

## **BAB II**

### **TEORI PENUNJANG**

#### **1.1 Teknik Informatika**

Perkembangan teknologi dan informasi membawa dampak ke dunia Pendidikan, sehingga minat masyarakat untuk memilih Pendidikan bidang informatika terus meningkat baik di tingkat sekolah menengah maupun di perguruan tinggi. Teknik informatika mempelajari serta menerapkan prinsip-prinsip ilmu komputer dan analisis matematis dalam perancangan, pengujian, pengembangan, dan evaluasi sistem operasi, perangkat lunak (software), dan kinerja komputer. Dalam kata lain, bidang studi ini akan melakukan pembahasan terkait pengolahan data dengan memanfaatkan teknologi komputer yang menggunakan prinsip dan proses logika. [1]

#### **1.2 Sistem Pendukung Keputusan (SPK)**

Sistem Pendukung Keputusan (Decision Support System) merupakan sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan. Sistem ini bertujuan untuk menyediakan informasi yang berfungsi membantu manajer dalam pengambilan keputusan semi terstruktur yang terkadang keputusan dapat diketahui terkadang keputusan yang akan dibuat tidak jelas dan keputusan tidak terstruktur dimana tidak seorang pun tahu keputusan apa yang harus dibuat. Pada awalnya Turban & Aronson (1998), mendefinisikan sistem penunjang keputusan (*Decision Support Systems – DSS*) sebagai sistem yang digunakan untuk mendukung dan membantu pihak manajemen melakukan pengambilan keputusan pada kondisi semi terstruktur dan tidak terstruktur [2]. Pada dasarnya konsep *DSS* hanyalah sebatas pada kegiatan membantu para manajer melakukan penilaian serta menggantikan posisi dan peran manajer. Konsep *DSS* pertama kali diperkenalkan pada awal tahun 1970-an oleh Michael Scott Morton, yang selanjutnya dikenal dengan istilah “*Management*

*Decision System*". Konsep *DSS* merupakan sebuah sistem interaktif berbasis komputer yang membantu pembuatan keputusan memanfaatkan data dan model untuk menyelesaikan masalah-masalah yang bersifat tidak terstruktur dan semi terstruktur. *DSS* dirancang untuk menunjang seluruh tahapan pembuatan keputusan, yang dimulai dari tahapan mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan yang digunakan dalam proses pembuatan keputusan sampai pada kegiatan mengevaluasi pemilihan alternatif. [2]

### **1.3 Metode *Analytical Hierarchy Process (AHP)***

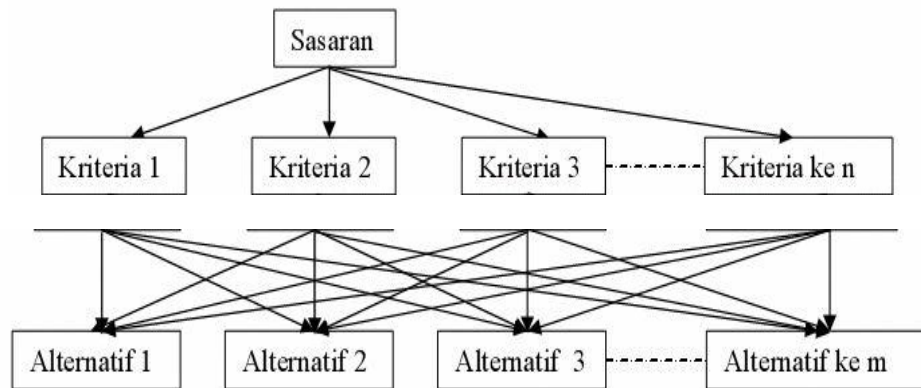
Merupakan salah satu metode pengambilan keputusan yang dikembangkan oleh Dr. Thomas L Saaty, seorang ahli matematika daripada Universitas Pittsburg Amerika Serikat pada awal tahun 1970-an. *AHP* merupakan salah satu model pengambilan keputusan yang dapat membantu kerangka berpikir manusia. Dasar berpikirnya metode *AHP* adalah proses membentuk skor secara numerik untuk menyusun ranking setiap alternatif keputusan yang berbasis pada bagaimana sebaiknya alternatif itu dicocokkan dengan kriteria pembuat keputusan. Pada hakekatnya *AHP* merupakan suatu model pengambilan keputusan yang komprehensif yang memperhitungkan hal-hal yang bersifat kualitatif dan kuantitatif. *AHP* juga memungkinkan ke struktur suatu sistem dan lingkungan ke dalam komponen saling berinteraksi dan kemudian menyatukan mereka dengan mengukur dan mengatur dampak dari komponen kesalahan sistem [3]

Metode *AHP* ini membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria, pihak yang berkepentingan, hasil dan dengan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas. Metode *AHP* merupakan penggabungan antara kekuatan dari perasaan dan logika yang bersangkutan pada berbagai persoalan, lalu mensintesis berbagai pertimbangan yang beragam menjadi hasil yang cocok dengan perkiraan kita secara intuitif sebagaimana yang dipresentasikan pada pertimbangan yang telah dibuat. [3]

### 1.3.1 Prosedur Perhitungan Metode AHP

Pada dasarnya terdapat beberapa tahapan atau prosedur yang harus dilakukan dalam proses perhitungan bobot dengan metode AHP. Adapun tahap-tahap dalam perhitungan bobo tantara lain :

1. Menyusun hirarki dari permasalahan yang dihadapi, yaitu mendefinisikan masalah dan membentuk solusi yang diinginkan. Kemudian membuat struktur hirarki yang diawali dengan tujuan umum, dilanjutkan dengan kriteria dan alternatif-alternatif pada tingkatan yang paling bawah. Seperti terlihat pada gambar II.1. [3]



Sumber : Saaty, 2008.

*Gambar 0.1 Hirarki Permasalahan*

2. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap masing-masing kriteria lain (Saaty, 2008), kriteria 1 sampai 9 adalah skala terbaik dalam membandingkan dalam membandingkan elemen. Tabel II.1 menunjukkan nilai dan definisi pendapat kualitatif dari skala perbandingan berpasangan Saaty. [3]

Tabel 0.1 Skala nilai perbandingan berpasangan

<b>Intensitas Kepentingan</b>	<b>Definisi</b>	<b>Penjelasan</b>
1	Elemen yang satu sama pentingnya dibanding dengan elemen yang lain (equal importance)	Kedua elemen menyumbang sama besar pada sifat tersebut
3	Elemen yang satu sedikit lebih penting dari pada elemen yang lain (moderate more importance)	Pengalaman menyatakan sedikit memihak pada satu elemen
5	Elemen yang satu jelas lebih penting dari pada elemen yang lain (essential, strong more importance)	Pengalaman menunjukkan secara kuat memihak pada satu elemen
7	Elemen yang satu sangat jelas lebih penting dari pada elemen yang lain (demonstrated importance)	Pengalaman menunjukkan secara kuat disukai dan didominasi oleh sebuah elemen tampak dalam
9	Elemen yang satu mutlak lebih penting dari pada elemen yang lain (absolutely more importance)	Pengalaman menunjukkan satu elemen sangat jelas lebih penting
2,4,6,8	Apabila ragu-ragu antara dua nilai yang berdekatan	Nilai ini diberikan bila diperlukan kompromi

	(grey area)	
1/(2-19)	Jika kriteria C1 mendapatkan satu angka bila dibandingkan dengan kriteria C2 memiliki nilai kebalikan bila dibandingkan C1	Jika kriteria C1 mempunyai nilai x bila dibandingkan dengan kriteria C2, maka kriteria C2 mendapatkan

Sumber : Saaty, 2008.

Contoh matriks berpasangan dapat dilihat pada tabel II.2 berikut. [3]

*Tabel 0.2 Matriks Berpasangan*

	K <sub>1</sub>	K <sub>2</sub>	...	K <sub>n</sub>
K <sub>1</sub>	Nilai perbandingan K <sub>11</sub>	Nilai perbandingan K <sub>12</sub>		Nilai perbandingan K <sub>1n</sub>
K <sub>2</sub>	Nilai perbandingan K <sub>21</sub>	Nilai perbandingan K <sub>22</sub>		Nilai perbandingan K <sub>2n</sub>
...				
K <sub>n</sub>	Nilai perbandingan K <sub>n1</sub>	Nilai perbandingan K <sub>n2</sub>		Nilai perbandingan K <sub>nn</sub>

Sumber : Saaty, 2008.

3. Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas atau *Total Priority Value (TPV)* [1].
- Menjumlahkan nilai-nilai dari setiap kolom pada matriks seperti terlihat pada tabel II.3 berikut.

*Tabel 0.3 Penjumlahan kolom*

	$K_1$	$K_2$	...	$K_n$
$K_1$	Nilai perbandingan $K_{11}$	+...	...	+...
$K_2$	Nilai perbandingan $K_{21}$	+...	...	+...
...	...	...	...	...
$K_n$	Nilai perbandingan $K_{n1}$	+...	...	+...
$\Sigma$ kolom	$\Sigma$ Kolom K1	$\Sigma$ Kolom K2	...	$\Sigma$ Kolom Kn

Sumber : Saaty, 2008.

- Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks, kemudian menjumlahkan nilai-nilai dari baris dan membaginya dengan jumlah elemen ( $n$ ) atau kriteria untuk mendapatkan nilai prioritas (*TPV*). Penjumlahan baris dapat dilihat pada tabel II.4 berikut:

Tabel 0.4 Penjumlahan baris

	$K_1$	$K_2$	...	$K_n$	TPV
$K_1$	Nilai perbandingan $K_{11}$	+...	...	+...	$\Sigma$ Baris $K_1/n$
$K_2$	Nilai perbandingan $K_{21}$	+...	...	+...	$\Sigma$ Baris $K_2/n$
...	...	...	...	...	
$K_n$	Nilai perbandingan $K_{n1}$	+...	...	+...	$\Sigma$ ...
$\Sigma$ kolom	$\Sigma$ Kolom $K_1$	$\Sigma$ Kolom $K_2$	...	$\Sigma$ Kolom $K_n$	$\Sigma$ Baris $K_n/n$

Sumber : Saaty, 2008.

- c. Memeriksa konsistensi (*Consistency Ratio* atau *CR*) matriks perbandingan suatu kriteria. Matrik perbandingan dinyatakan konsisten jika nilai  $CR \leq 0,1$ , tapi nilai  $CR > 0,1$  maka pertimbangan yang dibuat perlu diperbaiki dan diteliti kembali.
1. Bobot yang didapat dari nilai *TPV* dikalikan dengan nilai-nilai awal elemen matriks perbandingan yang telah diubah menjadi bentuk

desimal. Adapun tabel perkalian *TPV* dengan elemen matriks dapat dilihat pada tabel II.5 di bawah ini:

*Tabel 0.5 Perkalian TPV dengan elemen matriks*

	TPV $K_1$	TPV $K_2$	TPV $K_n$
$K_1$	Nilai perbandingan $K_{11} * TPV K_1$	...	Nilai perbandingan $K_{1n}$ $* TPV K_1$
$K_2$	Nilai perbandingan $K_{21} * TPV K_1$	...	Nilai perbandingan $K_{2n}$ $* TPV K_1$
...	....	...	...
$K_n$	Nilai perbandingan $K_{n1} * TPV K_1$	...	Nilai perbandingan $K_{nn}$ $* TPV K_1$

Sumber : Saaty, 2008.

- Menjumlahkan nilai setiap baris yaitu nilai hasil perkalian *TPV* dengan elemen matriks. Adapun tabel penjumlahan baris setelah perkalian *TPV* dapat dilihat pada tabel II.6 di bawah ini.

*Tabel 0.6 Penjumlahan baris setelah perkalian TPV*

	$K_1$	$K_2$	...	$K_n$	TPV
$K_1$	Nilai perbandingan	+...	...	+...	$\sum$ Baris



	$K_{11} * TPV K_1$				$K_1$
$K_2$	Nilai perbandingan $K_{21} * TPV K_1$	+...	...	+...	$\sum$ Baris $K_2$
...	....	...	...	...	...
$K_n$	Nilai perbandingan $K_{n1} * TPV K_1$	+...	...	+...	$\sum$ Baris $K_n$

Sumber : Saaty, 2008.

3. Hasil dari penjumlahan baris dibagi dengan nilai *TPV* setiap kriteria, sehingga didapatkan nilai  $\lambda$  Maks setiap baris

$$\begin{bmatrix} \sum \text{Baris } K_1 \\ \dots \\ \sum \text{Baris } K_2 \end{bmatrix} + \begin{bmatrix} TPV K_1 \\ \dots \\ TPV K_2 \end{bmatrix} = \begin{bmatrix} \lambda \text{ Maks } K_1 \\ \dots \\ \lambda \text{ Maks } K_n \end{bmatrix}$$

4. Nilai rata-rata  $\lambda$  Maks didapat dengan cara menjumlahkan semua nilai  $\lambda$  Maks setiap baris dan dibagi dengan jumlah kriteria (n) seperti pada rumus berikut :

$$\lambda \text{ Maks} = (\lambda \text{ Maks } K_1 + \dots + \dots + \lambda \text{ Maks } K_n) / n$$

Keterangan :

$\lambda$  Maks = Nilai rata-rata dari keseluruhan kriteria

n = Jumlah kriteria dalam matriks perbandingan

5. Setelah mendapatkan nilai  $\lambda$  Maks, Langkah selanjutnya mencari nilai *Consistency Index ( CI )* dengan persamaan sebagai berikut :

$$CI = (\lambda \text{ Maks}-n) / (n-1)$$

6. Setelah *CI* didapat, maka Langkah selanjutnya menghitung nilai *Consistency Ratio (CR)* dengan mengacu pada *random index (RI)* yang dapat diambil dengan ketentuan sesuai dengan jumlah kriteria yang digunakan. Nilai *RI* dapat dilihat pada tabel II.7. Adapun rumus *Consistency Ratio (CR)* adalah :

$$CR = CI/RI$$

Keterangan :

*CR* = *Consistency Ratio*

*CI* = *Consistency Index*

*RI* = *Random Index*

7. Langkah perhitungan untuk mendapatkan nilai sub kriteria sama seperti Langkah 2 dan 3 .
8. Menghitung nilai rating dari setiap sub kriteria dengan cara menormalisasikan, yaitu membagi nilai prioritas yang didapat dengan nilai prioritas terbesar.

#### **1.4 WEB**

*WEB* (bahasa Inggris: website) adalah sekumpulan halaman web yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi. Sebuah situs web biasanya ditempatkan setidaknya pada sebuah server web yang dapat diakses melalui jaringan seperti Internet, ataupun jaringan wilayah lokal (LAN) melalui alamat Internet yang dikenali sebagai URL. Gabungan atas semua situs yang dapat diakses publik di Internet disebut pula sebagai World Wide Web atau lebih dikenal dengan singkatan WWW. Meskipun setidaknya halaman beranda situs Internet umumnya dapat diakses publik secara bebas, pada praktiknya tidak semua situs memberikan kebebasan bagi publik untuk mengaksesnya, beberapa situs web mewajibkan pengunjung untuk melakukan pendaftaran sebagai anggota, atau bahkan

meminta pembayaran untuk dapat menjadi anggota untuk dapat mengakses isi yang terdapat dalam situs web tersebut, misalnya situs-situs yang menampilkan pornografi, situs-situs berita, layanan surel (e-mail), dan lain-lain. Pembatasan-pembatasan ini umumnya dilakukan karena alasan keamanan, menghormati privasi, atau karena tujuan komersial tertentu. [5]

### 1.5 *HTML ( Hypertext Markup Language )*

*HyperText Markup Language (HTML)* adalah bahasa inti dari hampir semua konten Web. Sebagian besar dari apa yang Anda lihat pada layar browser anda adalah sebuah deskripsi, secara mendasar, menggunakan *HTML*. Lebih tepatnya, *HTML* adalah bahasa yang menggambarkan struktur dan isi semantik dari sebuah dokumen Web. Konten dalam halaman Web ditandai dengan elemen *HTML* seperti `<img>`, `<title>`, `<p>`, `<div>`, dan sebagainya. *HTML* adalah standar internasional yang spesifikasinya didasari dari *World Wide Web Consortium* dan *WHATWG*. Ini dianggap sebagai "standar dasar" dan secara teknik selalu ditingkatkan [6]

Disebut *hypertext* karena di dalam *HTML* sebuah text biasa dapat berfungsi lain, kita dapat membuatnya menjadi link yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik text tersebut. Kemampuan text inilah yang dinamakan *hypertext*, walaupun pada implementasinya nanti tidak hanya text yang dapat dijadikan link. [6]

Disebut *Markup Language* karena bahasa *HTML* menggunakan tanda (mark), untuk menandai bagian-bagian dari text. Misalnya, text yang berada di antara tanda tertentu akan menjadi tebal, dan jika berada di antara tanda lainnya akan tampak besar. Tanda ini di kenal sebagai *HTML* tag. *HTML* merupakan bahasa dasar pembuatan web. Disebut dasar karena dalam membuat web, jika hanya menggunakan *HTML* tampilan web terasa hambar. Terdapat banyak bahasa pemrograman web yang ditujukan untuk memanipulasi kode *HTML*, seperti *Java Script* dan *PHP*. Akan tetapi sebelum anda belajar *JavaScript* maupun *PHP*, memahami *HTML* merupakan hal yang paling awal [6]

Cara kerja Dokumen *HTML* mirip dengan dokumen tulisan biasa. Akan tetapi, dalam dokumen *HTML*, sebuah tulisan dapat memuat instruksi yang ditandai dengan kode atau lebih dikenal dengan TAG tertentu. Contohnya, jika ingin membuat tulisan ditampilkan menjadi tebal, seperti TULISAN TEBAL, penulisan dalam dokumen dilakukan dengan cara TULISAN TEBAL. Tanda dipakai untuk mengaktifkan instruksi cetak tebal, kemudian diikuti tulisan yang ingin ditebalkan. Setelah itu, diakhiri dengan tanda untuk menonaktifkan cetak tebal tersebut. [6]

*HTML* lebih memfokuskan pada penggambaran komponen-komponen struktur dan format dalam halaman situs daripada menentukan penampilannya. Sementara, web browser digunakan untuk menginterpretasikan susunan halaman ke web browser dengan menggunakan jenis tulisan, warna, garis, dan teks yang dikehendaki ke komputer yang menampilkan halaman situs [6]

## 1.6 CSS (*Cascading Style Sheet*)

*CSS (Cascading Style Sheet)* adalah salah satu bahasa desain web (*style sheet language*) yang mengontrol format tampilan sebuah halaman web yang ditulis dengan menggunakan penanda(markup language). Biasanya *CSS* digunakan untuk mendesain sebuah halaman *HTML* dan *XHTML*, tetapi sekarang *CSS* bisa diaplikasikan untuk segala dokumen *XML*, termasuk *SVG* dan *XUL* bahkan *ANDROID*. [7]

*CSS* dibuat untuk memisahkan konten utama dengan tampilan dokumen yang meliputi layout, warna dan font. Pemisahan ini dapat meningkatkan daya akses konten pada web, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi halaman untuk sebuah formatting dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik tableless pada desain web. [7]

*CSS* juga memungkinkan sebuah halaman untuk ditampilkan dalam berbagai style dengan menggunakan metode pembawaan yang berbeda pula, seperti on-screen,

in-print, *by voice*, dan lain-lain. Sementaraitu, pemilik konten web bisa menentukan link yang menghubungkan konten dengan file *CSS*. [7]

Tujuan utama *CSS* diciptakan untuk membedakan konten dari dokumen dan dari tampilan dokumen, dengan itu, pembuatan ataupun pemrograman ulang web akan lebih mudah dilakukan. Hal yang termasuk dalam desain web diantaranya adalah warna, ukura dan formatting. Dengan adanya *CSS*, konten dan desain web akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan-tampilan tertentu dalam suatu web, sehingga akan memudahkan dalam membuat halaman web yang banyak, yang pada akhirnya dapat memangkas waktu pembuatan web. [7]

Fungsi utama *CSS* adalah merancang, merubah, mendisain, membentuk halaman wesite(blog juga website). dan isi dari halaman website adalah tag-tag html, logikanya css itu dapat merubah tag-tag html(yang sederhana) sehingga menjadi lebih fungsional dan menarik. [7]

Cara kerja *CSS* sangatlah mudah, karena *CSS* hanya membutuhkan style sebagai penentu dari font, warna, dan format-format lain untuk memformat atribut sebuah halaman web yang kita buat. Tiap style memiliki dua buah elemen dasar yaitu “selector” dan “declaration”. Sebuah selector biasanya adalah tag *HTML*, sementara declaration adalah satu atau beberapa perintah / nilai dari *CSS* yang menunjukkan type bentuk yang diaplikasikan pada selector. Declaration ini biasanya di tandai dengan tanda kurung kurawal “{ }” , dan perintah atau nilai *CSS* yang berbeda dipisahkan satu dengan yang lain dengan menggunakan titik-koma “;” seperti terlihat pada contoh berikut : [7]

```
< style type="text/css"> .teks {font-family:verdana; color:blue;} </style>
```

## **1.7 Firebase Firestore**

Firebase Firestore merupakan database atau media penyimpanan yang dirancang khusus dan disediakan oleh google. Firebase menyediakan berbagai

fungsionalitas, seperti analisis, database, pesan, dan pelaporan error sehingga Anda dapat bergerak dengan cepat dan fokus pada pengguna. Firebase dikembangkan di infrastruktur Google dan diskalakan secara otomatis, bahkan untuk aplikasi terbesar.

## 1.8 Javascript

Menurut Yeni Kustiyahningsih dan Devie Rosa Anamisa, *JavaScript* adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen *HTML*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap *HTML* dengan mengizinkan pengekseskuan perintah-perintah disisi user variabel atau fungsi dengan nama TEST berbeda dengan variabel dengan nama test dan setiap instruksi diakhiri dengan artinya disisi browser bukan disisi server web. *JavaScript* adalah bahasa yang “case sensitive” artinya memnedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh karakter titik koma. [5]

## 1.9 WEB Server

*Web server* merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman *web* yang umumnya berbentuk dokumen HTML, konsep *web server* antara lain:. [5]

- a. *Web server* merupakan mesin aplikasi atau software yang beroperasi dalam medistribusikan web page ke user, tentu saja sesuai dengan permintaan user.
- b. Hubungan antara *web server* dan *browser* internet merupakan gabungan atau jaringan komputer yang berada diseluruh dunia. Setelah terhubung secara fisik, *protocol TCP/IP (networking protocol)* yang memungkinkan semua komputer dapat berkomunikasi antar satu dengan lainnya. Pada saat aplikasi *browser* meminta data *web page* ke *server* maka instruksi permintaan data oleh *browser* tersebut dikemas dalam TCP yang merupakan *protocol transport* dan dikirim ke alamat yang merupakan *protocol* berikutnya yaitu *hyper text transfer protocol (HTTP)*. Data yang diparsing dari *browser* ke *web server* disebut sebagai

HTTP *request* yang meminta halaman *web* dan kemudian *web server* akan mencari data HTML yang dibutuhkan dan dikemas dalam TCP *protocol* kemudian dikirim kembali ke *browser*. Data yang dikirim dari *server* ke *browser* disebut sebagai HTTP *response*. Jika data yang diminta oleh *browser* tidak ditemukan pada *web server* maka akan menampilkan *error* pada *web page* yaitu *Error: 404 Page Not Found*. [5]