# BAB II

# LANDASAN TEORI

## 2.1 Konsep Dasar Sistem

Terdapat dua kelompok di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurenya dan yang menekankan pada komponen atau elemennya.

Menurut [Jogiyanto Hartono,1999]

Pendekatan sistem yang lebih menekankan pada prosedur mendefinisiksn sistem sebagai berikut :

*"suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersana-sama untuk melakukan suatu kegiatan atau untuk menyelsaikan suatu sasaran yang tertentu"*

Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefinisikan sistem sebagai berikut :

*"sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu"*

### 2.1.1 Karakteristik Sistem

Suatu sistem mempenyai karakteristik atau sifat-sifat yang tertenyu yaitu:

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling bekerja sama membentuk satu kesatuan.

2. Batasan Sistem

   Batasan sistem merupakan daerah yang membatasi antara satu sistem dengan sistem lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem

   Lingkungan luar dari suatu sistem adalah apapun batas dari sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem

   Penghubung sistem merupakan penghubung antara satu sub sistem dengan sub sistem yang lainnya.

5. Masukan Sistem

   Masukan sistem adalah energi yang dimasikan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) adalah energi yang dimasukan agar sistem tersebut dapat beroperasi, atau masulan sinyal (*signal input*) adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran Sistem

   Keluaran sistem adalah hasil dari energi yang diolah dan di klasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolahan Sistem

   Pengolahan sistem adalah suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah suatu masukan menjadi keluaran.

8. Sasaran Sistem

   Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*) , jika sistem tidak mempunyai sasatan, maka oper5asi sistem tidak akan ada gunanya.

### 2.1.2 Klasifikasi Sistem

Sistem dapat di klasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem di klasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*phsycal system*)
2. Sistem di klasifikasikan sebagai sistem alamiah (*nature system*) dan sistem buatan manusia (*human made system*)
3. Sistem di klasifikasikan sebagai sistem tentu (*Deterministic system*) dan sistem tak tentu (*Probabilistic System*)
4. Sistem di klasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*opened system*)

## 2.2 Konsep Dasar Informasi

*"Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan berarti bagi yang menerimanya"* **(Jogiyanto H.M., 2001 : 8).**

Sumber informasi adalah data. Data merupakan fakta atau kenyataan yang menggambarkan suatu kejadian-kejadian atau event nyata, kemudian dirumuskan

ke dalam sekelompok simbol atau lambang-lambang yang teratur yang menunjukkan kualitas, tindakan atau hal-hal lain.

Informasi ibarat darah yang mengalir di dalam tubuh suatu organisasi, sehingga informasi sangat penting di dalam suatu organisasi. Karena informasi dapat berguna bagi suatu organisasi atau seseorang dalam mengambil suatu keputusan menurut JOG. Informasi (*information*) dapat didefinisikan sebagai berikut :

*"Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya."*

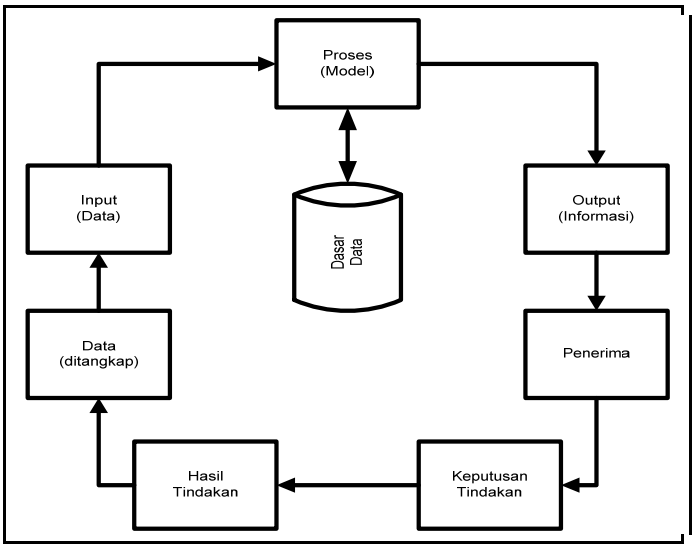
Sedangkan menurut AGS, informasi dapat didefinisikan sebagai berikut:

*"Informasi adalah suatu kerangka kerja dimana SDM (manusia, komputer) dikoordinasikan untuk mengubah masukan (data) menjadi keluaran (informasi) guna mencapai sasaran perusahaan."*

### 2.2.1 Siklus Informasi

Data merupakan bentuk yang masih mentah yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data yang diolah untuk menghasilkan informasi menggunakan suatu model proses yang tertentu. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut

dengan siklus informasi (*information cycle*). Siklus ini disebut juga dengan siklus data (*data processing cycles),* diperlihatkan pada gambar 2.1 di bawah ini.




**Gambar 2.1 Siklus Informasi**

***(Sumber : JOG)***

### 2.2.2 Kualitas Informasi

Informasi yang baik adalah informasi yang berkualitas. Informasi yang berkualitas ditentukan oleh hal-hal sebagai berikut:

1. Relevan (*Relevance*)

   Informasi tersebut mempunyai manfaat untuk pemakainya dan harus sesuai dengan yang dibutuhkan.

2. Dapat dipercaya (Realibility)

   Informasi yang akan diberikan dapat dipercaya kebenarannya dan mempunyai data-data yang lengkap dan jelas sumber-sumber datanya.

3. Tepat pada waktunya (*Timeliness*)

   Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak mempunyai nilai lagi, oleh karena itu informasi harus *up to date*.

4. Akurat (*Accurate*)

   Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan informasi, harus jelas mencerminkan maksudnya. Komponen-komponen data yang akurat adalah sebagai berikut:

   a. *Completeness*, yaitu informasi yang dihasilkan atau yang dibutuhkan memiliki kelengkapan yang baik, karena bila informasi yang dihasilkan sebagian-sebagian tentunya akan mempengaruhi dalam pengambilan keputusan atau menentukan tindakan secara keseluruhan.
   b. *Correctness*, yaitu kebenaran informasi dapat dipertanggungjawabkan dan mempunyai bukti-bukti dan fakta yang kuat.
   c. *Security* atau keamanan, dalam hal ini informasi yang dikirimkan ke setiap orang yang membutuhkannya perlu pengawasan karena struktur pengecekan dapat memutuskan jika informasi yang sensitif ditujukan kepada pemakai yang tidak sah kepada pihak yang salah.

5. Ekonomis

   biaya pembuatan informasi murah dan memberikan manfaat yang besar bagi pemakai.

## 2.3 Konsep Dasar Sistem Informasi

Sebuah sistem informasi merupakan kumpulan dari perangkat keras dan perangkat lunak komputer serta perangkat manusia yang akan mengolah data menggunakan perangkat keras dan perangkat lunak tersebut.

Selain itu sistem informasi dapat didefinisikan sebagai berikut :

1. Suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi.
2. Sekumpulan prosedur organisasi yang pada saat dilaksanakan akan memberikan informasi bagi pengambil keputusan dan atau untuk mengendalikan organisasi.
3. Suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, bersifat manajerial, dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

   Informasi merupakan hal yang sangat penting dalam pengambilan keputusan, permasalahannya adalah dimana informasi tersebut didapat.Informasi dapat diperoleh dari sistem informasi. Robert A Leitch dan K. Roscoe Davis mendefinisikan sistem informasi sebagai berikut:

   *"Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi*

*dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan."* **(Jogiyanto H.M., 2001 : 8).**

Komponen-komponen sistem informasi adalah sebagai berikut:

*1.* Perangkat keras *(hardware)*

Perangkat keras *(hardware)* terdiri dari komputer

*2.* Perangkat lunak *(software)*

Perangkat lunak *(software)* berupa program-program aplikasi yang akan digunakan, yaitu merupakan kumpulan dari perintah atau fungsi yang ditulis dengan aturan tertentu untuk memerintahkan komputer melaksanakan tugas tertentu.

3. Data

Data merupakan komponen dasar dari informasi yang akan diproses lebih lanjut untuk menghasilkan informasi.

4. Prosedur

Prosedur merupakan dokumentasi prosdedur atau proses sistem, tata cara atau penuntun operasional (aplikasi) dan teknis

5. Manusia

Manusia adalah pengguna dari sistem informasi.

## 2.4 Sistem Temu-kembali Informasi

Sistem temu-kembali informasi pada prinsipnya adalah suatu sistem yang sederhana. Misalkan ada sebuah kumpulan dokumen dan seorang user yang memformulasikan sebuah pertanyaan (*request* atau *query*). Jawaban dari pertanyaan tersebut adalah sekumpulan dokumen yang relevan dan membuang dokumen yang tidak relevan. Secara matematis hal tersebut dapat dituliskan sebagai berikut :

$Q \xrightarrow{2^n} D$, dimana Q = pertanyaan (queri), D = dokumen, n = jumlah dokumen, $2^n$ = jumlah kemungkinan himpunan bagian dari dokumen yang ditemukan. Sistem temu-kembali akan mengambil salah satu dari kemungkinan tersebut.

Sistem temu-kembali informasi pada dasarnya dibagi dalam dua komponen utama yaitu sistem pengindeksan (*indexing*) yang menghasilkan basis data sistem dan temu-kembali yang merupakan gabungan dari *user interface* dan look-up-table. Pada bagian selanjutnya akan dijelaskan berbagai macam sistem pengindeksan dan teknik-teknik temu-kembali informasi yang telah dikembangkan.

## 2.5 Sistem Temu-kembali Informasi Berbasis Hiperteks

Pada awalnya, hiperteks dan temu-kembali informasi merupakan bidang penelitian yang berbeda satu dengan yang lain. Hiperteks berkisar pada masalah *user-disorientation*, strategi pengembangan dokumen hiperteks, dan mekanisme konversi dokumen tekstual menjadi bentuk hiperteks (Ellist,1996). Sedangkan

temu-kembali informasi bergerak pada topik manipulasi kueri, konsep basis data tekstual, dan relevansi dokumen terhadap kueri (Bodhitama,1997). Penggabungan kedua bidang ini dapat memecahkan masalah-masalah dalam bidang temu-kembali informasi. Misalnya, sistem temu-kembali informasi yang didasarkan pada penggunaan operator *Boolean,* mengandalkan kemampuan pemakai dalam memformulasikan kueri. Hal ini sering mempersulit pengguna. Dengan adanya sistem hiperteks, hal ini dapat di permudah dengan penyediaan antar muka yang memakai pencarian dengan metode *browsing*.

Smeaton(1991) di dalam Ellist(1996) juga menyatakan bahwa hiperteks dan temu-kembali informasi itu saling berkomplemen satu sama lain. Hiperteks membutuhkan lebih banyak *searching* sedangkan temu-kembali informasi membutuhkan lebih banyak *browsing*. Hal yang dimaksud adalah hiperteks akan semakin baik jika disertai dengan fasilitas *search*, dan temu-kembali informasi membutuhkan *browsing* dalam melakukan pencarian yang efisien. Adapun maksud dari *searching* adalah berusaha mendapatkan atau mencapai tujuan spesifik sedangkan *browsing* adalah mengikuti suatu path sampai mencapai suatu tujuan. Menurut Brown(1988) didalam Agosti(1993), *browsing* itu bisa diibaratkan dengan ***From Where to What***. Maksudnya adalah kita tahu dimana posisi kita dalam *database* dan kita ingin tahu apa yang ada disana (*database*). Sedangkan *Searching* bisa diibaratkan dengan ***From What to Where***. Maksudnya adalah kita tahu apa yang kita inginkan dan kita ingin menemukan dimana dia didalam *database*.

Penggabungan sistem temu-kembali kedalam basis hiperteks lebih dikenal dengan nama search engine, dimana sistem ini dapat dibagi kedalam dua kategori berdasarkan sumber informasinya yaitu:

1. *Worldwide Search Engine*

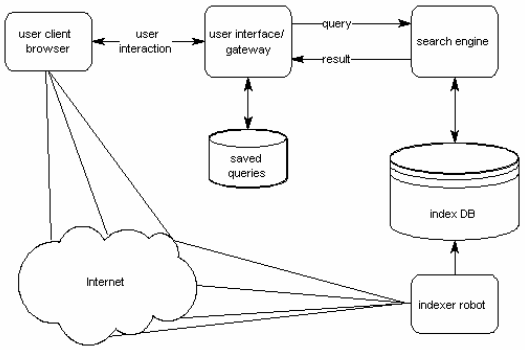
   *Worldwide Search Engine* adalah suatu sistem temu-kembali informasi yang mengambil data-data dari berbagai server di seluruh penjuru dunia. Data-data tersebut diambil melalui program yang disebut dengan "robot" atau "bot". Program inilah yang melakukan pencarian data pada setiap server, yang kemudian dikirim ke server pusat pada selang waktu tertentu.

2. *Local Search Engine*

   *Local search engine* adalah suatu sistem temu-kembali informasi yang mengambil data-data dari server tertentu saja. Kata "local", yang berarti lokal atau setempat, memberi penekanan akan lokasi sumber data yang akan digunakan. *Local search engine* tidak dirancang untuk mengarungi belantara internet seperti *worlwide search engine*. Tujuan implementasi local search engine dimaksudkan untuk pencarian pada objek spesifik dan lebih kecil lingkupnya dibandingkan internet sendiri.

Mengenai pemilihan penerapan sistem temu-kembali berbentuk *local search engine* atau *worlwide search engine* tergantung kepada masalah atau jenis informasi yang akan kita sediakan. Penerapan kedua kategori ini hanya akan mempengaruhi cara sistem pengindeksan dari temu-kembali. Sedangkan teknik *retrieval* dan rancangan penerapan teknik pada hiperteks akan sama saja, baik pengindeksannya secara *local search engine* ataupun *worldwide search engine*.

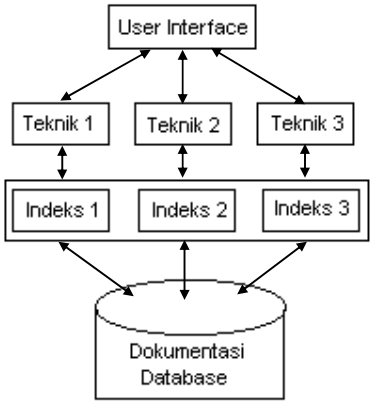
Penelitian mengenai penerapan sistem temu-kembali berbasis hiperteks telah mulai dilakukan seiring dengan perkembangan internet akhir-akhir ini. Penelitian yang dilakukan Yuwono(1995), menggunakan rancangan atau arsitektur seperti terlihat pada Gambar 2.2.




**Gambar 2.2 Arsitektur sistem temu-kembali (Yuwono,1995)**

Arsitektur yang dirancang ini terdiri dari dua komponen utama yaitu: *Index Builder* dan *Search Engine*. *Index builder* merupakan sebuah sistem pengindeksan yang memanfaatkan "robot" yang berkomunikasi dengan menggunakan HTTP (*Hypertext Transfer Protocol*) untuk mencari informasi yang akan di indeks. Sedangkan *Search engine* merupakan teknik dari temu-kembali dalam menemukan dokumen dan sekaligus mengeksekusi algoritma peringkat dalam menampilkan dokumen. Sedangkan komunikasi antara pemakai dan *search engine* dalam memformulasikan kueri dilakukan melalui *User Interface.* Setelah pemakai menemukan dokumen yang relevan dengan kueri, dapat langsung melakukan *browsing* ke sumber informasi dalam hal ini adalah alamat tempat *www*.

Adapun arsitektur dari sistem temu-kembali yang dikembangkan dalam penelitian ini dapat dilihat pada Gambar 2.3 di bawah ini.




**Gambar 2.3 Rancangan Sistem Temu-Kembali Berbasis Hiperteks**

Sistem temu-kembali yang dibangun terdiri dari berbagai macam teknik *retrieval* seperti teknik *Boolean* biasa dan *Boolean* berperingkat serta teknik *Extended Boolean* berdasarkan p-norm model. Sedangkan teknik pengindeksannya juga terdiri dari beberapa macam antara lain teknik berdasarkan frekuensi kemunculan istilah dan teknik pengindeksan yang dinormalisasi berdasarkan aturan Savoy(1993). Pada sistem ini, teknik *retrieval*, basis data indeks dan kumpulan dokumen berada dalam sebuah komputer *server* yang sama (*local*). Sedangkan antar muka dari sistem yang dikembangkan adalah berbasis hiperteks.

Implementasi program temu-kembali informasi berbasis hiperteks mempunyai kelebihan dibandingkan dengan sistem konvensional, terutama dalam hal navigasi sistem. Sebagai gambaran lebih jelas dapat di lihat pada Gambar 3. Untuk membaca isi dokumen, cukup dengan memilih *hiperlink* judul atau nama

pengarang. Demikian pula untuk melihat relasi istilah guna membentuk kueri baru, cukup dengan membuka *selection list* yang di dalamnya terdapat istilah lain yang berelasi. Relasi istilah ini telah di proses sewaktu pemakai melakukan pencarian berdasarkan istilah-istilah yang terdapat pada kueri pemakai.

Pada sistem temu-kembali berbasis hiperteks ini, selain menemukan dokumen yang relevan juga menampilkan relasi istilah pada kueri dalam satu proses. Semua modul yang ada pada sistem ini menjadi satu kesatuan mode, dimana dokumen yang ditemukan secara langsung merupakan input bagi modul lain seperti modul relasi istilah.

## 2.6 Tokenizer

Tokenizer adalah pemrosesan suatu unit dokumen yang mempunyai hasil akhir berupa *Tokens* unik dan banyaknya frekuensi *Tokens* yang terdapat dari suatu unit dokumen. Didalam proses *Tokenizer* terdapat dua proses yaitu proses *Tokenization* dan proses *Parsing*.

### 2.6.1 Tokenization

Dengan satu urutan karakter dan satu unit dokumen yang didefinisikan, tokenization adalah pekerjaan pemotongan satu urutan karakter menjadi beberapa bagian yang dinamakan *tokens* yang biasanya adalah kata, pada saat bersamaan proses tokenization membuang karakter tertentu, seperti pemberian tanda baca. *Tokens* ini sering dengan bebas menunjuk sebagai istilah atau kata, tetapi tanda baca ini kadang-kadang penting untuk membuat satu perbedaan *Tokens*. *Tokens*

adalah satu contoh (instance) dari satu urutan karakter didalam beberapa dokumen tertentu.

### 2.6.2 Parsing

Parsing adalah proses pengenalan dan pengambilan Token hasil Tokenization dari sekumpulan unit dokumen. Yang biasanya kata – kata. Proses parsing tidak hanya dapat dilakukan dalam proses Information retrieval, melainkan juga pada bidang lain seperti pada pembuatan sebuah compiler dan Bahasa Alami. Sebelumnya perlu diketahui arti dari istilah parser yaitu program yang melakukan proses parsing.

Untuk pemrosesan, dokumen dipilah menjadi unit-unit yang lebih kecil misalnya berupa kata, frasa atau kalimat. Unit pemrosesan tersebut disebut sebagai token. Parsing merujuk pada proses pengenalan token yang terdapat dalam rangkaian teks (Grossman, 2002). Oleh karena itu bagian dasar dalam parsing adalah algoritme pengambil token dari teks yang disebut tokenizer. Proses ini memerlukan pengetahuan bahasa untuk menangani karakter-karakter khusus, serta menentukan batasan satuan unit dalam dokumen.

### 2.6.3 Parser

Parser dapat diibaratkan sebagai “otak” dari sebuah kompiler: komponen inilah yang menginferensikan makna dari bahasa dalam string input berdasarkan grammaratautatabahasa yang telah ditentukan sebelumnya oleh pemrogram. Dari makna atau nilai semantik sebuah rangkaian token, parser dapat segera

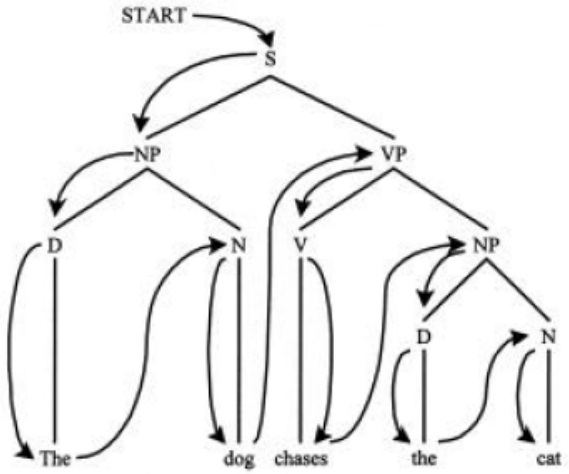
memproses (interpretasi, translasi) struktur implisit dalam rangkaian token tersebut, atau menunda pemrosesan sampai didapat struktur utuh dari string input.

#### 2.6.3.1 Top-down Parsing

Top-down parser bekerja dengan cara menguraikan sebuah kalimat mulai dari constituent yang terbesar yaitu sampai menjadi constituent yang terkecil. Hal ini dilakukan terus-menerus sampai semua komponen yang dihasilkan ialah constituent terkecil dalam kalimat, yaitu kata.

Sebagai contoh, dengan menggunakan contoh grammar, dapat dilakukan proses top-down parsing untuk kalimat *"The dog chased the cat"* yang ditunjukkan pada gambar 2.4. Dari gambar ini terlihat bahwa top-down parser menelusuri setiap node pada parse tree secara *pre-order*. Beberapa metode parsing yang bekerja secara top-down ialah:

1. Top-down parser biasa
2. Recursive-descent parser
3. Transition-network parser
4. Chart parser




**Gambar 2.4 Cara Kerja Top-down Parser**

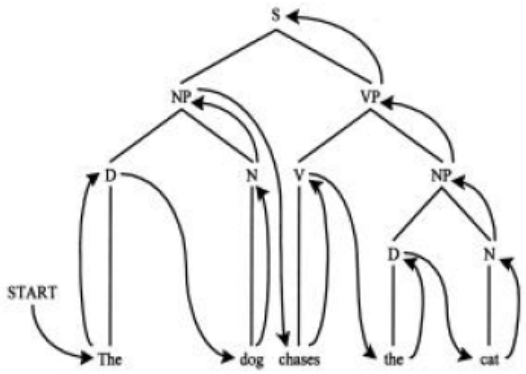
Top-down parser dapat diimplementasikan dengan berbagai bahasa pemrograman, namun akan lebih baik jika digunakan *declarative language* seperti Prolog atau LISP. Hal ini disebabkan oleh karena pada dasarnya proses parsing ialah proses searching yang dilakukan secara rekursif dan backtracking, dimana proses ini sudah tersedia secara otomatis dalam bahasa Prolog.

Dengan demikian parser yang ditulis dalam Prolog atau bahasa deklaratif lainnya akan menjadi jauh lebih sederhana daripada parser yang dibuat dalam bahasa prosedural biasanya seperti Pascal, C dan sebagainya. Program 1 menunjukkan implementasi top-down parser biasa dalam bahasa Prolog.

### 2.6.3.2 Bottom-Up Parsing

Bottom-up parser bekerja dengan cara mengambil satu demi satu kata dari kalimat yang diberikan, untuk dirangkaikan menjadi constituent yang lebih besar.

Hal ini dilakukan terus-menerus sampai constituent yang terbentuk ialah sentence atau kalimat. Dengan demikian metode bottom-up bekerja dengan cara yang terbalik dari top-down. Cara kerja bottom-up parser ditunjukkan pada gambar 2.5.




**Gambar 2.5 Cara Kerja Bottom-up Parser**

Metode parsing yang bekerja secara bottom-up antara lain ialah bottom-up parser biasa dan shift-reduce parser.

Perhatikan bahwa parser ini tidak membedakan antara *rule* (grammar) dan *word* (lexicon) sehingga cara kerjanya sangat sederhana namun sangat "bodoh" karena akan terus mengulang-ulang kesalahan yang sama.

Kesederhanaan metode ini terletak pada predikat untuk parsing, yaitu *parse* yang hanya memiliki sebuah argumen. Argumen ini berisi kalimat yang akan diparse dalam bentuk list dari symbol. Kata-kata dari input kalimat akan dirangkaikan sambil mencari aturan yang lebih luas, sampai tinggal sebuah simbol saja dalam list

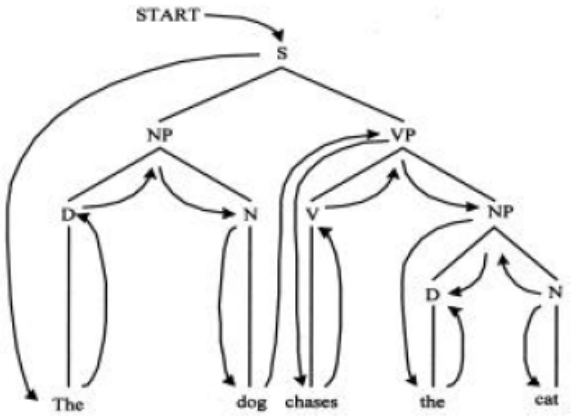
.

### 2.6.3.3 Gabungan Top-Down dan Bottom-Up Parsing

Baik top-down parsing mapun bottom-up parsing memiliki kekurangan dan kelebihannya masing-masing. Metode top-down mampu menangani grammar dengan *empty production* (misalnya *d→ 0*) namun tidak dapat menangani grammar dengan *left recursion* (misalnya *np→ np conj np*). Sedangkan metode bottom-up dapat menangani left recursion namun tidak dapat menangani empty production.

Dengan demikian metode parsing yang terbaik ialah metode yang dapat menggabungkan top-down dan *bottom-up* parsing. Ada beberapa metode yang dikembangkan yang menggabungkan kedua metode ini, di antaranya ialah *left-corner parsing* serta *Earley's parsing*.

Cara kerja left-corner parsing ialah dengan mula-mula menerima sebuah kata, menentukan jenis constituent apa yang dimulai dengan jenis kata tersebut, kemudian melakukan proses parsing terhadap sisa dari constituent tersebut secara top-down. Dengan demikian proses parsing dimulai secara bottom-up dan diakhiri secara top-down. Program 3 menunjukkan implementasi left-corner parser dalam Turbo Prolog, sedangkan alur kerjanya ditunjukkan pada gambar 2.6.




**Gambar 2.6 Cara Kerja Left-Corner Parser**

### 2.6.4 Grammer

Pada saat kompiler membaca string input, yang ditemuinya hanyalah rangkaian karakter finit. Meskipun rangkaian tersebut dapat dipecah menjadi kelompok-kelompok kecil seperti kata atau frase, tindakan tersebut masih belum cukup bagi kompiler untuk mendapatkan makna dari string tersebut untuk kemudian memprosesnya. Lalu bagaimana kompiler melakukannya? Yaitu dengan bantuan grammar. Grammar adalah set aturan produksi yang mengendalikan urutan pemunculan kata atau frase dalam sebuah kalimat. Sebuah aturan produksi biasa ditulis sebagai pasangan a::=b, dengan adan bberisi kategorisintaksis (nonterminal)atau karakter (simbolterminal), dan ::=berarti “didefinisikan sebagai” atau “dapat disubstitusikan dengan”: adidefinisikan sebagai b. Sebuah nonterminal dapat berisi simbol terminal atau nonterminal lain.

Ada berbagai tingkatan kelas grammar berdasarkan kemampuan ekspresifmasing-masing, namun ada dua tingkatan yang paling sering digunakan

untuk kebutuhan praktis perakitan kompiler: grammar regular dan grammar bebas konteks.

### 2.6.4.1 Grammer Regular

Dalam hierarki-bahasa Chomsky, grammar regular (Grammar Tipe-3) adalah grammar paling restriktif yang dapat membangkitkan sebuah kalimat. Dalam hierarki tersebut, grammar regular mempunyai kemampuan pembangkitan kalimat yang sangat minimal karena:

1. Sisi kiri hanya boleh berisi sebuah nonterminal,
2. Sisi kanan dalam setiap aturan produksinya hanya boleh berisi satu nonterminal, dan posisinya hanya boleh berada di akhiratausisi kanan rangkaian.

Deskripsi yang lebih sederhana: parser untuk grammar ini tidak dapat mengetahui konteks pemunculan nonterminal yang tengah didefinisikan, dan hanya dapat melihat symbol terminal yang ada tepat didepannya.Contoh dari grammar ini adalah pada persamaan 2.1:

$$T :: = \text{‘0’ } t \mid \text{’1’ } t \mid Є \qquad \text{persamaan (2.1)}$$

Yang mengekspresikan semua rangkaian digit biner dengan panjang arbitrer (simbol terminal dibatasi dengan tanda petik tunggal).

Karena ia hanya perlu dapat melihat satu simbol ke depan itulah grammar regular unggul dibanding tipe grammar lain untuk fungsi pengenalan pola. Grammar ini sering digunakan dalam versi ringkasnya, yang disebut sebagai ekspresi regular atau regex.

### 2.6.4.2 Grammer Bebas Konteks

Grammar bebas konteks (CFG: Context-free Grammar)setingkat lebih tinggi dibanding grammar regular, karena tidak ada batasan untuk sisi kanan aturan produksi. Grammar ini lazim digunakan untuk menspesifikasikan bahasa-bahasa yang memiliki kompleksitas moderat.

Hanya CFG dan grammar regular, kelas grammar yang dapat secara efisien diemulasikan karakteristiknya oleh program komputer; sebuah compiler dapat mengenali bahasa yang relatif kompleks, hanya dengan menggunakan kombinasi dari keduanya: CFG untuk spesifikasi grammar bahasa secara keseluruhan, dan regex untuk spesifikasi tokenatausimbol terminal.

### 2.6.4.3 Lexical Analyzer

Scanner atau lexical analyzer adalah komponen dari kompiler yang bersentuhan langsung dengan string input, mengubahnya ke dalam bentuk yang lebih mudah dikelola oleh parser. Kedua istilah tersebut umumnya merujuk pada komponen yang sama pada kompiler. Terkadang istilah satu lebih cenderung digunakan dibanding istilah lainnya untuk menyebut komponen tersebut, dengan pembedaan berdasarkan gugus tugas yang dibebankan: “scanner” membaca setiap karakter dalam string input satu-persatu dan mengumpulkan masing-masing karakter ke dalamkelompok-kelompok kecil yang disebut lexeme dan diasosiasikan dengan entitas tunggal yang disebut sebagai token, sementara “lexical analyzer” melakukan tindakan yang kurang-lebih sama, namun

melakukan pengolahan lebih jauh sedemikian sehingga deretan lexeme dalam string input tidak dapat dipetakan secara linear dengan deretan token dalam string output. Contoh termudah tentunya fasilitas preprosesor dalam bahasa pemrograman CatauC++: preprosesor mengolah setiap baris yang diawali dengan simbol '#' sesuai dengan makna semantik masing-masing, dan menghilangkan baris-baris komentar.

Secara umum pembedaan ini tidaklah terlalu relevan, karena pilihan pembebanan gugus tugas pada akhirnya adalah keputusan teknis yang bergantung terutama pada grammar bahasa yang akan diimplementasikan (komentar baris dapat diabaikan oleh scanner, namun komentar dalam tanda kurung mungkin membutuhkan grammar untuk menyeimbangkan delimiter dalam grammar meski harus mengorbankan unjuk kerja parser). Dalam tulisan ini, setidaknya, tidak ada pembedaan di antara kedua istilah tersebut.

## 2.7 Stoplist

Stoplist Adalah proses pembuangan atau menghilangkan kata-kata buang, yaitu : Kata depan, kata sambung, kata ganti, dll. seperti : di, dan, tetapi, dia, yaitu, sedangkan, dan sebagainya

## 2.8 Stemming

*Stemming* adalah proses penghilangan prefiks dan sufiks dari kueri dan istilah istilah dokumen (Grossman, 2002). *Stemming* dilakukan atas dasar asumsi bahwa kata-kata yang memiliki *stem* yang sama memiliki makna yang serupa sehingga

pengguna tidak keberatan untuk memperoleh dokumen-dokumen yang di dalamnya terdapat kata-kata dengan *stem* yang sama dengan kuerinya.

Teknik-teknik *stemming* dapat dikategorikan menjadi:

1. Berdasarkan aturan sesuai bahasa tertentu
2. Berdasarkan kamus
3. Berdasarkan kemunculan bersama.

Proses ini memiliki dua tujuan. Dalam hal efisiensi, *stemming* mengurangi jumlah kata-kata unik dalam indeks sehingga mengurangi kebutuhan ruang penyimpanan untuk indeks dan mempercepat proses pencarian. Dalam hal keefektifan, *stemming* meningkatkan *recall* dengan mengurangi bentuk-bentuk kata ke bentuk dasarnya atau *stem*-nya. Sehingga dokumen-dokumen yang menyertakan suatu kata dalam berbagai bentuknya memiliki kecenderungan yang sama untuk ditemukembalikan. Hal tersebut tidak akan diperoleh jika tiap bentuk suatu kata disimpan secara terpisah dalam indeks. Akan tetapi, *stemming* dapat menurunkan tingkat *precision* jika setiap bentuk suatu *stem* diperoleh, sedangkan yang relevan hanyalah bentuk yang sama dengan yang digunakan dalam kueri (Liddy, 2001).

*Parsing* dan *stemming* dalam suatu IRS tergantung pada bahasa yang digunakan dalam dokumen yang akan dicari. IRS untuk Bahasa Inggris kurang optimal untuk menangani dokumen dalam Bahasa Indonesia. Bahasa Indonesia memiliki daftar kata buang (*stoplist*) serta sistem pembentukan kata yang sangat berbeda dengan bahasa Inggris, sehingga diperlukan IRS yang khusus untuk

Bahasa Indonesia (Gunarso, 1998).

*Stemming* untuk Bahasa Indonesia telah dikembangkan antara lain yang menggunakan aturan berdasarkan algoritme Porter (1980) oleh Akhmadi (2002) yang hanya melakukan pemotongan prefiks dan oleh Ridha (2002) yang melakukan pemotongan prefiks dan sufiks. *Stemming* berdasarkan kamus untuk Bahasa Indonesia juga telah dikembangkan oleh Nazief (1996) (Nazief, 2000).

Stemming adalah proses penghilangan atau pemotongan imbuhan yang terdapat pada sebuah kata yang mempunyai imbuhan menjadi bentuk kata dasarnya saja, untuk Bahasa Indonesia imbuhan mempunyai peran penting dalam suatu kalimat, karena suatu kata dapat mempunyai arti yang berbeda apabila diberi suatu imbuhan. Yang mengakibatkan setiap kata berimbuhan mempunyai arti yang berbeda dan pembentukan kata baru. Contohnya : kata “diadaptasikan” atau “beradaptasi” menjadi bentuk kata dasar “adaptasi” sebagai istilah. Algoritma ini didahului dengan pembacaan tiap kata dari file sampel. Sehingga input dari algoritma ini adalah sebuah kata yang kemudian dilakukan, pemeriksaan semua kemungkinan bentuk kata. Setiap kata diasumsikan memiliki 2 Awalan (Prefiks) dan 3 Akhiran (Sufiks). Sehingga bentuknya menjadi : **Prefiks 1 + Prefiks 2 + Kata Dasar + Akhiran 1 + Akhiran 2 + Akhiran 3**.

## 2.9 Pembentukan Kata-kata Bahasa Indonesia

Ada banyak ragam pembentukan kata dalam Bahasa Indonesia. Sebagian besar kata dibentuk dengan cara menggabungkan beberapa komponen yang berbeda. Untuk memahami cara pembentukan kata-kata tersebut kita sebaiknya

mengetahui lebih dahulu beberapa konsep dasar dan istilah seperti yang dijelaskan di bawah ini. Untuk mempersingkat dan memperjelas pembahasannya, kami menggunakan kata-kata yang tidak bersifat gramatikal atau teknis untuk menjelaskan kata-kata tersebut sebanyak mungkin.

### 2.9.1 Definisi Istilah

**Kata dasar** (akar kata) = kata yang paling sederhana yang belum memiliki imbuhan, juga dapat dikelompokkan sebagai bentuk asal (tunggal) dan bentuk dasar (kompleks), tetapi perbedaan kedua bentuk ini tidak dibahas di sini.

**Afiks** (imbuhan) = satuan terikat (seperangkat huruf tertentu) yang apabila ditambahkan pada kata dasar akan mengubah makna dan membentuk kata baru. Afiks tidak dapat berdiri sendiri dan harus melekat pada satuan lain seperti kata dasar. Istilah afiks termasuk prefiks, sufiks dan konfiks.

**Prefiks** (awalan) = afiks (imbuhan) yang melekat di depan kata dasar untuk membentuk kata baru dengan arti yang berbeda.

**Sufiks** (akhiran) = afiks (imbuhan) yang melekat di belakang kata dasar untuk membentuk kata baru dengan arti yang berbeda.

**Konfiks** (sirkumfiks atau simulfiks) = secara simultan (bersamaan), satu afiks melekat di depan kata dasar dan satu afiks melekat di belakang kata dasar yang bersama-sama mendukung satu fungsi.

**Kata turunan (kata jadian)** = kata baru yang diturunkan dari kata dasar yang mendapat imbuhan.

**Keluarga kata dasar** = kelompok kata turunan yang semuanya berasal dari satu kata dasar dan memiliki afiks yang berbeda.

### 2.9.2 Imbuhan (Afiks) dalam Bahasa Indonesia

Imbuhan adalah suatu unsur struktural yang diikatkan pada sebuah kata dasar. Imbuhan terbagi atas : Awalan (Prefiks), Sisipan (Infiks), Akhiran (Sufiks), Imbuhan Terbelah (Konfiks).

#### 2.9.2.1 Awalan (Prefiks)

Imbuhan yang terdapat di awal suatu kata, seperti : "me-, *ber-*, *per-*, *meng-*, *di-*, *ter-*, *ke-* dan *se-*". Untuk awalan *me-* mempunyai aturan khusus, yaitu:

Awalan me-

1. tetap, jika huruf pertama kata dasar adalah l, m, n, q, r, atau w. Contoh: *me-* + luluh → *me*luluh, *me-* + makan → *me*makan.
2. *me-* → *mem-*, jika huruf pertama kata dasar adalah b, f, **p***, atau v. Contoh: *me-* + baca → *mem*baca, *me-* + pukul → *mem*ukul*, *me-* + vonis → *mem*vonis, *me-* + fasilitas + i → *mem*fasilitasi.
3. *me-* → *men-*, jika huruf pertama kata dasar adalah c, d, j, atau **t***. Contoh: *me-* + datang → *men*datang, *me-* + tiup → *men*iup*.
4. *me-* → *meng-*, jika huruf pertama kata dasar adalah huruf vokal, **k***, g, h. Contoh: *me-* + kikis → *meng*ikis*, *me-* + gotong → *meng*gotong, *me-* + hias → *meng*hias.

5. *me-* → *menge-*, jika kata dasar hanya satu suku kata. Contoh: *me-* + bom → *menge*bom, *me-* + tik → *menge*tik, *me-* + klik → *meng*eklik.
6. *me-* → *meny-*, jika huruf pertama adalah **s***. Contoh: *me-* + sapu → *meny*apu*.

Huruf dengan tanda ***** memiliki sifat-sifat khusus:

1. Dilebur jika huruf kedua kata dasar adalah huruf vokal. Contoh: *me-* + *ti*pu → *meni*pu, *me-* + *sa*pu → *menya*pu, *me-* + *ki*ra → *mengi*ra.
2. Tidak dilebur jika huruf kedua kata dasar adalah huruf konsonan. Contoh: *me-* + *kl*arifikasi → *mengkl*arifikasi.
3. Tidak dilebur jika kata dasar merupakan kata asing yang belum diserap secara sempurna. Contoh: *me-* + *k*onversi → *mengk*onversi.

#### 2.9.2.2 Sisipan (Infiks)

Terdapat empat Sisipan (Infiks), yaitu : *-el-, -em-, -er- dan –in-*.

#### 2.9.2.3 Akhiran (Sufiks)

Imbuhan yang terdapat di akhir sebuah kata.dalam bahasa Indonesia terdapat delapan sufiks yaitu: *-i, -kan, -an, -man, -wan , -wati, -wi (-wiah) dan –nya*. Selain itu juga terdapat sufiks yang berasal dari bahasa asing yaitu: *-wan, -wati, -in, -at, -isme, -(is)asi, -logi dan –tas*. Juga terdapat akhiran lain yang disebut partikel (P), yaitu: *lah, kah, tah, pun*. Sufiks juga dapat diberikan untuk menggantikan kata ganti kepunyaan (KK), misalnya: *-ku, -mu*, dan *–nya.*

#### 2.9.2.4 Imbuhan Terbelah (Konfiks)

adalah gabungan dari infiks dan sufiks tetapi menjadi satu kesatuan yang tidak dapat dipisahkan. Konfiks ini harus mengapit kata dasar. Terdapat beberapa konfiks yaitu: *ke - an, ber - an, peng - an, per - an, se - nya*.

#### 2.9.2.5 Imbuhan Gabungan (Simulfiks)

gabungan lebih dari satu awalan atau akhiran, contoh: *member - kan memberlakukan* dan *memberdayakan*

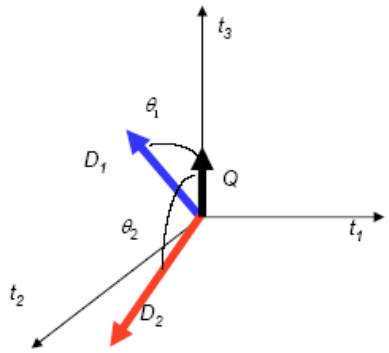
## 2.10 Term Weighting (Pembobotan Istilah)

### 2.10.1 Vektor Space Model

Salah satu hal yang mempengaruhi *recall* suatu sistem temu kembali informasi adalah faktor bahasa yang digunakan pada dokumen yang akan dicari. *Recall* adalah kemampuan sistem temu kembali informasi tersebut untuk menemukan sebanyak mungkin dokumen yang relevan dengan *query* yang diberikan oleh pengguna. *Recall* didefinisikan sebagai perbandingan antara jumlah dokumen yang relevan yang berhasil ditemukan dengan keseluruhan jumlah dokumen yang relevan. Sinonim kata adalah salah satu faktor dalam bahasa manusia yang dapat mempengaruhi *recall* suatu sistem temu kembali informasi. Adanya sinonim kata dapat menyebabkan turunnya *recall* dari sistem temu kembali informasi tersebut. Hal ini disebabkan adanya dokumen yang relevan dengan *query* pengguna namun tidak dapat ditemukan oleh sistem temu kembali informasi. Dokumen tersebut

tidak dianggap relevan karena dokumen itu tidak mengandung *keyword* yang dicari, melainkan sinonim dari *keyword* tersebut.

*Vector space model* adalah suatu model yang digunakan untuk mengukur kemiripan antara suatu dokumen dengan suatu *query* ditunjukan pada gambar 2.7. Pada model ini, *query* dan dokumen dianggap sebagai vektor-vektor pada ruang n-dimensi, dimana n adalah jumlah dari seluruh *term* yang ada dalam leksikon. Leksikon adalah daftar semua *term* yang ada dalam indeks.




**Gambar 2.7 Ukuran Kesamaan Antara Vektor Dokumen Dan Query, Dimana: t = kata di database, D = dokumen, Q = kata kunci**

1) Defenisi

   a. $w_{i,j}$ adalah bobot $(k_i,d_j)$, $w_{i,j} \in R+$

   b. $w_{i,q}$ adalah bobot $(k_{i,q})$, $w_i, q \in R+$

   c. Vektor query $q = (w_{1,q}, w_{2,q}, \ldots w_{t,q})$

   d. t : banyaknya index term

   e. Dokumen direpresentasikan sebagai vektor $d_j = (w_{1,j}, w_{2,j}, \ldots w_{t,j})$

f. Dapat dihitung ukuran kesamaan antara vektor $d_j$ dan q, disebut $sim(d_j,q)$

2) Term Weight

a) Term Frequency (tf): banyaknya kemunculan term ke-I pada dokumen ke-j ($tf_{ij}$)

b) Document Frequency (df): banyaknya dokumen dimana term ke-i muncul di dalamnya ($df_i$)

c) Inverse Document Frequency (idf): ukuran diskriminan kemunculan term ke-i dalam koleksi ➔ $idf_i$ = log ($Nataudf_i$), N : #dokumen

d) Banyak variasi dari term-weight yang ada sebagai hasil pengembangan dari tf-idf.

e) Salah satu yang dianggap baik:

$$w_{ij} = \frac{\left(\log tf_{ij} + 1.0\right) * idf_j}{\sum_{j=1}^{t} \left[\left(\log tf_{ij} + 1.0\right) * idf_j\right]^2} \qquad \text{Persamaan (2.2)}$$

3) Normalisasi:

1. N : banyaknya dokumen
2. $n_i$ : banyaknya dokumen memiliki term $k_i$
3. $Freq_{ij}$ : frekuensi term $k_i$ dalam $d_j$

$idf_i = \log$ persamaan (2.3)

$$f_{ij} = \frac{\text{freq}_{i,j}}{\max_i \text{freq}_{i,j}} \qquad \text{persamaan (2.4)}$$

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i} \qquad \text{persamaan (2.5)}$$

4) **Ukuran Kesamaan (*Similarity Measurement*)**

Ukuran kesamaan (*similarity*) istilah dalam model ruang vektor ditentukan berdasarkan *assosiative coefficient* berdasarkan *inner product* dari dokumen vektor dan kueri vektor, dimana *word overlap* menunnjukkan kesamaan istilah. *Inner product* umumnya sudah dinormalisasi. Metode Ukuran kesamaan yang paling populer adalah *cosine coefficient*, yang menghitung sudut antara vektor dokumen dengan vektor kueri.

Rumus = $\sum_{i=1}^{n} W_{i,q} \times W_{i,j}$ persamaan(2.6)

Untuk mengukur kesamaan antara vektor dan q yang digunakan hanya menggunakan 2 metode yaitu:

*a) Inner Product*

*b) Cosine*

***1) Inner Product***

$$SC(Q, D_i) = \sum_{j=1}^{t} w_{qj} \times d_{ij} \qquad \text{persamaan(2.7)}$$

Masalah: dokumen yang panjang cenderung akan memiliki koefisien kesamaan yang tinggi karena peluang term yang sesuai antara query dan dokumen cukup tinggi.

***2) Cosine***

Rumus =

$$SC(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} \times d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_i(d_{ij})^2 \quad \Sigma_{j=1}^{t}(w_{qj})^2}}$$ persamaan(2.8)

## 2.11 Pengindeksan

Sebuah bahasa indeks adalah bahasa yang digunakan untuk menggambarkan dokumen-dokumen dan permintaan-permintaan. Elemen bahasa indeks adalah istilah indeks, yang dapat diturunkan dari teks dokumen yang digambarkan atau dibuat secara mandiri (Rijsbergen, 1979).

Salton (1968) menunjukkan bahwa sistem pencarian dan analisa teks yang sepenuhnya otomatis tidak menghasilkan kinerja temu-kembali yang lebih buruk dibandingkan dengan sistem konvensional yang menggunakan pengindeksan dokumen manual dan formulasi pencarian manual.

## 2.12 Sejarah *Hypertext*

Istilah hypertext sendiri sudah digunakan sejak lebih dari 30 tahun yang lalu. Bahkan di tahun 1945, sudah ada tulisan yang memimpikan suatu mesin yang bisa

berfungsi sebagai mesin hypertext. Beberapa perkembangan yang dapat dicatat dalam sejarah antara lain sebagai mana dikemukakan oleh *Jacob Nielsen* dalam *Short Hystory of Hypertext*:

1. **1945** Vannevar Bush mengajukan proposal mesin *Memex*
2. **1965** Ted Nelson menggunakan istilah "*hypertext*" dalam buku *Literary Machines*
3. **1967** The *Hypertext Editing System* and FRESS, Brown University, Andy van Dam
4. **1968** Doug Engelbart dan beberapa peneliti mendemokan *NLS* system
5. **1978** *Aspen Movie Map* hypermedia videodisk pertama , Andy Lippman, MIT
6. **1984** Filevision dari Telos; hypermedia database dibuat untuk komputer Macintosh
7. **1985** *Symbolics Document Examiner*, Janet Walker
8. **1985** *Intermedia*, Brown University, Norman Meyrowitz
9. **1986** OWL memperkenalkan *Guide*, hypertext untuk umumpertama
10. **1987** Apple memperkenalkan *HyperCard*, Bill Atkinson
11. **1987** *Hypertext'87* menyelenggarakan konfrensi pertama mengenai hypertext
12. **1991** *World Wide Web* di CERN menjadi global hypertext pertama, Tim Berners-Lee
13. **1992** *New York Times Book Review* , cerita sampul hypertext fiksi
14. **1993** *Mosaic*, National Center for Supercomputing Applications

15. **1993** *A Hard Day's Night* film berformat hypermedia pertama
16. **1993** Hypermedia encyclopedias terjual lebih banyak dari bentuk cetakannya

*Nielsen* melihat perkembangan hypertext berdasarkan tonggak (mile stone) dimana terjadi perkembangan yang cukup signifikan dalam sejarah. Pengembangan ini berupa alat, teknologi, ataupun penggunaan hypertext itu sendiri. Nielsen menganggap ide dari Vannevar Bush mengenai mesin pintar dengan link-link nya yang bisa tersimpan sebagai ide awal pengembangan hypertext.

*Neil Ridgway* menganggap ada tiga tokoh utama yang paling penting dibalik pengembangan hypertext. Tokoh-tokoh tersebut adalah *Vannevar Bush, Engelbert*, serta *Nelson* dengan ide dan ciptaannya masing-masing.

1. Pertama adalah *Vannevar Bush* dengan mesin Memex-nya. Tahun 1945 Bush sudah memperkirakan akan pertumbuhan literature sain yang sangat pesat, dan dia berkeinginan untuk menciptakan suatu cara dimana informasi dalam jumlah besar dapat dilihat (browse) sekaligus. Dalam salah satu artikelnya, Bush menjelaskan tentang bagaimana pikiran manusia bekerja dengan merangkai informasi. Dia mengaplikasikan konsep ini menjadi suatu mesin yang disebut Memex, yang memungkinkan pengguna merangkai beberapa potongan yang relevan menjadi suatu informasi, dari dokumen yang berbeda. Ide ini dikenang orang sebagai ide pertama yang menjelaskan hypertext.

2. Kedua adalah *Doug Engelbarts* dengan mesin oN Line System (NLSatau Augment) yang dibuatnya. Pada tahun 1963, Engelbarts menjelaskan suatu sistem komputer yang akan memperkaya kemapuan intelektual manusia, dengan memungkinkan pengguna berinteraksi menggunakan beberapa perangkat kerjasama khusus. Hasilnya adalah peningkatan dalam jumlah informasi yang bisa dikelola secara efektiv oleh kemapuan dasar manusia tersebut. NLS ini diimplementasikan 5 tahun kemudian pada Stanford Research Institute. Mesin ini memungkinkan pengguna merelasikan bagian antar dokumen atau dalam dokumen itu sendiri.
3. *Nelson* dengan sistem XANADU nya. Pada saat NLS sedang dibuat, Ted Nelson juga sedang mematangkan sebuah ide mengenai mesin pemerkaya kemapuan tersebut. Sistem yang diciptakan Nelson memungkinkan pengeditan atau perubahan isi dari dokumen yang ada sesuai dengan format aslinya saja. Dengan menggunakan link ke belakanag, maka isi dokumen asli dapat diketahui. Dokumen ini disimpan dalam satu media penyimpanan sehingga perubahan-perubahannya bisa dilacak dengan mudah. Sistem yang diciptakan *Nelson* ini memungkinkan penggunan membuat hubungan atau keterkaitan antar bagian dokumen. Nelson menuangkan idenya ini melalui sebuah buku yang diberi judul *Literary Machines*

## 2.13 Pengertian *Hypertext*

Smeaton(1991) di dalam Ellist(1996) juga menyatakan bahwa hiperteks dan temu-kembali informasi itu saling berkomplemen satu sama lain. Hiperteks membutuhkan lebih banyak *searching* sedangkan temu-kembali informasi membutuhkan lebih banyak *browsing*. Hal yang dimaksud adalah hiperteks akan semakin baik jika disertai dengan fasilitas *search*, dan temu-kembali informasi membutuhkan *browsing* dalam melakukan pencarian yang efisien. Adapun maksud dari *searching* adalah berusaha mendapatkan atau mencapai tujuan spesifik sedangkan *browsing* adalah mengikuti suatu path sampai mencapai suatu tujuan. Menurut Brown(1988) didalam Agosti(1993), *browsing* itu bisa diibaratkan dengan *From Where to What*. Maksudnya adalah kita tahu dimana posisi kita dalam *database* dan kita ingin tahu apa yang ada disana (*database*). Sedangkan *Searching* bisa diibaratkan dengan *From What to Where*. Maksudnya adalah kita tahu apa yang kita inginkan dan kita ingin menemukan dimana dia didalam *database*.

Dalam terminologi yang diberikan oleh Konsorsium W3, hypertext diartikan sebagai suatu teks yang tidak dibatasi oleh linieritas (*Text which is not constrained to be linear)*. Definisi ini disamakan dengan Hypermedia, dimana Hypermedia dinyatakan sebagai Multimedia Hypertext, dan digunakan dengan arti yang sama atau dapat saling dipertukarkan.

*Neil Ridgway* menyatakan bahwa Hypertext adalah perluasan dari bentuk tradisional atau linier text menjadi text yang tidak linier. *Ridgway* mencontohkan penggunaan sistem komputer yang baru yang memungkinkan penggunanya

membuat referensi dari bagian mana saja didalam teksnya ke suatu tempat, baik dalam dokumen atau file yang sama ataupun ke dokumen atau file eksternal. Hypertext juga sering disebut sebagai non linier text, karena dalam bagian-bagian tertentu bisa merujuk ke bagian lain secara tidak sekuen sesuai dengan alamat rujukan yang diberikan. Rujukan atau *link* ini diantaranya yang membedakan dengan teks konvensional sebelumnya. Pembaca bisa mengikuti jalur sesuai dengan preferensinya, dan tidak harus melalui jalur yang sama.

*Nancy Kaplan* dalam tulisannya yang dimuat dalam jurnal *Computer-Mediated Communication Magazine* mendefinisikan Hypertext sebagai : *multiple structurations within a textual domain*. Kaplan mencontohkan dengan sebuah buku yang bisa dibaca kapan saja, dimulai dari bagian mana saja, dan bagian-bagian tersebut bisa memiliki hubungan dengan bagian lain. Dia mencontohkan suatu bentu yang disebutnya sebagai *proto* hypertext dalam bentuk cetakan seperti *Ensiklopedia*, kamus, dan bentuk-bentuk buku manual atau buku panduan. Dimana pembaca bisa mulai dari bagian mana saja sesuai dengan keinginannya, dan setelah itu bisa pergi ke bagian atau halaman mana saja sesuai dengan kebutuhannya. Contoh lain yang diberikan *Kaplan* adalah program pertelivisian di Inggris yang disebut *CEEFAX*, dimana dengan menggunakan beberapa tombol yang ada di remote kontrol kita bisa mengakses atau melihat daftar acara yang disajikan.

*George P. Landow* mendefinisikan hypertext sebagai : *text composed of blocks of words (or images) linked electronically by multiple paths, chains, or trails in an open-ended, perpetually unfinished textuality described by the terms*

*link, node, network, web , and path*. *Landow* mendefinisikan sebagai satu kumpulan kata-kata atau gambar, yang terhubung secara elektronik dengan banyak jalur, kaitan, atau jejak yang terbuka, secara terus-menerus tidak pernah selesai secara tekstual, yang dijelaskan dengan terminologi link, simpul, jaringan, web dan jalur. Defenisi *Landow* menekankan pada kumpulan kata atau gambar. Kata ini saling terkait, dan bisa melalui beberapa jalur (bukan satu jalur saja). Kaitan antar kumpulan ini terus berkembang dan kemungkinan tidak ada habisnya. *Landow* juga memberikan penekanan pada terminologi link yang menunjukkan hubungan, node yang menunjukkan masing-masing simpul atau bagian, jaringan dan web yang berguna untuk menjadi prasarana penghubung masing-masing simpul.

*Darlene Cardillo & Kimberly Kenyon* dari University at Albany menjelaskan Hypertext sebagai suatu bentuk presentasi nonlinier dengan banyak pilihan jalur informasi, yang memungkinkan pembaca berinteraksi secara interaktif dengan setiap teks. Lebih jauh dijelaskan hypertext telah membuat bentuk baru dari lingkungan membaca dan menulis yang mendukung pengembangan materi pembelajaran secara interaktif seperti jurnal akademik, ensiklopedia, referensi, serta bentuk elektronik teks lainnya.

Dari beberapa defenisi tersebut, dapat dilihat adanya beberapa persamaan mengenai hypertext. Dimana hypertext dianggap sebagai bentuk teks yang tidak linier, memiliki link atau terhubung ke teks lain, baik internal maupun eksternal, dan dapat dikembangkan secara bersama-sama. Hypertext juga mengandung unsur elektronik, baik dalam pembuatan, pengaksesan, maupun pengembangan.

Walaupun ada bentuk cetak dari hypertext, tetapi sebagian besar menekankan hypertext sebagai media yang diakses secara elektronik.

## 2.14 *Search Engine*

Satu dekade ini internet berkembang demikian pesat. Jumlah situs tumbuh secara eksponensial dan nyaris tak terkendali. Jutaan topik dan layanan disuguhkan untuk memenuhi kebutuhan manusia, dan hampir tidak satu topik pun yang tidak dimiliki *internet*.

Dengan melimpahnya sajian di internet, hal itu tidak lantas membuat mudah bagi orang-orang atau tepatnya *user* untuk menemukan apa yang mereka cari. Sering kali, karena begitu banyak pilihan yang ditawarkan, *user* justru menjadi bingung apa yang mesti dilakukan dan dari mana memulai? Kondisi ini lebih merepotkan lagi bagi mereka yang belum bisa berhadapan dengan “Dunia Maya”.

Manusia dengan setumpuk idenya berusaha menemukan jalan keluar dari setiap masalah. Berbagai metode dicoba untuk menemukan solusi yang dapat mengeliminasi kendala-kendala diatas dan lagi-lagi hal ini dimaksudkan agar bisa menguntungkan manusia, atau setidaknya membuat enteng pekerjaan manusia. Dalam konteks di mana manusia harus menemukan satu titik informasi diantara lautan informasi, sudah barang tentu yang dibutuhkan adalah sebuah mesin pencari yang pintar, yang dapat menyuguhkan apa-apa yang *direquest* dalam waktu yang sesingkat-singkatnya dan dengan tingkat akurasi yang dapat diandalkan.

Apabila dilihat dari karakteristik, yang harus mampu melakukan pencarian atas berbagai topik dalam kecepatan tinggi, yang dibutuhkan manusia adalah apa yang disebut "*Search Engine*". *Search Engine* tidak lain sebuah mesin pencari yang ulet dan teliti, yang melakukan eksplorasi atas informasi-informasi yang *direquest* tanpa memandang kapan, di mana dan oleh siapa itu dilakukan. *Search Engine* dirancang oleh insinyur-insinyur teknologi informasi sefleksibel mungkin, mudah digunakan dan dengan konstruksi yang dapat dikostumasi.

### 2.14.1 Klasifikasi *Web Search Service*

Sebenarnya agak sulit mengklasifikasikan situs-situs mesin pencari. Disamping karena belum adanya referensi formal yang disepakati, kita juga memiliki banyak kriteria untuk membedakan antara engine satu dengan yang lainnya. Hal ini banyak dipengaruhi oleh corak atau 'warna' yang diperlihatkan oleh engine-engine itu sendiri. Meski demikian, bila kita meninjaunya secara umum dengan mengambil titik tolak dari *content*, fitur-fitur, desain, serta kemudahan penggunaannya, setidaknya kita mendapatkan atiga kategori engine berikut:

1. *Search Engine*
2. *Directory*
3. *Library*

#### 2.14.1.1 *Search Engine*

Dengan *search engine*, user memasukkan *keyword* baik berupa kata, kalimat, angka, kode, atau kominasi dari semuanya untuk menampilkan daftar dokumen atau alamat situs yang berhubungan dengan *keyword* yang *diinput*. Pencarian dalam *search engine* tidak berbatas dan user dapat menginput *query* paling spesifik sekalipun. Beberapa *search engine* yang termasuk kategori ini diantaranya:

1. Google yakni mesin pencari dengan database yang sangat besar, memiliki *interface simple* dan banyak orang menyukainya. Disamping menawarkan dokumen, Google juga mensupport pencarian image. Ini adalah keunggulan google diantara engine-engine lainnya.
2. GoatauInfoseek yakni *search engine* dengan menawarkan fitur *advance form* yang kaya.
3. Alta Vista yakni *search engine* dengan database yang besar dan komplit
4. Northern Light yakni *search engine* yang juga komplit dan besar
5. Savvy Search yakni merupakan *multi search engines* terbaik.

#### 2.14.1.2 *Directory*

Mesin *Directory* adalah pilihan terbaik untuk tujuan eksplorasi situs, tetapi kurang tepat untuk dokumen. Di sini *user* dapat melakukan pencarian berdasarkan kategori, seperti ekonomi dan bisnis, komputer dan ilmu pengetahuan, kesehatan,

pendidikan dan lain sebagainya. Beberapa engine yang termasuk kategori ini diantaranya:

a. *Yahoo* yakni Mesin d*irectory* dengan kategori paling lengkap, dan yahoo menyajikan katalog engine yang besar. Semua orang mengenal yahoo.
b. *Magellan* yakni Mesin *directory* dengan fitur cerdas, memberikan perankingan dan review situs
c. *LookSmart* yakni Mesin *directory* seleksi terbaik
d. *Netguide* yakni Mesin directory dengan fasilitas review situs

**2.14.1.3 *Library***

*Online Libraries* merupakan direktory dari kumpulan direktory. Database yang disuguhkan memuat file-file dokumen atau referensi. Umumnya koleksi library dengan sangat hati-hati dipilih dan dievaluasi oleh pakar-pakar kepustakaan, dengan sasaran validitas dan kualitas.

a) World Wide Web Virtual Library yakni Merupakan Web library yang besar dan terkelola dengan baik
b) Argus Clearinghouse yakni Membuat sederet koleksi resource yang lengkap

Disamping klasifikasi atau kategori diatas, terdapat juga apa yang disebut situs *metasearch* atau *metacrawler,* yang bekerja mengirim aksi-aksi pencarian ke beberapa *search engine*. Oleh karena metasearch engine tidak memungkinkan user untuk menginput banyak variable pencarian, maka penggunaan terbaiknya adalah untuk menemukan item-item yang samar atau sekedar melihat-lihat apa

yang berkembang diinternet. Beberapa yang terkenal dari kategori ini adalah Dogpile, Mamma, Metacrawler, dan Savvy Search.

### 2.14.2 Mengevaluasi Aplikasi *Search Engine*

Sebuah *search engine* akan berhadapan langsung dengan *interface user*, melayani *user* menemukan *resource-resource* spesifik melalui berbagai metode pencarian. Dalam hal ini kebanyakan *user* tidak ambil peduli dengan apa sesungguhnya yang dilakukan *search engine* guna memenuhi *request-request* yang masuk kepadanya. Yang penting begitu pijit tombol, search engine harus menyodorkan hasilnya dalam satu atau beberapa detik.

Bila kita kaji secara teknis, sebuah aplikasi search engine sebetulnya memikul beban kerja yang berat menangani satu buah *query* saja. Sebagaimana dijelaskan sebelumnya, search engine akan melewati tahapan-tahapan proses yang kompleks untuk menemukan hasil akhir. Disamping itu, ia juga harus memperhatikan faktor-faktor ketepatan, dan ini bukanlah tugas yang ringan. Hanya aplikasi-aplikasi cerdas saja yang mampu melakukannya.

#### 2.14.2.1 Fitur-fitur Umum *Search Engine Internet*

Fitur search engine internet sebenarnya sangat relative dan melibatkan banyak factor. Di samping tergantung pada teknologi yang digunakan search engine, yang terus dilakukan perbaikan dan pengembangan dari waktu ke waktu oleh pengelolahnya, hal itu juga tergantung dari sudut mana kita menilai.

### 2.14.3 Anatomi *Search Engine*

Bila kita tinjau dari anatomi dan strukturnya, sebuah aplikasi search engine dibentuk oleh sekumpulan program terotomasi. Mereka dikenal sebagai *spider* atau *crawlers*, yang berfungsi mengambil informasi dari internet. Kesatuan dari fungsi-fungsi ini sering juga disebut *crawling*.

Secara garis besar, crawling search engine pada umumnya terdiri dari lima bagian utama

1. *Crawler*
2. *Spider*
3. *Indexer*
4. *Database (the "index")*
5. *Result Engine*

#### 2.14.3.1 *Crawler*

Crawler adalah program terotomasi yang memproses *link-link* yang ditemukan dalam halaman-halaman web, yang kemudian menunjukkan spider untuk mengunjungi situs-situs tertentu yang baru ditemukan. Saat *spider* mendownload halaman-halaman, ia melakukan 'pengintaian' atas link-link. Mereka dapat dengan mudah melakukannya karena selalu menemukan *item* yang sama. Selanjutnya *crawler* menunjukkan ke mana spider harus pergi (didasarkan link-link dan list URL yang ada). Seringkali link-link baru yang dia temukan saat kunjungan kembali ke sebuah situs kemudian ditambahkan ke dalam list. Saat

anda menambahkan sendiri sebuah URL ke search engine, rogram crawler akan mengecek *request* anda dengan mengunjungi situs tersebut.

#### 2.14.3.2 *Spider*

Spider adalah bagian program otomatis yang berperan untuk *mendownload* dokumen-dokumen yang ditemukan dalam suatu web atas referensi *crawler*.

Program spider bekerja sangat sibuk dan dalam kecepatan tinggi. Layaknya sebuah browser, ia melakukan *download* banyak halaman (dalam *environment* yang besar bisa mencapai ratusan ribu). Kebanyakan spider tidak melakukan download atas *image*, dan tidak diperintahkan untuk mengirim. Jika anda penasaran apa yang dilihat dan diseleksi spider saat berkunjung ke sebuah halaman *web*, silahkan klik kanan *button mouse* anda, kemudian pilih *view source* pada menu yang muncul. Anda akan melihat kode-kode *script* dari halaman web tersebut. Inilah yang dipelajari oleh spider.

#### 2.14.3.3 *Indexer*

Program indexer memiliki tugas "membaca" halaman-halaman yang telah di download spider. Di sini indexer mempelajari tentang apakah subjek dari *site* anda tersebut. Beberapa kata yang terkategori umum akan di *reject* (seperti *and*, *it, the*, dan semacamnya). Indexer akan memeriksa kode HTML guna menemukan kata-kata penting yang dikandung oleh situs yang dibaca. Kata-kata yang dicetak tebal (*bold*), miring (*italic*), dan *tag-tag header* akan lebih diperhatikan. Analisis juga

akan difokuskan terhadap informasi-informasi *meta*, termasuk *tag-tag keyword* dan deskripsi.

#### 2.14.3.4 *Results Engine*

Sebagai program penutup dan sekaligus berperan dalam menggenerasikan hasil pencarian (dari database) atas setiap *query* yang diinput *user*, program ini adalah bagian terpenting dalam search engine.

*Result Engine* adalah porsi *customer facing*. Oleh sebab itu disini diperlukan usaha optimasi yang maksimal karena ia akan berhadapan langsung dengan *interface user*. Result Engine harus mampu memperhatikan output yang akurat dan relevan dengan apa yang direquest user.

Saat seorang user mengetik sebuah keyword atau kalimat yang dicari, result engine harus memutuskan halaman-halaman mana saja dari sekian ribu halaman yang lebih mendekati dengan keinginan user. Metode yang berperan mengolah keputusan ini adalah apa yang disebut “algoritma”. Sebagai informasi tambahan, spider dan crawler sering dipanggil juga “robots” terutama dalam konteks dokumen-dokumen *official robots exclusion standar*.

## 2.15 Analisis dan Perancangan Sistem

Perancangan sistem merupakan tahap setelah analisis dan siklus pengembangan sistem, pendefinisian kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi, menggambarkan suatu sistem yang akan dibentuk.

## 2.16 Sejarah *Internet*

Pada tahun 1969 ARPA (Advanced Research Project Agency), sebuah bagian dalam kementerian pertahanan Amerika Serikat amemulai sebuah proyek, yang disatu sisi menciptakan jalur komunikasi yang tak dapat dihancurkan dan disisi lain memudahkan kerja sama antar badan riset diseluruh negeri, seperti juga industri senjata. Maka terbentuklah ARPANet.

Proyek ARPANET merancang bentuk jaringan, kehandalan, seberapa besar informasi dapat dipindahkan, dan akhirnya semua standar yang mereka tentukan menjadi cikal bakal untuk pengembangan protokol baru yang sekarang dikenal sebagai TCPatauIP *(Tranmission Control Protocol atau Internet Protocol)*

ARPANET dibentuk secara khusus oleh empat universitas besar di Amerika, yaitu Stanford Research Institute, University of california at Santa Barbara, University of California at Los Angeles, dan University of Utah, dimana mereka membentuk suatu jaringan terpadu di tahun 1969, dan secara umum ARPANET diperkenalkan pada bulan Oktober 1972. Pada tahun 1981, jumlah komputer yang tergabung dalam ARPANET hanya 213 komputer, kemudian di awal tahun 1986, bertambah menjadi 2308 komputer, dan 1,3 juta komputer pada tahun 1993. Pada awal tahun delapan puluhan seluruh jaringan yang tergabung dalam ARPANET diubah menjadi TCPatauIP, karena proyeknya sendiri sudah dihentikan, dan jaringan ARPANET inilah yang merupakan koneksi utama *(backbone)* dari *internet*.

Proyek percobaan tersebut akhirnya dilanjutkan dan dibiayai oleh NSF *(National Science Foundation),* suatu lembaga pengetahuan seperti LIPI di

Indonesia. NSF lalu mengubah nama jaringan ARPANET menjadi NSFNET dimana *backbone*nya memiliki kecepatan tinggi yang dihubungkan ke komputer-komputer yang ada di universitas dan lembaga penelitian terbesar di Amerika. Setelah itu pemerintah Amerika Serikat memberikan ijin ke arah komersial pada awal tahun 1950. jelaslah bahwa awalnya *internet* dikenal sebagai suatu wadah bagi para peneliti untuk saling bertukar informasi kemudian dimanfaatkan oleh perusahaan-perusahaan komersial sebagai sarana bisnis mereka. Sat ini pengguna *internet* tersebar di seluruh dunia dengan jumlah mencapai lebih dari 250 juta orang.

### 2.16.1 Fasilitas *Internet*

Saat ini jika orang berbicara tentang *internet*, yang mereka maksud adalah *word wide web (www).* Pada kenyatannya *internet* memiliki banyak bagian yang lain, yaitu:

1. *World wide web* disingkat dengan web adalah bagian yang paling menarik dari *internet*. Melalui *web* kita bisa mengakses informasi yang tidak hanya berupa teks, tetapi juga gambar-gambar, suara-suara, film, dan lain-lain. Untuk mengakses web dibutuhkan *software* yang disebut *browser*. *Browser* yang terpopuler saat ini adalah microsoft Internet Explorer dan Netscape Communicator.
2. *Electronic Mail* disingkat *E-mail* adalah surat elektronik yang dikirim melalui *internet*. Dengan fasilitas ini kita bisa mengirim atau menerima *email* dari dan ke pengguna *internet* di seluruh dunia. Jika dibandingkan

dengan pos, fasilitas *email* jauh lebih cepat kita bisa mengirim email ke teman yang ada di luar negeri hanya dalam beberapa menit dan tidak perlu khawatir, surat tersebut akan rusak karena hujan atau hal-hal lainnya. Untuk mengirim atau menerima *e-mail* dibutuhkan program *e-mail* yang terdapat di Microsoft Internet Explorer dan Netscape Communication. Yang perlu dibutuhkan hanya mengetik surat tersebut dan mengirimkannya. Di *internet* juga banyak tersedia *mailing list*, fasilitas yang memungkinkan kita untuk saling berbagi informasi mengenai topik-topik tertentu dengan orang lain. Jika kita bergabung dengan salah satu *mailing list*, kita bisa menerima dan mengirim *e-mail* ke semua anggota *mailing list* tersebut.

3. Telnet. Dengan menggunakan telnet kita bisa menggunakan komputer untuk berhubungan dengan komputer lain dan mencari atau mengambil informasi-informasi yang ada di komputer tersebut.
4. *File Transfer Protokol* disingkat FTP. Melalui *software* FTP, anda bisa mengirim data atau file di sebuah komputer ke komputer lain. Proses mengirim file dari sebuah komputer ke komputer anda disebut dengan proses *download*, sedangkan proses mengirim file dari komputer anda ke komputer lain disebut dengan *upload*. Jika anda pernah membuat *homepage*, maka kata *upload* tidak asing lagi.
5. *Gopher* adalah sistem dimana pemakai dapat mengakses informasi dari komputer lain. Beda *gopher* dengan *web* adalah tidak bisa menampilkan

gambar, melainkan hanya teks. Oleh sebab itu *gopher* banyak ditingalkan para pemakai *internet* saat ini.

6. *Chat Group* atau *Internet Relay Chat* (IRC) adalah forum dimana pemakai dapat saling berdiskusi atau berbincang-bincang dengan pemakai lain.
7. *News Group* bisa disebut ruang percakapan bagi para anngota yang mempunyai kemungkinan yang sama. Di *internet* tersedia bermacam-macam news group dengan tema yang berbeda-beda. Untuk dapat menikmati fasilitass ini kta harus terkoneksi ke ISP yang menyediakan fsilitas *news group*, karena tidak semua ISP menyediakannya. Adapun fasilitas yang anda gunakan semuanya merupakan bagian dari *internet*. Melalui fasilitas-fasilitas *internet* itu, kita bisa melakukan banyak hal yang akan menambah pengetahuan dan wawasan kita.

### 2.16.2 Syarat Tergabung ke *Internet*

Untuk berhubungan dengan i*nternet* tentunya harus mempunyai komputer. *Internet* adalah jaringan komputer, oleh karena itu jalan satu-satunya untuk berhubungan dengan *internet* adalah melalui komputer. Selain itu juga diperlukan *harddisk* yang berisi *software* untuk berhubungan dengan *Internet Service Provider* dan internet seperti *web browser*, dan *harddisk* untuk menyimpan informasi-informasi yang anda *download*. Untuk mengakses *internet* tidak cukup hanya dengan komputer saja, tetapi ada alat bantunya seperti:

1. ***Modem***. *Modem* adalah perangkat *hardware* tambahan untuk komputer (baik jenis *card internal* maupun *eksternal* yang terletak di luar komputer).

Pada dasarnya *modem* adalah alat yang memungkinkan komputer anda untuk berbicara dengan komputer lain melalui kabel telepon. Kata *modem* berasal dari kata Modulasi Demodulasi yang bisa diartikan sebagai proses perubahan denyut elektronis dari komputer menjadi suara (modulasi) sehingga dapat dikirimkan melalui kabel telepon. Modem penerima akan mengubah nada suara tadi menjadi denyut elektronis kembali untuk diproses selanjutnya oleh komputer (demodulasi).

2. **Telepon *Modem*** memerlukan telepon untuk melakukan tugasnya. Proses pada saat *modem* terhubung dengan telepon dan anda memulai hubungan *internet service privider* pada dasarnya sama dengan proses anda menelepon biasa. Jika ada yang menelepon anda pada saat menggunakan modem, maka dia akan menerima nada sibuk dan tidak bisa menghubungi anda, karena *modem* dan telepon berkaitan erat, kecepatan *modem* juga akan berpengaruh terhadap rekening telepon yang harus anda bayar setiap bulan, sehingga *modem* yang cepat akan menghemat uang daripada *modem* yang lambat.
3. ***Software***. *Software* diperlukan untuk berhubungan dengan *internet*. Tanpa *software*nya sesuai, anda tidak bisa mengakses *internet*. Contoh *software* yang diperlukan adalah:
   a. *Microsoft Internet Explorer* dan *Netscape Communicator* adalah *browser* untuk mengakses *web* yang paling baik dan terpopuler saat ini. Kedua *browser* ini juga sekaligus berfungsi sekaligus berfungsi sebagai FTP*, mail* dan *news group*.

b. Cute FTP, Mozilla dan WSFTP, adalah *software* khusus untuk FTP

c. Untuk *e-mail* anda bisa menggunakan *Internet Mailatau Outlook express* yang merupakan bagian dari *Netscape Communicator*.

d. MIRC, *Yahoo Messenger*, atau ICQ adalah program untuk *chating* yang paling popular saat ini. Dengan program ini anda bisa berbincang-bincang dengan orang lain melalui *keyboard*.

*e.* *Internet Service Provider*. Untuk bisa bergabung dengan *internet*, anda harus mempunyai akses dengan cara berlangganan ke penyedia jasa *internet* atau yang lebih sering disebut *Internet Service Provider* (ISP). ISP adalah perusahaan yang menawarkan jasa pelayanan kepada anda untuk berhubungn dengan *internet*. Untuk mengakses, anda perlu cukup menghubungi Internet Service Privider melalui komputer dan modem, selanjutnya *Internet Service Privider* akan mengurus detail-detail yang diperlukan untuk berhubungan dengan internet termasuk biaya SLJJ koneksi tersebut.

f. Sambungan langsung ke Network Anda dapat menggunakan sebuah komputer yang secara langsung mempunyai hubungan ke *INTERNET*. Sebagai contoh, Anda mungkin menggunakan sebuah PC yang merupakan bagian dari sebuah jaringan komputer yang mempunyai hubungan ke *INTERNET*. Dalam kasus ini, sistem Anda menjadi host *INTERNET* penuh, yaitu mempunyai alamat elektronik tersendiri.

g. Sambungan dengan menggunakan SLIPatauPPP. Untuk menggunakan hubungan dial-up telepon, Anda memerlukan sebuah alat untuk

mengkonversi sinyal komputer (digital) menjadi sinyal telepon (analog), dan sebaliknya. Alat untuk mengkonversi sinyal digital ke sinyal analog disebut *modulator*. Sedang, alat untuk mengkonversikan sinyal analog ke sinyal digital disebut *demodulator*. Untuk mengakses ke *INTERNET* melalui hubungan telepon, Anda memerlukan sebuah modem (*modulator-demodulator*). Selain itu diperlukan juga TCPatauIP dan software SLIP atau PPP seperti Linux, Warp, dll.

h. Sambungan langsung ke On-line Service seperti BBS, Compuserve. Untuk menjadi sebuah host *INTERNET* tanpa harus memiliki hubungan full-time ke *INTERNET* (yang umumnya sangat mahal), ada sebuah cara mensetup sebuah host *INTERNET* melalui hubungan telepon. Untuk melakukan hal tersebut, Anda perlu mengadakan perjanjian dengan sebuah host *INTERNET* yang lain yang bertindak sebagai titik hubungan. Selanjutnya, diperlukan sejumlah program yang disebut sebagai PPP (Point to Point Protocol) dan SLIP (Serial Line Internet Protocol) dalam workstation. Setelah workstation menghubungi host *INTERNET* melalui jalur telepon, PPP menyediakan kemampuan TCPatauIP untuk workstation tersebut.

## 2.17 Web Server

Web server merupakan salah satu tulang punggung penyedia layanan di Internet atau server di Internet. Apache merupakan web server yang terpopuler saat ini dimana survey netcraft lebih dari 50% situs di Internet menggunakan

Apache sebagai Web Server. Selain karena Apache lebih cepat juga luwes karena dapat di implementasikan untuk berbagai system operasi, mudah dikonfigurasi, dapat ditambahkan modul lain melalui module Apache.

Apache menyediakan fasilitas yang kaya, yang sangat dibutuhkan suatu server serius, seperti otentikasi, pengaturan akses direktori, virtual host, kemampuan URL rewritting, dan juga alias. Kemampuan melakukan content negotiation membuat Apache mampu melayani beragam client secara otomatis, baik untuk berbagai browser yang memiliki kemampuan berbeda. Fungsi log yang dimiliki oleh Apache dapat dikirim melalui proses piping, sehingga dapat dilakukan rotasi log, fillter log, serta melakukan pemisahan log secara langsung.

Awalnya Apache dikembangkan berdasakan keinginan untuk memperbaiki Web server yang saat itu populer (NCSA web server). Tetapi akhirnya mengalami perombakan dan penulisan ulang dan menjadi Web server yang berdiri sendiri dan berbeda dengan NCSA. Kini malah mengalahkan kepopuleran NCSA Web server.

Pada tahun 1999 dibentuk Apache Software Foundation untuk mengurus perkembangan Apache ini. Apache telah membuktikan sebagai web server yang cepat, stabil dengan fitur yang paling kaya di antara web server lainnya. Saat ini proyek Apache telah berkembang dan tidak hanya sekedar Web server. Apache dapat memberikan layanan yang cukup banyak bagi penggunakanya karena memiliki program pendukung yang cukup banyak seperti control akses, CGI, PHP dan lain sebagainya.

## 2.18 Alat Bantu Pemodelan

### 2.18.1 Flowchart

*Flowchart* adalah bagan alir yang menunjukkan arus dari dokumen berupa inputan, proses, dan output..

### 2.18.2 Diagram Konteks

*Diagram Konteks* adalah merupakan alat-alat untuk struktur analisis. Dan pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan.

### 2.18.3 Data Flow Diagram (DFD)

*Data Flow Diagram* adalah suatu jaringan dari proses dengan tempat penyimpanan data serta dihubungkan satu dengan lainnya, atau kumpulan simbol-simbol yang menggambarkan jalannya aliran data dari sistem atau suatu diagram yang mudah dimengerti dan merupakan suatu gambaran mengenai tata letak lokasi dan semua kegiatan-kegiatan dan aktivitas yang terjadi di dalam proses aktivitas tersebut.

### 2.18.4 Kamus Data

Kamus data adalah katalog fakta tentang data kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan kamus data analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis

sistem, kamus data digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir ke sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang *input,* merancang laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD.

Kamus data tidak menggunakan notasi grafik sebagaimana halnya DFD. Kamus data berfungsi membantu pelaku sistem untuk mengerti aplikasi secara presisi, sehingga pemakai dan penganalisa sistem punya dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

Cara mendefinisikan kamus data adalah:

1. Menggambarkan arti aliran data atau penyimpanan yang ditunjukan dalam *Diagram Flow Datagram.*
2. Menggambarkan komponen dari kumpulan data yang mengalir, yaitu kumpulan komponen yang mengkin bisa dipecahkan menjadi data elementer.
3. Menggambarkan data yang tersimpan.
4. Menentukan nilai di bagian elementer dari informasi yang relevan *Diagram Flow Datagram* dan *file*.

**2.18.4.1 Basis Data**

Basis data terdiri atas dua kata yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), 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 :

a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.

b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan.

c. Kumpulan fileatautabelatauarsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

## 2.19 HTML

*Hypertext Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman website. HTML merupakan pengembangan dari standard pemanfaatan dokumen teks yaitu *Standar Generalized Markup Language* (SGML).

Dokumen HTML sebenarnya hanya berupa *file* teks biasa yang dilengkapi dengan tanda-tanda khusus (tag) yang menentukan bagaimana teks tersebut ditampilkan. Karena itu, dokumen HTML dapat dibuat menggunakan teks editor sederhana seperti Notepad, EditPlus, dan sebagainya. Namun sekarang ini telah

banyak visual editor untuk HTML seperti: Microsoft Front Page, Macromedia Dreamweaver, dan lain-lain.

Tag HTML biasanya adalah tag-tag yang berpasangan dan ditandai dengan simbol "<" dan ">", sedangkan akhir perintah dari sebuah tag ditandai dengan tanda "atau". Dokumen HTML biasanya disimpan dengan extension .HTM atau .HTML. Untuk membuka dokumen HTML ini, diperlukan sebuah Web Browser.

Struktur sebuah dokumen HTML pada dasarnya dibagi menjadi dua bagian besar, yaitu *header* dan *body*. Masing-masing ditandai oleh pasangan *container tag* <*head*> dan <*body*>. Bagian *head* berisikan judul dokumen dan informasi-informasi dasar lainnya sedangkan bagian *body* adalah data dokumennya. Pengaturan format teks dan pembentukan *link* dilakukan terhadap objeknya langsung dengan ditandai oleh *tag-tag* HTML.

HTML diatur oleh konsorsium WWW (W3C) semua perubahan atas standar bahasa HTML harus disahkan terlebih dahulu oleh WWW. Sejauh ini, HTML telah mengalami berbagai revisi sepanjang hidupnya, standar paling akhir yang sekarang diperkenalkan adalah standar HTML 4.0, yang mendukung antara lain CSS *(Cascading Style Sheet)*, *Dynamic Content Positioning* (Penempatan isi secara dinamis).

## 2.20 CSS

CSS merupakan kependekan dari **Cascading Style Sheet** yang memungkinkan untuk mendesain (style) tampilan dokumen (terutama HTML) dengan memisahkan isi dari dokumen HTML dengan kode untuk menampilkannya (CSS). Jika memiliki banyak file HTML, hanya perlu satu file CSS, sehingga ketika mengganti jenis huruf pada file CSS maka semua file HTML yang berhubungan pada file CSS tersebut akan berubah. CSS distandarisasi oleh W3C (World Wide Web Consortium). CSS dapat dipasang pada dikumen HTML yang telah jadi.

Ada 3 cara untuk memasang CSS pada dokumen HTML yaitu: External Style Sheet (file CSS berbeda dari file HTML), Internal Style Sheet (Kode CSS dipasang di dalam tag head HTML) dan Inline Style Sheet (Kode CSS langsung dipasang di tag HTML, tidak direkomendasikan). namun biasanya standar yang digunakan adalah External Style Sheet karena lebih mudah dalam mengelolanya.

Contoh Kode program dari CSS:

```
<html>
<head>
<title>Belajar CSS<atautitle>
<style type="textataucss">
h1 { font-family: verdana; }
<ataustyle>
<atauhead>
<body>
```

<h1>Belajar CSS<atauh1>
<ataubody>
<atauhtml>

CSS terdiri dari dua bagian utama yaitu: selector, dalam hal ini H1 dan deklarasi yang berada diantara kurung kurawal {font-family: verdana}. Didalam deklarasi juga terbagi menjadi dua bagian yaitu property dalam hal ini font-family dan value dalam hal ini verdana. Dalam contoh diatas hanya mengubah sebuah tag yaitu tag <h1> menjadi teks dimana jenis hurufnya menjadi verdana. Agar dapat mengkombinasikan berbagai macam style menjadi satu.

**2.21 XAMPP**

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 menginstalasi 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 pengunaan *Apache web server.*

*XAMPP sendiri menawarkan* berberapa paket pengistallan antara lain :

1. Apache 2.2.6
2. MySQL 5.0.45
3. PHP 5.2.4 + PHP 4.4.7 + PEAR
4. PHP-Switch win32 1.0
5. XAMPP Control Version 2.5 from www.nat32.com
6. XAMPP Security 1.0
7. SQLite 2.8.15
8. OpenSSL 0.9.8e
9. phpMyAdmin 2.11.1
10. ADOdb 4.95
11. Mercury Mail Transport System v4.01b
12. FileZilla FTP Server 0.9.23
13. Webalizer 2.01-10

Cara mengaktifkan Xampp:

1. Instal XAMP, lalu aktifkan web server Apache dan MySQL dari control panel XAMPP.

2. Jalankan browser (IE, Mozilla Firefox atau Opera) lalu ketikkan alamat web berikut : http:atauataulocalhostatauphpmyadminatau pada address bar lalu tekan Enter.

3. Apabila telah nampak interface (tampilan antar muka) phpMyAdmin bisa memulainya dengan mengetikkan nama database, nama tabel dan seterusnya.

## 2.22 PHP

PHP singkatan dari *Personal Home Page Tools*, adalah sebuah bahasa *scripting* yang dibundel dengan HTML, yang dijalankan disisi server. Sebagian besar perintahnya berasal dari C, Java dan dengan beberapa tambahan fungsi khusus PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan server. PHP dapat dijalankan pada berbagai macam Operating System (OS), misalnya Windows, Linux dan Mac OS. Selain Apache, PHP juga mendukung beberapa web server lain, misalnya Microsoft IIS, Caudium, PWS dan lain-lain.PHP dapat memanfaatkan database untuk menghasilkan halaman web yang dinamis. Sistem manajemen database yang sering digunakan bersama PHP adalah MySQL. Namun PHP juga mendukung system manajemen Database Oracle, Microsoft Acces, Interbase, d-Base, PostgreSQL dan sebagainya.

PHP menawarkan solusi yang lebih luwes. Dengan PHP, developer tidak perlu lagi berurusan dengan dua buah file terpisah seperti itu. *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 menetukan apakah ada kode di dalam file tersebut yang akan dieksekusi. Bila ada, kode-kode tersebut akan dieksekusi. Hasilnya akan

dimasukkan di dalam dokumen yang sama. Server web bekerja secara langsung terhadap *file* yang bersangkutan, tidak memanggil *script* terpisah seperti pada metode CGI. Seluruh kode dieksekusi di Server (oleh karena itu disebut *server-side script*).

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

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* dapat berkonsentrasi langsung pada penampilan dokumen *web*nya.
2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi objek (*Object Oriented*).
4. Sintaksis pemogramannya mudah dipelajari, sangat menyerupai C dan Perl.
5. Integrasi yang sangat luas ke berbagai server database.

PHP tidak terbatas untuk hanya menghasilkan keluaran HTML. Ia juga bisa digunakan untuk menghasilkan gambar GIF, atau bahkan sumber gambar GIF yang dinamis.

Hingga kini PHP sudah berkembang hingga versi ke 5. PHP 5 mendukung penuh ***Object Oriented Programing(OOP)***, integrasi XML, mendukung semua ekstensi terbaru MySQL, pengembangan web services dengan SOAP dan REST, serta ratusan peningkatan kemampuan lainnya dibandingkan versi sebelumnya. Sama dengan web server lainnya PHP juga bersifat open source sehingga setiap orang dapat menggunakannya dengan gratis.

## 2.23 MySQL

Pada awal perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah database. SQL pertama kali didefinisikan oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen database yang bersifat open source. MySQL adalah pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia.

MySQL dapat digunakan untuk membuat dan mengola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sisitem manajemen database yang bersifat ***at relational.*** Artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat.

MySSQL (*My Strukture Query Language*) atau yang biasa dibaca " mai-se-kuel" adalah salah satu jenis *database* server yang sangat terkenal. MYSQL sangat popular karena MYSQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*. MYSQL bersifat *free* pada berbagai *platform* (kecuali pada Windows, yang bersifat *shareware* atau anda perlu membayar setelah melakukan evaluasi yang memutuskan untuk digunakan untuk keperluan produksi). Perangkat lunak MYSQL sendiri bisa di download dari http:atauatauwww.mysql.org atau http:atauatauwww.mysql.com.

MySQL dalam operasi klien-server melibatkan server daemon MySQL di sisi server dan berbagai macam program serta pustaka yang berjalan di sisi klient. MySQL mampu menangani data yang 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. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah ***Structured Query Language (SQL)*** untuk mengelola database-database yang ada di dalamnya. Hingga kini, MySQL sudah berkembang hingga versi 5. MySQL 5 sudah mendukung trigger untuk memudahkan pengelolaan tabel dalam database.

### 2.24 Dreamweaver

Dreamweaver dalah suatu bentuk program editor web yang dibuat oleh makromedia. Dengan program ini seorang programmer web dapat dengan mudah membuat dan mendesain webnya.

Dreamweaver merupakan editor yang komplit, yang dapat digunakan untuk membuat animasi sederhana yang berbentuk layer. Dengan program ini pembuatnya tidak akan susah-susah mengetik script-script format HTML, PHP, ASP maupun bentuk program yang lainnya.

Dengan kelebihan ini, seorang programmer dapat langsung melihat hasil buatannya tanbpa harus di browser. Dreamweaver MX juga memiliki dua bentuk layer, yaitu bentuk halaman *Design* dan halaman *Code*. Dua bentuk layer ini akan

mempermudah dalam menambahkan *script* yang berbasis PHP maupun javascript. Dreamweaver MX, selain mendukung pembuatan web yang berbasis HTML, juga dapat mendukung program–program web yang lain diantaranya PHP, ASP, Perl, Javascript, dan lain-lain.