

BAB III

LANDASAN TEORI

3.1 *Haversine Formula*

Teorema *Haversine Formula* adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dari *longitude* dan *latitude*. Ini merupakan bentuk persamaan khusus dari trigonometri bola, *law of haversines*, mencari hubungan sisi dan sudut pada garis segitiga dalam bidang bola (Sulistio, 2019).

Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza y Rios di tahun 1801. Istilah *haversine* ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Rios menggunakan *haversine* pertama kali dalam penelitiannya tentang “Masalah Utama Astronom Nautical”, Proc.Royal Soc, Dec 22. 1796. *Haversine* digunakan untuk menemukan jarak antar bintang (Sulistio, 2019).

Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari = $R = 6.371$ km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah lon_1 , lat_1 , dan lon_2 , lat_2 , maka rumus *Haversine* dapat ditulis dengan persamaan berikut ini (Palupi et al., 2021):

Rumus *Haversine*:

$$\begin{aligned} X &= (lon_2 - lon_1) * \cos((lat_1 + lat_2)/2) \\ Y &= (lat_2 - lat_1) \\ d &= \sqrt{x^2 + y^2} * R \end{aligned}$$

Keterangan:

X = *Longitude* (Lintang)

Y = *Latitude* (Bujur)

d = Jarak

R = Radius Bumi = 6371 km

1 derajat= 0.0174532925 radian

Contoh menghitung jarak yaitu jarak SMAN 3 Yogyakarta ke Toko Buku Gramedia.

Dengan penyelesaian sebagai berikut :

a. Menentukan lokasi awal

Lokasi : SMAN 3 Yogyakarta

lon1 : 110.37335841453164

lat1 : -7.786349554216108

b. Menentukan Lokasi Tujuan

Lokasi : Toko Buku Gramedia

lon2 : 110.37450694359015

lat2 : -7.783334667218786

c. Merubah derajat keradian

lon1 = 110.37335841453164 x 0.0174532925 radian

= 1.92637850862 radian

lat1 = 7.786349554216108 x 0.0174532925 radian

= -0.13589743627 radian

lon2 = 110.37450694359015 x 0.0174532925 radian

= 1.92639855423 radian

lat2 = -7.783334667218786 x 0.0174532925 radian

= -0.13584481657 radian

d. Mencari Nilai x dan y

x = (lon2-lon1) * cos ((lat1+lat2)/2)

= (1.92639855423 - 1.92637850862)* cos

((-0.13589743627 +(-0.13584481657))/2)

= 0.00001986086

y = (lat2 - lat1)

= (-0.13584481657 -(-0.13589743627))

= 0.0000526197

e. Mencari Nilai d (jarak)

d = sqrt (x*x + y*y) * R

= sqrt (0.00001986086 x 0.00001986086

+ 0.0000526197 x 0.0000526197) x 6371

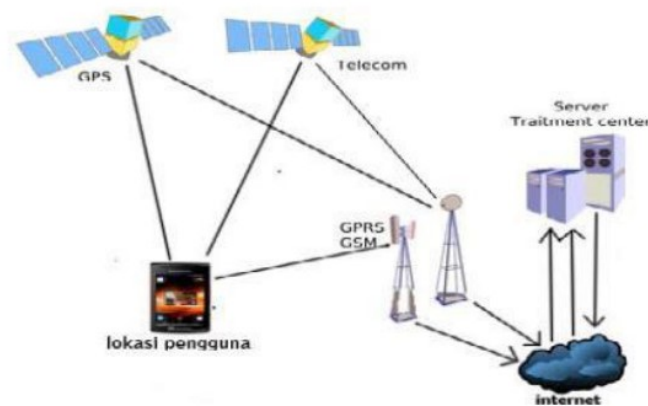
= 0.0000562431 x 6371

= 0.355287679 km

= 358.324790 m

3.2 Geolocation

Geolocation adalah identifikasi lokasi geografis suatu objek pada dunia nyata. *Geolocation* mempunyai kaitan erat dengan *positioning*, perbedaannya adalah *geolocation* lebih spesifik dalam menentukan sebuah lokasi (misalnya alamat jalan) dibandingkan dengan *positioning* yang hanya mencakup sekumpulan koordinat geografis. Suatu lokasi geografis mengandung nilai *latitude* dan *longitude* (Widiastuti & Azizah Widiastuti, 2018).



Gambar 3.1 *Geolocation*

3.3 Confusion Matrix

Confusion Matrix merupakan sebuah teknik yang digunakan dalam *data mining* dan *machine learning* untuk menghitung seberapa baik sebuah model dapat memprediksi label dari sebuah data. Teknik ini sering digunakan dalam *evaluasi model classification* yang mana model harus memprediksi label dari sebuah data berdasarkan atribut-atribut yang ada. *Confusion Matrix* adalah sebuah tabel yang menggambarkan seberapa sering model memprediksi label yang benar dan salah. Setiap baris dari tabel tersebut mewakili sebuah label *actual*, sedangkan setiap kolom mewakili label yang diprediksi oleh model. *Confusion Matrix* membantu mengukur kinerja dimana output algoritma dapat berada dalam dua kategori atau lebih biasa disebut *Positive* atau *Negative*, Ya atau Tidak. Setiap tabel terdiri dari

empat sel, masing-masing mewakili kombinasi unik dari nilai prediksi atau *actual* (KantinIT, 2023). Empat hasil potensial tersebut adalah:

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

1. **True Positive (TP)**

Merupakan data positif yang diprediksi benar. Contohnya, pasien menderita hipertensi (*class 1*) dan model membuat prediksi bahwa pasien tersebut hipertensi (*class 1*).

2. **True Negative (TN)**

Merupakan data *negative* yang diprediksi benar. Contoh, pasien tidak menderita hipertensi (*class 2*) dan model membuat prediksi bahwa pasien tersebut tidak menderita hipertensi (*class 2*).

3. **False Positive (FP) – Type I Error**

Merupakan data *negative* namun diprediksi sebagai data positif. Contoh, pasien tidak menderita hipertensi (*class 2*) tetapi model memprediksi pasien tersebut menderita hipertensi (*class 1*).

4. **False Negative (FN) – Type II Error**

Merupakan data positif namun diprediksi sebagai data *negative*. Contoh, pasien menderita hipertensi (*class 1*) tetapi model memprediksi pasien tersebut tidak menderita hipertensi (*class 2*).

A. Accuracy

Accuracy merupakan metode pengujian berdasarkan tingkat kedekatan antara nilai prediksi dengan nilai *actual*. Dengan mengetahui jumlah data yang diklasifikasikan secara benar maka dapat diketahui akurasi hasil prediksi.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\%$$

B. Precision

Precision merupakan metode pengujian dengan melakukan perbandingan jumlah informasi relevan yang didapatkan sistem dengan jumlah seluruh informasi yang terambil oleh sistem baik yang relevan maupun tidak.

$$Precision = \frac{TP}{TP+FP} \times 100\%$$

C. Recall

Recall merupakan metode pengujian yang membandingkan jumlah informasi relevan yang didapatkan system dengan jumlah seluruh informasi relevan yang ada dalam koleksi informasi (baik yang terambil atau tidak terambil oleh system).

$$Recall = \frac{TP}{TP+FN} \times 100\%$$

D. F-Measure atau F1-Score

F-Measure atau disebut dengan *F1-Score* menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Accuracy* tepat kita gunakan sebagai acuan performansi algoritma jika data set kita memiliki jumlah data *false negative* dan *false positive* yang sangat mendekati (*symmetric*). Namun jika jumlahnya tidak mendekati, maka sebaiknya kita menggunakan *F1-Score*.

$$F1 = \frac{2 \times Recall \times Precision}{Recall+Precision} \times 100\%$$

Berikut contoh perhitungan *Confusion Matrix* untuk model memprediksi pasien yang menderita hipertensi.

	Prediksi “Hipertensi”	Prediksi “Tidak Hipertensi”
Aktual “Hipertensi”	50	10
Aktual “Tidak Hipertensi”	5	35

Dari table di atas dapat dilihat bahwa model tersebut memprediksi 50 pasien hipertensi dengan benar, namun terdapat 10 pasien yang sebenarnya tidak hipertensi tapi diprediksi sebagai hipertensi oleh model. Sebaliknya model tersebut memprediksi 35 pasien yang sebenarnya tidak hipertensi dengan benar, namun terdapat 5 pasien yang sebenarnya hipertensi tapi diprediksi sebagai tidak hipertensi oleh model.

Untuk perhitungan akurasi, presisi dan recall sebagai berikut.

$$\begin{aligned} \text{Akurasi} &= \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \\ &= \frac{50+35}{50+10+5+35} \times 100\% = 85\% \end{aligned}$$

$$\begin{aligned} \text{Presisi “Hipertensi”} &= \frac{TP}{TP+FN} \times 100\% \\ &= \frac{50}{50+5} \times 100\% = 91\% \end{aligned}$$

$$\begin{aligned} \text{Recall “Hipertensi”} &= \frac{TP}{TP+FP} \times 100\% \\ &= \frac{50}{50+10} \times 100\% = 81\% \end{aligned}$$

$$\begin{aligned} \text{Presisi “Tidak Hipertensi”} &= \frac{TN}{TN+FP} \times 100\% \\ &= \frac{35}{35+10} \times 100\% = 78\% \end{aligned}$$

$$\begin{aligned} \text{Recall “Tidak Hipertensi”} &= \frac{TN}{TN+FN} \times 100\% \\ &= \frac{35}{35+5} \times 100\% = 88\% \end{aligned}$$

Dari hasil di atas, dapat kita lihat bahwa model tersebut memiliki akurasi sebesar 85%, presisi sebesar 91% untuk memprediksi label hipertensi dan *recall* sebesar

83% untuk label hipertensi. Namun, untuk label tidak hipertensi, model tersebut hanya memiliki presisi sebesar 78% dan *recall* sebesar 88%.

3.4 HTML

HyperText Markup Language (HTML) adalah bahasa web. Saat kita membuka halaman web di *browser* web seperti Internet Explorer, Firefox, atau Safari, *browser* akan mengunduh dan menampilkan HTML. Pada intinya, HTML hanyalah sebuah dokumen, sama seperti dokumen yang dibuat di *word*. Program seperti Microsoft Word digunakan untuk melihat dokumen pengolah kata karena Microsoft Word tahu cara membaca dan menampilkan teks. Demikian juga dengan web, *browser* web adalah program yang mengetahui cara membaca dan menampilkan dokumen yang dibuat dengan HTML. Dokumen Word Process dapat dibuat dan dibaca dengan satu program. Di sisi lain, dokumen HTML membutuhkan program yang berbeda untuk pembuatan dan pembacaan; Kita tidak dapat membuat dokumen HTML menggunakan *browser*. Untuk membuat dokumen HTML kita membutuhkan program yang disebut *editor*. Editor ini bisa sesederhana program Notepad yang disertakan dengan Microsoft Windows atau serumit Eclipse dan Microsoft Visual Studio. Umumnya kita dapat menggunakan program yang sama untuk membuat dokumen HTML yang digunakan untuk membuat program PHP (Sholikhah, 2022).

HTML, sering dikenal sebagai *Hyper-Text Markup Language*, adalah bahasa utama yang digunakan di seluruh dunia oleh situs web. Hampir semua *website* yang tersedia di internet dibuat dengan menggunakan variasi HTML. Pengembang dapat menambahkan teks, multimedia, dan tautan ke *browser* yang kompatibel dengan atribut atau elemen lain dengan menggunakan HTML. Dari elemen yang dapat terhubung ke dokumen Anda (hiperteks) hingga elemen yang membuat dokumen Anda lebih interaktif, semuanya terdapat dalam satu file HTML. Dengan kata lain, HTML (*Hypertext Markup Language*) adalah bahasa yang biasa digunakan untuk menandai dokumen yang akan dipublikasikan secara online. Dalam bahasa sederhana, HTML mengilustrasikan struktur halaman Web, yang terdiri dari

serangkaian elemen HTML, untuk menjelaskan kepada browser web cara menampilkan konten yang telah dipilih. HTML memiliki standar yang dibuat oleh W3C (*World Wide Web Consortium*) pada tahun 1997 dan semakin lama semakin bermanfaat. Dalam HTML, ada tag yang digunakan untuk menunjukkan struktur. Tag dan elemen dapat dilindungi dengan menggunakan karakter `<` dan `>` (Kurniawan, 2023).

Dokumen HTML dapat disimpan di folder *Documents* dalam komputer. Namun jika dibuka, hanya kita saja yang bisa melihat isinya. Untuk menyasati hal ini, kita memerlukan lebih banyak sumber daya, sebut saja sebagai *server* web. Menyimpan dokumen di *server* web memungkinkan orang lain untuk dapat melihat dokumen kita. *Server* web adalah komputer yang menjalankan perangkat lunak khusus yang mengetahui cara mengirim (atau menyajikan) halaman web ke banyak orang pada saat yang bersamaan. Dokumen HTML diatur dalam urutan tertentu, dengan bagian-bagian tertentu. Mereka terstruktur sedemikian rupa agar *browser* web dapat membaca dan menampilkannya. Karena hal tersebut, ketika kita membuat dokumen HTML, kita harus mengikuti strukturnya dan menyiapkan dokumen agar browser dapat membacanya. Browser web dapat menampilkan beberapa jenis dokumen, tidak hanya HTML, jadi saat membuat dokumen web, hal pertama yang harus dilakukan adalah memberi tahu browser tentang jenis dokumen apa yang akan digunakan. Kita harus menetapkan jenis dokumen dengan baris khusus HTML di bagian atas dokumen. Dalam istilah teknis, tipe dokumen disebut Document Text Declaration, atau disingkat DTD. Dalam versi HTML sebelumnya, developer terus-menerus menyalin dan menempelkan jenis dokumen ke dalam dokumen karena panjang dan rumit. Dengan dirilisnya versi terbaru HTML, yang disebut HTML5, jenis dokumen menjadi sangat disederhanakan (Sholikhah, 2022). Jenis dokumen untuk HTML5 adalah:

```
<!doctype html>
```

Ini akan menjadi baris pertama dari setiap dokumen HTML yang kita buat, sebelum membuat dokumen lainnya. Setiap kali, kita perlu menampilkan HTML dan menyertakan jenis dokumennya yang kadang-kadang disebut *doctype*. Bisa jadi

kita akan tergoda untuk menggunakan `<!doctype html5>`, tetapi tidak ada nomor versi yang terkait dengan jenis dokumen HTML5. Karena ketika versi HTML berikutnya keluar, kita tidak perlu lagi memperbarui atau mengubah semua jenis dokumen yang telah kita buat pada HTML sebelumnya ke versi HTML6. Ada berbagai jenis dokumen lain yang lebih lama, diantaranya adalah :

HTML 4.01 Strict `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"`

`"http://www.w3.org/TR/html4/strict.dtd">`

HTML 4.01 Transitional `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"`

`"http://www.w3.org/TR/html4/loose.dtd">`

XHTML 1.0 Strict `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"`

`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

XHTML 1.1 DTD `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"`

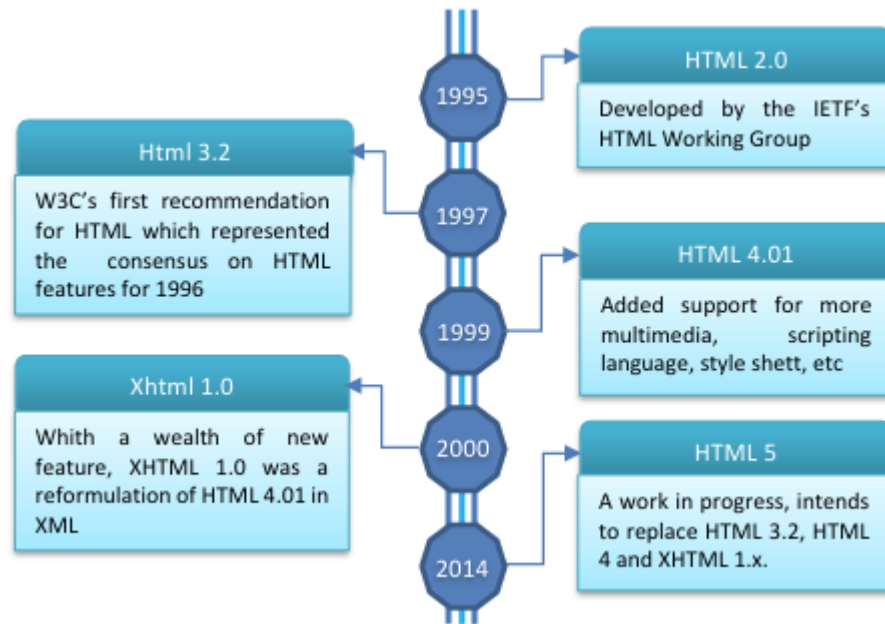
`"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">`

3.4.1 Sejarah Singkat HTML

Sejarah singkat HTML menurut (Kurniawan, 2023) yaitu :

1. Awal mula kemunculan HTML yaitu dimulai dari seorang ilmuwan yang bernama Tim Berners-Lee yang bekerja di CERN.
2. CERN bukanlah bisnis yang berhubungan dengan teknologi atau bahkan internet. CERN adalah singkatan dari Dewan Riset Nuklir Eropa, atau singkatnya CERN. Ini mengacu pada Komisi Eropa untuk Fisika Nuklir.
3. Sejak awal, HTML telah mengalami banyak bug, dan versi HTML terbaru saat tulisan ini dibuat adalah HTML5. Dalam hal bahasa *markup*, HTML5 telah memimpin akhir-akhir ini dan sudah menyertakan beberapa fitur yang ada di versi HTML sebelumnya, serta fitur yang dihilangkan tetapi ternyata

ada di versi XHTML sebelumnya. Meskipun tidak ada pembaruan versi khusus untuk setiap rilis baru, fitur baru ditambahkan ke HTML5 hampir setiap hari. Baik audio maupun video pada dasarnya berbeda antara HTML dan HTML5, tetapi di setiap fitur HTML, masing masing adalah produk dari HTML5, seperti yang dapat dilihat pada contoh gambar di bawah ini.



Gambar 3.2 Perubahan Versi HTML

Halaman web yang baik disusun dalam urutan yang logis. Ini berarti elemen-elemen harus ditempatkan diberbagai urutan tertentu agar dapat dibaca dengan benar oleh browser web. Ketika aturan, ketika membuka setiap elemen maka harus menutup elemen tersebut juga menggunakan tag tertentu yang digunakan dengan tanda kurung siku dan garis miring.. Menggunakan elemen yang sesuai Halaman web menggunakan beberapa elemen halaman yang juga disebut sebagai tag. Beberapa elemen tersebut disajikan pada tabel di bawah ini (Sholikhan, 2022) :

Tabel 3.1 Berbagai Jenis Elemen dan Penggunaannya

Elemen	Deskripsi	Jenis Penggunaan
<a>	<i>Anchor</i>	Membuat link ke halaman lain atau bagian dari dokumen yang sama.
 	Jeda baris	Memasuki jeda baris atau karakter kembali.
<div>	Bagian dari halaman	Membuat area keseluruhan atau pembagian logis pada halaman, seperti bagian heading/menu, area konten, atau <i>footer</i> .
<form>	<i>Form web</i>	Membuat formulir web untuk menerima input pengguna.
<h1> hingga <h6>	<i>Heading</i>	Membuat wadah untuk judul, seperti teks judul.
<hr>	<i>Hard Rule</i>	Membuat garis horizontal.
	<i>Image</i>	Wadah untuk gambar.
<input>	<i>Input</i>	Elemen untuk menerima input pengguna.
<link>	<i>Link sumber</i>	Tautan ke sumber daya untuk halaman tersebut; jangan

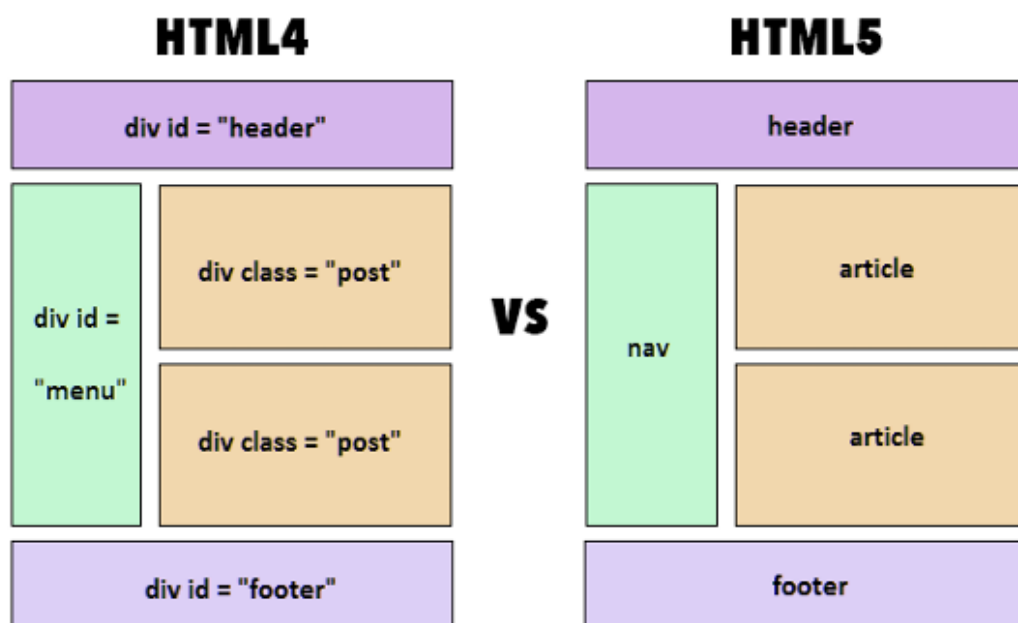
Elemen	Deskripsi	Jenis Penggunaan
		bingung dengan elemen <i>Anchor</i>
<p>	Paragraf dalam halaman	Membuat paragraf tekstual atau area dan wadah lain untuk teks.
<script>	Tag skrip	Menunjukkan skrip web atau program. Juga sering ditemukan di bagian kepala.
	Span	Membuat wadah untuk elemen. Sering digunakan bersama dengan informasi gaya.

3.4.2 HTML dan HTML5

HTML dan HTML5 menurut (Kurniawan, 2023) yaitu :

1. HTML5 mendukung SVG, kanvas, dan grafik vektor lainnya, sedangkan dalam HTML, grafik vektor hanya dapat digunakan dengan teknologi lain seperti Flash, VML (*Vector Markup Language*), *Silverlight*, dan lain-lain. SVG (*Scalable Vector Graphic*) adalah format gambar yang menggunakan XML (*eXtensible Markup Language*) sebagai dasar pembuatan gambar vektor dua dimensi. HTML5 menggunakan *database* web SQL, sedangkan HTML hanya memiliki *cache browser* yang dapat digunakan untuk tujuan ini. HTML5 juga menggunakan aplikasi untuk *caching* data.

2. Perbedaan lain antara HTML dan HTML5 adalah bahwa HTML5 tidak mencegah JavaScript berjalan di browser itu sendiri (berjalan di atas terpisah di bagian luar *browser*), sedangkan HTML5 melakukannya.
3. HTML5 mendukung kontrol formulir baru, termasuk waktu dan tanggal, email, nomor telepon, jangkauan, telegraf, url, pencarian, dan lain-lain.



Gambar 3.3 Perbandingan HTML4 dengan HTML5

3.4.3 Perbedaan Antara *Element DIV* dan *SPAN*

(Sholikhan, 2022) Elemen dan keduanya adalah jenis wadah, tetapi dengan beberapa kualitas yang berbeda. Secara *default*, elemen memiliki lebar tak terbatas (setidaknya ke tepi *browser*), yang dapat kita lihat dengan menerapkan batas ke salah satunya, seperti ini:

```
<div style="border:1px solid green;">Hello</div>
```

Namun, elemen hanya selebar teks yang dikandungnya. Oleh karena itu, baris HTML berikut membuat batas hanya di sekitar kata Hello, yang tidak meluas ke tepi kanan *browser*.

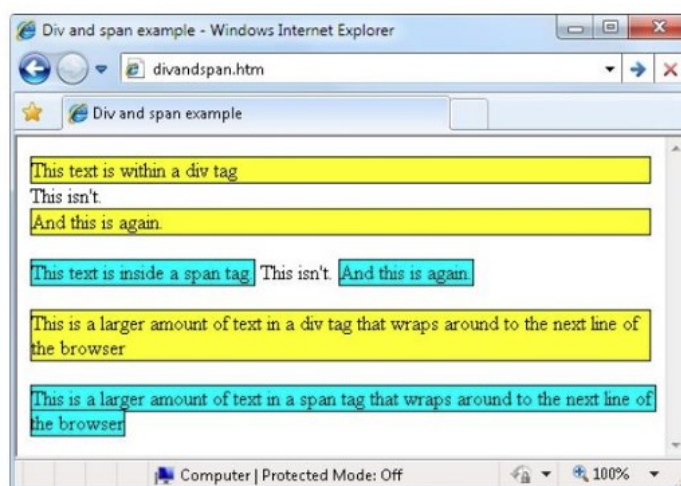
```
<span style="border:1px solid green;">Hello</span>
```

Juga, elemen `` mengikuti teks atau objek lain saat mereka membungkusnya, dan karena itu dapat memiliki batas yang rumit. Misalnya, pada contoh koding html dibawah, saya menggunakan CSS untuk membuat latar belakang semua elemen `<div>` menjadi yellow, untuk membuat semua elemen `` menjadi cyan, dan untuk menambahkan batas pada keduanya, sebelum kemudian membuat beberapa contoh `` dan `<div>` bagian.

Contoh

```
<!DOCTYPE html>
<html>
<head>
<title>Div and span example</title>
<style>
div, span { border :1px solid black; }
div { background-color:yellow; }
span { background-color:cyan; }
</style>
</head>
<body>
<div>This text is within a div tag</div>
This isn't. <div>And this is again.</div><br>
<span>This text is inside a span tag.</span>
This isn't. <span>And this is again.</span><br><br>
<div>This is a larger amount of text in a div that wraps around
to the next line of the browser</div><br>
<span>This is a larger amount of text in a span that wraps around
to the next line of the browser</span>
</body>
</html>
```

Gambar 3.4 menunjukkan seperti apa contoh ini di *browser* web. Meskipun hanya dicetak dalam nuansa abu-abu dalam buku ini, gambar tersebut dengan jelas menunjukkan bagaimana elemen meluas ke tepi kanan *browser*, dan memaksa konten berikut untuk muncul di awal posisi pertama yang tersedia di bawahnya.



Gambar 3.4 Berbagai Elemen dengan Lebar Berbeda

Gambar tersebut juga menunjukkan bagaimana elemen menjaga dirinya sendiri dan hanya menggunakan ruang yang diperlukan untuk menyimpan kontennya, tanpa memaksa konten berikutnya muncul di bawahnya. Misalnya, di dua contoh gambar di bawah, kita juga dapat melihat bahwa ketika elemen membungkus tepi layar, elemen tersebut mempertahankan bentuk persegi panjang, sedangkan elemen hanya mengikuti alur teks (atau konten lainnya) mereka mengandung. Catatan: Karena tag hanya bisa berbentuk persegi panjang, tag ini lebih cocok untuk memuat objek seperti gambar, kotak, kutipan, dan sebagainya, sedangkan *tag* paling baik digunakan untuk menyimpan teks atau atribut lain yang ditempatkan satu demi satu inline lain, dan yang harus mengalir dari kiri ke kanan (atau kanan ke kiri dalam beberapa bahasa).

3.4.4 *List HTML*

(Kurniawan, 2023) Dalam HTML, terdapat 3 jenis *tag list* istilah, yaitu berurutan (*ordered list*), tidak berurutan (*unordered list*) dan definition list.

- a. *Ordered list* akan ditampilkan menggunakan huruf atau angka dan menggunakan tag “”, sedangkan tag “” akan digunakan untuk daftar itu sendiri. Ordered list mempunyai atribut *type* untuk mengganti tipe list yaitu 1, A, a, I, i. contoh : <ol type=”i”>
- b. *Unordered list* ditampilkan dengan sebuah lingkaran atau persegi dan menggunakan tag , sedangkan untuk *list* sendiri menggunakan tag . *Unordered list* mempunyai atribut *type* untuk mengganti tipe list yaitu *disc*, *circle*, *square*, *none*. Contoh : <ul type=”square”>
- c. *Definition list* menampilkan daftar formulir dengan deskripsi masing-masing formulir. Tag “<dl>” dan “<dt>” mendefinisikan deskripsi dokumen, “<dt>” mendefinisikan nama institusi, dan “<dd>” mendeskripsikan masing-masing institusi; namun, tag “” digunakan untuk mendefinisikan daftar itu sendiri.

Perhatikan contoh di bawah ini agar lebih mudah memahaminya.

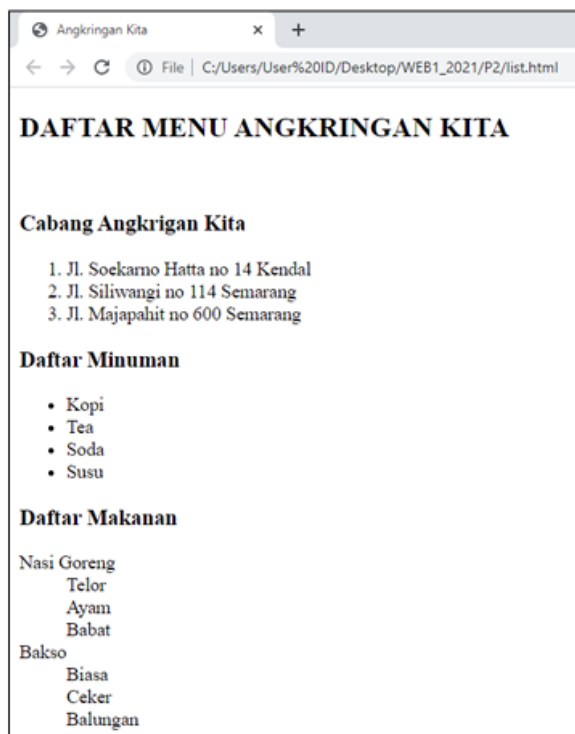
```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Angkringan Kita</title>
5 </head>
6 <body>
7   <!-- judul konten -->
8   <h2>DAFTAR MENU ANGKRINGAN KITA</h2>
9
10  <!-- enter -->
11  <br>
12
13  <!-- Contoh ordered list Number -->
14  <h3>Cabang Angkringan Kita</h3>
15  <ol>
16    <li>Jl. Soekarno Hatta no 14 Kendal</li>
17    <li>Jl. Siliwangi no 114 Semarang</li>
18    <li>Jl. Majapahit no 600 Semarang</li>
19  </ol>
20
21  <!-- contoh Unordered list Bullets -->
22  <h3>Daftar Minuman</h3>
23  <ul>
24    <li>Kopi</li>
25    <li>Tea</li>
26    <li>Soda</li>
27    <li>Susu</li>
28  </ul>
29
30  <!-- contoh Definition List -->
31  <h3>Daftar Makanan</h3>
32  <dl>
33    <dt>Nasi Goreng</dt>
34    <dd>Telor</dd>
35    <dd>Ayam</dd>
36    <dd>Babat</dd>
37    <dt>Bakso</dt>
38    <dd>Biasa</dd>
39    <dd>Ceker</dd>
40    <dd>Balungan</dd>
41  </dl>
42
43 </body>
44 </html>

```

Gambar 3.5 Contoh Sintak Membuat *List*

Hasilnya dapat dilihat pada gambar berikut ini:



Gambar 3.6 Hasil Contoh Sintak Membuat *List*

3.4.5 Menambahkan Tautan

Tautan ditambahkan dengan `<a>`, atau elemen *anchor*. Atribut `href` memberi tahu elemen *anchor* tujuan tautan. Tautan itu sendiri dapat ditambahkan ke apa saja di halaman. Misalnya, kita dapat menautkan setiap pohon yang disebutkan di bagian sebelumnya ke artikel tentang masing-masing jenis pohon tersebut. Ketika sesuatu ditautkan, browser akan memberikan umpan balik *visual* bahwa ada tautan dengan menyorot dan menggarisbawahi area yang ditautkan. Seperti elemen HTML lainnya, elemen `<a>` memiliki tag penutup `` yang sesuai yang digunakan untuk memberi tahu browser kapan harus berhenti menyorot dan menggarisbawahi tautan (Sholikhah, 2022).

3.4.6 HTML Images

Menurut (Kurniawan, 2023), *tag* `img` digunakan untuk meletakkan gambar dalam dokumen HTML dan terlihat seperti ini:

Menurut standar teknis, gambar tidak ditambahkan ke database situs web; sebaliknya, mereka hanya ditambahkan ke database situs web. Tag `img` membuat ruang kosong tempat gambar yang relevan dapat ditampilkan.

Tag `` memiliki beberapa atribut yang di gunakan :

- a. *Src* : Digunakan untuk memberi tahu *browser* tempat untuk menemukan gambar
- b. *Alt* : Memberikan informasi teks alternatif untuk gambar
- c. *Title* : Memberikan nama pada gambar pada saat *cursor* berada di wilayah gambar
- d. *Width* : Menentukan lebar gambar, ukuran gambar default (*px pixel* dan persen (%))
- e. *Height* : Menentukan tinggi gambar, ukuran gambar default (*px pixel* dan persen (%))

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Image</title>
5 </head>
6 <body>
7   <!-- gambar 1 -->
8   <img src = "gambar/logostekom.png" alt="Logo Universitas Stekom">
9
10  <br>
11
12  <!-- gambar 2 -->
13  
15 </body>
16 </html>

```

Gambar 3.7 Contoh Sintak Menyisipkan Gambar



Gambar 3.8 Hasil Contoh Sintak Menyisipkan Gambar

(Sholikhan, 2022) Gambar ditambahkan dengan elemen ``. Sama seperti elemen `<a>`, elemen `` menggunakan atribut untuk memberi tahu *browser* lebih banyak informasi tentang dirinya sendiri. Atribut `src` digunakan untuk memberi tahu *browser* di mana menemukan gambar. Sebelumnya, pada gambar dibawah atas. HTML membawa gambar tersebut ke halaman *website* dengan bentuk coding seperti ini:

```

```

Seperti yang kita lihat, elemen `` menambahkan atribut `src`, yang kemudian merujuk ke lokasi gambar di server web. Elemen `` tidak memiliki tag penutup `` karena elemen ini tidak memiliki kontennya sendiri, berbeda dengan elemen `<p>` dan `<a>` yang keduanya membutuhkan konten untuk masuk ke dalamnya dan oleh karena itu harus ditutup. Terkadang kita mungkin melihat elemen seperti `` ditutup dengan `/>` alih-alih hanya `>`, seperti pada contoh. Keduanya adalah cara yang dapat diterima dan valid untuk menutup elemen jenis ini. Elemen `` harus selalu memiliki atribut `alt`. Atribut `alt` memberi tahu seach engine dan teknologi bantu tentang gambar yang digunakan. Saat digunakan dengan elemen ``, atribut `alt` terlihat seperti ini :

```

```

3.4.7 Form

Dalam HTML, formulir adalah bagian dari konten yang berfungsi sebagai kolom *input* atau sebagai permintaan pengguna; itu akan segera dihapus atau diubah agar dapat digunakan sesuai kebutuhan. Pertimbangkan prosedur transfer data, *browsing*, hapus, penguncian data, dan yang serupa lainnya. Prosedur pengisian formulir HTML dijelaskan di sini (Kurniawan, 2023).

```
<form action="...URL.." name="..Nama Form.." method="GET|POST"> Isi Form
</form>
```

"Nama Formulir" mengacu pada nama formulir yang sekarang digunakan, "Metode" mengacu pada metode atau sarana yang digunakan untuk mentransfer data ke *server*, dan "Action" mengacu pada alamat atau file lain yang digunakan untuk menerima *input* dari form di *server*.

a) Media *Input* Dalam Form

Salah satu kegunaan pembuatan media inputan adalah untuk membantu user agar lebih mudah dalam memasukan data kedalam halaman web. Media inputan terdiri dari Kontrol berupa *text*, *radio button*, *checkbox*, *dropdown list* dan *button*.

1) Kontrol Berupa *Text*

Menggunakan dua *textarea* dan *input tag* anak, formulir HTML dapat digunakan untuk memasukkan informasi berbasis teks. *Tag input* digunakan untuk memasukkan teks yang hanya berisi satu *bar*, sedangkan *tag textarea* digunakan untuk memasukkan teks yang memiliki beberapa baris.

```
<textarea name=",,nama textarea.." id="..nama id textarea.." cols="30"
rows="10">
```

Isi *textarea*

```
</textarea>
```

Didalam *text area* terdapat beberapa atribut yaitu :

- a. *Name*, digunakan untuk memberi nama pada tag *textarea*
- b. *Id*, digunakan untuk memberi nama pada *textarea*

- c. *Cols*, digunakan untuk menentukan panjang dari *textarea*
- d. *Rows*, digunakan untuk menentukan tinggi dari *textarea*

Dengan menggunakan properti *height* dan *width* CSS, Anda juga dapat mengubah *margin* kiri dan kanan *textarea*. Meskipun dapat menampung jumlah kata yang banyak, *textarea* tidak dapat digunakan untuk semua situasi entri data. Biasanya menginginkan pengguna untuk memasukkan data satu digit dari sejumlah besar teks. Anda dapat menggunakan elemen masukan untuk jenis masukan seperti :

```
<input type="text">
```

Ketahui bahwa elemen ini akan menggunakan tipe atribut untuk menunjukkan tipe data yang akan didistribusikan. Ada banyak tipe data yang bisa diakses, dan browser akan mencocokkan tipe teks yang tersedia dengan atribut yang Anda tentukan. Mendefinisikan istilah untuk jenis atribut meliputi:

1. <i>color</i>	9. <i>time</i>
2. <i>datetime-local</i>	10. <i>datetime</i>
3. <i>number</i>	11. <i>month</i>
4. <i>tel</i>	12. <i>search</i>
5. <i>week</i>	13. <i>url</i>
6. <i>date</i>	14. <i>password</i>
7. <i>email</i>	15. <i>text</i>
8. <i>range</i>	16. <i>fil</i>

2) Kontrol Berupa *Radio Button*

Tombol radio digunakan untuk menunjukkan pilihan dengan komponen emosional yang kuat. Biasanya digunakan di situs web untuk pertanyaan komprehensif seperti yang ditanyakan saat pendaftaran *online*, pilihan ini biasanya berkaitan dengan keyakinan seseorang dalam hal agama,

politik, atau topik lainnya. Tombol radio dibuat menggunakan input elemen HTML dan atribut dengan type : `radio`

```
<input type="radio" name="jeniskelamin" value="pria"> Pria
```

```
<br>
```

```
<input type="radio" name="jeniskelamin" value="wanita"> Wanita
```

Dalam proses pembuatan tombol radio elemen, dua atribut lainnya nama dan nilai harus ditentukan. Atribut nama digunakan untuk memberi tahu browser bahwa tombol radio tertentu milik grup tombol radio tertentu, sehingga pengguna tidak dapat memilih dua opsi untuk tombol radio yang dipilih.

3) Kontrol Berupa *Checkbox*

Berbeda dengan tombol radio yang hanya memungkinkan pengguna memilih satu opsi, kotak yang menyala memungkinkan pengguna memilih satu, beberapa, atau tidak ada opsi sama sekali. Ini biasanya digunakan untuk mengumpulkan data operasional berkualitas tinggi. Anda dapat menggunakan "Centang kotak" untuk *masking* atribut. Kotak centang *input* adalah alat untuk memilih opsi atau pilihan tertentu. Jika Anda mengisinya, pengguna akan memeriksanya (mencentang), dan jika tidak, Anda hanya akan mendapatkan peringatan sederhana. Mirip dengan tombol radio, elemen ini dibuat menggunakan elemen *input* yang jenis atributnya menentukan kotak centang.

```
<input type="checkbox" name="jurusan" value="sistemkomputer">
```

Sistem Komputer

```
<br>
```

```
<input type="checkbox" name="jurusan" value="teknikkomputer">
```

Teknik Komputer

```
<br>
```

```
<input type="checkbox" name="jurusan" value="sisteminformatika">
```

Teknik Informatika

4) Kontrol Berupa *Dropdown List*

Penciptaan dua komponen menu *drop-down*, pilih dan opsi, dilakukan. Elemen *select* menyertakan semua opsi elemen yang tersedia untuk membuat menu *drop-down*. Elemen sebenarnya dari opsi adalah pengidentifikasi *dropdown* yang diinginkan. Perhatikan kode di bawah ini:

```
<select name="satuan" id="satuan">
  <option value="lembar">Lembar</option>
  <option value="botol">Botol</option>
  <option value="sachet">Sachet</option>
  <option value="box">Box</option>
  <option value="unit">Unit</option>
</select>
```

b) *Button*

Kontrol *input* bertipe *button* pada HTML mewakili tombol yang dapat diklik, yang biasanya digunakan untuk mengirimkan formulir ke alamat tujuan atau *action* dalam *form*.

1) *Submit*

Tujuan tombol ini adalah menampilkan URL yang telah disetujui untuk tindakan atribut. Pembuatan tombol *submit* juga melibatkan penggunaan elemen *input*, yang jenis atributnya diidentifikasi oleh nomor *submit*, seperti yang terlihat di bawah ini:

```
<input type="submit" name="submit" value="Tambah">
```

2) *Reset*

Tombol ini bertujuan untuk mengirim *form* ke tahap awal (mengosongkan nilai semua elemen yang ada pada *form*). Mirip dengan kirim tombol, pembuatan tombol dilakukan dengan kondisi yang sama, tetapi dengan nilai tipe atribut yang berbeda yang ditunjukkan oleh tombol *reset* pada tombol saat ini.

```
<input type="reset" name="reset" value="Batal">
```

c) Organisasi Elemen Form

HTML juga menyediakan elemen yang dapat digunakan sebagai penanda elemen yang digunakan dalam formulir untuk memberikan informasi kepada pengguna. *Label*, *fieldset*, dan *legend* adalah satu-satunya elemen yang dapat digunakan untuk tujuan ini.

1) *Label*

Label elemen menawarkan kemampuan untuk melampirkan teks ke label dengan menggunakan formulir elemen. Unsur dalam kalimat ini ditentukan oleh deskripsi yang dimaksudkan dari atribut kalimat itu. Atribut untuk label elemen memiliki nilai numerik yang sama dengan id atribut untuk formulir pengiriman elemen. Nilai serupa akan menyertakan elemen input dan label yang disorot, memungkinkan pengguna mengklik teks yang telah disediakan oleh elemen label untuk menghasilkan nilai.

```
<label for="username">Username</label>
<input type="text" name="username" id="username">
```

2) *Fieldset*

Elemen *fieldset* adalah alat yang digunakan untuk mengelompokkan sejumlah elemen yang berhubungan dengan bentuk untuk menunjukkan bahwa masing-masing elemen ini adalah bagian dari satu grup atau jika tidak terhubung.

```
<fieldset>
<label for="username">Username</label>
<input type="text" name="username" id="username">
<label for="password">Password</label>
<input type="text" name="password" id="password">
</fieldset>
```

3) *Legend*

Legenda elemen digunakan untuk memberikan *fieldset* dengan judul. Penggunaan elemen legenda sangat mudah dan hanya perlu mengacu pada

elemen sebagai anak pertama dari fieldset, seperti yang terlihat di bawah ini:

```
<fieldset>
<legend>Login</legend>
<label for="username">Username</label>
<input type="text" name="username" id="username">
<label for="password">Password</label>
<input type="text" name="password" id="password">
</fieldset>
```

d) Validasi Masukan *Form*

Validasi formulir digunakan untuk memeriksa keakuratan data yang dimasukkan ke dalam formulir. Misalnya, jika kami ingin memvalidasi bahwa bidang email dalam formulir memerlukan alamat *email*, kami memerlukan pengguna untuk memasukkan alamat *email* dalam format yang valid. Jika pengguna memasukkan alamat *email* dalam format yang tidak *valid*, pesan kesalahan akan muncul di lapangan dan data tidak dapat diproses. Validasi formulir memastikan bahwa *browser* dapat mengeksekusi data yang dimasukkan pengguna dalam format yang konsisten dengan yang ditentukan dalam standar HTML.

1) Atribut *required*

Atribut ini digunakan untuk mengidentifikasi bahwa *form* harus diisi. Jika pengguna tidak mengisi *form* yang sudah ada atribut ini maka akan tampil pesan peringatan.

```
<form action="" method="POST">
<input type="text" name="username" required>
<input type="button" name="username" value="Username">
</form>
```

2) Atribut *maxlength*

Atribut ini digunakan untuk mengatur berapa jumlah karakter yang dapat di *input form*. Penggunaannya kita harus menambahkan atribut *maxlength* dan menentukan value yang merupakan jumlah *max* karakter inputan.

```
<form action="" method="POST">
<input type="text" name="username" maxlength="6">
<input type="button" name="username" value="Username">
</form>
```

3) Atribut *type* : *|number|url|email|*

Penggunaan atribut *type* mempunyai fungsi bermacam-macam, jika kita menginginkan *form* hanya bisa diisi dengan angka saja maka gunakan *number* pada *typenya*. Jika kita menginginkan yang diisikan tipe url atau *email*, ganti dengan url atau *email*. Sebagai contoh perhatikan sintaks di bawah ini:

```
<form action="" method="POST">
<input type="number" name="username">
<input type="button" name="username" value="Username">
</form>
```

3.5 CSS

Menurut (Sholikhan, 2022) salah satu hal yang paling mendasar tentang properti CSS adalah bahwa mereka mengalir, itulah sebabnya mereka disebut *Cascading Style Sheets*. Tapi apa artinya ini? *Cascading* adalah metode yang digunakan untuk menyelesaikan potensi konflik antara berbagai jenis style sheet yang didukung *browser*, dan menerapkannya dalam urutan prioritas menurut siapa yang membuatnya, metode yang digunakan untuk membuat gaya, dan jenis properti yang dipilih. Ada tiga jenis utama style sheet yang didukung oleh semua browser modern. Dalam urutan prioritas dari tinggi ke rendah, mereka adalah:

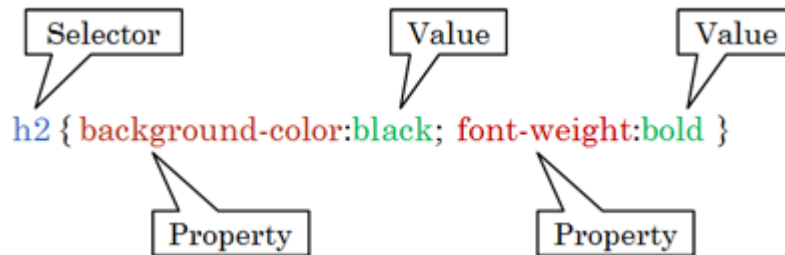
1. dibuat oleh penulis dokumen
2. dibuat oleh pengguna

3. dibuat oleh *browser*

Ketiga set *style sheet* ini diproses dalam urutan terbalik. Pertama, default di *browser* web diterapkan ke dokumen. Tanpa default ini, halaman web yang tidak menggunakan *style sheet* akan terlihat buruk. Mereka termasuk wajah font, ukuran, dan warna; jarak elemen; batas dan spasi tabel; dan semua standar wajar lainnya yang diharapkan pengguna. Selanjutnya, jika pengguna telah membuat gaya apa pun untuk digunakan alih-alih gaya standar, gaya ini diterapkan, menggantikan gaya *default browser* apa pun yang mungkin bertentangan. Terakhir, gaya apa pun yang dibuat oleh penulis dokumen saat ini kemudian diterapkan, menggantikan gaya apa pun yang telah dibuat baik sebagai *default browser* atau oleh pengguna.

Menurut (Kurniawan, 2023) CSS singkatan dari "*Cascading Style Sheets*". Sesuai dengan namanya, CSS memiliki definisi "bahasa *style sheet*" yang mengacu pada bahasa pemrograman yang digunakan dalam desain web. CSS adalah satu-satunya bahasa pemrograman yang paling banyak digunakan untuk membuat situs *web header*. Saat mendesain *header* situs web, CSS menggunakan id dan penanda yang dapat dikenali oleh pengguna secara sekilas. Dalam bidang pemrograman dan teknologi yang berkembang pesat, CSS tidak lagi hanya digunakan bersamaan dengan HTML dan XHTML, namun masih mampu diterapkan pada pengembangan aplikasi Android. CSS dapat digunakan menggunakan *id*, *class*, atau *tag*, yang biasanya disebut sebagai "pemilik". *Font*, ukuran *font*, gaya *font*, dan format *font* semuanya dapat diubah dengan CSS. Buat tata letak yang sempit, tinggi, dan ringan, buat situs web yang responsif, ubah bidang formulir, dan lakukan beberapa tugas lain dengan CSS tetapi saya tidak akan membahasnya secara mendalam dalam pelajaran mempelajari prinsip-prinsip CSS ini. *Margin* dapat digunakan untuk mengubah ukuran *font*, warna dapat diubah untuk mengubah warna huruf, dan definisi *font* dapat diubah untuk mengubah jenis huruf. Menggunakan "latar belakang" dapat digunakan untuk memindahkan teks atau gambar lebih jauh dari penampil. Gunakan "ukuran *font*" untuk mengubah ukuran *font*. Jenis *font* yang menggunakan "*font-family*" dan beberapa gaya *font* lainnya.

a) Selector

Gambar 3.9 *Selector* CSS

Karena CSS digunakan untuk mengelola atau bahkan memodifikasi konten HTML, diperlukan metode untuk "gaiting" atau menghubungkan kode CSS dengan *tag* HTML yang sesuai. Dalam CSS, inilah yang dimaksud dengan *Selector*. Sesuai dengan namanya, sebuah pemilih dapat digunakan untuk mencari halaman online yang perlu diubah ukurannya atau ditata. Misalnya, "cari seluruh *tag* <p>", "cari seluruh *tag* HTML yang memiliki atribut *class*="warning", atau "cari seluruh *link* yang ada di dalam *tag* <p>".

CSS paling sederhana untuk digunakan adalah *tag* dari dokumen HTML standar, seperti *tag* <p>, <h1>, , dan lain-lain. Pemilih CSS dapat menjadi objek yang kompleks tergantung pada kebutuhan pengguna.

1) *Universal Selector*

Hanya satu pemilih universal tanda "*" yang tersedia di CSS. Selektor ini dimaksudkan untuk "mencari" semua tag yang tersedia. CSS Pemilih Universal Contoh:

```
* {
    Color: blue;
    Background-color: white; }
```

Kode CSS yang disediakan di sini efektif untuk membuat setiap *tag* HTML berwarna coklat dan tampilan latar menjadi hijau. Pada *selector*

universal ini biasanya juga digunakan untuk mereset settingan *default* yang ada pada *web browser*.

2) *Element Type Selector*

Sintaks untuk pemilihan yang pada dasarnya adalah *tag* untuk HTML itu sendiri adalah Pemilih Jenis Elemen, juga dikenal sebagai Pemilih *Tag*. Setiap *tag* HTML dapat digunakan sebagai pemilih, dan pilihan khusus ini akan mencakup semua tag yang ditemuinya. Menggunakan Pemilih Tipe Elemen Contoh CSS:

```
h1 {
    text-decoration: underline;
}

p {
    font-size: 14px;
}
```

Kode CSS pada contoh ini akan membuat semua *tag* dengan *tag* h1 bergerak ke kiri dan semua tag dengan *tag* p memiliki lebar 14 piksel. Pengaruh pemilih jenis elemen adalah dari *tag* pertama hingga *tag* terakhir. Jika ada *tag* I di dalam tag "<p>", maka *tag* tersebut juga akan memiliki lebar 14 piksel hingga diikuti oleh *tag* penutup "</p>".

3) *Class Selector*

Class Selector adalah satu-satunya pemilih yang paling populer dan sering digunakan. Pemilih Kelas akan "mencari" setiap *tag* yang memiliki atribut kelas dengan nilai yang sesuai. Untuk menggunakan Pemilih Kelas, Anda harus memiliki *tag* HTML dengan atribut kelas. Contohnya:

```
<p class="paragraph_pertama"> Ini adalah sebuah paragraph pertama</p>
<h1 class="judul">Judul Artikel</h1>
```

```
<h2 class="penting">Sub Judul Artikel</h2>
```

Ingatlah bahwa kita menambahkan nama kelas itu sendiri sebagai kelas atribut untuk setiap *tag* pada daftar di atas. Nama kelas dapat diberikan oleh lebih dari satu *tag*, dan sebuah *tag* dapat mencakup lebih dari satu kelas. Misalnya pada paragraf terakhir dari contoh di atas, *tag* h2 memiliki atribut `class="judul penting berwarna"`. *Tag* ini terdiri dari tiga kelas: *judul*, *penting*, dan kelas *berwarna*. Sebaliknya, kode Pemilih Kelas CSS adalah sebagai berikut:

```
.paragraf_pertama {
    color: red;
}

.judul {
    font-size:20px;
}

.penting{
    color:red;
    font-size:1em;
}
```

Di CSS, pemilih kelas diterapkan dengan memberi tanda centang di depan nama kelas. Sebagai contoh kita, setiap kelas yang memiliki nilai "paragraf pertama" akan menyebabkan teks menjadi merah. Dan setiap judul kelas akan memiliki jenis huruf 20 piksel.

4) *ID Selector*

Selector yang paling dasar dan sering digunakan adalah *ID Selector*, yang bekerja berdampingan dengan *class Selector* (walaupun tidak sesering *class selector*). Penggunaan pemilih ID hampir identik dengan penggunaan pemilih kelas, kecuali bahwa dengan pemilih kelas, atribut seperti kelas digunakan untuk mengidentifikasi tag HTML, dengan

pemilih ID, atribut seperti id digunakan. Contoh penggunaan ID atribut dalam tag HTML:

```
<p id="paragraph_pembuka"> Ini adalah sebuah paragraph
pembuka</p>
```

```
<h1 id="judul_utama">Judul Artikel</h1>
```

```
<h2 id="sub_judul">Sub Judul Artikel</h2>
```

Selain digunakan sebagai pemilih CSS, atribut id juga berfungsi sebagai *tag* unik untuk masing-masing *tag* (terutama dipakai untuk kode JavaScript). Karena itu, setiap ID yang digunakan harus unik dan tidak boleh sama dengan yang lain. Dengan kata lain, sebuah ID hanya boleh digunakan satu kali di dalam satu halaman web dan tidak boleh sama. Contoh penggunaan kode pemilih id pemilih kelas CSS adalah sebagai berikut:

```
#paragraf_pembuka {
    color:red;
}

#judul_utama {
    font-size:20px;
}
```

Di CSS, menggunakan tanda hash "#" untuk menunjukkan bahwa sedang mencari *tag* dengan id tertentu.

5) *Attribute Selector*

Dibandingkan dengan selektor-selektor sebelumnya, selektor ini sedikit lebih rumit. Pemilih atribut ini digunakan untuk menemukan setiap tag dengan atribut yang telah diformat dengan benar. Konteks penggunaan pemilih atribut CSS adalah sebagai berikut:

```
[href] {
    font-size:20px;
```

```

    }
    [type="submit"] (
        width:30px;
    }

```

Seperti yang terlihat dari contoh di atas, setiap atribut yang digunakan harus ditempatkan di antara tanda kurung "[" dan "]". [href] akan berfungsi dengan semua *tag* yang memiliki atribut bernama "href", terlepas dari nilainya (yang biasanya ditemukan pada *tag* "<a>"). Misalnya, [type=" kirim "] Ini akan berfungsi dengan *tag* yang memiliki atribut dengan nilai kirim, yang dalam hal ini akan berfokus pada tombol kirim formulir. Meskipun memiliki kemampuan untuk mengidentifikasi *tag* yang cukup spesifik, atribut ini sering digunakan.

b) *Property*

Properti CSS adalah elemen atau gaya yang akan dipisahkan dari *tag* HTML. CSS memiliki beberapa properti yang dapat diterapkan untuk menghasilkan hasil akhir yang kita inginkan. Hampir setiap properti CSS dapat diterapkan ke semua pengguna. Jika pemilih digunakan, misalnya, untuk "menemukan setiap *tag* <p>", maka propertinya adalah "apa pun yang ingin Anda ekstrak dari *tag* <p> itu", seperti ukuran teks, warna, dan jenis font.

c) *Value*

Nilai CSS adalah ukuran properti. Misalnya, nilai atau ciri properti *backgroundcolor*, yang digunakan untuk mengubah warna latar belakang suatu pilihan, dapat berupa kombinasi merah, biru, hitam, atau putih.

3.5.1 Mengimpor *Style Sheet*

Menurut (Sholikhah, 2022) saat kita ingin menata seluruh situs, daripada satu halaman, cara yang lebih baik untuk mengelola *style sheet* adalah dengan memindahkannya sepenuhnya dari halaman web ke file yang terpisah, lalu mengimpor yang kita butuhkan. Ini memungkinkan menerapkan *style sheet* yang

berbeda untuk *layout* yang berbeda (seperti web dan cetak), tanpa mengubah HTML. Ada beberapa cara berbeda untuk mencapai ini, bagian pertama adalah dengan menggunakan arahan CSS `@import` seperti ini:

```
<style>
@import url('styles.css');
</style>
```

Pernyataan ini memberitahu browser untuk mengambil *style sheet* dengan nama `styles.css`. Perintah `@import` cukup fleksibel karena kita dapat membuat *style sheet* yang menarik daripada *style sheet* lain, dan seterusnya. Kita hanya perlu memastikan bahwa tidak ada *tag* `<style>` atau `</style>` di salah satu *style sheet* eksternal Anda, atau mereka tidak akan berfungsi. Sampai sini kita telah melihat bagaimana gaya sebaris menambahkan informasi gaya ke setiap elemen satu per satu, ini bisa menjadi membosankan. Dilanjutkan dengan cara menggunakan *style sheet* internal untuk membuat gaya informasi untuk halaman secara keseluruhan. Browser membaca *style sheet* eksternal ini seperti halnya membaca gaya yang diterapkan di dalam halaman itu sendiri, dan menerapkan gaya tersebut sesuai dengan itu. Kita menambahkan atau menyertakan *style sheet* eksternal dengan elemen `<link>`, yang berada di area `<head>` sebuah halaman HTML. Elemen `<link>` khas untuk menambahkan CSS terlihat seperti ini:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

Baris itu menyertakan file bernama `style.css` di direktori saat ini dan memasukkannya ke dalam halaman. Semua informasi `<style>` dan gaya sebaris dapat dihapus menggantikan satu baris di bagian `<head>` halaman. Di dalam *style sheet* eksternal terdapat aturan yang harus diterapkan dan hanya aturan yang harus diterapkan. Kita tidak perlu menyertakan atribut gaya atau bahkan elemen pembuka atau penutup `<style>` dalam *style sheet* eksternal.

```
div {
  font-weight: bold;
}
span {
```

```
font-style: italic;
}
```

Sekarang *style sheet* eksternal dapat dibagikan diantara beberapa file HTML. Jika kita perlu membuat perubahan gaya, kita hanya perlu mengedit satu file CSS, dan itu secara otomatis menerapkan gaya ke semua halaman yang menggunakan file CSS itu. Seperti yang kita lihat, file CSS eksternal membuat pemeliharaan situs web menjadi lebih mudah. *Style sheet* eksternal adalah metode yang disarankan untuk menggunakan CSS, dan dengan hanya beberapa pengecualian, sisa buku ini menggunakan CSS yang disertakan dari *style sheet* eksternal.

3.5.2 Memulai CSS

Menurut (Kurniawan, 2023) untuk memulai CSS yaitu :

a) Mengubah Background dengan CSS

Background atau batas bawah *header* website hanyalah dua elemen yang harus didesain agar menghasilkan *header* website yang terlihat tajam dan kekinian. Ada berbagai metode untuk mengubah latar belakang situs web atau tajuknya. Memanfaatkan gambar sebagai latar belakang situs web dimungkinkan. Jika menggunakan gambar bukan yang Anda inginkan, Anda juga dapat mengubah warna tampilan latar menggunakan CSS jika Anda tidak ingin menggunakannya. Gunakan properti latar belakang CSS untuk mengubah warna latar belakang. Selain digunakan untuk mengubah *backdrop body* website, properti *background* juga dapat digunakan untuk mengubah gambar *background* dari elemen HTML tertentu. Dalam pelajaran CSS, metode untuk menghapus latar belakang menggunakan CSS akan dibahas. cara menghilangkan background dengan css. Lanjutkan proses memodifikasi CSS untuk header. Buat file HTML baru dengan nama unik Anda sendiri. Di lokasi ini saya membuatnya dengan nama *index3.html*. lalu buat *file* CSS dengan nama yang sama seperti Anda, tetapi saya sudah memberinya nama *style3.css* dalam contoh ini.

1. Mengubah warna background dengan CSS

Perhatikan informasi di bawah ini tentang cara mengubah warna latar belakang menggunakan CSS.

index3.html

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Merubah warna background dengan CSS</title>
5     <link rel="stylesheet" type="text/css" href="style3.css">
6   </head>
7
8   <body>
9     <h1>Tutorial mengubah warna background dengan CSS</h1>
10  </body>
11 </html>

```

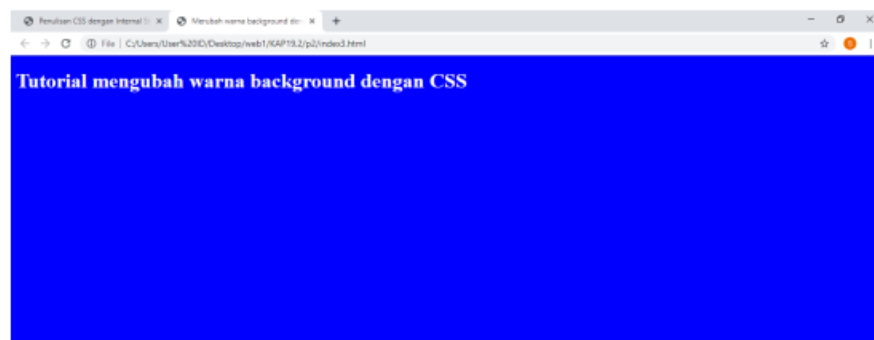
style3.css

```

1 body{
2   background: blue;
3   color: white;
4 }

```

Hasilnya :



Gambar 3.10 Mengubah Warna Background

Pada kode CSS diatas, kita menentukan *body* yang akan dimodifikasi. Kami kemudian menentukan warna latar belakang menjadi biru (biru), dan warna *font* menjadi putih (putih).

2. Mengubah gambar *background* dengan CSS

Untuk melakukannya, Anda harus menyediakan *file* grafik untuk digunakan sebagai gambar latar situs web. Dalam hal ini, saya memberikan gambar menghadap ke belakang yang akan saya gunakan sebagai foto sepia. Untuk mencegah hal ini terjadi, tempatkan *file* gambar dalam satu folder dengan file html dan CSS. Nama *file* gambarnya adalah stekom.png.



Gambar 3.11 File HTML Gambar dan CSS

index4.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Membuat background dengan gambar</title>
5   <link rel="stylesheet" type="text/css" href="style4.css">
6 </head>
7
8 <body>
9   <h1>Membuat background gambar dengan CSS</h1>
10 </body>
11 </html>

```

style4.css

```

1 body{
2   background: url('stekom.png');
3   background-size: 300px;
4   color: #fff;
5 }
6 h1{
7   color: red;
8   background-color: white;
9 }

```

Perhatikan sintaks CSS di atas. Gunakan sintaks berikut untuk membuat gambar latar:

background: url('stekom.png');

Hasilnya :



Gambar 3.12 Mengubah Background dengan Gambar

b) *Hyperlink* dengan CSS

Hyperlink dibuat untuk mengubah halaman target saat diklik. Dalam HTML, *hyperlink* atau kalimat dibuat dengan *tag* "a" dan diakhiri dengan *tag* "/a". Dan menggunakan CSS untuk membuat, memodifikasi, atau membuat *hyperlink* cukup sederhana. Dalam pembelajaran CSS ini, prosedur pembuatan *hyperlink* yang disempurnakan dengan CSS akan dibahas. Lanjutkan dengan menambahkan peringatan pada tautan, mengubah bentuk tautan menjadi tombol, membuat teks *hyperlink* menjadi kuning saat diklik, dan seterusnya. Ada dua status yang dikontrol oleh HTML dan dapat diubah menggunakan CSS. Yaitu:

1. *Link* : Ini adalah koneksi default yang aktif.
2. *Visited* : ini adalah status link yang sudah dibuka.
3. *Hover* : adalah status tautan saat penunjuk *mouse* digerakkan di atasnya.
4. *Active* : status tautan atau *hyperlink* saat diklik.

Sebaliknya, status penggunaan atau pemeliharaan tautan dapat dinyatakan menggunakan sintaks berikut:

1. a:link = untuk link biasa
2. a:visited = mewakili status tautan yang baru saja diklik.

3. a:hover = Saat mouse digerakkan di atas tautan, status tautan itu ditunjukkan oleh atribut.
4. a:aktif = Status tautan atau hyperlink saat diklik ditunjukkan dengan atribut.

index.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Hyperlink CSS</title>
5   <link rel="stylesheet" type="text/css" href="style.css">
6 </head>
7 <body>
8   <a class="link" target="_blank" href="https://stekom.ac.id/">Klik di sini</a>
9 </body>
10 </html>

```

style.css

```

1 .link{
2   font-size: 20pt;
3 }
4 .link:hover{
5   color: red;
6 }
7

```

Hasil ketika tidak ditunjuk kursor



Gambar 3.13 *Hyperlink* dengan CSS

Hasil ketika ditunjuk kursor



Gambar 3.14 *Hyperlink* dengan CSS ketika Hover

3.5.3 *Comment* CSS

Menurut (Sholikhah, 2022) di dalam aturan CSS yang ditampilkan di dekatnya, ada komentar: */* CSS Goes Here */*. Sama seperti di HTML di mana kita dapat menggunakan komentar untuk membantu menjelaskan bagian kode tertentu, demikian juga kita dapat menggunakan komentar di CSS untuk membantu menjelaskan CSS. Seperti komentar HTML, komentar dalam CSS tidak terlihat di *output* halaman tetapi, juga seperti komentar HTML, komentar CSS dapat dilihat dengan melihat sumber dari dokumen HTML atau CSS itu sendiri. Ini berarti pengunjung juga dapat melihat *comment*. Komentar dalam CSS dibuka dengan */** dan ditutup dengan **/*. Segala sesuatu yang muncul di antara */** dan **/* diperlakukan sebagai komentar. Seperti id, kelas diterapkan terlebih dahulu ke elemen HTML dengan atribut. Atributnya adalah kelas dengan judul yang tepat, seperti dalam contoh ini:

```
<div class="boldText">This text has a class.</div>
```

Seperti pada contoh id, kelas juga peka huruf besar/kecil. Kasing yang digunakan dalam HTML harus cocok dengan yang ada di CSS. Sedangkan *selector* ID menggunakan tanda *pound* (#) di CSS, kelas menggunakan satu titik atau titik. Dalam contoh sebelumnya, di mana kelas diberi nama *boldText* di HTML, itu akan direferensikan seperti ini di CSS:

```
.boldText {
  /* CSS Goes Here */
}
```

Dalam contoh ini, kelas *boldText* dipilih. Kelas dapat digunakan untuk memecahkan masalah yang ditemukan sebelumnya (di bagian "Menggunakan *style sheet* internal"), di mana *font* tebal diterapkan ke semua elemen `<div>` karena CSS menggunakan *selector* `div`. Kita dapat menggunakan kelas dalam HTML untuk menargetkan hanya elemen-elemen yang ingin kita targetkan. Merupakan ide yang baik untuk mengomentari aturan CSS Anda, bahkan jika kita hanya menjelaskan kelompok utama pernyataan daripada semua atau sebagian besar dari mereka. Kita

dapat melakukan ini dengan dua cara berbeda. Pertama, kita dapat menempatkan komentar di dalam sepasang tag `/* ... */`, seperti ini:

```
/* This is a CSS comment */
```

Atau kita dapat memperpanjang komentar melalui banyak baris, seperti ini:

```
/*  
A Multi  
line  
comment  
*/
```

3.6 Microsoft Visual Studio Code

Visual Code Studio adalah editor kode gratis yang dapat digunakan pada sistem operasi *desktop* berbasis Windows, Linux, dan Macintosh. Kode untuk editor ini dibuat oleh Microsoft, penyedia teknologi terdepan di dunia. Kode Visual adalah editor perangkat lunak yang tangguh, namun terkadang mengalami kegagalan fungsi saat digunakan. Dia mampu membuat dan menulis kode ringkasan dalam berbagai bahasa pemrograman. Yakni, JavaScript, TypeScript, dan Node.js. Juga kompatibel dengan bahasa lain dan lingkungan *runtime*, seperti PHP, Python, Java, and.NET, adalah Visual Code Studio. Ini terkait dengan ekonomi yang kuat dan perluasan pengetahuan yang mengancam. Anda bisa mendapatkan Visual Studio Code dari situs ini : <https://code.visualstudio.com/download>. Selain itu masih banyak lagi aplikasi text editor lainnya. Dalam praktikum pemrograman web ini nantinya bebas ingin menggunakan aplikasi *text editor* yang mana saja karena pada dasarnya perbedaan text editor hanya terletak pada fasilitas yang di milikinya. Intinya anda harus menyimpan file tersebut dengan ekstensi penyimpanan file tersebut (Kurniawan, 2023).

Visual Studio Code (VS Code) adalah editor kode sumber terbuka gratis untuk pengembangan dan *debugging* aplikasi *cloud* dan web *modern* yang tersedia gratis di Linux, OS X dan Windows. VS Code mendukung lebih dari 30 bahasa pemrograman, markup, dan database yang berbeda, beberapa di antaranya adalah

JavaScript, C#, C++, PHP, Java, HTML, R, CSS, SQL, Markdown, TypeScript, Less, Sass, JSON, XML dan Python. *Editor* secepat kilat tidak hanya memuaskan pengembang dengan *debugging* terintegrasi (termasuk Aplikasi ASP.NET 5 dan Node.js): Bahkan dukungan Git tersedia dengan menekan satu tombol. Banyak sekali pintasan keyboard menjaga produktivitas Anda di tempat kerja. Berbeda dengan Visual Studio 2015 dan pendahulunya, Visual Studio Code tidak berfungsi file proyek, tetapi dengan file dan folder. Lingkungan dapat diperkaya secara fleksibel melalui penggunaan ekstensi yang tetap independen dari sistem operasi yang mendasarinya. Kode benar-benar memenuhi syarat sebagai tambahan sempurna untuk alat pengembangan pilihan. Berkat fungsionalitas pembaruan terintegrasi, Visual Studio Code selalu terkini. Bulanan siklus pembaruan mencerminkan kemajuan dinamis yang didorong secara paralel untuk semua dukungan platform. Dan untuk Visual Studio Code Insiders, bahkan tersedia edisi pratinjau fitur itu dapat hidup berdampingan dengan aman dan terjamin dengan build reguler di mesin yang sama (Kahlert & Giza, 2016)..

3.7 PHP

PHP adalah bahasa *script* di sisi *server* yang didesain secara khusus untuk jaringan. Dalam halaman HTML, bisa dimasukan kode PHP yang akan dieksekusi tiap kali halaman di kunjungi. Kode PHP akan di terjemahkan di jaringan *server* dan *generate* HTML atau *output* lain yang dilihat oleh pengunjung. Secara khusus, PHP di rancang untuk membentuk web dinamis. Artinya ia dapat membentuk suatu tampilan berdasarkan permintaan terkini. Misalnya, anda dapat menampilkan isi *database* ke halaman Web. Pada prinsipnya, PHP mempunyai fungsi yang sama dengan skrip-skrip seperti ASP (*Active Server Page*), Cold Fusion ataupun Perl. Kelahiran PHP bermula saat Ramus Lerdorf membuat sejumlah skrip Perl yang dapat mengamati siapa saja yang melihat-lihat daftar riwayat hidupnya, pada tahun 1994. Skrip-skrip ini selanjutnya dikemas menjadi tool yang disebut "*Personal Home Page*". Paket inilah yang menjadi cikal-bakal PHP. Pada tahun 1995, Rasmus menciptakan PHP/FI Versi 2. Pada versi inilah pemrograman dapat

menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks sambil jalan. Pada saat ini, PHP cukup populer sebagai piranti pemrograman Web, terutama di lingkungan Linux. Walaupun demikian, PHP sebenarnya juga dapat berfungsi pada *server-server* yang berbasis UNIX, Windows NT, dan Macintosh. Bahkan versi untuk Windows 95/98 juga tersedia. Menurut dokumen resmi PHP, PHP singkatan dari PHP *Hypertext Preprocessor*. PHP merupakan bahasa berbentuk skrip yang di tempatkan dalam server dan diproses di *server*. Hasilnya yang di kirimkan ke klien, tempat pemakai menggunakan *browser*. Pada awalnya, PHP di rancang untuk diintegrasikan dengan *Web server* Apache. Namun, belakangan ini PHP juga dapat bekerja dengan *Web server* seperti PWS (*Personal Web Server*), IIS (*Internet Information Server*), dan Xitami. Untuk mencoba PHP, anda tidak perlu menggunakan komputer berkelas *server* (Faisal, 2021).

PHP Pertama kali ditemukan pada 1995 oleh seorang *Software Developer* bernama Rasmus Lerdorf. Ide awal PHP adalah ketika itu Radmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. *Script* yang dikembangkan baru dapat melakukan dua pekerjaan, yakni merekam informasi *visitor*, dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang di milis mendiskusikan script buatan Rasmus Lerdorf, hingga akhirnya rasmus mulai membuat sebuah *tool/script*, bernama *Personal Home Page* (PHP). Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik, dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP diubah menjadi akronim berulang PHP: *Hypertext Preprocessor*. Pada Juni 2004, Zend merilis PHP 5.0. Dalam versi ini, inti dari interpreter PHP mengalami perubahan besar, versi ini juga memasukkan model pemrograman berorientasi objek ke dalam PHP untuk menjawab perkembangan bahasa pemrograman ke arah paradigma berorientasi

objek. Versi terbaru dari bahasa pemrograman PHP adalah versi 7.1 (Siswanto, 2021).

PHP atau PHP *Hypertext Preprocessor* adalah sebuah bahasa *script* berbasis *server (server-side)* yang mampu mem-*parsing* kode php dari kode web dengan ekstensi .php, sehingga menghasilkan tampilan website yang dinamis di sisi *client (browser)*. Dengan menambahkan skrip PHP, anda bisa menjadikan halaman HTML menjadi lebih *powerful*, dinamis dan bisa dipakai sebagai aplikasi lengkap, misalnya *web portal, e-learning, e-library* (Ani Oktarini Sari, Ari Abdilah, 2019).

Berikut ini adalah beberapa alasan yang membuat bahasa Pemrograman PHP yang termasuk dalam kategori bahasa yang banyak digunakan oleh *programmer* di dunia (Siswanto, 2021) :

1. Gratis, PHP bersifat *Open Source*, sehingga kita bebas menggunakan PHP tanpa harus membayar.
2. Lintas *Platform*, PHP dapat dijalankan pada sistem operasi apapun, seperti Linux, Windows, OpenBSD, FreeBSD, Mac OS, Solaris dan lain-lain.
3. Didukung berbagai *Web Server* Handal seperti Apache, IIS (Windows), Xitami, Lighttpd, OmniSecure, WebWeaver, dan lain-lain.
4. Mampu koneksi dengan berbagai *database, support* untuk aplikasi *database* seperti MySQL, Ms. SQL Server, ORACLE, PostgreSQL, Interbase, SQLite, dBase, dan lainnya.
5. Memiliki tingkat akses paling cepat dibanding pemrograman web lainnya.

Kode PHP diproses melalui pemrosesan dari sisi *server*, makanya PHP disebut skrip *server-side*. Sehingga kode PHP tidak bisa diberikan langsung ketika ada permintaan dari *client (browser)*. Kode PHP dimasukkan ke dalam kode HTML dengan cara menyelipkannya di dalam kode HTML. Untuk membedakan kode PHP dengan kode HTML, di depan kode PHP tersebut diberi *tag* pembuka dan di akhir kode PHP diberi *tag* penutup. Dengan adanya kode PHP, sebuah halaman web bisa melakukan banyak hal yang dinamis, seperti mengakses *database*, membuat gambar, membaca dan menulis *file*, dan sebagainya. Hasil pengolahan kode PHP

akan dikembalikan lagi dalam bentuk kode HTML untuk ditampilkan di *browser* (Ani Oktarini Sari, Ari Abdilah, 2019).

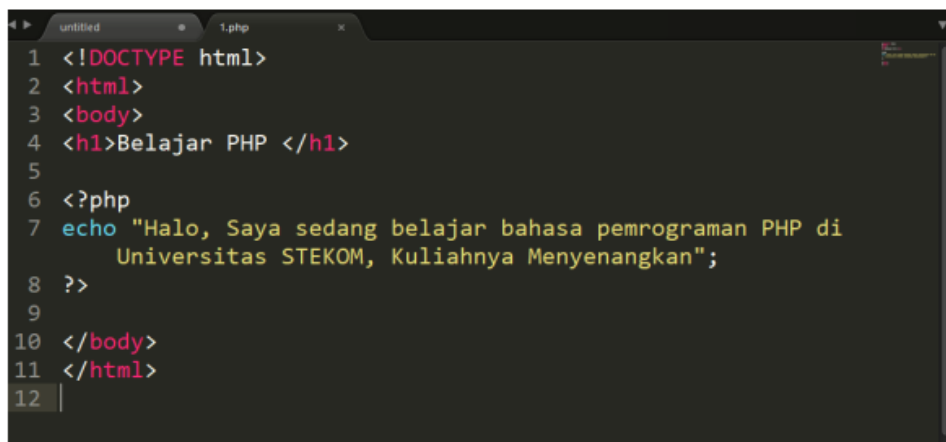
3.7.1 Aturan Penulisan Skrip PHP

Menurut (Siswanto, 2021), dalam penulisan skrip PHP harus dibuka oleh tag `<?php` dan ditutup dengan `?>`.

```
<?php
// Tuliskan Skrip PHP Disini
?>
```

Ekstensi file default untuk file PHP adalah ".php". Sebuah file PHP biasanya berisi *tag* HTML, dan beberapa kode skrip PHP. Skrip PHP memiliki kemampuan untuk mengintegrasikan format tampilan dengan *database* dalam membangun sebuah sistem. Untuk menyisipkan file HTML ke dalam skrip PHP atau sebaliknya, kita dapat menggunakan 2 cara, yaitu :

1. *Embedded Script*



```
1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h1>Belajar PHP </h1>
5
6 <?php
7 echo "Halo, Saya sedang belajar bahasa pemrograman PHP di
   Universitas STEKOM, Kuliahnya Menyenangkan";
8 ?>
9
10 </body>
11 </html>
12 |
```

2. *Non-Embedded Script*

```
<?php
echo "<html>
<body>
<h1>Halaman PHP Pertamaku</h1>
<p>Halo, Saya sedang belajar bahasa pemrograman PHP di universitas
STEKOM </p>
```

```

</body>
</html>";
?>

```

Adapun cara yang akan digunakan, silahkan pilih sesuai selera, karena baik menggunakan cara *Embedded Script* maupun *Non-Embedded Script* hasilnya sama saja.

Ada 4 jenis tag yang bisa digunakan untuk memasukkan kode PHP (Ani Oktarini Sari, Ari Abdilah, 2019)

Tabel 3.2 Jenis *Tag*

Jenis <i>Tag</i>	<i>Tag</i> Pembuka	<i>Tag</i> Penutup
<i>Tag</i> Standar	<?php	?>
<i>Tag</i> Pendek	<?	?>
<i>Tag</i> ASP	<%	%>
<i>Tag</i> Script	<script language = "php">	</script>

Contoh *Script* PHP

Buka *file* baru di *Text Editor*. Kemudian ketikkan *script* seperti di bawah ini :

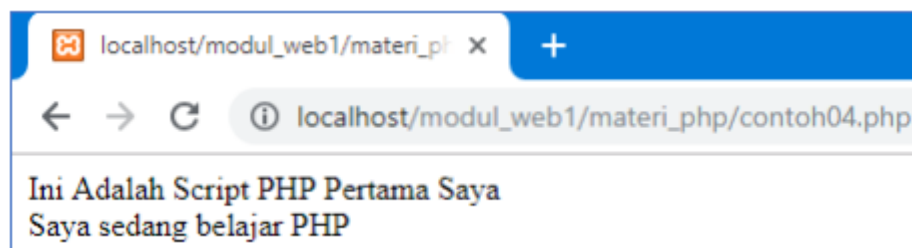
```

<?php
echo "Ini Adalah Script PHP Pertama Saya <br>";
echo "Saya sedang belajar PHP";
?>

```

Simpan *file* dengan nama *contoh04.php*

Pilih *file* *contoh04.php* maka akan tampil hasilnya :



Gambar 3.15 Hasil *contoh04.php*

Contoh04.php merupakan contoh *script* php yang berdiri sendiri tanpa ada tambahan *script* yang lain. Perintah *echo* merupakan perintah yang digunakan untuk mencetak. *Script* PHP bisa juga digabung dalam *tag* HTML.

3.7.2 Memberikan Komentar pada Skrip PHP

Menurut (Siswanto, 2021), skrip komentar pada PHP digunakan untuk memberikan keterangan yang tidak terbaca pada saat dieksekusi oleh program. Hal ini sangatlah penting bagi seorang *programmer* untuk menandai atau memberikan keterangan pada masing-masing fungsi skrip. Fungsi komentar pada skrip PHP digunakan untuk :

1. Agar orang lain memahami apa yang dilakukan pada fungsi kode program tertentu.
2. Mengingatkan pada diri sendiri tentang bagian dari program yang sudah dikerjakan.

Ada dua cara memberikan komentar pada skrip PHP, yaitu :

1. Komentar satu baris menggunakan tanda // atau #.
2. Komentar lebih dari satu baris menggunakan tanda /* ... */.
3. Berikut adalah contoh latihan menggunakan komentar pada skrip PHP

```
<?php
/*
Berikut adalah Program untuk menampilkan
Tanggal saat ini menggunakan
Fungsi date
*/
// Membuat variabel berisi tanggal hari ini
$tanggal = date("d-M-Y");
# Menampilkan tanggal di browser menggunakan fungsi echo
echo "Tanggal hari ini : $tanggal";
```

?>

3.7.3 Dasar-Dasar PHP

Dasar-dasar PHP menurut (Ani Oktarini Sari, Ari Abdilah, 2019) yaitu :

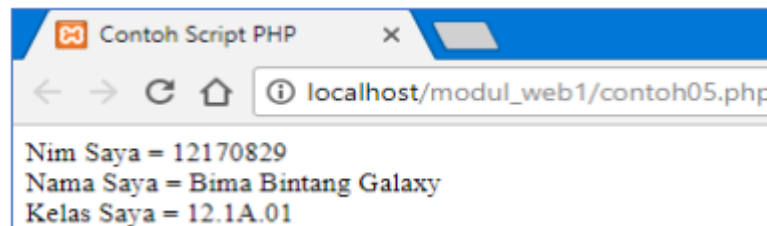
a. Variable

Variable merupakan sebuah istilah yang menyatakan sebuah tempat yang menampung nilai-nilai tertentu di mana nilai di dalamnya bisa diubah-ubah. *Variable* merupakan tempat untuk menyimpan data dalam tipe tertentu, *variable* bisa berupa null (belum ada isinya), angka, *string*, objek, *array*, *Boolean*, dan isinya bisa diubah-ubah nantinya. *Variable* penting karena tanpa adanya *variable* tidak bisa menyimpan nilai tertentu untuk diolah. *Variable* ditandai dengan adanya tanda dolar (\$) yang kemudian bisa diikuti dengan angka, huruf, dan *underscore*. Namun *variable* tidak bisa mengandung spasi. Berikut ini contoh pendefinisian *variable* : \$nama, \$no_telp, \$pekerjaan. Untuk mendefinisikan *variable*, hanya perlu menuliskannya maka otomatis *variable* dikenali oleh PHP. Contoh skrip *Variable* dalam PHP :

```
<html>
<head>
    <title> Contoh Script PHP</title>
</head>
<?php    //opening penulisan script php
    //script dibawah ini adalah contoh deklarasi variabel
    $nim ="12170829";
    $nama ="Bima Bintang Galaxy";
    $kelas ="12.1A.01";
//script dibawah ini digunakan sebagai output
    echo "Nim Saya = $nim<br>";
    echo "Nama Saya = $nama<br>";
    echo "Kelas Saya = $kelas<br>";
```

```
//script dibawah ini adalah closing penulisan script php
?>
</body>
</html>
```

Hasil :



Gambar 3.16 Hasil contoh05.php

b. Tipe Data

Berbeda dengan bahasa pemrograman lain, *variable* di PHP lebih fleksibel. Kita tidak perlu mendefinisikan jenisnya ketika mendefinisikan pertama kali. Ada 6 Tipe data dasar yang dapat diakomodasi di PHP, seperti berikut ini :

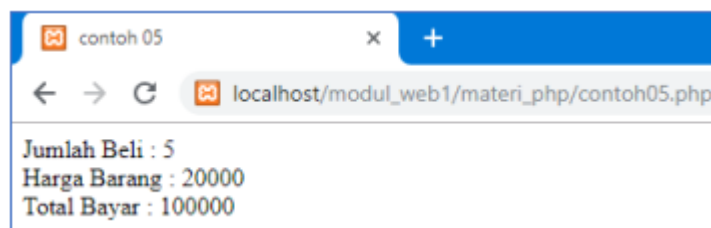
Tabel 3.3 Tipe Data

Tipe	Contoh	Penjelasan
Integer	134	Semua angka bukan pecahan
Double	5.1234	Kumpulan karakter
String	“asep”	Salah satu nilai True atau False
Boolean	False	Sebuah instance dari class
Object		Larik
Array		

Contoh05.php

```
<html>
<head>
    <title> contoh 05 </title>
</head>
<body>
<?php
    $jumlah = 5;
    $harga = 20000;
    $total=$harga*$jumlah;
    echo "Jumlah Beli : $jumlah <br>";
    echo "Harga Barang : $harga <br>";
    echo "Total Bayar : $total <br>";
?>
</body>
</html>
```

Hasil tampilan di *browser* :



Gambar 3.17 Hasil dari contoh05.php

c. Konstanta

Selain *variable*, sebuah program umumnya juga memungkinkan adanya konstanta. Konstanta fungsinya sama seperti *variable* namun nilainya statis/konstan dan tidak bisa berubah. Cara untuk mendefinisikan konstanta adalah :

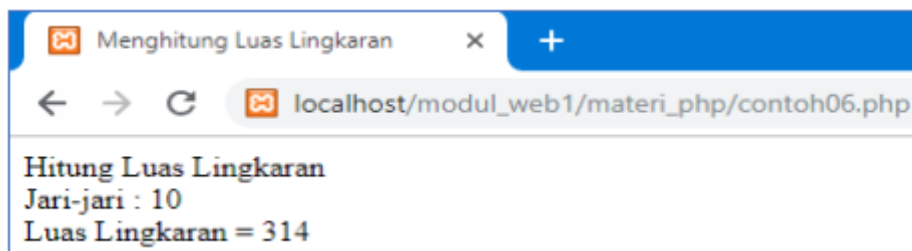
Define ("NAMA_KONSTANTA", nilai_konstanta);

Setelah didefinisikan, kita dapat langsung menggunakannya dengan mengetikkan nama konstanta tersebut. Nama konstanta umumnya diketik menggunakan huruf besar.

Contoh penggunaan Konstanta dan Komentar : contoh06.php

```
<html>
<head>
  <title> Menghitung Luas
  Lingkaran</title>
</head>
<body>
<?php
//konstanta untuk nilai judul
define("Judul","Hitung Luas
Lingkaran");
//konstanta untuk nilai phi
define("PHI",3.14);
echo "<br> Jari-jari : $r <br>";
$luas = PHI*$r*$r;
echo "Luas Lingkaran = $luas";
?>
</body>
</html>
```

Hasil tampilan di *browser* :



Gambar 3.18 Hasil dari contoh06.php

3.7.4 Operator Aritmatika dengan PHP

(Siswanto, 2021) Dalam mempelajari bahasa pemrograman dasar, Kita selalu dihadapkan dengan istilah operator aritmatika. Operator Aritmatika adalah operator matematis yang terdiri dari operator penambahan, pengurangan, perkalian, pembagian, modulus, plus, dan minus. Dan pada tutorial kali ini, saya akan memberikan contoh latihan operator aritmatika dengan menggunakan bahasa pemrograman PHP.

A. Jenis Operator Aritmatika dalam PHP

Tabel 3.4 Operator Aritmatika

$\$a + \b	Penambahan	Total dari \$a dan \$b
$\$a - \b	Pengurangan	Selisih dari \$a dan \$b
$\$a * \b	Perkalian	Hasil kali dari \$a dan \$b
$\$a / \b	Pembagian	Hasil bagi dari \$a dan \$b
$\$a \% \b	Mod / Sisa hasil bagi	Sisa dari pembagian \$a dan \$b

Contoh Penggunaan Operator Aritmatika di dalam PHP

Nama file : aritmatika.php

```
<?php
```

```
$a = 35;
```

```
$b = 10;
```

```
$hasil1 = $a + $b; // Penjumlahan
```

```
$hasil2 = $a - $b; // Pengurangan
```

```
$hasil3 = $a * $b; // Perkalian
```

```
$hasil4 = $a / $b; // Pembagian
```

```
$hasil5 = $a % $b; // Modulus
```

```
echo "Nilai a : $a<br>
```

```
Nilai b : $b <br>
```

Hasil penjumlahan $\$a + \$b = \$hasil1$

Hasil pengurangan $\$a - \$b = \$hasil2$

Hasil perkalian $\$a \times \$b = \$hasil3$

Hasil pembagian $\$a / \$b = \$hasil4$

Sisa dari pembagian $\$a / \$b = \$hasil5$;

?>

Contoh Penggunaan Operator Aritmatika dengan *Form*

Nama file : form_jumlah.php

```
<form method="POST" action="hasil.php">
```

```
Nilai a : <input type="text" name="a"><br><br>
```

```
Nilai b : <input type="text" name="b"><br><br>
```

```
<input type="submit" value="Jumlahkan">
```

```
</form>
```

Nama file : hasil.php

```
<?php
```

```
// Ambil variabel dari form
```

```
 $\$a = \$_POST['a'];$ 
```

```
 $\$b = \$_POST['b'];$ 
```

```
 $\$hasil = \$a + \$b;$ 
```

```
echo "Nilai a :  $\$a$ <br>
```

```
Nilai b :  $\$b$ <br>
```

```
Hasil Penjumlahan  $\$a + \$b = \$hasil$ ";
```

```
?>
```

Dalam menggunakan operator aritmatik, kita harus berhati-hati terutama ketika menggunakan lebih dari satu operator yang berbeda dalam satu *statement* perhitungan. Sebagai contoh skrip berikut ini:

```
<?php
```

```
 $\$a = 3 + 4 * 5 - 6;$ 
```

```
echo $a;
```

```
?>
```

Apabila skrip di atas dijalankan, maka hasil yang muncul bukan 29, tapi 17. Mengapa demikian? itu dikarenakan operasi aritmatik yang dikerjakan terlebih dahulu adalah perkalian (*). Mengapa demikian? Karena perkalian memiliki tingkat presedensi yang lebih tinggi daripada + dan -. Setelah perkalian(*) dikerjakan, baru dikerjakan operasi penjumlahan(+) atau pengurangan(-). Operasi penjumlahan(+) dan pengurangan(-) memiliki tingkat presedensi yang sama. Oleh karenanya, maka yang dikerjakan lebih dahulu adalah yang terletak di bagian yang lebih kiri, yaitu penjumlahan (+). Bagaimana dengan operator pembagian (/)? Operator ini memiliki tingkat presedensi yang sama dengan perkalian(*). Keduanya memiliki tingkat presedensi yang lebih tinggi daripada penjumlahan (+) dan pengurangan (-). Sedangkan operator modulus (%) levelnya juga sama dengan perkalian (*) dan pembagian (/). Jika kita ingin yang mengerjakan penjumlahan terlebih dulu, caranya adalah dengan memberikan tanda kurung seperti contoh di bawah ini.

```
<?php
```

```
$a = (3 + 4) * 5 - 6;
```

```
echo $a;
```

```
?>
```

B. Kombinasi Operator Aritmatik dan *Assignment*

Selain bentuk operator aritmatik yang dibahas sebelumnya, ada juga operator yang merupakan kombinasi antara operator aritmatik dengan *assignment*. Dalam pemrograman seringkali dijumpai proses yang melibatkan proses *increment* (kenaikan nilai). Misalkan kita menginginkan proses *increment* dengan tingkat kenaikan 1, maka perintah yang dituliskan dapat berupa

```
$counter = $counter + 1;
```

Maksud dari perintah di atas adalah, nilai variabel `$counter` yang baru diperoleh dari nilai `$counter` yang lama ditambah 1. Dalam PHP, perintah di atas dapat ditulis dalam satu perintah singkat sebagai

```
$counter += 1;
```

Dari contoh di atas tampak bahwa operator yang digunakan (`+=`) merupakan gabungan dari operator aritmatika dan *assignment*. Berikut ini adalah bentuk-bentuk operator lain jenis ini.

Tabel 3.5 Operator Aritmatika dan *Assignment*

Operator	Contoh	Operator yang ekuivalen
<code>+=</code>	<code>\$x += 2;</code>	<code>\$x = \$x + 2;</code>
<code>-=</code>	<code>\$x -= 4;</code>	<code>\$x = \$x - 4;</code>
<code>*=</code>	<code>\$x *= 3;</code>	<code>\$x = \$x * 3;</code>
<code>/=</code>	<code>\$x /= 2;</code>	<code>\$x = \$x / 2;</code>
<code>%=</code>	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
<code>.=</code>	<code>\$var_str.="hello";</code>	<code>\$var_str = \$var_str . "hello";</code>

C. Operator *Pre/Post Increment* dan *Decrement*

Operator jenis ini merupakan pengembangan dari operator jenis sebelumnya. Operator ini hanya digunakan pada proses *increment* maupun *decrement* dengan tingkat 1.

Berikut ini adalah operator yang termasuk jenis ini :

```
$x++;
```

ekuivalen dengan `$x += 1;` atau `$x = $x + 1;`

```
$x--;
```

ekuivalen dengan `$x -= 1;` atau `$x = $x - 1;`

Contoh :

```
<?php
$x = 4;
$x++;
echo "Nilai x yang baru : ". $x;
$x = 4;
$x--;
echo "Nilai x yang baru : ". $x;
?>
```

3.7.5 Pengenalan *Form* dan Penggunaan HTTP *SERVER*

Pengenalan *Form* dan penggunaan HTTP *server* menurut (Ani Oktarini Sari, Ari Abdilah, 2019) yaitu :

1. Komponen *Form*

Sebuah website dinamis seringkali memerlukan interaksi antara *browser client* dan *server* bisa berupa pemasukan data teks, angka, atau *upload file* untuk diproses oleh *server*. Untuk mewedahi suatu data yang dikirimkan oleh *browser client*, dibutuhkan adanya *FORM* HTML. Penggunaan *form* misalnya untuk pendaftaran keanggotaan, pemasukan kode kartu kredit, login user, transaksi perbelanjaan, dan *upload file*.

2. Pengolahan Data dari *Form*

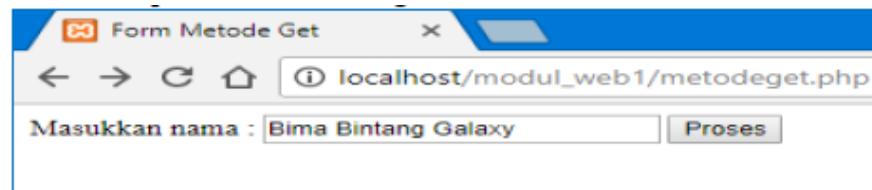
Form di HTML dikenal dengan adanya *tag* <FORM> dan ditutup dengan *tag* </FORM>. Di dalam *tag* pembuka <FORM> diikuti dengan atribut *action* dan *method*. Pada atribut *action* menerangkan ke halaman yang akan digunakan pada proses *input*, sedangkan *method* digunakan untuk mengatur cara mem-*parsing* konten. Pada halaman web akan menerima *inputan* dari user atau para pengunjung dengan menggunakan metode GET dan POST. GET akan mengirimkan data bersama dengan URL, sedangkan POST akan mengirimkannya secara terpisah. User mengirimkan data *input* dengan mengisi teks atau pilihan pada attribut *form* html.

- a. Proses Form menggunakan Metode GET.

File metodeget.php

```
<html>
<head>
    <title> Form Metode Get</title>
</head>
<body>
<form action="metodegetproses.php" method="get">
    Masukkan nama : <input type = "text" name="nama" size="25">
<input type="submit" value="Proses">
</form>
</body>
</html>
```

Hasil dari skrip di atas adalah sebagai berikut:



Gambar 3.19 Hasil metodeget.php

Buat file untuk memproses *Variable* yang diberikan oleh file metodeget.php, beri nama filenya : metodegetproses.php

```
<html>
<head>
    <title> Form Metode Get Proses</title>
</head>
<body>
Data Nama Yang Diinputkan adalah : <?php echo
$_GET["nama"];?>
</body>
</html>
```


Hasil dari skrip di atas adalah sebagai berikut:



Gambar 3.20 Hasil metodegetproses.php

Perhatikan tulisan pada url di web browser karena menggunakan metode GET, maka data yang dikirimkan akan ditampilkan di URL

b. Proses *Form* menggunakan metode POST

Untuk membuat inputan, dan beri nama file : metodepost.php

```
<html>
```

```
<head>
```

```
    <title> Form Metode Get</title>
```

```
</head>
```

```
<body>
```

```
<form action="metodepostproses.php" method="post">
```

```
Masukkan nama : <input type = "text" name="nama" size="25">
```

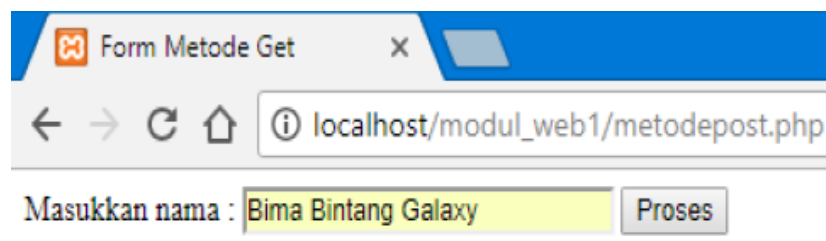
```
<input type="submit" value="Proses">
```

```
</form>
```

```
</body>
```

```
</html>
```

Hasil dari skrip di atas adalah sebagai berikut:

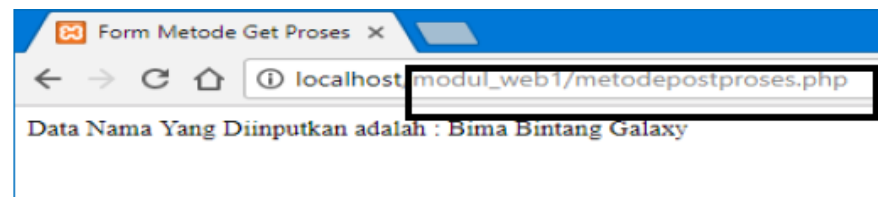


Gambar 3.21 Hasil metodepost.php

Buat file untuk memproses *Variable* yang diberikan oleh file `metodepost.php` beri nama filenya : `metodepostproses.php`

```
<html>
<head>
    <title> Form Metode Post Proses</title>
</head>
<body>
Data Nama Yang Diinputkan adalah : <?php echo $_POST["nama"];?>
</body>
</html>
```

Hasil dari skrip di atas adalah sebagai berikut:



Gambar 3.22 Hasil `metodepostproses.php`

Perhatikan tulisan pada url di web *browser* karena menggunakan metode POST, maka data yang dikirimkan akan ditampilkan di URL.

c. Contoh Pembuatan *Form* dengan *Input*, Proses, dan *Output* Dalam Satu *File*

Untuk membuat *form input* dan halaman untuk menampilkan dalam satu *file*, maka bisa menggunakan *statement* :

```
If (!Empty (nama_Variable))
```

Artinya jika *Variable* yang dicari tidak kosong (alias ada) maka baru ditampilkan, sementara jika tidak ada maka tidak akan ditampilkan.

Yang perlu diketahui adalah digunakannya alamat *action* :

```
$_server ['php_self']
```

Artinya alamat *action* akan mengacu ke halaman itu sendiri dan bukan di halaman yang lain. Dengan demikian, jika *form* di *submit*, maka halaman yang dipanggil tetap halaman yang sama

Contoh formdatadiri.php

```
<html>
<head>
    <title> Masukan Data</title>
</head>
<body>
<h1> Masukan Identitas Anda</h1>
<form action="<?php echo $_SERVER['PHP_SELF']; ?>"
method="post">
<pre>
Isikan Nama : <input type="text" name="nama">
Isikan No Telp: <input type="text" name="notelp">
Isikan Alamat : <textarea name="alamat" rows="5"
cols="40"></textarea>
<input type="submit" value="TAMPIL"><input type="reset"
value="BATALL">
</pre>
<?php
$nama = $_POST['nama'];
$stelp = $_POST['notelp'];
$alamat = $_POST['alamat'];
if(!empty($nama)){
echo "Nama : $nama <br>"; }
if(!empty($stelp)){
echo "No Telp : $stelp <br>"; }
if(!empty($alamat)){
echo "Alamat : $alamat <br>"; }
```

```
?>
</body>
</html>
```

Hasil dari skrip di atas adalah sebagai berikut:



The screenshot shows a web browser window with the title 'Masukan Data'. The address bar shows 'localhost/modul_web1/FormDataDiri.php'. The page content is titled 'Masukan Identitas Anda'. It contains three input fields: 'Isikan Nama' with the value 'Bima Bintang Galaxy', 'Isikan No Telp' with the value '08791234567', and 'Isikan Alamat' with the value 'Jalan Anggrek No. 9 Bandung'. At the bottom, there are two buttons: 'TAMPIL' and 'BATAL'.

Gambar 3.23 Hasil formdatadiri.php

Selanjutnya setelah ada hasil formdatadiri maka KLIK Tombol Tampil, maka akan tampil kiriman datanya.



The screenshot shows the same web browser window as in Gambar 3.23. The 'TAMPIL' button has been clicked, and the form data is now displayed below the input fields. The displayed data is: 'Nama : Bima Bintang Galaxy', 'No Telp : 08791234567', and 'Alamat : Jalan Anggrek No. 9 Bandung'. The 'TAMPIL' and 'BATAL' buttons are still visible at the bottom.

Gambar 3.24 Hasil Kiriman Form Data Diri

3.8 MySQL

MySQL adalah database server relasional yang gratis di bawah lisensi GNU General Public License. Dengan sifatnya yang *Open Source*, memungkinkan user juga untuk melakukan modifikasi pada source code-nya untuk memenuhi kebutuhan spesifik mereka sendiri. MySQL merupakan database server multi-user

dan multi-threaded yang tangguh (robust). Dengan memiliki banyak feature MySQL bias bersaing dengan database komersial sekalipun. Tidak mengejutkan, MySQL menjadi database pilihan untuk banyak pengguna PHP. MySQL dikembangkan oleh MySQL AB, sebuah perusahaan komersial yang membangun layanan bisnisnya melalui database MySQL. Awal mula pengembangan MySQL adalah penggunaan mSQL untuk koneksi ke table mempergunakan rutin level rendah (ISAM) setelah beberapa pengujian diperoleh kesimpulan mSQL tidak cukup cepat dan fleksibel untuk memenuhi kebutuhan. Sehingga dihasilkan suatu antarmuka SQL baru pada *database* tetapi dengan API yang mirip mSQL. API ini dipilih sedemikian rupa sehingga memudahkan porting kode. Tentang penggunaan nama MySQL sampai saat ini belum jelas betul. *Base directory* dan sejumlah *library* serta *tool* pada saat pengembangan telah diawali dengan “my” selama sekitar 10 tahun. Bagaimanapun pemberian nama MySQL sampai saat ini masih menjadi sebuah “misteri”. MySQL adalah suatu sistem manajemen *database*. Untuk menambahkan, mengakses, dan memproses data yang tersimpan pada suatu database komputer anda memerlukan sistem manajemen *database* seperti MySQL. Karena komputer sangat unggul dalam menangani sejumlah besar data, system manajemen database memainkan suatu peranan yang peting dalam komputansi, baik sebagai *utility stand-alone* maupun bagian dari aplikasi lainnya. MySQL adalah sistem manajemen *database* relasional. Suatu *database* relasional menyimpan data dalam table-table terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa tabel pada suatu permintaan. Bagian SQL dari kata MySQL berasal dari “*Structured Query Language*” bahasa paling umum yang dipergunakan untuk mengakses *database*. Konektivitas, kecepatan dan keamanannya membuat MySQL cocok untuk pengaksesan *database* pada internet. MySQL merupakan sistem *client/server* yang terdiri dari *SQL server multi heraded* yang memungkinkan *backed* yang berbeda, sejumlah program *client* dan *library* yang berbeda, *tool administrative*, dan beberapa antarmuka pemrograman. MySQL juga tersedia sebagai *library* yang bisa digabungkan keaplikasi (Faisal, 2021).

3.8.1 Perbedaan SQL dan MySQL

(Siswanto, 2021) SQL (*Structured Query Language*) adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua *server* basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. Secara umum, SQL terdiri dari dua bahasa, yaitu *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML).

3.8.2 Data Definition Language (DDL)

DDL digunakan untuk mendefinisikan, mengubah, serta menghapus basis data dan objek-objek yang diperlukan dalam basis data, misalnya tabel, *view*, *user*, dan sebagainya. Secara umum, DDL yang digunakan adalah *CREATE* untuk membuat objek baru, *USE* untuk menggunakan objek, *ALTER* untuk mengubah objek yang sudah ada, dan *DROP* untuk menghapus objek. DDL biasanya digunakan oleh administrator basis data dalam pembuatan sebuah aplikasi basis data.

3.8.3 Data Manipulation Language (DML)

DML digunakan untuk memanipulasi data yang ada dalam suatu tabel. Perintah yang umum dilakukan adalah:

1. *SELECT* untuk menampilkan data
2. *INSERT* untuk menambahkan data baru
3. *UPDATE* untuk mengubah data yang sudah ada
4. *DELETE* untuk menghapus data

Sedangkan MySQL adalah sebuah perangkat lunak sistem manajemen basis data atau DBMS (*Database Management System*) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL dapat Anda dapatkan secara gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL. Selain MySQL, Anda juga mungkin pernah mendengar perangkat lunak DBMS lainnya seperti : Microsoft SQL Server, Sybase, Interbase,

XBase, Firebird, PostgreSQL, Microsoft Access, dBase III, Paradox, FoxPro, Visual FoxPro, Visual dBase, Lotus Smart, db2, MongoDB dan lain-lain. Jadi bisa disimpulkan secara sederhana bahwa SQL adalah struktur bahasa untuk mengakses *database*, sedangkan MySQL adalah salah satu aplikasi database atau *software* untuk manajemen *database*.

3.8.4 Keunggulan MySQL

Berikut adalah beberapa keunggulan dari MySQL, antara lain :

1. Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. *Open Source*

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.

3. *Multi User*

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance Tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam Tipe Data

MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. *Perintah dan Fungsi*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

7. Keamanan

MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

8. Skalabilitas dan Pembatasan

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Konektivitas

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau *Named Pipes* (NT).

10. Lokalisasi

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

11. Interface

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

12. *Clients* dan *Tools*

MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

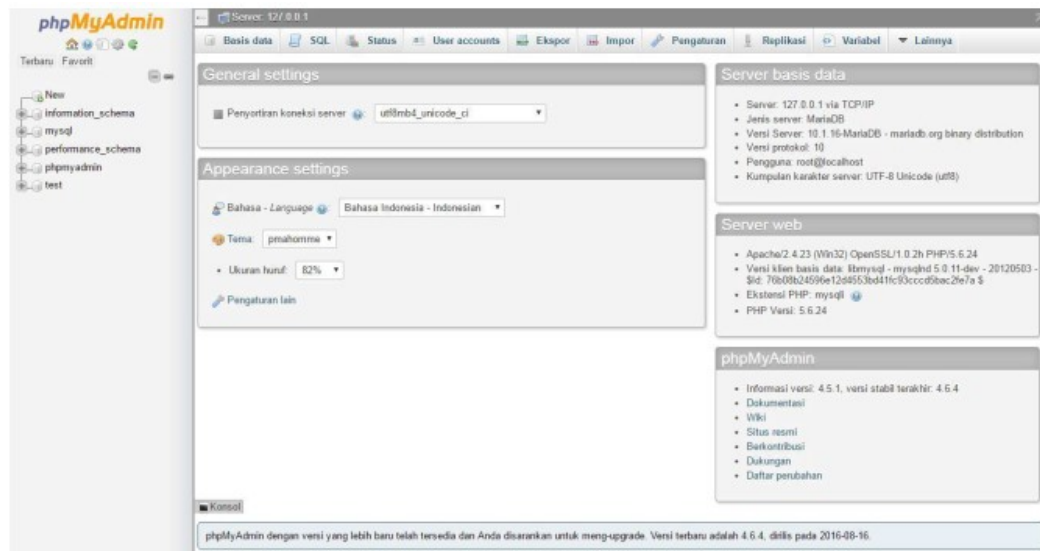
13. Struktur tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani *ALTER TABLE*, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle. Sejak diakuisisinya MySQL oleh Oracle pada September 2010, Monty Program sebagai penulis awal kode sumber MySQL memisahkan diri dari pengembangan dan membuat versi yang lebih

mandiri yakni MariaDB. MariaDB merupakan versi pengembangan terbuka dan mandiri dari MySQL.

3.8.5 Mengelola Database Melalui phpMyAdmin

PHP dapat bekerja dengan berbagai macam *software database*, seperti MS. Access, SQL Server, Oracle dan lainnya. Namun, pada kesempatan kali ini akan coba membahas *software database* bawaan XAMPP yaitu MariaDB. XAMPP saat ini tidak lagi menyertakan MySQL sebagai default instalasi *database*. MySQL digantikan oleh MariaDB. Untuk menjaga kompatibilitas, pihak pembuat tidak banyak mengganti *environment* MySQL. Jadi Anda tidak perlu khawatir dengan dengan banyaknya perubahan, karena semua connector, library dan aplikasi yang bekerja pada MySQL, dapat bekerja pada MariaDB. Pada dasarnya, mengelola basis data dengan MariaDB/MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu. Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola basis data dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat database, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya. Untuk membuka phpMyadmin, buka browser favorit Anda, lalu ketikkan alamat <http://localhost/phpmyadmin>, maka akan muncul halaman phpMyAdmin seperti berikut :



Gambar 3.25 Halaman phpMyAdmin

3.8.6 Tipe Data

Berikut adalah tipe-tipe data yang sering digunakan di dalam sebuah tabel :

1. *INT*, data berupa angka.
2. *VARCHAR*, data berupa karakter.
3. *DATE*, data berupa tanggal.
4. *TIME*, data berupa waktu.
5. *TEXT*, data berupa karakter dalam jumlah yang banyak.
6. *ENUM*, data yang sudah ditentukan isinya, misalnya Y (Yes) atau N (No) saja.

3.9 JavaScript

JavaScript merupakan suatu bahasa *script* yang banyak digunakan dalam dunia teknologi terutama internet, bahasa ini dapat bekerja di sebagian besar *web browser* seperti Internet Explorer (IE), Mozilla Firefox, Netscape, Opera dan *web browser* lainnya. bahasa JavaScript dapat dideskripsikan dalam bentuk fungsi (*Function*) yang ditaruh di bagian dalam tag <head> yang dibuka dengan tag <script language =” javascript”> Isi dari script JavaScript sama dengan konsep yang sudah dipelajari dalam materi PHP, yakni ada deklarasi *Variable*, penggunaan operator,

percabangan, *looping*, dan fungsi. Di dalam JavaScript juga sebuah komponen *Alert* yang digunakan untuk menampilkan kotak pesan pada browser ketika fungsinya di jalankan (Ani Oktarini Sari, Ari Abdilah, 2019).

Menurut (Kurniawan, 2023) tampilan JavaScript berbeda dari HTML dan CSS. CSS dan HTML masing-masing digunakan untuk mengubah format konten tabular. oleh karena itu desain keseluruhannya sangat berbeda. Ada beberapa contoh orang yang menggunakan HTML, CSS, dan JavaScript secara bersamaan, termasuk konstruksi rumah, dekorasi interior, dan lampu serta pintu-pintu. Akibatnya, HTML digambarkan sebagai sebuah blok bangunan yang dapat digunakan untuk merenungkan segala sesuatu, bahkan tempat-tempat dengan alamat tetap seperti tembok, pintu, atap, dan lain-lain. Sebaliknya, CSS adalah alat dekorasi rumah yang dapat mengubah tampilan *wallpaper*, karpet, dan hal-hal lain di rumah. Javascript berguna untuk hal-hal yang perlu responsif dan menghidupkan, seperti menyalakan lampu atau membuat pintu. JavaScript membuat situs web lebih responsif karena, meski tidak ada masalah, isinya masih akan berubah secara signifikan. Tidak adanya lampu atau pintu mungkin tidak menjadi masalah, tetapi rumah besar mungkin menjadi redup dan mulai mengeluarkan sedikit kelembapan. Juga, kehadiran JavaScript membuat situs web agak tidak menarik. Bahasa ini termasuk yang paling mudah dipelajari para segelintir. Beberapa website populer yang sekarang tersedia menggunakan bahasa Netscape. Menggunakan JavaScript memungkinkan Anda mengirim elemen ke halaman web yang diinginkan. Selain kompatibel dengan berbagai *platform* dan *browser* web, JavaScript cukup serbaguna. Penggunaan memorinya sangat kecil, ringan, dan cepat dibandingkan dengan bahasa pengkodean lainnya. Namun, JavaScript tidak seaman bahasa pemrograman lainnya.

3.9.1 Sejarah

Seperti yang telah disebutkan sebelumnya, JavaScript adalah bahasa pemrograman yang dikembangkan oleh Netscape. Proses desain membutuhkan waktu sepanjang hari. Nama JavaScript adalah Mocha, yang dibuat oleh Mona, Live Script, dan

kemudian JavaScript. Hanya komunitas Netscape yang menggunakan Javascript versi asli, September 1995, dan fungsinya telah diperbarui. Namun seiring berjalannya waktu, JavaScript terus berkembang hingga bisa berfungsi seperti sekarang. Pada tahun 1996, JavaScript sebagian besar disebut sebagai ECMAScript. Situasi ini dimulai pada tahun 1998. ECMAScript 2 dan ECMAScript 3 dibuat masing-masing pada tahun 1998 dan 1999. ECMAScript terus berkembang hingga akhirnya berevolusi menjadi JavaScript yang digunakan saat ini. Hampir 92 persen situs web pada tahun 2016 menggunakan JavaScript. Mungkin jumlahnya akan lebih tinggi dari saat ini. Bolehkah saya mau? Rupanya, selama dua tahun ke depan, JavaScript berkembang sebagai satu-satunya bahasa pemrograman yang paling populer dan penting. Jika Anda sering menggunakan internet, Anda mungkin tahu apa itu JavaScript, familiar dengan namanya, atau lebih kabur dan belum pernah melihat sintaksnya.

3.9.2 Cara Kerja dan Fungsi JavaScript

Untuk lebih memahami siapa kami sebagai sebuah tim, kami akan meninjau fungsi dan metode kerja JavaScript. HTML, CSS, PHP, dan bahasa pengkodean lainnya akan sering diperiksa oleh *server* sebelum dikirim ke pengguna. Lebih tepatnya, ketika Anda membuat situs web menggunakan HTML, CSS, PHP, dan bahasa lainnya, bahasa situs web tersebut akan diterjemahkan secara otomatis dan diperiksa oleh *server* web setiap kali pengunjung meminta akses ke sana. *Server* web akan segera mulai menghapus *cookie* yang ada, setelah itu mereka akan ditinjau dengan cermat oleh pengunjung situs web sebelum dikirim ke mereka, memungkinkan mereka untuk mendapatkan data dan informasi lainnya dari situs yang baru dikunjungi. Metode kerja ini berbeda dengan metode kerja yang digunakan oleh JavaScript. Cara kerja dari JavaScript adalah bahwa pengguna situs web akan dikirim ke file dengan ekstensi *.js* atau ke halaman yang telah disematkan secara permanen. Oleh karena itu, dimungkinkan untuk menyebutnya sebagai bahasa klien. Lain kali klien mengunjungi situs web, skrip yang disebutkan di atas akan aktif di titik akhir klien. Aksara yang sama juga digunakan pada dimakat yang

sedang digunakan. Oleh karena itu, masalahnya bukan terletak pada *server* web melainkan pada masing-masing klien atau pengunjung situs web. Namun, tersedia *browser* yang menawarkan opsi bagi pengguna untuk menentukan apakah mereka ingin mengaktifkan atau tidak. Beberapa fitur tidak akan berfungsi atau, lebih buruk lagi, tidak akan tersedia jika hal ini tidak terjadi. Dan Anda sebagai awam mungkin bisa memahami hal ini. Jika Anda sering menggunakan internet, mungkin ada situs web yang meminta Anda untuk mengaktifkan fungsi JavaScriptnya. Itulah satu-satunya hal yang disorot dalam paragraf ini. Namun, ada browser web yang menawarkan opsi aktif yang tidak ada.

3.9.3 Kelebihan JavaScript

JavaScript adalah satu-satunya bahasa pemrograman dengan banyak kekurangan. Jika dibandingkan dengan bahasa pemrograman bahasa lain, keterampilan lain lebih mudah dipelajari. Penanganan dan koreksi kesalahan juga lebih mudah. Selain itu, Anda tidak memerlukan kompiler karena *browser* web dapat mengartikannya menggunakan HTML. JavaScript juga dapat diaktifkan pada halaman web atau peristiwa yang ditargetkan, seperti selama klik atau gerakan mouse. *File* JavaScript juga dapat digunakan di berbagai *platform* dan *browser* web, menjadikannya cukup populer dan penggunaannya hampir universal. Selain itu, JavaScript dapat digunakan untuk memvalidasi informasi saat ini, membuatnya kurang diinginkan untuk memasukkan data secara *manual*. Dibandingkan dengan bahasa pemrograman lain, JavaScript lebih ringkas, lebih cepat, dan lebih efisien saat membandingkan penggunaan memori. Alasan yang disebutkan di atas hanya menjadikan JavaScript sebagai bahasa pemrograman yang kurang disukai. Setiap tindakan memiliki beberapa risiko, bahkan menggunakan JavaScript.

3.9.4 Kekurangan JavaScript

Jenis ketidakpastian lainnya adalah adanya risiko ledakan yang tinggi. Seperti yang sudah kita ketahui, JavaScript sangat populer. Kemungkinan hal ini akan mendorong peretas, penipu, atau aktor jahat lainnya untuk menyerang dan mencari

potensi kelemahan dalam sistem. Situasi saat ini baru-baru ini menarik perhatian organisasi JavaScript, tetapi juga membawa risiko yang signifikan, sehingga yang menggunakannya harus waspada. Selain itu, JavaScript dapat digunakan untuk mengaktifkan kode komputer khusus pengguna. Setiap kali mereka menggunakannya, pengguna harus berhati-hati. Masalah lain yang tidak terkait dengan popularitas masalah ini adalah keamanan *browser*. Meskipun sebagian besar *browser* web mendukung JavaScript, masih ada beberapa yang tidak dan tidak dapat digunakan dalam konteks saat ini. Ini biasanya melibatkan seseorang mematikan *browser* web mereka. Jika dinyatakan, apakah ada tanda-tanda masalah? Karena baik pemilik situs web maupun pengunjung tidak dapat memaksimalkan konten mereka, inilah satu-satunya alasan.

3.9.5 Perbedaan JavaScript dengan Bahasa Pemrograman lainnya

Bahasa yang akan dibahas bersama adalah PHP, HTML, CSS, dan JavaScript. Elemen pertama adalah JavaScript, atau JS mungkin disertakan. Fungsi dasar JavaScript adalah membuat situs web apa pun lebih menarik dan menarik. JS adalah bahasa pemrograman yang diaktifkan oleh browser pengguna, berbeda dengan bahasa pemrograman lain yang diaktifkan oleh *server*. Pengembang umumnya tidak perlu melanjutkan prosedur asli untuk menyediakan fungsionalitas yang diperlukan karena JS biasanya beralih ke posisi bawahan organisasi lain. Anda mungkin sudah mengetahui bahwa HTML, juga dikenal sebagai *Hyper Text Markup Language*, sering digunakan di situs web dan berfungsi sebagai titik awal untuk berbagai bagian situs web. Setiap orang yang memiliki situs web mungkin merasa canggung menggunakan tag HTML "img" untuk gambar, "p" untuk paragraf, dan beberapa simbol lainnya. Sebaliknya, CSS, juga dikenal sebagai *Cascading Style Sheets*, sering digunakan untuk mengubah gaya dan format konten. Dalam HTML, setiap elemen harus dimasukkan secara manual, namun dalam CSS, hal ini tidak diperlukan untuk menampilkan konten apa pun. Anda tidak diharuskan

untuk memahami berungki tampilan elemen. PHP sering disebut sebagai "sisi server" karena sebagian besar pemrosesan datanya terjadi di *server*, tidak seperti JavaScript, yang diluncurkan dari komputer klien atau biasanya disebut sebagai "sisi klien". Bahasa pemrograman PHP sering disertakan dalam WordPress dan sistem manajemen lainnya. PHP sering digunakan dalam pengembangan *back end* karena menawarkan toples terbaik untuk memfasilitasi proses transfer informasi.

3.9.6 Cara Menulis Kode JavaScript di HTML

Pada contoh di bawah ini, kami telah memasukkan kode JavaScript menggunakan teknologi *embedding* HTML (ditempel). Meskipun semua telah dikatakan, masih ada beberapa hal yang memerlukan perhatian kita :

- a. *Embed* (kode HTML untuk Javascript diuji terus menerus. Konteks: saat ini)
- b. *Inline* (Kode JavaScript tertanam dalam atribut HTML).
- c. Eksternal (File HTML dan kode Javascript kompatibel).

1) Penulisan Kode JavaScript Embed

Dalam hal ini, kode Javascript tertanam dalam HTML dengan menggunakan elemen "<script>". Tag ini dapat ditulis dalam tag "<head>" dan "<body>".

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Universitas Stekom</title>
5     <script>
6       // ini adalah penulisan kode javascript
7       // di dalam tag <head>
8       console.log("Hello Stekom, dari Head");
9     </script>
10  </head>
11  <body>
12    <p>Tutorial Javascript untuk Pemula</p>
13    <script>
14      // ini adalah penulisan kode javascript
15      // di dalam tag <body>
16      console.log("Hello Stekom, dari body");
17    </script>
18  </body>
19 </html>
20

```

Meskipun dapat di tulis pada bagian <head> ataupun <body>, namun demikian, banyak orang menyarankan untuk menempatkannya di <body> karena akan membuat website memuat lebih cepat.

2) Penulisan Kode JavaScript *Inline*

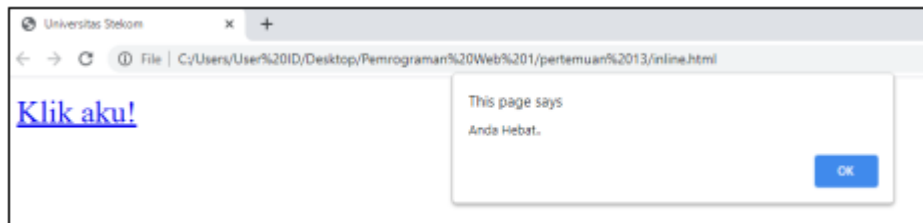
Dengan metode ini, kami akan menyematkan kode JavaScript dalam atribut HTML. Metode ini sering digunakan untuk mengaktifkan fungsi utama acara saat ini. Saat ditautkan, diklik, misalnya.

```
<a href="#" onclick="alert('Yey!')">Klik aku!</a>
```

Atau bisa juga seperti :

```
6 <body>
7 <a href="javascript:alert('Anda Hebat..')">Klik aku!</a>
8 </body>
9 </html>
```

Hasilnya :



Gambar 3.26 *Inline* JavaScript

Keterangan :

1. Pada contoh pertama di atas, kode JavaScript dituliskan pada *event* atribut. Satu-satunya atribut *event* yang ada dalam HTML adalah *onclick*. Oleh karena itu, kode JavaScript dapat ditulis dalam *event* atribut.
2. Pada HTML, atribut *event* ditulis terlebih dahulu dengan huruf on, seperti Atribut HTML yang dikenal sebagai *onclick* digunakan untuk *event clicks*. Silakan baca contoh di bagian bawah penjelasan ini.
3. Pada contoh di bawah ini, aktifkan fungsi `alert()`. Fungsi ini adalah salah satu untuk menyajikan dialog. Kemudian, dengan menggunakan JavaScript, kami menambahkan fungsi `alert()` ke atribut `href`. Atribut `href` biasanya digunakan untuk menentukan tautan atau alamat URL. Karena kami ingin menyematkan kode JavaScript di sini, kami akan

mengonversi alamat tautan menjadi JavaScript dan kemudian menerapkan fungsionalitas yang dimaksud.

Contoh atribut *event* HTML *onkeyup*

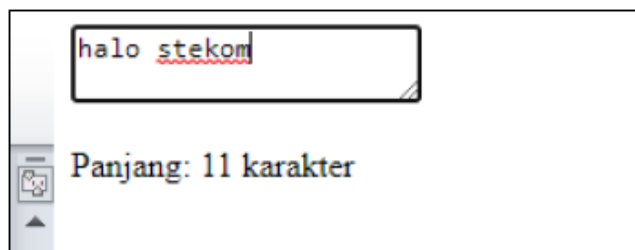
```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Universitas Stekom</title>
5 </head>
6 <body>
7   <textarea onkeyup="document.getElementById('counter').innerText = this.value.
   length"></textarea>
8   <p>Panjang: <span id="counter">0</span> karakter</p>
9 </body>
10 </html>

```

Dalam contoh ini, kami menggunakan *event keyup* yang mendukung *onkeyup*. Kejadian ini terjadi saat kami menekan tombol pada keyboard. Sederhananya, ketika kita mengklik elemen "*textarea*", kode JavaScript yang ada di sana akan dieksekusi. Kode JavaScript ini digunakan untuk menampilkan berbagai karakter dinamis.

Hasil:



Gambar 3.27 Jumlah Karakter dengan JavaScript

Tabel 3.6 *Event* Sistem dalam JavaScript

<i>Event</i>	Keterangan
<i>onError</i>	<i>Event</i> ini dibangkitkan ketika terjadi kesalahan. <i>Event</i> ini dimiliki oleh Window dan

<i>Event</i>	Keterangan
<i>onLoad</i>	<i>ImageEvent</i> ini dibangkitkan ketika suatu objek selesai ditampilkan pada halaman web. Objek yang dimaksud pada keterangan ini adalah objek <i>Window</i> , <i>Frame</i> dan <i>Image</i>
<i>onUnload</i>	<i>Event</i> ini dibangkitkan ketika suatu objek telah dikeluarkan dari ruang memori, atau telah selesai digunakan. Objek ini adalah objek <i>Window</i> , <i>Frame</i> dan <i>Image</i>

Tabel 3.7 *Event* Pengguna Beserta Keterangan

<i>Event</i>	Keterangan
<i>onAbort</i>	<i>Event</i> yang dimiliki oleh objek <i>image</i> ini akan dibangkitkan ketika sebuah gambar dihentikan proses pemunculannya yang dimungkinkan akibat penekanan tombol <i>stop</i> pada <i>browser</i>
<i>onBlur</i>	<i>Event</i> ini dibangkitkan ketika sebuah <i>element</i> (dalam hal ini adalah element <i>window</i> , <i>frame</i> , <i>select</i> , <i>text</i> , dan <i>textarea</i>) kehilangan <i>focusnya</i>
<i>onChange</i>	<i>Event</i> ini dibangkitkan sebuah <i>element</i> (dalam hal ini adalah element <i>select</i> , <i>text</i> dan <i>textarea</i>) telah diubah nilainya sebelumnya element tersebut kehilangan <i>focusnya</i>
<i>onClick</i>	<i>Event</i> ini dibangkitkan ketika terjadi aksi klik (dilakukan oleh pengguna) terhadap element yang dimaksud adalah semua elemen dari <i>form</i> yang dapat diklik seperti <i>element button</i> (tombol). Aksi klik yang dimaksud

<i>Event</i>	Keterangan
	adalah adanya penekanan tombol klik kiri <i>mouse</i> terhadap suatu <i>element</i> .
<i>onContextMenu</i>	<i>Event</i> ini terbangkitkan ketika terjadi aksi klik kanan dari tombol <i>mouse</i> terhadap suatu element
<i>onDblClick</i>	Hampir sama dengan <i>event onClick</i> , namun akan dibangkitkan ketika terjadi aksi klik ganda terhadap suatu <i>element</i>
<i>onFocus</i>	<i>Event</i> ini adalah kebalikan dari <i>event onBlur</i> , dimana akan dibangkitkan ketika suatu <i>element</i> dikenai fokus. <i>Element-element</i> yang dimaksud adalah <i>element window, frame, select, text</i> dan <i>textarea</i>
<i>onHelp</i>	<i>Event</i> ini dibangkitkan ketika terjadi penekanan terhadap tombol F1, yang umumnya digunakan berbagai aplikasi sebagai tombol <i>shortcut</i> untuk menampilkan fasilitas bantuan (<i>help</i>). Anda dapat menggunakan <i>event</i> ini untuk mengalihkan fitur <i>help</i> milik <i>browser</i> menjadi fitur <i>help</i> yang Anda buat sendiri untuk halaman web Anda
<i>onKeyDown</i>	<i>Event</i> ini dibangkitkan ketika terjadi penekanan ke bawah tombol <i>keyboard</i>
<i>onKeyPress</i>	<i>Event</i> ini dibangkitkan ketika terjadi aksi penekanan tombol <i>keyboard</i> . Aksi penekanan disini adalah sebuah aksi lengkap penekanan tombol <i>keyboard</i> , mulai dari

Event	Keterangan
	memencat tombol <i>keyboard</i> tersebut hingga melepaskannya
<i>onKeyUp</i>	<i>Event</i> ini dibangkitkan ketika tombol <i>keyboard</i> yang awalnya ditekan lalu dilepas, ketika tombol dilepas <i>event</i> ini bangkit
<i>onMouseDown</i>	<i>Event</i> ini dibangkitkan, ketika tombol mouse ditekan ke bawah
<i>onMouseMove</i>	<i>Event</i> ini dibangkitkan ketika <i>mouse</i> digerakkan (panah <i>mouse</i> berubah posisinya)
<i>onMouseout</i>	<i>Event</i> ini dibangkitkan ketika panah <i>mouse</i> keluar dari daerah lingkupan suatu <i>element</i> , yaitu <i>element hyperlink</i> dan <i>element area</i>
<i>onMouseover</i>	<i>Event</i> ini dibangkitkan ketika panah <i>mouse</i> berada di atas <i>element</i> , yaitu <i>element hyperlink</i> dan <i>element area</i>
<i>onReset</i>	<i>Event</i> yang dimiliki secara khusus oleh objek <i>form</i> ini akan dibangkitkan ketika terjadi penekanan terhadap tombol <i>reset</i> yang dimiliki <i>form</i> yang bersangkutan
<i>onResize</i>	<i>Event</i> ini dibangkitkan ketika jendela <i>browser</i> diubah ukurannya, baik diperbesar maupun diperkecil
<i>onSelect</i>	<i>Event</i> ini dibangkitkan ketika terjadi pemilihan teks pada <i>element</i> , <i>text</i> dan <i>textarea</i> dengan cara memberinya <i>highlight</i> (diblok)

<i>Event</i>	Keterangan
<i>onStop</i>	<i>Event ini dibangkitkan ketika pengguna melakukan penekanan terhadap tombol stop dari browsernya</i>
<i>onSubmit</i>	<i>Event ini dibangkitkan setelah terjadi penekanan tombol submit yang dimiliki sebuah form</i>

3) Penulisan Kode JavaScript Eksternal

Dalam metode ini, kami akan menyertakan kode JavaScript dalam *file* HTML yang diformat dengan ketat. Karena lebih mudah mengelola kode proyek dengan metode ini, cara ini biasanya digunakan untuk proyek-proyek besar. Buat file html dan *file* js dalam 1 *folder* atau menyesuaikan *folder*.

Isi *file* kode.js

```

1 alert("Halo, ini saya sedang belajar eksternal JS");

```

Isi *file* eksternal.html

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Universitas Stekom</title>
5 </head>
6 <body>
7   <h2>Belajar Javascript Dasar</h2>
8
9   <!-- sisipkan file js disini -->
10  <script src="kode.js"></script>
11 </body>
12 </html>
13

```

Untuk menyisipkan file JavaScript, gunakan *tag* `<script>...</script>`

Hasilnya:



Gambar 3.28 *Alert JavaScript*

Selain itu kita juga dapat menyisipkan JavaScript yang ada di internet dengan memberikan alamat URL lengkapnya seperti :

```
<script src="https://kit.fontawesome.com/a076d05399.js"></script>
```

3.10 Google Maps API

Google Maps API adalah suatu *library* yang berbentuk JavaScript yang berguna untuk memodifikasi peta yang ada di Google Maps sesuai kebutuhan informasi mengenai lokasi fasilitas umum seperti bengkel tambal ban, toko sparepart, dan ATM sangat diperlukan untuk pengguna *smartphone* Android. Informasi lokasi yang telah di definisikan tersebut dapat dilihat pada aplikasi Google Maps Android, namun data lokasi yang terkait masih terbatas. Adanya teknologi API dan pemetaan Google yaitu Google Maps dapat digunakan untuk mengembangkan suatu aplikasi pemetaan lokasi bengkel tambal ban, toko sparepart, dan ATM. Untuk itu diperlukan perancangan sistem berbasis android untuk mendapatkan sistem pencarian berdasarkan lokasi. Hasil dari aplikasi yang dikembangkan adalah dapat memberikan informasi jarak, waktu, dan rute menuju lokasi tertentu (Palupi et al., 2021).

(Aisa, 2021) Google Maps API merupakan aplikasi *interface* yang dapat diakses lewat JavaScript agar Google Maps dapat ditampilkan pada halaman web yang sedang kita bangun. Untuk dapat mengakses Google Maps, Kita harus melakukan pendaftaran Api *Key* terlebih dahulu dengan data pendaftaran berupa nama domain web yang kita bangun. Banyak sekali kegunaan Google Maps untuk website yang kita buat, diantaranya dapat digunakan untuk menampilkan lokasi

pemilik website (pada *about us*), lokasi *event*/kegiatan, atau dapat juga digunakan untuk aplikasi GIS berbasis web. Pada Google Maps API terdapat 4 jenis pilihan model peta yang disediakan oleh Google, diantaranya adalah:

1. *ROADMAP*, ini yang saya pilih, untuk menampilkan peta biasa 2 dimensi
2. *SATELLITE*, untuk menampilkan foto satelit
3. *TERRAIN*, untuk menunjukkan relief fisik permukaan bumi dan menunjukkan seberapa tingginya suatu lokasi, contohnya akan menunjukkan gunung dan sungai
4. *HYBRID*, akan menunjukkan foto satelit yang di atasnya tergambar pula apa yang tampil pada *ROADMAP* (jalan dan nama kota)