain. Penulisan nama host secara lengkap disebut sebagai *Full Qualifield Domain Name (FQDN).*

DNS bekerja secara hirarki dan berbentuk seperti pohon (tree). Bagian atas adalah *Top Level Domain* (TLD) seperti COM, ORG, EDU, MIL dsb. Seperti pohon DNS mempunyai cabang-cabang yang dicari dari pangkal sampai ke ujung. Pada waktu kita mencari alamat misalnya linux.or.id pertama-tama DNS bertanya pada TLD server tentang DNS Server yang melayani domain .id misalnya dijawab ns1.id, setelah itu dia bertanya pada ns1.id tentang DNS Server yang bertanggung jawab atas .or.id misalnya ns.or.id kemudian dia bertanya pada ns.or.id tentang linux.or.id dan dijawab 64.29.24.175

Sedangkan untuk mengubah IP menjadi nama *host* melibatkan domain in-addr.arpa. Seperti domain lainnya domain in-addr.arpa pun bercabang-cabang. Yang penting diingat adalah alamat IP-nya ditulis dalam urutan terbalik di bawah in-addr.arpa. Misalnya untuk alamat IP 64.29.24.275 prosesnya seperti contoh linux.or.id: cari server untuk arpa, cari server untuk in-addr.arpa, cari server untuk 64.in-addr.arpa, cari server 29.64.in-addr.arpa, cari server untuk 24.29.64.in-addr.arpa. Dan cari informasi untuk 275.24.29.64.in-addr.arpa. Pembalikan urutan angkanya memang bisa membingungkan.

2.2.6.5 *Hypertext Transfer Protocol* (HTTP)

Server dan *browser web* berkomunikasi satu sama lain dengan protokol yang memang dibuat khusus untuk ini yaitu HTTP. HTTP bertugas menangani permintaan-permintaan *(request)* dan *browser* untuk mengambil dokumen-dokumen *web*. HTTP bisa dianggap sebagai sistem yang bermodel *client-server*.

2.2.7 *Hypertext Markup Language* (HTML)

Dokumen HTML adalah file teks regular (disebut juga ASCII) yang diciptakan dengan menggunakan editor teks (misalnya, Emacs dalam Unix, Gnotepad pada Linux, atau Notepad pada Windows) atau dengan menggunakan word processor, tetapi harus disimpan dengan format *"text only with line breaks".*

Sesuai dengan namanya, bahasa ini menggunakan tanda (MarkUp) untuk menandai perintah-perintahnya. Saat ini banyak sekali aplikasi-aplikasi yang dapat digunakan untuk membuat web page, seperti Microsof Frontpage, Adobe GoLive, Adobe Dreamweaver, Ultradev dan lain-lainnya. Namun untuk seorang web develover kemampuan dasar perintah HTML tetap dibutuhkan.

Homepage dibangun oleh sebuah file yang didalamnya terdapat kode-kode dan informasi dan dibaca oleh sebuah aplikasi yang bernama browser. Oleh browser, kode dan informasi akan disajikan sebagai sebuah halaman web yang kita kenal sebagai homepage. Kode tersebut haruslah ditulis menurut suatu aturan tertentu yang dapat dimengerti browser. Aturan tersebut dikenal dengan "bahasa pengkodean".

Untuk membuat homepage website, bahasa pengkodean yang digunakan adalah HTML. Sekalipun banyak orang menyebutnya sebagai salah satau bahasa pemrograman, HTML sebenarnya sama sekali bukan bahasa pemrograman karena HTML merupakan bahasa Mark Up (pengkodean) yang digunakan untuk menentukan format atau style dan teks yang ditandai. Simbol Mark Up yang digunakan oleh HTML ditandai dengan tanda lebih kecil (<) dan lebih besar (>),keduanya dinamai tag. Sebagai contoh akan ditampilkan teks yang tercetak tebal, maka mark up yang digunakan adalah sebagai berikut:

```
<b> Teks ini bercetak tebal </b>
```

Untuk menandai bahwa sebuah file teks merupakan file HTML, maka ciri yang terlihat jelas adalah ekstensi filenya yaitu .htm atau .html. Namun

labih jauh dari pada itu didalam file tersebut harus mengandung struktur sebagai berikut:

```
<HTML>
…….
…….
</HTML>
```

Tag **<HTML>** harus diletakan pada bagian awal dan tag **</HTML>** harus diletakan pada bagian terakhir dari suatu file HTML. Tag-tag HTML tidak bersifat case sensive artinya penggunaan huruf besar maupun huruf kecil tidak menjadi masalah. Jadi tag **<HTML>** akan sama dengan tag **<html>.**

2.2.8 *Hypertext Preprocessor* (PHP)

PHP adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang berjalan disisi *server*. Sebagian besar perintahnya berasal dari bahasa *C, Java* dan *Perl* dengan beberapa tambahan fungsi khusus PHP.

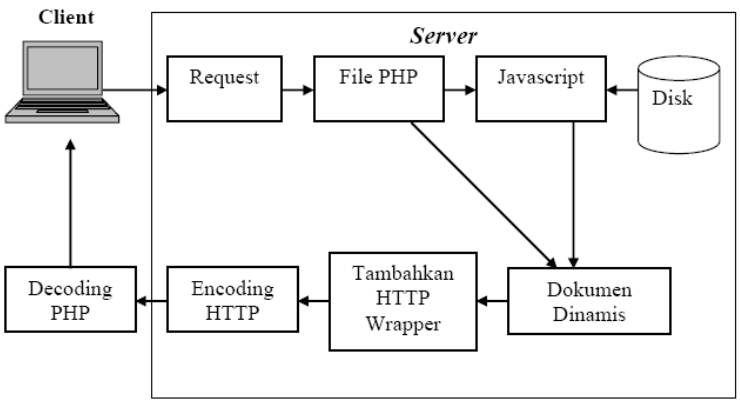
Bahasa ini memungkinkan para pembuat aplikasi web yang menyajikan HTML yang dinamis dan interaktif dengan cepat dan mudah, yang di hasilkan server. PHP biasa berinteraksi dengan hampir semua teknologi *web* yang sudah ada. *Developer* biasa menulis sebuah program PHP yang mengeksekusi suatu program CGI di *server* lain. Fleksibilitas ini amat bermanfaat bagi pemilik situs-situs web yang besar dan sibuk, karena pemilik masih menggunakan aplikasi-aplikasi yang sudah terlanjur dibuat dimasa lalu dengan menggunakan CGI, ISAP atau dengan *script* seperti *Perl*, *awk* atau *Phyton* selama proses migrasi ke aplikasi baru yang di buat dengan menggunakan PHP.

Kode program PHP menyatu dengan tag-tag HTML dalam satu file. Kode PHP diawali dengan tag <? Dan ditutup dengan tag ?>. File yang berisi tag HTML dan kode PHP ini diberi ekstensi .php atau ekstensi lainnya yang ditetapkan pada *Apache/web server*. Berdasarkan ektensi ini, pada saat file diakses, server akan tahu bahwa file ini mengandung kode PHP. Server akan menerjemahkan kode ini dan menghasilkan output dalam bentuk tag

HTML yang akan dikirim ke *browser client* yang mengakses file tersebut. Contohnya dapat dituliskan seperti ini :

```
<?php echo ( "
<html>
<head>
<title>Aplikasi Hello World </title>
</head>
<body>
<b>Hello World</b>
</body>
</html>
");  ?>
```

Dengan PHP, *developer* tidak perlu lagi berurusan dengan dua buah file yang terpisah seperti pada CGI. *Browser web* mengacu secara langsung ke file yang dituju, yang lalu dibaca oleh *server* sebagaimana file HTML statis biasa. Bedanya, sebelum dikirim balik ke *browser web, server web* memeriksa isi file dan menentukan apakah ada kode didalam file tersebut yang harus dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya akan dimasukkan kedalam dokumen yang sama. *Server web* bekerja secara langsung terhadap file yang bersangkutan tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode di eksekusi di *server* (oleh karena itu disebut *server-side script*). Lihat Gambar 2.1 dibawah ini untuk melihat kerja dari PHP dan *Apache*.



Gambar 2.6 Prinsip Kerja PHP dan *Apache*

PHP membuat proses pengembangan aplikasi menjadi mudah karena kelebihan-kelebihannya, yaitu :

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* biasa berkonsentrasi langsung pada penampilan webnya.
2. Tidak ada proses *compiling* dan *linking.*
3. Berorientasi objek *(Object Oriented).*
4. Sintaksis pemrogramannya udah dipelajari, sangat menyerupai *C* dan *Perl.*

Integrasi yang sangat luas ke berbagai *server database*. Menulis *web* yang terhubung ke *database* menjadi sangat sederhana. Database yang didukung oleh PHP : *Oralce, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePRo, Velocis, Informic, dBase, UNIX dbm.*

2.2.9 MySQL

MySQL adalah multi *user database* yang menggunakan bahasa *Strured Query Language* (SQL). MySQL merupakan database server multi user dan multi *threaded* yang tangguh. Dengan memiliki banyak feature MySQL bisa bersaing dengan datgabase komsersial sekalipun. Tidak mengejutkan MySQL menjadi database pilihan untuk banyak pengguna PHP. MySQL termasuk RDBMS (*Relational Database Management System*) itulah sebabnya istilah seperti tabel, baris dan kolom digunakan pada MySQL. Pada MySQL sebuah database mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom.

MySQL mampu menangani data yagn cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 *Gigabyte* data. SQL adalah sebuah standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industry.

Dengan menggunakan MySQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan menggunakan *dBASE* atau *Clipper* yang

masih menggunakan perintah-perintah pemrograman. MySQL merupakan *software database* yang masih yang masih popular di lingkungan Linux, kepopuleran ini sangat ditunjang oleh performa *query* dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermaslah. MySQL ini juga sudah berjalan pada lingkungan windows.

2.2.10 *Cascading Style Sheet* (CSS)

CSS adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C).

CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

CSS memungkinkan halaman yang sama untuk ditampilkan dengan cara yang berbeda untuk metode presentasi 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.

2.2.11 JavaScript

JavaScript adalah bahasa script yang digunakan pada halaman HTML. *JavaScript* adalah varian dari *java* yang sangat terkenal dalam lingkungan pemrograman *web*. *JavaScript* mempunyai keunggulan pada sisi *client*, maupun diakses lebih cepat dari *script server*. Tetapi kelemahannya pada

script ini belum tentu mampu diproses oleh *browser client* tergantung kompetibilitas *browser* tersebut. Cara untuk menjalankan j*avascript* hanya dengan *javascript-enabled* yaitu *browser* yang mampu menjalankan *javascript* seperti *Netscape Navigator* (versi 2.0 keatas) atau *internet ekplorer* (MSIE-versi 3.0 ke atas).

Beberapa *hal* mengenai JavaScript :

1. *JavaScript* didesain untuk menambah interaktif suatu *web*.
2. *JavaScript* merupakan sebuah bahasa *scripting* dengan bahasa pemrograman yang ringan.
3. *JavaScript* biasanya disisipkan (*embedded*) dalam halaman HTML.
4. *JavaScript* adalah bahasa interpreter (yang berarti skrip dieksekusi tanpa proses kompilasi.

Ada beberapa cara untuk menuliskan JavaScript yaitu :

1. *Script* pada bagian *head*

   Skrip ini dieksekusi ketika dipanggil (biasanya berbentuk *function*) atau dipanggil berdasarkan *event* tertentu. Peletakan *script* di *head* akan menjamin *script* di-*load* terlebih dahulu sebelum digunakan.

```
<html>
<head>
<script type=”text/javascript”>
Function message ()
{
    Alert (“Alert ini dipanggil saat even dijalankan”);
}
<script>
</head>
</html>
```

2. *Script* pada *Body*

   Skrip akan dieksekusi ketika halaman di-*load* sampai dibagian <body> dan ketika ditempatkan berarti antara isi dan *javacript* dijadikan satu bagian.

```
<html>
<head>
</head>
<body>
<script type=”text/javascript”>
    Document.write (“Pesan ini akan tampil”);
</script>
</body>
</html>
```

3. *External JavaScript*

Untuk memudahkan menulis *script* yang berulang-ulang di setiap halaman maka *JavaScript* dapat ditulis di *file* secara *ekternal*. Jadi, antara HTML dan *JavaScript* dipisah, kemudian *file* tersebut dipanggil dari dokumen HTML. *File JavaScript* tersebut disimpan dengan ektensi .js.

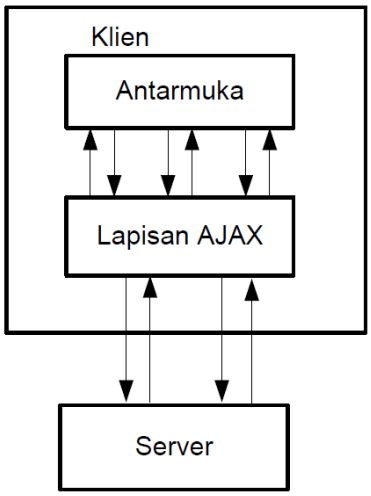
```
<html>
<head>
</head>
<body>
<script src=”test.js”>
</script>
</body>
</html>
```

2.2.12 *Asynchronous JavaScript and* XML (AJAX)

AJAX (*Asynchronous JavaScript and XML*) sesungguhnya bukanlah suatu bahasa pemrogarman, melainkan hanya suatu teknik yang memungkinkan untuk membuat aplikasi *web* yang interaktif. Dengan menggunakan AJAX, aplikasi *web* dapat berinteraksi dengan server di latar belakang sehingga tidak mempengaruhi halaman *web* secara keseluruhan. Sebagai gambaran, Anda mungkin sering mendaftarkan diri pada suatu situs

web sebagai anggota baru. Setelah Anda selesai mengetikan *user ID* saat registrasi, sistem bisa memberitahukan bahwa nama yang Anda pakai sudah digunakan oleh orang lain atau belum.

Bagaimana *aplikasi web* ini bisa tahu hal itu padahal Anda belum mengklik tombol apapun dalam halaman *web*. Rahasianya terletak pada pemrosesan di latar belakang yang dilakukan oleh AJAX. Gambar 2.2 Menunjukan aplikasi *web* yang berbasis AJAX.



Gambar 2.7 Model Aplikasi Web dengan AJAX

Nampak bahwa pada aplikasi *web* yang berbasis AJAX terdapat bagian yang disebut lapisan AJAX (atau terkadang disebut mesin AJAX). Lapisan inilah yang mengatur komunikasi antara *client* dan *server*.

Perlu diperhatikan dalam lapisan AJAX :

1. Lapisan AJAX meminta layanan ke *server* atau tidak, tergantung pada jenis layanan antarmuka. Sebagai contoh, untuk validasi terhadap objek formulir tidak perlu melakukan permintaan ke *server*. Cukup ditangani di sisi *client*. Namun untuk pemeriksaan keabsahan *password*, mau tidak mau harus dilayangkan ke *server* karena harus berhubungan dengan database.
2. Sekalipun penambahan lapisan AJAX berkesan memperlambat proses (karena komunikasi ke server tidak lagi dilakukan secara langsung), pada

kenyataanya yang terjadi adalah sebaliknya. Hal ini karena permintaan yang dilakukan oleh lapisan AJAX menghasilkan informasi yang jauh lebih sedikit dibanding kalau yang diminta adalah keseluruhan halaman.

3. Lapisan AJAX bekerja dengan pendekatan asinkron. Asinkron berarti bahwa *client* bisa meminta layanan dari *server* dan tidak perlu menunggu sampai *server* melayaninya. Dengan demikian, saat informasi dari *server* belum didapat, layanan kepada pemakai tetap bisa dilakukan.
4. Komunikasi antara *client* dan *server* tidak perlu mengubah keseluruhan antarmuka dalam halaman *web*, tetapi cukup hanya bagian tertentu yang terpengaruh. Hal seperti ini tidak bisa dilakukan pada aplikasi *web* yang tidak menggunakan AJAX.

AJAX pertama kali ditemukakan pada tahun 2005 oleh Jesse James Garret, presiden dan pendiri perusahaan Adaptive Path. Menurut beliau, nama AJAX diberikan untuk memudahkan dalam berkomunikasi dengan klien daripada kalau menyebutnya "*Asynchronous JavaScript + CSS + DOM + XMLHttp Request*". Teknik ini sebenarnya sudah ada sejak 1999. Pada masa itu Microsoft membuat objek bernama *XMLHttpRequest* (Objek yang kelak akan dilibatkan dalam membuat AJAX) sebagai control *ActiveX* pada Internet Explorer 5. AJAX menjadi sangat popular baru setelah digunakan intensif oleh Yahoo! Dan Google.

Menurut Jesse James Garret, di dalam artikelnya yang berjudul "*AJAX A New Approach to Web Applications*", AJAX yang lengkap mencakup :

1. XHTML (*Extensible HyperText Markup Language*), sebagai peluasan dari HTML.
2. CSS (*Cascading Style Sheets*) sebagai pemformat bagian dari dokumen.
3. DOM (*Document Object Model*) untuk mengatur interaksi dan penampilan isi HTML dan XHTML secara dinamis.
4. XML (Extensible *Markup Language*) memungkinkan untuk melakukan pertukaran data.
5. XSLT (*Extensible Stylesheet Language* Transformation) adalah bahasa berbasis XML yang ditujukan untuk melakukan transformasi dokumen XML ke bentuk yang lain.

6. *XMLHttpRequest* untuk melakukan pertukaran data secara asinkron.
7. *JavaScript* sebagai bahasa untuk membentuk lapisan AJAX.

Memang nampaknya sangat kompleks karena melibatkan banyak hal. Namun tidak pelu khawatir bahwa hal itu akan menyusahkan dalam mewujudkannya. Dalam praktik, AJAX bisa diimplementasikan tanpa menggunakan XML, XSLT, maupun CSS.

2.2.13 *Virtual Instrumentation Engineering Workbench* (LabView)

LabVIEW adalah perangkat lunak komputer untuk pemrosesan dan visualisasi data dalam bidang akuisisi data, kendali instrumentasi serta automasi industri yang pertama kali dikembangkan oleh perusahaan National Instruments pada tahun 1986. Perangkat lunak ini dapat dijalankan pada sistem operasi Linux, Unix, Mac OS X dan Windows.