

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Sistem Pakar**

Menurut Ahmad Syatibi (2012), Sistem pakar merupakan cabang dari AI (*Artificial Intelligent*) yang membuat ekstensi khusus untuk spesialisasi pengetahuan guna memecahkan suatu permasalahan pada *Human Expert*. *Human Expert* merupakan seseorang yang ahli dalam suatu bidang ilmu pengetahuan tertentu, ini berarti bahwa *expert* memiliki suatu pengetahuan atau *skill* khusus yang dimiliki oleh orang lain. *Expert* dapat memecahkan suatu permasalahan yang tidak dapat dipecahkan oleh orang lain dengan cara efisien.

##### **2.1.1 Kelebihan Sistem Pakar**

Sistem pakar memiliki beberapa fitur menarik yang merupakan kelebihannya, seperti:

1. Meningkatkan ketersediaan.
2. Mengurangi biaya (*reduced cost*), biaya yang diperlukan untuk menyediakan keahlian satu orang *user* menjadi berkurang.
3. Mengurangi bahaya (*reduced danger*), sistem pakar dapat digunakan di lingkungan yang mungkin berbahaya bagi manusia.
4. Permanen (*permanence*), sistem pakar dan pengetahuan yang terdapat didalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.

5. Keahlian *multiple (multiple expertise)*, pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinu menyelesaikan suatu masalah setiap saat. Tingkat keahlian/pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.
6. Meningkatkan kehandalan (*increased reliability*), sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternative pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku, jika sistem dibuat oleh salah seorang pakar sehingga akan selalu sama dengan pendapat pakar tersebut kecuali jika sang pakar melakukan yang mungkin terjadi pada saat tertekan atau stres.
7. Penjelasan (*explanation*), sistem pakar dapat menjelaskan detail proses penalaran (*reasoning*) yang dilakukan hingga mencapai suatu kesimpulan. Seorang pakar mungkin saja terlalu lelah atau tidak mampu melakukannya setiap waktu. Dan hal ini akan meningkatkan tingkat kepercayaan bahwa kesimpulan yang dihasilkan adalah benar.
8. Respon yang cepat (*fast response*), respon yang cepat/*real-time* diperlukan pada beberapa aplikasi. Meskipun bergantung pada *hardware* dan *software* yang digunakan, namun sistem pakar relatif memberikan respon yang lebih cepat dibandingkan seorang pakar.
9. Basis data cerdas (*intelligent database*), sistem pakar dapat digunakan untuk mengakses basis data secara cerdas.

### 2.1.2 Kekurangan Sistem Pakar

Disamping memiliki beberapa kelebihan dan manfaat, sistem pakar juga memiliki beberapa kelemahan antara lain:

1. Biaya yang diperlukan untuk membuat dan memeliharanya mahal.
2. Sulit dikembangkan. Hal ini tentu saja erat kaitannya dengan ketersediaan pakar di bidangnya.
3. Sistem pakar tidak 100% bernilai benar.

### 2.2 Penelitian Terkait

Penelitian yang dilakukan oleh Andi Bekti Widiyanto pada tahun 2015 berjudul Sistem Pakar Untuk Mengidentifikasi Jenis Kayu dengan Metode *Certainty Factor* dan *Forward Chaining*. Penerapan *Certainty Factor* sebagai metode untuk pengambilan kesimpulan akhir sudah sesuai dengan hasil perhitungan manual dan hasil yang diberikan oleh sistem. Dari jumlah hasil identifikasi sistem pakar yang benar dan jumlah kayu yang diuji, maka penelitian tersebut didapatkan Akurasi =  $(36 \div 40) \times 100\% = 90\%$ . Hasil perhitungan menyatakan bahwa sistem pakar mampu menghasilkan akurasi sebesar 90%. Penulis mengacu pada jurnal ini untuk perhitungan hasil pengujian.

### 2.3 Metode Faktor Kepastian (*Certainty Factor*)

Menurut Denok Pustpitasari (2011), *Certainty Factor* merupakan suatu metode yang digunakan untuk menyatakan kepercayaan dalam sebuah kejadian (fakta atau hipotesis) berdasarkan bukti atau penilaian pakar. Secara konsep, *Certainty Factor* (CF) merupakan salah satu teknik yang digunakan untuk mengatasi ketidakpastian dalam pengambilan keputusan. *Certainty Factor* (CF)

dapat terjadi dengan berbagai kondisi. Diantara kondisi yang terjadi adalah terdapat beberapa antensenden (dalam *rule* yang berbeda) dengan satu konsekuen yang sama. Dalam kasus ini, kita harus mengagregasikan nilai CF keseluruhan dari setiap kondisi yang ada. Pada konsep *Certainty Factor* ini juga sering dikenal dengan adanya believe dan disbelieve. *Believe* merupakan keyakinan, sedangkan *Disbelieve* merupakan ketidakyakinan.

Adapun notasi atau rumusan dasar dari *Certainty Factor*, sebagai berikut.

$$\text{CF}[h,e] = \text{MB}[h,e] - \text{MD}[h,e]$$

$$\text{MB}[h, e_1 \Delta e_2] = \begin{cases} 0 & \text{MD}[h, e_1 \Delta e_2] = 1 \\ \text{MB}[h, e_1] + \text{MB}[h, e_2] \cdot (1 - \text{MB}[h, e_1]) & \text{lainnya} \end{cases}$$

$$\text{MD}[h, e_1 \Delta e_2] = \begin{cases} 0 & \text{MB}[h, e_1 \Delta e_2] = 1 \\ \text{MD}[h, e_1] + \text{MD}[h, e_2] \cdot (1 - \text{MD}[h, e_1]) & \text{lainnya} \end{cases}$$

Keterangan :

$\text{CF}[h,e]$  = *Certainty Factor* dalam hipotesis h yang dipengaruhi oleh fakta e.

$\text{MB}[h,e]$  = *Meansure of Believe*, merupakan nilai kenaikan dari kepercayaan hipotesis h dipengaruhi oleh fakta e.

$\text{MD}[h,e]$  = *Meansure of Disbelieve*, merupakan nilai kenaikan dari ketidakpercayaan hipotesis h dipengaruhi oleh fakta e.

H = Hipotesis.

e = Evidence.

Sebagai contoh, berikut ini adalah sebuah aturan dengan CF yang diberikan oleh seorang pakar:

IF Nafas sesak

AND Nafas ngorok

AND Bersin-bersin

AND Batuk

AND Mata berair

THEN Batuk berdarah

### **2.3.1 Kelebihan *Certainty Factor***

- a. Metode ini cocok dipakai dalam sistem pakar untuk mengukur sesuatu apakah pasti atau tidak pasti dalam mendiagnosa penyakit sebagai salah satu contohnya.
- b. Perhitungan dengan menggunakan metode ini dalam sekali hitung hanya dapat mengelola dua data saja sehingga keakuratan data dapat terjaga.

### **2.3.2 Kekurangan *Certainty Factor***

- a. Ide umum dari pemodelan ketidakpastian manusia dengan menggunakan numerik metode *certainty factor* biasanya diperdebatkan. Sebagian orang akan membantah pendapat bahwa formula untuk metode *certainty factor* diatas memiliki sedikit kebenaran.
- b. Metode ini hanya dapat mengolah ketidakpastian/kepastian hanya dua data saja. Perlu dilakukan beberapa kali pengolahan data untuk data yang lebih dari dua buah.
- c. Nilai CF yang diberikan bersifat subyektif karena penilaian setiap pakar bisa saja berbeda-beda tergantung pengetahuan dan pengalaman pakar.

## 2.4 Definisi Hepatitis

Istilah Hepatitis dipakai untuk semua jenis peradangan pada sel-sel hati, yang bisa disebabkan oleh infeksi (virus, bakteri, parasite), obat-obatan (termasuk obat tradisional), konsumsi alkohol, lemak yang berlebih dan penyakit autoimmune. Ada 5 jenis Hepatitis Virus yaitu Hepatitis A, B, C, D, dan E. Antara Hepatitis yang satu dengan yang lain tidak saling berhubungan. Pada penelitian ini hanya dibahas mengenai penyakit Hepatitis A, B, dan C yang sering ditemui dikarenakan Virus Hepatitis D dan E masih jarang ditemui.

Tabel 2.1 Prevalensi Hepatitis Menurut Karakteristik di Indonesia Th 2013

Karakteristik	Prevalensi Hepatitis (%)
<b>Kelompok umur (tahun)</b>	
<1	0,5
1 - 4	0,8
5 - 14	1,0
15 - 24	1,1
25 - 34	1,3
35 - 44	1,3
45 - 54	1,4
55 - 64	1,3
65 - 74	1,4
>75	1,3
<b>Jenis Kelamin</b>	
Laki-laki	1,3
Perempuan	1,1
<b>Pekerjaan</b>	
Tidak Bekerja	1,1
Pegawai	1,0
Wiraswasta	1,2
Petani/Nelayan/Buruh	1,6
	1,4

Sumber: Riskesdas Tahun 2013, Balitbangkes, Kemenkes

Dari table di atas terlihat karakteristik prevalensi Hepatitis tertinggi pada kelompok umur 45-54 dan 65-74 (1,4%). Penderita Hepatitis baik pada laki-laki maupun perempuan, proporsinya tidak berbeda secara bermakna. Jenis pekerjaan

juga mempengaruhi prevalensi Hepatitis, penderita Hepatitis banyak ditemukan pada petani atau nelayan serta buruh dibandingkan jenis pekerjaan yang lain.

#### **2.4.1 Hepatitis A**

WHO memperkirakan di dunia setiap tahunnya ada sekitar 1,4 juta penderita Hepatitis A. Di Amerika insidens Hepatitis A adalah 1 per 100.000 penduduk, dengan estimasi 21.000 orang (Tahun 2009). Di Eropa insidens Hepatitis A adalah 3,9 per 100.000 penduduk (Publikasi tahun 2008). Di Indonesia, Hepatitis A sering muncul dalam Kejadian Luar Indonesia (KLB). Tahun 2010 tercatat 6 KLB dengan jumlah penderita 279, jumlah kematian 0, CFR 0 sedangkan tahun 2011 tercatat 9 KLB, jumlah penderita 550, jumlah kematian 0, CFR 0. Tahun 2012 sampai bulan Juni, telah terjadi 4 KLB dengan jumlah penderita 204, jumlah kematian 0, CFR 0.

Menurut hasil Riset Kesehatan Dasar (2007), data lain menunjukkan pada tahun 1998, di Kabupaten Bogor, Jawa Barat telah terjadi KLB Hepatitis A dengan jumlah kasus 74 orang (AR = 1,4%) dan golongan umur terbanyak 19-25 tahun (AR = 3,4%), di Provinsi Jawa Timur yaitu di Kabupaten Bondowoso (Kecamatan Sukosari) dan Kabupaten Malang (Kecamatan Wonosari) di 7 desa dengan jumlah kasus 998, tahun 2004 di Kecamatan Tegal Ampel, Kabupaten Bondowoso, Jawa Timur 47 kasus. Tahun 2006 di Kecamatan Pakem, Kabupaten Bondowoso, Jawa Timur 65 kasus. (Surveilans Prop Jawa Timur). Tahun 2008 di Provinsi DIY tercatat 1.160 kasus dengan hasil pemeriksaan anti-HAV positif yaitu di Kodya Yogyakarta 287 kasus, Kabupaten Bantul 48 kasus, Kulon Progo 6

kasus, Gunung Kidul 11 kasus dan Sleman 808 kasus serta KLB di Pulau Panggang dengan 57 kasus. Tahun 2009 di Kabupaten Ngawi dengan 146 kasus.

#### **2.4.1.1 Etiologi**

Penyebab penyakit adalah virus Hepatitis A (VHA), termasuk *family picornaviridae* berukuran 27 nanometer, *genus hepatovirus* yang dikenal sebagai *enterovirus 72*, mempunyai 1 *serotype* dan 4 *genotype*, merupakan RNA virus. Virus Hepatitis A bersifat termostabil, tahan asam dan tahan terhadap empedu. Virus ini diketahui dapat bertahan hidup dalam suhu ruangan selama lebih dari 1 bulan. Pejamu infeksi VHA hanya terbatas pada manusia dan beberapa binatang primata. Virus dapat diperbanyak secara *in vitro* dalam kultur sel primer monyet kecil atau secara *invivo* pada simpanse.

#### **2.4.1.2 Cara Penularan**

Menurut Direktorat Jenderal Kementerian Kesehatan RI (2012), virus Hepatitis A ditularkan secara *fecal-oral*. Virus ini masuk ke dalam saluran pencernaan melalui makanan dan minuman yang tercemar tinja penderita VHA. Virus kemudian masuk ke hati melalui peredaran darah untuk selanjutnya menginvasi sel-sel hati (*hepatosit*), dan melakukan replikasi di *hepatosit*. Jumlah virus yang tinggi dapat ditemukan dalam tinja penderita sejak 3 hari sebelum muncul gejala hingga 1-2 minggu setelah munculnya gejala kuning pada penderita. Ekskresi virus melalui tinja pernah dilaporkan mencapai 6 bulan pada bayi dan anak. Sebagian besar kasus kemungkinan tidak menular lagi pada minggu pertama setelah ikterus. Ekskresi kronis pada VHA tidak pernah dilaporkan.



Infeksi Hepatitis A sering terjadi dalam bentuk Kejadian Luar biasa (KLB) dengan pola *common source*, umumnya sumber penularan berasal dari air minum yang tercemar, makanan yang tidak dimasak, makanan yang tercemar, dan sanitasi yang buruk. Selain itu, walaupun bukan merupakan cara penularan yang utama, penularan melalui transfusi atau penggunaan jarum suntik bekas penderita dalam masa inkubasi juga pernah dilaporkan.

#### **2.4.2 Hepatitis B**

Menurut hasil Riset Kesehatan Dasar (2007), menunjukkan bahwa Hepatitis klinis terdeteksi di seluruh propinsi di Indonesia dengan prevalensi sebesar 0,6% (rentang: 0,2%-1,9%). Hasil Riskesdas Biomedis tahun 2007 dengan jumlah sampel 10.391 menunjukkan bahwa persentase HBsAg positif 9,4%. Persentase Hepatitis B tertinggi pada kelompok umur 45-49 tahun (11,92%), umur >60 tahun (10,57%) dan umur 10-14 tahun (10,02%). HBsAg positif pada kelompok laki-laki dan perempuan hampir sama (9,7% dan 9,3%). Hal ini menunjukkan bahwa 1 dari 10 penduduk Indonesia telah terinfeksi virus Hepatitis B.

Dari data yang telah terkumpul, angka prevalensi HBsAg pada donor darah di Indonesia tahun 1981 dengan metode pemeriksaan RPHA (Reverse Passive Haemagglutination) menunjukkan rata-rata 5,2% (rentangan 2,4-9,1%), dan tahun 1993 dengan metode pemeriksaan ELISA rata-rata 9,4%, rentangan 2,5 -36,1% (Sulaiman et al., 1998).

Angka penularan secara vertikal dari ibu pengidap Hepatitis B kepada bayinya cukup tinggi. Berdasarkan penelitian beberapa rumah sakit di Indonesia, prevalensi HBsAg pada ibu hamil berkisar 2,1—5,2% (Soewignyo, 1992).

Data di RSUP Sanglah, Denpasar menunjukkan bahwa dari hasil uji survei 3.943 ibu hamil didapatkan hasil 80 ibu hamil dengan HBsAg positif, prevalensi HBsAg 2,03% dan HBeAg positif 50 %. Hasil pemeriksaan HBsAg tali pusat positif 12 % dari ibu hamil pengidap Hepatitis B (Surya, 1995). Peneliti lain melaporkan bahwa hasil uji saring pada 1.800 wanita hamil di Indonesia ditemukan 61 ibu hamil (3,4%) dengan HBsAg positif (Suparyatmo).

#### **2.4.2.1 Etiologi**

Penyebab penyakit adalah virus Hepatitis B (VHB) yang termasuk *family Hepadnavirus* dan berukuran sangat kecil (42 nm). Virus Hepatitis B merupakan virus DNA dan sampai saat ini terdapat 8 genotip VHB yang telah teridentifikasi, yaitu genotip A–H. VHB memiliki 3 jenis morfologi dan mampu mengkode 4 jenis antigen, yaitu HBsAg, HBeAg, HBcAg, dan HBxAg. Virus Hepatitis B yang menginfeksi manusia bisa juga menginfeksi simpanse. Virus dari *Hepadnavirus* bisa juga ditemukan pada bebek, marmut dan tupai tanah, namun virus tersebut tidak bisa menginfeksi manusia.

#### **2.4.2.2 Cara Penularan**

Menurut Direktorat Jenderal Kementrian Kesehatan RI (2012), virus Hepatitis B dapat ditemukan pada cairan tubuh penderita seperti darah dan produk darah, air liur, cairan *serebrospinalis*, *peritonea*, *pleural*, cairan *amniotik*, semen, cairan vagina dan cairan tubuh lainnya. Namun tidak semuanya memiliki kadar virus yang infeksius. Secara umum, penularan bisa terjadi secara vertikal maupun horizontal. Untuk saat ini, penularan VHB yang utama diduga berasal dari hubungan intim dan transmisi perinatal. Transmisi horizontal adalah penularan

dari satu individu ke individu lainnya. Selain lewat hubungan seksual tidak aman, transmisi horizontal Hepatitis B juga bisa terjadi lewat penggunaan jarum suntik bekas penderita Hepatitis B, transfusi darah yang terkontaminasi virus Hepatitis B, pembuatan tato, penggunaan pisau cukur, sikat gigi, dan gunting kuku bekas penderita Hepatitis B. Sementara itu, berpelukan, berjabat tangan, atau berciuman dengan penderita Hepatitis B belum terbukti mampu menularkan virus ini.

Penularan secara vertikal adalah penularan yang terjadi pada masa *perinatal* yaitu penularan dari ibu kepada anaknya yang baru lahir, jika seorang ibu hamil karier Hepatitis B dan HBeAg positif maka bayi yang di lahirkan 90% kemungkinan akan terinfeksi dan menjadi karier juga. Kemungkinan 25% dari jumlah tersebut akan meninggal karena Hepatitis kronik atau kanker hati. Transmisi *perinatal* ini terutama banyak terjadi di negara-negara Timur dan negara berkembang. Infeksi perinatal paling tinggi terjadi selama proses persalinan dan diduga tidak berhubungan dengan proses menyusui

### **2.4.3 Hepatitis C**

Berdasarkan hasil Surveilans Hepatitis C oleh Direktorat Jenderal Pengendalian Penyakit dan Penyehatan Lingkungan pada tahun 2010-2011 yang dilaksanakan di 21 propinsi, 53 rumah sakit, 49 laboratorium dan 26 Unit Transfusi Darah (UTD) PMI, dengan jumlah 1.825.823 sampel, kasus positif 29.480 orang, jumlah kasus terbanyak didapatkan pada golongan umur 20-40 tahun sebanyak 58,5% sedangkan proporsi menurut jenis kelamin menunjukkan bahwa pada kelompok laki-laki 83% dan 17% pada perempuan.

Prevalensi anti-VHC pada beberapa donor di Indonesia berkisar 0,5-3,4%. Prevalensi Anti-VHC pada virus Hepatitis Akut 9,5-20%, prevalensi Anti-VHC pada sirosis hati berkisar 30,8-89,2 persen.

Data ko-infeksi diperoleh dari beberapa penelitian, Rino S Gani (FK-UI, RSCM) penderita dengan HIV (IVDU), ko infeksi 80%, penderita ko infeksi dengan Hepatitis B berkisar 10-19%, Suryanto Sidik (RS Mintoharjo) pada penderita dengan HIV, 31,6% ko-infeksi dengan VHC.

#### **2.4.3.1 Etiologi**

Penyebab penyakit Hepatitis C adalah virus Hepatitis C (VHC) yang termasuk *family Flaviviridea genus Hepacivirus* dan merupakan virus RNA. Setidaknya 6 genotip dan lebih dari 50 *subtipe* VHC yang berbeda telah ditemukan.

#### **2.4.3.2 Cara Penularan**

Menurut Direktorat Jenderal Kementerian Kesehatan RI (2012), cara penularan VHC yang paling umum adalah secara *parenteral*, yaitu berkaitan dengan penggunaan bersama jarum suntik yang tidak steril terutama pada pengguna obat-obatan terlarang, tato, tindik, penggunaan alat pribadi seperti pisau cukur, sikat gigi bersama penderita, transfusi darah, operasi, transplantasi organ, dan melalui hubungan seksual. VHC adalah penyebab utama dari Hepatitis yang diderita setelah transfusi darah. Walaupun begitu, peraturan yang memperketat pemeriksaan darah bagi donor darah telah menurunkan risiko infeksi secara drastis. Penularan dapat terjadi dalam waktu 1 minggu atau lebih setelah timbulnya gejala klinis yang pertama pada penderita.

Penularan vertikal dari ibu ke bayi selama proses kelahiran sangat jarang (sekitar 5-6%) dan menyusui tidak meningkatkan resiko penularan VHC dari seorang ibu yang terinfeksi ke bayinya. Hepatitis C tidak dapat menular melalui jabat tangan, ciuman, dan pelukan.

## 2.5 MySQL

Menurut Yenie Kustiyahningsih (2010, p.145) Basis data adalah sekumpulan informasi yang diatur agar mudah dicari. Dalam arti umum basis data adalah sekumpulan data yang diproses dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan tepat, yang dapat digambarkan sebagai aktivitas dari satu atau lebih organisasi yang berelasi.

MySQL merupakan suatu database. MySQL dapat juga dikatakan sebagai database yang sangat cocok bila dipadukan dengan PHP. Secara umum, database berfungsi sebagai tempat atau wadah untuk menyimpan, mengklasifikasikan data secara profesional. MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Itu dapat diartikan bahwa MySQL merupakan standar penggunaan database di dunia untuk pengolahan data.

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Sedangkan RDBMS sendiri akan lebih banyak mengenal istilah seperti tabel, baris, dan kolom digunakan dalam perintah-perintah di MySQL. MySQL merupakan sebuah basis data yang mengandung satu atau sejumlah tabel. Tabel terdiri atas sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Di dalam PHP telah menyediakan fungsi untuk koneksi ke basis data dengan sejumlah fungsi untuk pengaturan baik menghubungkan maupun memutuskan

koneksi dengan server database MySQL sebagai sarana untuk mengumpulkan informasi.

Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *select* (mengambil), *insert* (menambah), *update* (mengubah), dan *delete* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat database, *field*, ataupun index guna menambah atau menghapus data.

## 2.6 PHP

Menurut Anhar (2010), PHP atau yang memiliki kepanjangan (*Hypertext Preprocessor*) merupakan suatu bahasa pemrograman yang difungsikan untuk membangun suatu website dinamis. PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya sehingga dengan adanya PHP tersebut, web akan sangat mudah di-*maintenance*.

PHP berjalan pada sisi server sehingga PHP disebut juga sebagai bahasa *Server Side Scripting*. Artinya bahwa dalam setiap/untuk menjalankan PHP, wajib adanya *web server*.

PHP ini bersifat *open source* sehingga dapat dipakai secara cuma-cuma dan mampu lintas platform, yaitu dapat berjalan pada sistem operasi Windows maupun Linux. PHP juga dibangun sebagai modul pada *web server apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

Keunggulan PHP dari sifatnya yang *server-side* tersebut antara lain :

- a. PHP merupakan program *open source* (tidak memerlukan biaya lisensi).
- b. *Multi-platform* (dapat dijalankan pada sistem operasi yang berbeda-beda).

- c. Adanya penggunaan *session* / sesi.
- d. Tidak diperlukan kompatibilitas *browser* atau harus menggunakan *browser* tertentu, karena *server*-lah yang akan mengerjakan skrip PHP. Hasil yang dikirimkan kembali ke *browser* umumnya bersifat teks atau gambar saja sehingga pasti dikenal oleh *browser* apapun.
- e. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh *server*, misalnya koneksi ke *database*.

## 2.7 HTML

Menurut A. Saputra (2011), HTML kependekan dari *HyperText Markup Language*. Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai *web page*. Dokumen HTML merupakan dokumen yang disajikan dalam *browser web surfer*. Dokumen ini umumnya berisi informasi atau interface aplikasi di dalam Internet. Ada dua cara untuk membuat sebuah *web page*: dengan *HTML editor* atau dengan *editor* teks biasa (misalnya notepad).

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). Versi terakhir dari HTML adalah HTML 5.01, meskipun saat ini telah berkembang XHTML yang merupakan pengembangan dari HTML.

HTML berupa kode-kode tag yang menginstruksikan browser untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah *file* yang merupakan *file* HTML dapat dibuka dengan menggunakan *web browser* seperti *Mozilla Firefox* atau *Microsoft Internet Explorer*. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan *browser*.

## 2.8 CSS

Menurut Jayan (2010), CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan *file* gambar. CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan web. Perlu diingat, CSS hanyalah berupa kumpulan *script* yang tujuannya bukan untuk menggantikan HTML, melainkan pelengkap agar dokumen HTML bisa tampil lebih cantik dan dinamis.

Sejak ditemukannya CSS pada awal dekade 90an, CSS terus dikembangkan dan diserap oleh *web developer*. Hingga sekarang telah mencapai versi ke-3. Kode CSS bersifat lintas *platform*, yang berarti *script* ini dapat dibaca oleh berbagai macam sistem operasi dan *browser*. Hanya saja *browser* seperti *Internet Explorer*, seringkali salah mengartikan *script* CSS yang menyebabkan ketidaksempurnaannya tampilan dokumen HTML. *Script* CSS perlu dioptimalkan agar tampil maksimal pada *browser internet explorer*.



## 2.9 Rekayasa Perangkat Lunak

Menurut Shalahuddin (2011), perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumen kebutuhan, model desain, dan cara pengguna (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak.

Perangkat lunak adalah suatu instruksi (program komputer) yang dibuat oleh sebuah perusahaan komputer untuk mengerjakan tugas tertentu. Memberikan fungsi dan unjuk kerja seperti yang diinginkan atau mengatur struktur data memungkinkan program memanipulasi informasi secara proporsional, dan mengatur dokumen yang menggambarkan operasi dan kegunaan program.

## 2.10 UML

Menurut Timotius Witono Merliana (2010), UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

Tujuan Penggunaan UML, menurut Suhendar (2011):

1. Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
2. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
3. Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
4. UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang *coding* program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reverse engineering*).

#### **2.10.1. Static View**

*Static View* termasuk diagram yang memberikan gambaran dari unsur-unsur dari sistem tetapi tidak memberitahu bagaimana elemen akan berperilaku. Hal ini sangat mirip *Blueprint*. *Blueprint* itu komprehensif, tetapi mereka hanya menunjukkan apa yang tetap diam, maka disebut *Static View*. *Static View* dibentuk oleh dua diagram, yaitu *Class Diagram* dan *Object Diagram*.

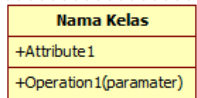



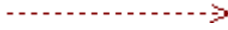

#### **2.10.3.2 Class Diagram**

*Class diagram* membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain (dalam *logical view*) dari suatu sistem. Selama proses analisis, class diagram memperlihatkan aturan-aturan dan tanggung jawab entitas

yang menentukan perilaku sistem, selama tahap desain *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

*Class diagram* juga merupakan fondasi untuk *component diagram* dan *deployment diagram*, dalam sebuah model mungkin terdapat beberapa diagram kelas dengan spesifikasi tersendiri. Simbol-simbol yang digunakan dalam *class diagram*:

Tabel 2.2 Simbol *Class Diagram*

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem
Paket/package 	Paket/package merupakan sebuah bungkus dari satu atau lebih kelas (kumpulan kelas)
Asosiasi 	<i>Asosiasi</i> merupakan hubungan antar kelas dengan makna umum, asosiasi biasanya jua disertai dengan <i>multiplicity</i>
Generalisasi 	<i>Generalisasi</i> merupakan hubungan generalisasi dan spesialisasi (umum khusus) antara dua kelas dimana fungsi yang satu adalah fungsi yang lebih umum dari fungsi yang lainnya
Dependency 	<i>Dependency</i> merupakan hubungan antarkelas yang saling bergantung, membutuhkan satu sama lain.
Agregasi 	<i>Agregasi</i> merupakan hubungan antar kelas dimana satu kelas merupakan semua bagian dari kelas-kelas yang lain.


### 2.10.2. *Dynamic View*



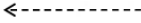


Pada *Dynamic View* meliputi diagram yang mengungkapkan bagaimana benda berinteraksi dengan satu sama lain dalam respon terhadap lingkungan. Ini termasuk *Sequence Diagram* dan *Collaboration Diagram*, yang kolektif disebut sebagai diagram interaksi. Mereka secara khusus dirancang untuk menjelaskan bagaimana benda berbicara satu sama lain. Ini juga mencakup *Statechart Diagram*, yang menunjukkan bagaimana dan mengapa perubahan objek dari waktu ke waktu dalam menanggapi lingkungan.

#### 2.10.3.2 *Sequence Diagram*

Semua *Sequence diagram* lebih dimodelkan pada tingkat objek daripada tingkat kelas untuk memungkinkan skenario yang menggunakan lebih dari satu instance dari kelas yang sama dan bekerja pada tingkat fakta, data uji, dan contoh. *Sequence Diagram* menggunakan tiga elemen notasi mendasar: *object*, *message/stimuli*, dan *object lifeline*. Simbol-simbol yang digunakan dalam *sequence diagram* :

Tabel 2.3 Simbol *Sequence Diagram*

Simbol	Keterangan
Objek dan kelas 	Objects; mewakili peserta
<i>Lifeline</i>	<i>Lifeline</i> merupakan siklus hidup sebuah pesan/message berdasarkan waktu

	
<i>Pesan/message</i> 	<i>Pesan/message</i> merupakan komunikasi antara objek yang satu dengan objek yang lainnya dimana objek yang satu meminta objek yang lain untuk melakukan sesuatu. Pesan ini diimplementasikan menjadi metode
<i>Return message</i> 	<i>Return message</i> merupakan balasan/hasil yang berisi nilai dari sebuah objek yang meminta (mengirim pesan)
<i>Self message</i> 	<i>Self message</i> merupakan pesan dari sebuah objek kepada objek itu sendiri untuk melakukan suatu aksi
<i>Return self message</i> 	<i>Return self message</i> sebagaimana <i>return message</i> merupakan balasan/hasil dari <i>self message</i> yang berisi suatu nilai kepada objek itu sendiri.

### 2.10.3. *Functional View*



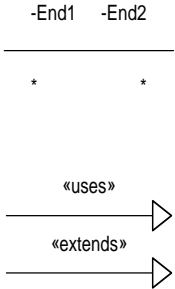
*Functional View* terbentuk oleh *Use Case Diagram* dan *Activity Diagram*.

#### 2.10.3.2 *Use Case Diagram*

*Use case diagram* menjelaskan manfaat sistem jika dilihat menurut pandangan orang yang berada diluar sistem (*actor*). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem berinteraksi dengan dunia luar.

*Use case diagram* dapat digunakan selama proses analisis untuk menangkap *requirements system* dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case diagram* menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Sebuah model mungkin terdapat satu atau beberapa *use case diagram*. Simbol-simbol yang digunakan dalam *use case diagram* :

Tabel 2.4 Simbol *Use Case Diagram*








No	Simbol	Nama	Deskripsi
1.		<i>Case</i>	Menggambarkan proses / kegiatan yang dapat dilakukan oleh aktor
2.		<i>Actor</i>	Menggambarkan entitas / subyek yang dapat melakukan suatu proses
3.		<i>Relation</i>	Relasi antara case dengan actor ataupun case dengan case lain.

### 3.10.3.2 *Activity Diagram*

*Activity diagram* memodelkan alur kerja sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena kita dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). *Activity diagram*

juga sangat berguna ketika kita ingin menggambarkan perilaku paralel atau menjelaskan bagaimana perilaku dalam berbagai *use case* berinteraksi. Simbol-simbol yang digunakan dalam *activity diagram* :

Tabel 2.5 Simbol *Activity Diagram*

No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Mempelihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Start State</i>	Bagaimana objek dibentuk atau diawali
4		<i>End State</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>State Transition</i>	Menunjukkan kegiatan apa berikutnya setelah suatu kegiatan
6		<i>Fork</i>	Percabangan yang menunjukkan aliran pada <i>Activity Diagram</i>
7		<i>Join</i>	Penggabungan yang menjadi arah aliran pada <i>Activity Diagram</i>
8		<i>Decision</i>	Pilihan untuk mengambil keputusan
9		<i>Flow Final</i>	Aliran Akhir