

## BAB II

### LANDASAN TEORI

#### 2.1 Penyebab Flu Burung

Penyebab flu burung (*Bird Flu/Avian Influenza*) adalah virus influenza tipe A. Virus influenza termasuk famili Orthomixoviridae. Virus influenza tipe A dapat berubah-ubah bentuk (Drift, shift) dan dapat menyebabkan epidemi dan pandemi. Berdasarkan sub tipenya terdiri dari *Hemagglutinin* (H) dan *Neuramidase* (N). Kedua huruf ini digunakan sebagai identifikasi kode subtipe flu burung yang banyak jenisnya.

Pada manusia hanya terdapat jenis H1N1, H2N2, H3N3, H5N1, H9N2, H1N2, H7N7. Sedangkan pada binatang H1-H5 dan N1-N9. *Strain* yang sangat *Virulen/ganas* dan menyebabkan flu burung adalah dari sub tipe A H5N1. Virus tersebut dapat bertahan hidup di air sampai 4 hari pada suhu 22<sup>0</sup> C dan lebih dari 30 hari pada suhu 0<sup>0</sup> C. Virus akan mati pada pemanasan 60<sup>0</sup> C selama 30 menit atau 56<sup>0</sup> C selama 3 jam dan dengan detergent, desinfektan misalnya formalin, serta cairan yang mengandung Iodin.

##### 2.1.1 Cara Penularan Flu Burung

Flu burung menular dari unggas ke unggas, dan dari unggas ke manusia. Penularan flu burung pada manusia diantaranya bisa melalui air liur, lendir dari hidung dan *Feces* (tinja) atau debu yang dicemari tinjanya. Penyakit ini dapat menular melalui udara yang tercemar virus H5N1 yang berasal dari kotoran atau sekreta burung/unggas yang menderita flu burung. Penularan dari unggas ke manusia juga bisa terjadi jika bersentuhan langsung dengan unggas yang

terinfeksi flu burung. Misalnya pekerja di peternakan ayam, pemotong ayam. Tetapi memakan daging ayam dan telur matang tidak menyebabkan tertular flu burung.

### 2.1.2 Cara Mendiagnosa Flu Burung

Flu burung memiliki gejala yang bervariasi. Pada kasus yang sangat ganas (*Akut*) ditandai dengan kematian tinggi tanpa disertai gejala klinis. Hewan tampak sehat tetapi tiba-tiba mati. Namun pada umumnya gejala yang ditimbulkan oleh infeksi virus flu burung akan menunjukkan gejala-gejala, sebagai berikut:

1. Kasus suspek (tersangka)
  - a. Kasus Suspek adalah kategori dari penyakit flu burung yang paling ringan. Biasanya seseorang yang menderita Infeksi Saluran Pernafasan Akut (ISPA) dengan gejala: demam (temperatur lebih dari  $38^{\circ}\text{C}$ ), batuk dan atau sakit tenggorokan dan hidung beringsus.

2. Kasus Probable

Kasus "*Probable*" adalah kasus suspek dengan salah satu keadaan sebagai berikut:

- a. 7 hari (seminggu) terakhir sebelum sakit mengunjungi peternakan yang sedang terjangkit flu burung.
- b. 7 hari (seminggu) sebelum sakit kontak dengan unggas sakit atau mati atau menggunakan produk mentah unggas seperti pupuk kandang dan lain-lain.
- c. Kontak dengan kasus konfirmasi flu burung dalam masa penularan.
- d. Bekerja pada suatu laboratorium yang sedang memproses spesimen manusia atau binatang yang dicurigai menderita flu burung.

- e. Cluster (kelompok) radang paru berat (*pneumonia* berat).
- f. Pemeriksaan darah : *Leukosit* jumlah kurang dari 5000, *Limfositopenia* dan *Trombositopenia*.
- g. Hasil pemeriksaan dengan HI tes positif pada spesimen tunggal atau kenaikan titer sepasang spesimen kurang dari 4 kali.

### 3. Kasus Konfirmasi

Kasus konfirmasi adalah kasus suspek atau "*Probable*" disertai oleh salah satu hasil pemeriksaan laboratorium:

- a. Kultur virus influenza A/H5N1 positif.
- b. RT-PCR influenza (H5) positif.
- c. Peningkatan titer antibodi H5 sebesar 4 kali atau lebih pada pemeriksaan spesimen kedua dengan *Mikro Neutralization* tes.
- d. IFA tes positif (+) dengan antibodi *Monoklonal/influenza A/H5*.

### 4. Gejala Klinis / Observasi

Gejala klinis yang ditemui seperti gejala pada umumnya, yaitu: demam, sakit tenggorokan, batuk, beringsus, nyeri otot, sakit kepala, lemas. Dalam waktu singkat penyakit ini dapat menjadi lebih berat yaitu peradangan di paru-paru (*pneumonia*), dan apabila tidak cepat ditangani dengan baik dapat menyebabkan kematian.

Masa Inkubasi flu burung dapat dibedakan juga pada manusia dan unggas

- a. Pada unggas : 1 minggu
- b. Pada manusia : 1-3 hari, masa infeksi 1 hari sebelum sampai 3-5 hari sesudah timbul gejala. Pada anak sampai 2 hari.

## 2.2 Sistem Pakar

Menurut Arhami (2005:3) sistem pakar adalah suatu cabang dari AI yang membuat penggunaan secara luas *knowledge* yang khusus untuk penyelesaian masalah tingkat manusia yang pakar. Sistem pakar bertindak sebagai penasihat atau konsultan pintar dengan mengambil pengetahuan yang di simpan dalam *Knowledge Base*.

Menurut Suparman (1991:13) tujuan utama sistem pakar bukan untuk mengganti kedudukan seorang ahli atau seorang pakar, tetapi hanya untuk memasyarakatkan pengetahuan dan pengalaman pakar-pakar yang sangat langka itu. Sistem pakar bertindak sebagai penasihat atau konsultan pintar dengan mengambil pengetahuan yang di simpan dalam *Knowledge Base*. Seorang pemakai yang tidak berpengalaman sekalipun asalkan mengetahui secara umum tentang cara kerja peralatan yang didiagnosa bisa memecahkan suatu masalah yang rumit dan bisa mengambil keputusan yang tepat dan akurat layaknya yang dilakukan seorang pakar. Pada dasarnya sistem pakar bisa memecahkan masalah yang rumit, sekalipun tidak ada seorang ahli.

Perkembangan teknologi Sistem Pakar sangat kuat dipengaruhi oleh ilmu kognitif dan matematika. Cara manusia menyelesaikan masalah dan landasan formal khususnya logika dan penalaran. *Production Rules* sebagai mekanisme representasi, tipe *Rules IF ... THEN* dapat mendekati penalaran manusia dan dapat dimanipulasi oleh komputer. Kesesuaian antara potongan-potongan pengetahuan dapat digunakan untuk manusia dan komputer.

*Rules* dapat digunakan untuk formulasi suatu teori dari pemrosesan informasi manusia. Seperti kita ketahui memori manusia dapat dibagi menjadi tiga

yaitu *Sensory Memory*, *Short Term Memory (STM)/Working memory*, *Long Term Memory (LTM)*. *Sensory Memory* berfungsi sebagai buffer untuk menerima stimulasi dari panca indera. *Long Term Memory (LTM)* berfungsi untuk menyimpan pengetahuan secara permanen, *Rules* disimpan pada *long Term memory*. *Short Term Memory (STM)* adalah tempat untuk menyimpan pengetahuan sementara. Memberikan suatu input atau pemikiran membangkitkan aturan aktivasi menggunakan *Triger* dari aktivasi yang sebelumnya.

### 2.3 Komponen Utama Sistem Pakar

Komponen utama sistem pakar yaitu basis pengetahuan (*knowledge base*), dan Mesin Inferensi (*Inference Engine*).

#### 1. Basis Pengetahuan

Merupakan data yang diperlukan untuk membuat suatu keputusan yang memuat fakta-fakta dan juga teknik dalam menerangkan masalah yang disusun dalam urutan yang logis. Basis pengetahuan (*knowledge base*) memuat informasi essensial tentang domain masalah dan sering direpresentasikan sebagai fakta (*Fact*) dan aturan (*Rules*).

##### a. Fakta

Fakta adalah suatu kenyataan atau kebenaran yang diketahui. Fakta menyatakan hubungan (relasi) antara dua objek atau lebih.

##### b. Aturan

Dalam menerangkan masalah digunakan aturan (*Rule*) untuk menentukan hal apa yang harus dilakukan dalam situasi tertentu dan aturan tersebut terdiri dari dua bagian yaitu IF dan THEN, dimana IF merupakan kondisi yang mungkin

benar atau tidak benar, sedangkan THEN adalah tindakan yang dilakukan jika kondisi benar.

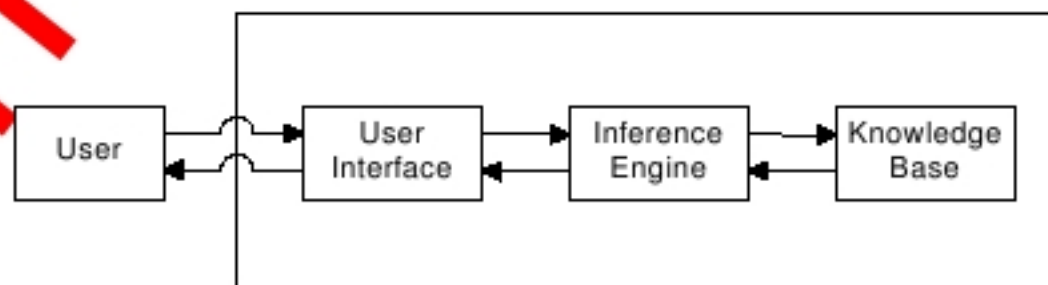
## 2. Mesin Inferensi

Mekanisme untuk menurunkan pengetahuan baru dari basis pengetahuan dan informasi yang disediakan oleh user. Selama proses penalaran, mekanisme inferensi menguji aturan-aturan dari basis pengetahuan satu per satu, dan pada saat kondisi itu benar tindakan tertentu diambil dan jika saat kondisi aturan itu salah akan diabaikan. Ada dua metode utama yang dibuat untuk *Inference Engine* yang digunakan untuk menguji aturan tersebut yaitu penalaran maju (*Forward Chaining*) dan penalaran mundur (*Backward Chaining*).

Elemen-elemen dari Sistem Pakar adalah sebagai berikut:

1. Basis Pengetahuan (*knowledge Base*)
2. Mesin Inferensi (*Inference Engine*)
3. Memori Kerja (*Working Memory*)
4. Agenda
5. Fasilitas Penjelasan (*Explanation Facility*)
6. Fasilitas akuisisi pengetahuan (*Knowledge Acquisition Facility*)
7. *User Interface*

Untuk lebih jelasnya dapat dilihat pada gambar 2.1.



Gambar 2.1. Struktur Sistem Pakar

## 2.4 Sistem Berbasis Aturan

Sistem berbasis aturan merupakan suatu sistem pakar yang menggunakan aturan-aturan untuk menyajikan pengetahuannya. Dengan kata lain bahwa sistem berbasis aturan adalah suatu perangkat lunak yang menyajikan keahlian pakar dalam bentuk aturan-aturan pada suatu domain tertentu untuk menyelesaikan suatu permasalahan.

## 2.5 Komponen Utama Sistem Berbasis Aturan

Suatu program komputer yang memproses informasi masalah khusus yang ada dalam *Working Memori* dan himpunan aturan dalam *Knowledge Base* dengan menggunakan mesin inferensi untuk memperoleh informasi baru. Untuk membangun suatu sistem berbasis aturan diperlukan beberapa komponen, secara umum dapat dilihat pada Gambar 2.2.



Gambar 2.2. Komponen Sistem Pakar Berbasis Aturan

- A. *Knowledge Base* yaitu pengetahuan yang menjadi dasar bagi pembuatan aturan-aturan dalam sistem, yang mencakup aturan-aturan itu sendiri, fakta-fakta yang terkait, serta atribut-atributnya.

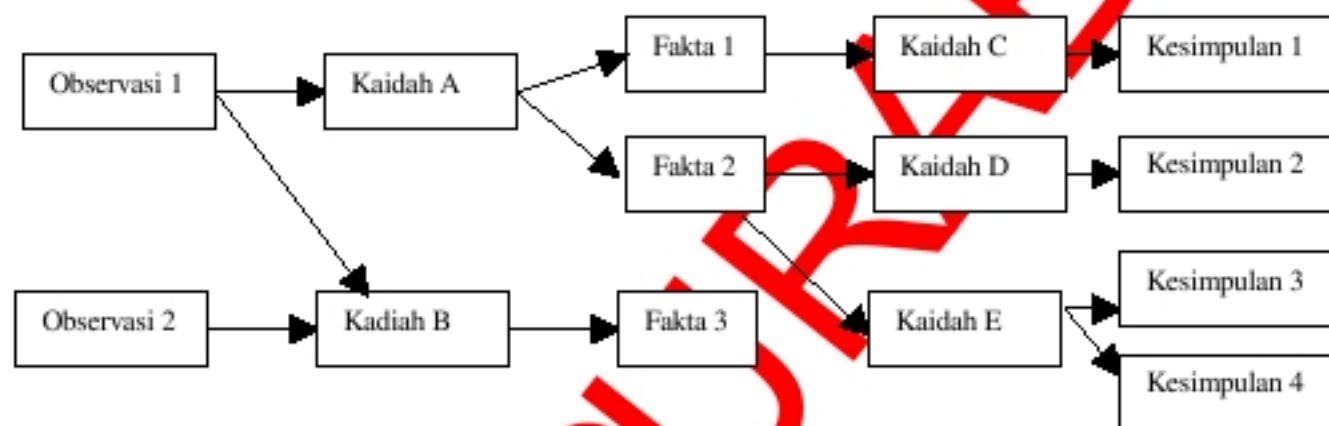
- B. Mekanisme Inferensi berfungsi untuk mensimulasikan strategi penyelesaian masalah dari seorang pakar. Sebuah konklusi akan dicapai dengan menjalankan suatu aturan tertentu pada fakta yang ada.
- C. Komponen Penjelas berfungsi menjelaskan strategi penyelesaian masalah bagi *User* yang meliputi:
1. Pertanyaan apa yang akan diajukan pada pemakai dan jika diperlukan mengapa mengajukan pertanyaan tersebut.
  2. Alasan bagaimana sistem tersebut memperoleh hasil demikian
  3. Karakteristik apa yang dimiliki tiap-tiap obyek.
- D. *User Interface* yaitu bagian program yang berhubungan langsung dengan pemakai, baik selama konsultasi maupun untuk pengembangan sistem. Oleh karena itu sistem haruslah menggunakan bahasa dan sistem pengoperasian yang mudah dimengerti.
- E. Komponen Akuisisi berfungsi untuk menyusun dan mengimplementasikan pengetahuan dalam basis pengetahuan. Komponen ini memiliki beberapa karakteristik antara lain:
1. Pengetahuan yang terdiri dari pengetahuan dan fakta harus mudah untuk dimasukkan.
  2. Metode Penyajian informasi dalam basis pengetahuan harus mudah dimengerti.
  3. Sangat baik jika memiliki sistem pengecekan atas format yang salah.

## 2.6 Forward Chaining

Menurut Andi (2003:15) *Forward Chaining* atau disebut juga penalaran maju adalah aturan-aturan diuji satu demi satu dalam urutan tertentu. *Inference*



*Engine* akan mencocokkan fakta atau statement dalam *Knowledge Base* dengan situasi yang dinyatakan dalam rule bagian IF. Jika fakta yang ada dalam *Knowledge Base* sudah sesuai dengan kaidah IF, maka *rule* itu distimulasi dan *rule* berikutnya diuji. Proses pengujian rule satu demi satu berlanjut sampai satu putaran lengkap melalui seluruh perangkat rule. Untuk lebih jelasnya dapat kita lihat alur dari metode *Forward Chaining* pada gambar 2.3.



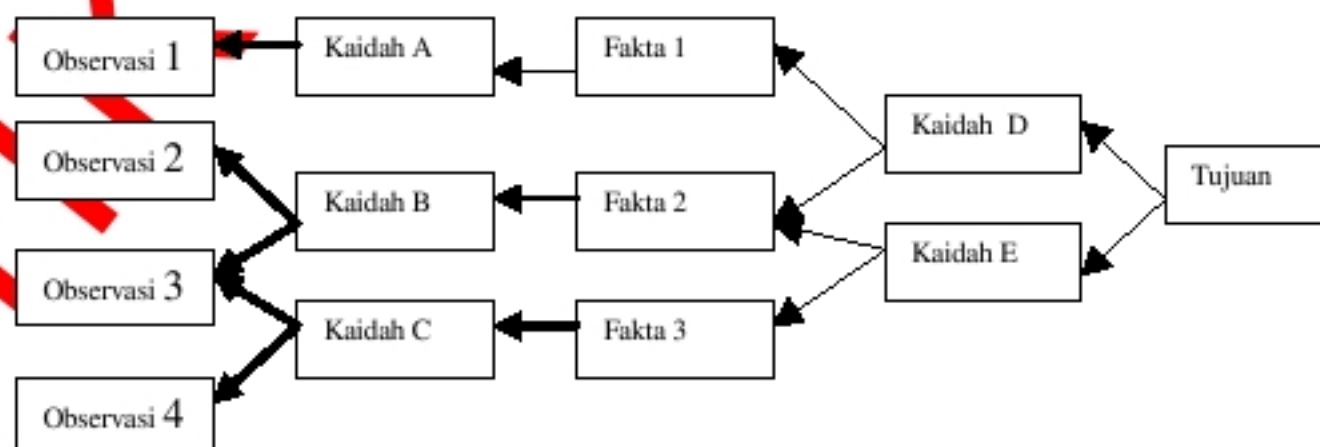
Gambar 2.3. Metode *Forward Chaining*

Pada gambar 2.3 menunjukkan pangkalan kaidah yang terdiri dari 5 buah yaitu Kaidah A, Kaidah B, Kaidah C, Kaidah D, Kaidah E. Sedangkan pangkalan data terdiri dari pengawalan fakta yang sudah diketahui, yaitu Fakta 1, Fakta 2, Fakta 3. Melalui observasi 1 mulai melacak pangkalan kaidah untuk mencari *Premis* dengan menguji semua kaidah secara berurutan. Pada observasi 1 pertama-tama melacak kaidah A dan kaidah B. Motor inferensi mulai melakukan pelacakan, mencocokkan kaidah A dalam pangkalan pengetahuan terhadap informasi yang ada di dalam pangkalan data yaitu fakta 1, fakta 2. Jika pelacakan pada kaidah A tidak ada yang cocok dengan fakta 1, maka terus bergerak menuju kaidah C yang kemudian menghasilkan kesimpulan 1, demikian seterusnya.

## 2.7 Backward Chaining

Metode *Backward Chaining* merupakan kebalikan dari metode *Forward Chaining* atau disebut penalaran mundur yaitu suatu metode yang digunakan dalam *Inference Engine* untuk melakukan pelacakan atau penalaran dari sekumpulan hipotesa menuju fakta-fakta yang mendukung kesimpulan tersebut.

Menurut Arhami (2005:19) *Backward Chaining* adalah pendekatan yang dimotori tujuan. Dalam pendekatan ini pelacakan dimulai dari tujuan, selanjutnya dicari aturan yang memiliki tujuan tersebut untuk kesimpulannya. Selanjutnya proses pelacakan menggunakan premis untuk aturan tersebut sebagai tujuan baru dan mencari aturan lain dengan tujuan baru sebagai kesimpulannya. Proses berlanjut sampai semua kemungkinan ditentukan. Jadi interpreter kaidah mulai menguji kaidah sebelah kanan yaitu *Then*. Mesin inferensi akan melacak bukti-bukti yang mendukung hipotes awal. Jika ternyata sesuai, maka basis data akan mencatat kondisi terhadap status sistem yang berlaku. Semua sisi kaidah *If* yang benar-benar sesuai digunakan untuk menghasilkan hipotesis yang baru dan keadaan tujuan, yang kemudian direkam dalam basis data. Keadaan diatas terus berlangsung sampai hipotesa terbukti kebenarannya. Untuk lebih jelasnya dapat dilihat pada gambar 4.



Gambar 2.4. Metode *Backward Chaining*

Dalam melakukan penelusuran dengan metode *Backward Chaining* berawal dari goal atau pada gambar disebut sebagai tujuan, kemudian barulah mencari informasi untuk memenuhi goal tersebut. Pertama-tama mulai dengan memberitahu sistem, bahwa kita ingin membuktikan keadaan tujuan. Motor inferensi melihat pangkalan data yaitu fakta untuk dicocokkan dengan pangkalan kaidah.

## 2.8 Verifikasi

Suatu kualitas dari basis pengetahuan dapat dilihat dari ukuran, kompleksitas dan sifat kritikal dari aplikasi-aplikasi yang ada. Semuanya itu dapat diwujudkan dari proses-proses Verifikasi. Elemen ini sangat penting bagi suatu sistem berbasis pengetahuan. Verifikasi adalah membangun sistem yang benar. Verifikasi itu sendiri terdiri dari 2 proses yaitu:

1. Memeriksa pelaksanaan suatu sistem secara spesifik.
2. Memeriksa konsistensi dan kelengkapan dari basis pengetahuan.

Verifikasi dijalankan ketika ada penambahan atau perubahan pada *Rule*, karena *Rule* tersebut sudah ada pada sistem. Sedangkan tujuan Verifikasi adalah untuk memastikan adanya kecocokan antara sistem dengan apa yang sistem kerjakan (*Rule Base*) dan juga untuk memastikan bahwa sistem itu terbebas dari error.

Berikut ini adalah yang akan di cek dalam suatu basis pengetahuan:

1. *Redundant Rules*

Dikatakan *Redundant Rule* jika dua *Rule* atau lebih mempunyai *Premise* dan *Conclusion* yang sama.

*Rule 1:* If napas sesak And batuk

Then cek saluran pernafasan

*Rule 2:* If batuk And napas sesak

Then cek saluran pernafasan

## 2. *Conflicting Rules*

*Conflicting Rules* terjadi ketika dua buah *Rule* atau lebih mempunyai *Premise* yang sama, tetapi mempunyai *Conclusion* yang berlawanan.

*Rule 1:* If napas sesak And batuk

Then cek saluran pernafasan

*Rule 2:* If batuk And napas sesak

Then cek tenggorokan

## 3. *Subsumed Rules*

Suatu *Rule* dapat dikatakan *subsumed* jika *Rule* tersebut mempunyai *constraints* yang lebih atau kurang tetapi mempunyai *Conclusion* yang sama.

*Rule 1:* If napas sesak And batuk And dahak

Then cek saluran pernafasan

*Rule 2:* If sesak And batuk

Then cek saluran pernafasan

## 4. *Circular Rules*

*Circular Rules* ialah suatu keadaan dimana terjadinya proses perulangan dari suatu *Rule*. Ini dikarenakan suatu *Premise* dari salah satu *Rule* merupakan *Conclusion* dari *Rule* yang lain, atau sebaliknya.

*Rule 1:* If cek suhu badan > 38

Then cek kontak unggas

*Rule 2:* If cek kontak unggas

Then cek suhu badan > 38

#### 5. *Unnecessary IF conditions*

*Unnecessary IF* terjadi ketika dua *Rule* atau lebih mempunyai *Conclusion* yang sama, tetapi salah satu dari *Rule* tersebut mempunyai *Premise* yang tidak perlu dikondisikan dalam *Rule* karena tidak mempunyai pengaruh apapun.

*Rule 1:* If tenggorokan nyeri and mata merah

Then cek saluran pernafasan

*Rule 2:* If tenggorokan nyeri and mata tidak berair

Then cek saluran pernafasan

#### 6. *Dead-end Rules*

*Dead-end Rules* adalah suatu *Rule* yang konklusinya tidak diperlukan oleh *Rule* lainnya

### 2.9 Internet

Internet adalah jaringan komputer dunia yang disusun oleh dua entity:

1. *Client Web* berupa berupa *Browser Web* seperti *Internet Explorer* dan *Netscape Navigator* yang akan meminta menampilkan suatu halaman *Web*, *File* atau data dari komputer lain yaitu *Server*.
2. *Server Web* suatu aplikasi yang ada di *Server* dan menangani permintaan *Client*. Secara sederhana *Server* adalah komputer yang “melayani” sesuatu. *Server*

umumnya dapat menangani permintaan *Client* dalam jumlah yang banyak secara simultan. *Server Web* adalah *Server* yang mampu menangani *Web* atau permintaan HTTP. Pada Microsoft, *Server Web* merupakan bagian dari *Internet Information Services (IIS)*.

Komunikasi antara *Browser Web* dan *Server* dilakukan menggunakan protokol yang disebut *Transmission Control Protocol/Internet Protocol (TCP/IP)*. Protokol adalah serangkaian aturan dan prosedur yang menentukan bagaimana dua entiti saling berkomunikasi. Sebenarnya terdiri dari dua bagian yaitu TCP dan IP. TCP sering disingkat *Transport Protocol*, akan membagi data yang besar menjadi *Packet-Packet* dan memastikan data diterima dalam keadaan yang sama ketika dikirim. IP, suatu *Network Protokol*, bertanggung jawab untuk mengirimkan *Packet* melalui suatu *Network* seperti *Internet*.

## 2.10 World Wide Web

*World Wide Web (WWW)* bukanlah *Internet*, demikian pula sebaliknya. Namun *WWW* dan *Internet* sangat berkaitan satu dengan lainnya. *Internet* adalah suatu jaringan *Internet* global, sedangkan *WWW* bukan sekedar jaringan tetapi didalamnya terdapat suatu set aplikasi komunikasi dan sistem perangkat lunak.

## 2.11 HyperText Markup Language

*HyperText Markup Language (HTML)* digunakan untuk membangun suatu halaman web. HTML adalah bahasa *Markup*, yaitu bahasa yang digunakan untuk *Markup* terhadap sebuah dokumen teks. HTML dibuat menggunakan teks yang diformat dan disebut *Tag*. *Tag* penulisannya diapit oleh lambang  $\diamond$ . *Tag*

biasanya ditampilkan dalam pasangan berupa *Tag* pembuka untuk menjalankan fitur pemformatan dan *Tag* penutup untuk menutup pemformatan. *Tag* penutup namanya sama dengan *Tag* pembuka tetapi penulisannya didahului dengan karakter *"/*".

Untuk membuat dokumen HTML, anda harus mulai dengan *Tag* `<HTML>` dan diakhiri dengan *Tag* `</HTML>`. Didalam *Tag* tersebut terdapat bagian HEAD dan BODY. Kepala dokumen yang ditandai dengan *Tag* `<HEAD>` dan `</HEAD>` dapat diisi dengan Judul dokumen, informasi yang berkaitan dengan dokumen seperti *Tag* META dan BASE serta *Script*. Bagian berikutnya adalah badan halaman yang ditandai dengan tag `<BODY>` dan `</BODY>`. Pada bagian inilah terdapat isi dari halaman *Web* yang hendak ditampilkan.

Struktur dari dokumen HTML sebagai berikut:

```
<HTML>
  <HEAD>
    <TITLE> Judul halaman web</TITLE>
  </HEAD>
  <BODY>
    Isi dari halaman Web
  </BODY>
</HTML>
```

## 2.12 Active Server Page

Untuk keperluan pemrograman, penulis menggunakan *Active Server Page* (ASP). ASP merupakan suatu *Script* bersifat *Server-Side* yang ditambahkan pada HTML untuk membuat sebuah *Web* menjadi lebih menarik, dinamis dan interaktif. ASP dapat mengolah data yang diambil dengan sebuah form, membuat aplikasi-aplikasi tertentu dalam sebuah *Web*, ataupun membuat *Database* dalam sebuah *Web*.

ASP bersifat *Server-Side*, berarti proses pengerjaan *Script* terjadi di server, bukan di *Browser/Client*. Untuk memanggil file ASP melalui *Browser*, maka *Browser* tersebut mengirimkan permintaan ke *Web Server*, kemudian *Server* mengeksekusi *Script* yang ada dan hasilnya dikirim kembali ke *Browser*. Pada sebuah PC biasa yang berbasis *Windows*, ASP bisa digunakan apabila PC tersebut disimulasikan menjadi sebuah *Web Server* dengan menginstal *Microsoft Personal Web Server (PWS)* atau *Microsoft Internet Information Service (IIS)*.

STIKOMMP SURABAYA