



PEMROGRAMAN WEB dengan HTML



HERU KARTIKA CANDRA



Diterbitkan Atas Kerjasama
Deepublish dengan Politeknik Banjarmasin



PEMROGRAMAN WEB DENGAN HTML

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. Penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. Penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

PEMROGRAMAN WEB DENGAN HTML

Heru Kartika Candra



PEMROGRAMAN WEB DENGAN HTML

Penulis :
Heru Kartika Chandra

ISBN :
978-623-7694-06-9

ISBN Elektronis :
978-623-7694-34-2

Editor dan Penyunting :
Adi Pratomo

Desain Sampul dan Tata Letak :
Rahma Indera; Eko Sabar Prihatin

Penerbit :
POLIBAN PRESS
Anggota APPTI (Asosiasi Penerbit Perguruan Tinggi Indonesia)
no.004.098.1.06.2019
Cetakan Pertama, 2020

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk
dan dengan cara apapun tanpa ijin tertulis dari penerbit

Redaksi :
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,
Pangeran, Komp. Kampus ULM, Banjarmasin Utara
Telp: (0511)3305052
Email: press@poliban.ac.id

Diterbitkan pertama kali oleh :
Poliban Press, Banjarmasin, Oktober 2020

Dicetak oleh :
PERCETAKAN DEEPUBLISH
Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoarjo, Ngaglik, Sleman
Jl.Kaliurang Km.9,3 – Yogyakarta 55581
Telp/Faks: (0274) 4533427
Website: www.deepublish.co.id
www.penerbitdeepublish.com
E-mail: cs@deepublish.co.id

Katalog Dalam Terbitan (KDT)

Heru Kartika Chandra —Cet. 1. — **Pemrograman Web dengan HTML**: Poliban Press,
Oktober 2020.

ix; 50 hlm.; 15.5x23 cm

UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada Poliban Press karena telah mempercayakan proses percetakan buku *Pemrograman Web dengan HTML* kepada Penerbit Deepublish. Semoga buku ini dapat memberikan manfaat kepada seluruh pembaca dan kerja sama ini dapat terus terjalin.



KATA PENGANTAR

Puji syukur ke hadirat Allah Swt. atas limpahan rahmat dan karunianya sehingga buku *Pemrograman Web dengan HTML* tahun 2020 telah dapat diselesaikan. Buku ini merupakan pengantar bagi Mahasiswa Komputer Akuntansi.

Terima kasih disampaikan kepada Joni Riadi S.S.T., M.T. selaku Direktur Politeknik Negeri Banjarmasin dan Nurmahaludin, S.T., M.T. selaku Ketua Pusat Penelitian dan Pengabdian Masyarakat beserta sekretaris dan staf. Terima kasih juga disampaikan kepada Faris Ade Irawan, Reza Fauzan, Eko Sabar Prihatin dan Rahma Indera yang telah berkontribusi dalam editing serta seluruh tim Poliban Press dan semua pihak yang telah ikut membantu dalam penyelesaian buku ini.

Kami menyadari masih terdapat kekurangan dalam buku ini untuk itu kritik dan saran terhadap penyempurnaan buku ini sangat diharapkan. Semoga buku ini dapat memberi manfaat bagi semua pihak.

Banjarmasin, September 2020

Poliban Press

PRAKATA

Assalamualaikum warahmatullahi wabarakatuh.

Puji syukur alhamdulillah kami panjatkan ke hadirat Tuhan Yang Maha Esa yang telah melimpahkan nikmat, taufik, serta hidayah-Nya yang sangat besar sehingga bahan ajar *Pemrograman Web dengan HTML* dapat diselesaikan.

Mudah-mudahan buku ajar ini bisa membantu mahasiswa untuk memahami secara kontekstual bagaimana merancang sistem berbasis web. Penulis yakin bahwa materi dalam buku ajar ini masih jauh dari kesempurnaan, sehingga terbuka untuk mendapatkan kritik dan saran untuk perbaikan dari sisi penulisan maupun pengembangan materi. Dan harapan penulis buku ajar ini dapat bermanfaat sebagai pengenalan dasar tentang pemrograman web.

Wassalamualaikum warahmatullahi wabarakatuh

Banjarmasin, 05 Mei 2020

Heru Kartika Candra, S.Si., M.T.

DAFTAR ISI

UCAPAN TERIMA KASIH.....	v
KATA PENGANTAR.....	vi
PRAKATA	vii
DAFTAR ISI.....	viii
BAB I PENDAHULUAN.....	1
1.1. Web Browser.....	1
1.2. Word Wide Web.....	1
1.3. Pengertian HTML.....	2
1.4. Software Pendukung.....	4
1.5. Struktur Document HTML.....	5
1.6. Atribut.....	9
1.7. Soal Latihan	9
BAB II TEXT & STYLES	10
2.1. Heading.....	10
2.2. Paragraph	12
2.3. <i>Text Formatting</i>	13
2.4. <i>Background Color</i>	15
2.5. <i>Text Color</i>	16
2.6. <i>Border Color</i>	18
2.7. <i>Fonts</i>	19
2.8. <i>Text Size</i>	20
2.9. <i>Text Alignment</i>	21
2.10. Soal Latihan	22
BAB III LIST & LINK.....	23
3.1. <i>Unordered List</i>	23

PEMROGRAMAN WEB DENGAN HTML

3.2.	<i>Ordered List</i>	24
3.3.	<i>Definition List</i>	25
3.4.	<i>Text Link</i>	27
3.5.	Gambar Link	28
3.6.	Soal Latihan	30
BAB IV	TABLES.....	31
4.1.	Tables.....	31
4.2.	<i>Border Tabel</i>	33
4.3.	<i>Colspan</i>	34
4.4.	<i>Rowspan</i>	35
4.5.	Soal Latihan	37
BAB V	FORM.....	38
5.1.	Pengertian Form	38
5.2.	Label	38
5.3.	Text Input.....	40
5.4.	Text Area	41
5.5.	Combo Box	42
5.6.	Checkbox	43
5.7.	Radio Button	45
5.8.	<i>Submit Button</i>	46
5.9.	<i>Reset Button</i>	47
5.10.	Soal Latihan	48
DAFTAR PUSTAKA		49
TENTANG PENULIS		50

BAB I

PENDAHULUAN

Capaian Pembelajaran

1. Mampu memahami *Word Wide Web* (WWW).
2. Mampu memahami struktur HTML.
3. Mampu memahami *tag* dan atribut HTML.

1.1. Web Browser

Web browser merupakan perangkat utama yang digunakan untuk menampilkan halaman web. Pada dasarnya halaman web dibuat dengan menggunakan HTML dan CSS. Setiap *browser* memiliki perbedaan dalam hal menampilkan halaman web dan fitur-fitur yang didukung dalam HTML dan CSS (Ariona, 2013).

1.2. Word Wide Web

Internet merupakan suatu jaringan global yang berfungsi untuk menghubungkan suatu *network* dengan *network* lainnya di seluruh dunia (Wempen, 2011). TCP/IP (*transmission control protocol/internet protocol*) menjadi protokol penghubung antara jaringan-jaringan yang beragam di seluruh dunia untuk dapat berkomunikasi. WWW (*world wide web*) merupakan bagian dari internet yang paling cepat berkembang dan paling populer. WWW bekerja berdasarkan pada tiga mekanisme berikut (Wempen, 2011).

1. Protokol adalah standard pada komputer *networking* yang di gunakan untuk berkomunikasi sedangkan HTTP (*hypertext transfer protocol*) adalah *protocol* untuk WWW.
2. *Address* www memiliki aturan penamaan alamat web yaitu URL (*uniform resource locator*) yang di gunakan sebagai standar alamat internet.

3. HTML digunakan untuk membuat dokumen yang bisa di akses melalui halaman web yang digunakan oleh pengguna.

1.3. Pengertian HTML

HTML (*Hypertext Markup Language*) merupakan bahasa yang digunakan untuk membuat sebuah halaman web dengan tujuan untuk menampilkan atau berbagai informasi dalam sebuah web yang akses melalui internet (Wempen, 2011). Semua halaman web yang sering anda buka, seperti facebook.com, twitter.com, google.com akan ditampilkan menggunakan HTML. Jadi bisa dikatakan HTML adalah bahasa dasar untuk menampilkan halaman web pada *web browser* (Ariona, 2013).

HTML terdiri dari 3 komponen kata yaitu *Hypertext*, *Markup* dan *Language*. Kata *Hypertext* dari HTML berarti “*text*” yang tidak hanya berfungsi sebagai *text* biasa tetapi juga dapat berfungsi sebagai penghubung ke halaman lain atau yang sering dikenal dengan istilah *link*. Selain *text* dapat dijadikan *link*, sebuah gambar juga dapat dijadikan sebagai *link* untuk penghubung ke halaman lain (Pratama, 2016).

Kata kedua dari HTML adalah *Markup* yang diambil dari kata bahasa inggris yaitu *mark* yang artinya “**tanda atau penanda**”. HTML menggunakan tanda-tanda khusus yang diperlukan untuk mengatur dan membuat struktur halaman *web* seperti tanda-tanda `<p>`, `<a>` dan masih banyak tanda-tanda khusus yang digunakan. Tanda-tanda ini pada HTML dikenal sebagai *Tag* yang berfungsi memberikan informasi kepada *web browser* (Pratama, 2016).

Kata ketiga adalah *Language* yang berarti “**bahasa**”. HTML tidak memiliki struktur dasar seperti variabel, kondisi *IF*, *function* atau *class* layaknya sebuah bahasa pemrograman komputer yang menggunakan *programming language*. Sehingga secara tidak langsung HTML dapat dikatakan bukanlah bahasa pemrograman (Pratama, 2016).

Dapat disimpulkan bahwa HTML (*hypertext markup language*) adalah sekumpulan simbol-simbol atau *tag* yang dituliskan dalam sebuah *file* dengan tujuan untuk menampilkan informasi dari yang dituliskan dalam bentuk simbol atau tersebut pada halaman *web browser* (Duckett,

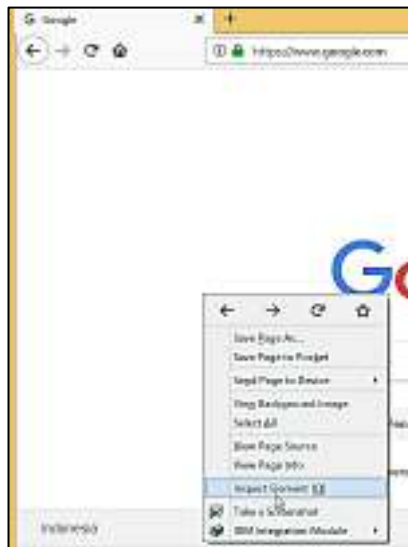
PEMROGRAMAN WEB DENGAN HTML

2011). *Tag* tersebut untuk memberikan informasi kepada *browser* untuk menampilkan halaman *web* dengan lengkap kepada pengguna.

Tag HTML selalu diawali dengan *tag* pembuka `<x>` dan *tag* penutup `</x>` (Duckett, 2011). Sebuah halaman website diapit oleh *tag* pembuka `<html>` dan *tag* penutup `</html>` untuk memulai menuliskan perintah-perintah HTML. *File*-file HTML selalu berakhiran dengan ekstensi `*.htm` atau `*.html`. Jadi jika anda mengetik sebuah naskah dan menyimpannya dengan ekstensi `*.html` maka anda membuat *file* yang berformat HTML.

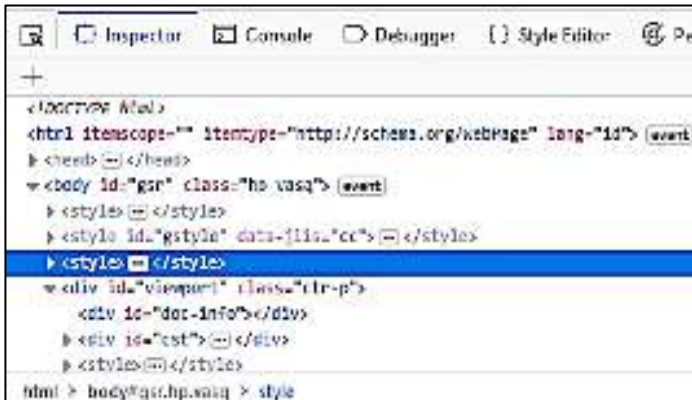
Buka salah satu *web browser* yang anda gunakan pada komputer kemudian buka sebuah situs seperti `http://www.google.com`.

1. Klik kanan pada area *browser* internet yang digunakan (buku ini saya menggunakan browser Mozilla Firefox dan pilih “*inspect element*” pada menu paling akhir seperti Gambar 1.1.



Gambar 1.1 Inspect Element

2. Perhatikan kode hasil *inspect element* pada Gambar 2.1.

A screenshot of a web browser's developer tools interface. The 'Inspector' tab is active, showing the DOM tree. The root element is <!DOCTYPE html>. Below it is the <html> tag with attributes 'itemscope=""', 'itemtype="http://schemas.org/webPage"', and 'lang="id"'. The <head> tag is expanded, showing <body> with attributes 'id="gar"', 'class="ho vasq"', and 'data-jlisa="cc"'. The <body> tag is further expanded, showing <div id="viewport" class="tr-p">, <div id="doc-info">, <div id="test">, and <style>. The breadcrumb at the bottom reads 'html > body>gar.hp.vasq > style'.

Gambar 1.2 Kode HTML dari Situs yang Bersangkutan

Sekarang perhatikan baris pertama pada HTML diawali dengan <!DOCTYPE HTML>, ini menandakan bahwa dokumen yang sedang dibuka dibuat menggunakan HTML. Sedangkan pada baris kedua ditulis dengan <HTML>, kode-kode tersebut merupakan *tag* pembuka untuk kode HTML.

1.4. Software Pendukung

1. *Browser*

Browser merupakan *software* yang diinstal di mesin *client* dan berfungsi untuk menampilkan *tag* HTML yang berada pada server untuk melakukan interaksi melalui halaman *web*. *Browser* yang sering digunakan biasanya Internet Explorer, Netscape Navigator, Mozilla Firefox atau Chrome dan masih banyak yang dapat digunakan.

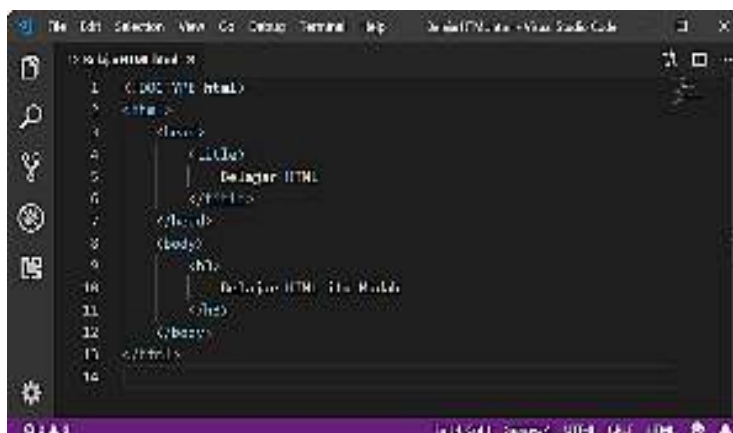
2. *Text Editor*

Program yang di gunakan untuk membuat dokumen HTML, ada banyak HTML *editor* yang bisa anda gunakan di antaranya: Microsoft FrontPage, Dreamweaver, Notepad, Visual Studio Code. Pada buku ini kita menggunakan Visual Studio Code. Anda dapat *download* Visual Studio Code di alamat ini <https://code.visualstudio.com/>.

Text editor Visual Studio Code merupakan *text editor* yang dibuat oleh Microsoft untuk sistem operasi *multiplatform* yang tersedia untuk sistem operasi Linux, Mac dan Windows. *Text editor* Visual Studio Code juga mendukung bahasa pemrograman JavaScript, Typescript, dan Node.js, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via *marketplace* Visual Studio Code (seperti C++, C#, Python, Go, Java, dst) sehingga banyak digunakan oleh *programmer*.

1.5. Struktur Document HTML

Suatu dokumen HTML juga mempunyai aturan dalam penulisannya, seperti *tag* `<!DOCTYPE HTML>` adalah *tag* awal dari setiap dokumen HTML, *tag* ini berfungsi untuk memberikan informasi pada *browser* bahwa dokumen yang sedang dibuka merupakan dokumen HTML. *Tag* ini dicantumkan pada awal dokumen HTML yang dibuat dengan tujuan untuk memberikan informasi kepada pembaca bahwa dokumen tersebut dibuat menggunakan HTML (Hadi, 2014).



Gambar 1.3 *Text Editor* Visual Studio Code

Dokumen HTML terdiri dari elemen-elemen yang berkaitan untuk membentuk dokumen HTML. Elemen-elemen HTML terbagi menjadi tiga

bagian yaitu elemen HEAD, elemen TITLE, dan elemen BODY. Masing-masing elemen diapit dengan tanda yang disebut delimiter <> sementara elemen yang telah diapit tanda delimiter disebut *Tag*.

1. HTML

Tag HTML merupakan kode awal dari isi dokumen yang dibuat menggunakan, semua kode HTML akan ditulis di dalam *tag* HTML dan setiap *tag* HTML juga memiliki *tag* pembuka dan *tag* penutup (Hadi, 2014). HTML memiliki cara untuk memperkenalkan *text* yang telah dituliskan kepada *web browser* baik *text* itu berupa *list*, paragraf, *link* dan sebagainya. Sekarang perhatikan Tabel 1.1 *tag* utama HTML.

Tabel 1.1 Tag Utama HTML

<html>	Tag pembuka
</html>	Tag penutup

2. HEAD

Tag head digunakan untuk menyimpan berbagai macam informasi tentang dokumen yang dituliskan dengan HTML. *Tag head* tidak akan ditampilkan secara langsung pada halaman *web browser*. *Tag head* juga digunakan untuk membuat *header* dari dokumen HTML dengan diapit oleh *tag* pembuka <HEAD> dan *tag* penutup </HEAD>.

Selain itu, *tag head* juga digunakan untuk menampilkan judul dari halaman *web* pada *tab browser* dengan diapit *tag* pembuka <TITLE> dan *tag* penutup </TITLE> (Howe, 2014). Sekarang tuliskan kode di bawah ini menggunakan *text editor* Visual Studio Code dan simpan file tersebut dengan berekstensi *.html.

```
<html>
  <head>
    <title>
      page title
    </title>
  </head>
</html>
```


Selanjutnya buka *file* yang berekstensi *.html tersebut dengan *browser* internet yang digunakan. Pada buku ajar ini menggunakan *browser* internet Firefox maka dapat dilihat seperti Gambar 1.4.



Gambar 1.4 Output Tag HEAD

Sekarang perhatikan kode HTML di mana *page title* diapit oleh *tag* <title>. Apa pun yang dituliskan antara *tag* <title> pembuka dan penutup, akan tampil di tab bagian atas pada *browser* yang digunakan untuk memberikan keterangan judul dari halaman *web* tersebut. Jadi fungsi dari *tag title* adalah untuk memberikan keterangan judul dari sebuah *web* untuk mempermudah pengguna untuk mengakses *web* tersebut.

3. BODY

Tag body merupakan inti dari semua dokumen HTML yang ingin ditampilkan pada halaman *browser*. *Tag body* berfungsi untuk memberikan informasi kepada pengguna berupa *text*, *image*, *link*, judul (*heading*), paragraf, gambar, tabel dan berbagai informasi lainnya yang akan ditampilkan pada halaman *web browser* (Howe, 2014). Bagian *body* dinyatakan dengan *tag* pembuka <BODY> dan *tag* penutup </BODY>. Di dalam sebuah dokumen HTML hanya terdapat satu *tag body*. Sekarang tuliskan kode di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan *name* belajar_html.

PEMROGRAMAN WEB DENGAN HTML

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>
      Belajar HTML
    </title>
  </head>
  <body>
    <h3>
      Belajar HTML itu Mudah
    </h3>
  </body>
</html>
```

Selanjutnya buka *file* belajar_html tersebut dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 1.5.

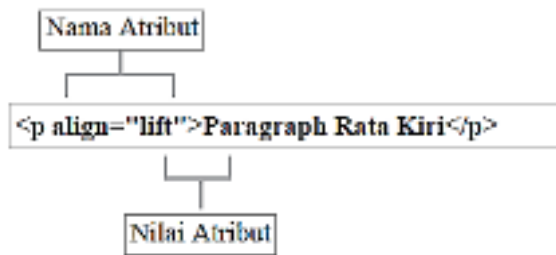


Gambar 1.5 Output Tag BODY

Sekarang perhatikan kode HTML yang diapit *tag body* pembuka dan penutup. Di mana pada *tag body* tersebut ditambahkan *tag h3 (header)* pembuka dan penutup. Sementara pada *tag h3* bertuliskan keterangan “belajar HTML itu mudah”. Apapun yang dituliskan antara *tag body* akan tampil di halaman *browser*. Jadi fungsi dari *tag body* adalah untuk menampilkan semua informasi yang berada pada halaman *browser* internet yang dapat memberikan informasi kepada pengguna.

1.6. Atribut

Atribut memberikan informasi tambahan tentang isi *tag* yang akan muncul pada tag pembuka untuk yang menyatakan karakteristik dari *tag* tersebut. Atribut terdiri dari dua bagian nama dan nilai, dipisahkan oleh tanda sama dengan (Powell, 2010). Sebagai contoh `<P STYLE="text-align:left">` digunakan untuk membuat rata kiri suatu paragraf. Untuk lebih jelas perhatikan Gambar 1.6 di bawah ini.



Gambar 1.6 Atribut

1.7. Soal Latihan

1. Jelaskan fungsi dari *web browser*?
2. Sebutkan dan jelaskan fungsi dari *www*?
3. Jelaskan apa yang anda ketahui tentang HTML?
4. Sebutkan dan jelaskan struktur dokumen HTML?
5. Apa fungsi atribut pada *tag* HTML?

BAB II

TEXT & STYLES

Capaian Pembelajaran

1. Mampu memahami *heading*, *paragraph* dan *font*.
2. Mampu membuat *background*, mengganti *color* pada *text*.
3. Mampu membuat *border*, mengatur ukuran dan jenis-jenis *font*.
4. Mampu mengatur tata letak *text*.

2.1. Heading

Tag heading biasanya digunakan untuk membuat judul pada halaman *web*, *tag heading* pada HTML terdiri dari 6 level yaitu h1, h2, h3, h4, h5, dan h6. *Tag heading* secara *default* di tampilkan oleh *browser* dengan huruf tebal (*bold*), *tag heading* yang paling besar adalah h1 dan yang paling kecil adalah h6 (Robbins, 2012).

Selain itu, mesin pencari melihat level *heading* sebagai bagian dari algoritme (informasi level *heading* yang lebih tinggi mungkin membuat mesin pencarian lebih lama). Untuk alasan ini, disarankan untuk menggunakan level 1 (h1) untuk membuat judul. Untuk lebih jelas penggunaan *heading* dapat dilihat pada Tabel 2.1.

Tabel 2.1 *Tag Heading*

Tag	Keterangan
<h1></h1>	Digunakan untuk judul halaman utama
<h2></h2>	Untuk sub judul halaman utama. Jika ada sub judul di bawah, makan gunakan tag h3 dan seterusnya.
<h3></h3>	Heading level 3
<h4></h4>	Heading level 4
<h5></h5>	Heading level 5
<h6></h6>	heading level 6

PEMROGRAMAN WEB DENGAN HTML

Sekarang tuliskan kode di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut. Pada *file name* isikan dengan judul *heading* dan pada *save as type* pilih HTML.

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>Belajar HTML Heading</title>
  </head>
  <body>
    <h1>Ini Heading level 1</h1>
    <h2>Ini level 2</h2>
    <h3>Inilevel 3</h3>
    <h4>Ini level 4</h4>
    <h5>Ini level 5</h5>
    <h6>Inilevel 6</h6>
  </body>
</html>
```

Selanjutnya buka *file* heading.html tersebut dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.1. Sekarang perhatikan adanya judul *heading* level 1 dimaksudkan agar suatu dokumen HTML lebih terstruktur layaknya sebuah dokumen resmi seperti skripsi atau paper yang mengharuskan adanya perbedaan antara bab utama dengan sub-bab. *Heading* juga memudahkan pengguna untuk memahami isi dari halaman *web* yang disampaikan oleh penulis.



Gambar 2.1 *Heading*

Sekarang perhatikan kode HTML yang diapit *tag* h1 yang memiliki tulisan lebih besar dibandingkan dengan yang lain h2 dan seterusnya. Sementara *tag* h6 menjadi tulisan yang paling kecil. Jadi fungsi dari *tag* h1 sampai dengan *tag* h6 untuk memberikan perbedaan level setiap tulisan.

2.2. Paragraph

Paragraph atau paragraf adalah elemen yang paling dasar dalam sebuah dokumen. Untuk membuat paragraf pada halaman *web* menggunakan *tag* <p> untuk awal paragraf dan *tag* </p> untuk penutup akhir dari paragraph (Robbins, 2012). *Tag* paragraf untuk mendefinisikan sebuah paragraf dengan mengatur *attribute left* untuk rata kiri, *attribute right* untuk rata kanan, *attribute center* untuk rata tengah dan *attribute justify* untuk rata kiri dan kanan. Sementara untuk berpindah ke baris berikutnya menggunakan *tag*
.

Tabel 2.2 *Tag dan Attribute Paragraph*

Tag	Atribut	Keterangan
<p></p>	Left	Rata kiri
	Right	Rata kanan
	Center	Rata tengah
	Justify	Rata kiri kanan
 		Memisahkan bagian <i>text</i> dengan <i>text</i> lain

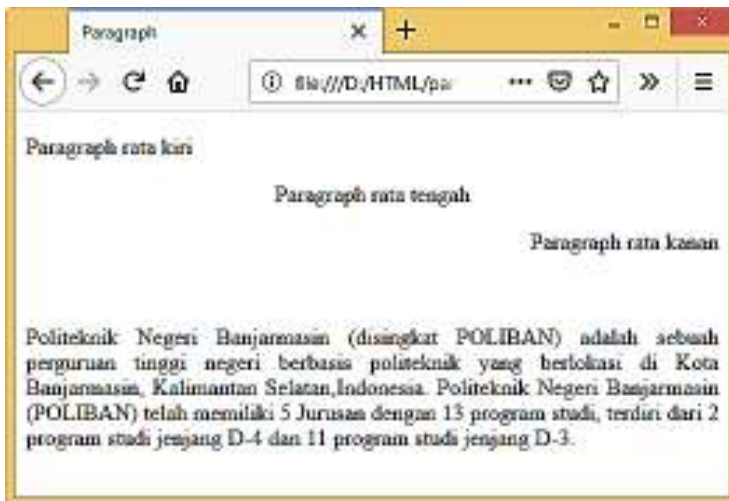
Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut menjadi *paragraph.html*.

```
<!DOCTYPEhtml>
<html>
<head><title>Paragraph</title></head>
<body>
<palgin="left">Paragraph rata kiri</p>
<palgin="center">Paragraph rata tengah</p>
<palgin="right">Paragraph rata kanan</p>
<palgin="justify">Politeknik Negeri Banjarmasin
```

PEMROGRAMAN WEB DENGAN HTML

(disingkat POLIBAN) adalah sebuah perguruan tinggi negeri berbasis politeknik yang berlokasi di Kota Banjarmasin, Kalimantan Selatan, Indonesia. Politeknik Negeri Banjarmasin (POLIBAN) telah memiliki 5 Jurusan dengan 13 program studi, terdiri dari 2 program studi jenjang D-4 dan 11 program studi jenjang D-3.</p>

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



Gambar 2.2 *Output Paragraph*

Sekarang perhatikan kode HTML yang diapit tag `<p>` dengan attribute `align` dan nilai attribute `left`, `center`, `right` dan `justify`. Masing-masing nilai attribute berfungsi untuk membuat mengatur tata letak *text* yang akan ditampilkan pada halaman *web*.

2.3. Text Formatting

Text formatting digunakan untuk melakukan pengaturan atau konfigurasi pada teks yang dijadikan isi dari sebuah halaman *web*.

PEMROGRAMAN WEB DENGAN HTML

Pengaturan *text* terlihat secara *real* pada halaman *web* yang diinputkan pada dokumen HTML. Tujuan penggunaan *text formatting* untuk membuat tampilan teks pada halaman *web* menjadi lebih baik atau mudah dipahami oleh pengguna. Secara umum pengaturan teks *formatting* meliputi pengaturan tebal, miring, garis bawah, teks berpangkat (Robbins, 2012). Penggunaan *tag text formatting* dapat dilihat pada Tabel 2.3.

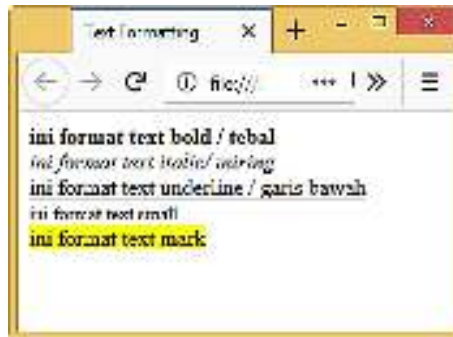
Tabel 2.3 Tag Formatting

Tag	Keterangan
	Format text bold / tebal
<i>	Format text italic / miring
<u>	Format text underline / garis bawah
<small>	Format text small
<mark>	Format text mark

Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan nama *formatting.html*.

```
<!DOCTYPEhtml>
<html>
  <head>
    <title>Text Formatting</title>
  </head>
  <body>
    <b> ini format text bold / tebal </b><br>
    <i> ini format text italic/ miring </i><br>
    <u> ini format text underline / garis bawah </u><br>
    <small> ini format text small </small><br>
    <mark> ini format text mark</mark>
  </body>
</html>
```

Selanjutnya buka *file* belajar_html tersebut dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.3.



Gambar 2.3 *Output Text Formatting*

Dari *output* terlihat perbedaan jenis font seperti cetak tebal, cetak miring, garis bawah dan *mark*. Penggunaan *text formatting* untuk memudahkan pengguna ketika berinteraksi dengan halaman *web* yang pertama kali di akses.

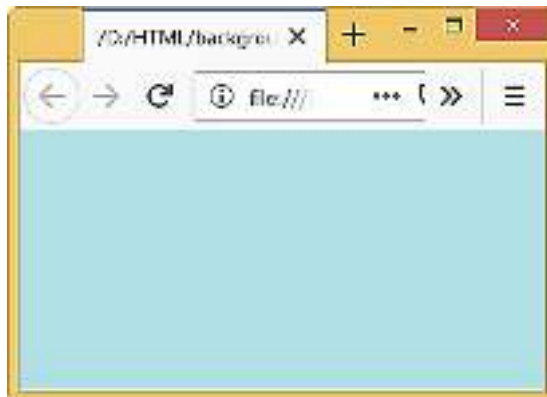
2.4. Background Color

Background adalah salah satu komponen desain yang sangat penting untuk halaman *web*. Dengan membuat warna *background* yang menarik dapat memberikan dekorasi visual terhadap sebuah halaman *web* sehingga dapat memberikan kemudahan nalar bagi pengguna untuk memahami konten yang disampaikan dari sebuah halaman *web* (Robbins, 2012).

Background juga memiliki pengaruh yang sangat besar terhadap keseluruhan desain dari sebuah *website* untuk memudahkan pengguna dalam berinteraksi dengan tampilan halaman *web* yang baru pertama digunakan oleh pengguna. Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan *name* `background_color.html`.

```
<!DOCTYPEhtml>
<html>
<bodystyle="background-color:powderblue;">
</body>
</html>
```

Selanjutnya buka *file* background_color.html tersebut dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.4.



Gambar 2.4 Output Background Color

Dari *output* Gambar 2.4 perhatikan *tag body* yang ditambahkan *attribute style* dengan *value* “background-color:powderblue” kode html tersebut berfungsi untuk mengganti warna *background* halaman *web* menjadi biru.

2.5. Text Color

Text color berfungsi untuk memberikan warna tulisan atau teks untuk desain halaman *web* terlihat lebih indah. Untuk mengganti warna pada teks dengan menggunakan *tag* dan mengganti nama atribut menjadi *color*. Setelah mengganti atribut *color* selanjutnya mengisi nilai atau memilih warna *color text*. Untuk pengisian nilai *color* terdapat dua pilihan. Yang pertama: menuliskan secara langsung nama *color* yang akan dipilih. Sebagai contoh, ingin mengganti warna *text* menjadi “hitam”, maka ditulis dengan “*black*”, yang untuk menyisipkan nilai warnanya. Yang kedua: menuliskan kode warna, sebagai contoh ingin mengganti warna *text* menjadi “hitam”. Maka ditulis kode *text* warna hitam “#000000”.

PEMROGRAMAN WEB DENGAN HTML

Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan nama `text_color.html`.

```
<!DOCTYPEhtml>
<html>
<body>
  <fontcolor="black">Ini Adalah Warna Hitam</font><br>
  <fontcolor="blue">Ini Adalah Warna Biru</font><br>
  <fontcolor="red">Ini Adalah Warna Biru</font><br>
  <fontcolor="green">Ini Adalah Warna Hijau</font><br>
  <fontcolor="yellow">Ini Adalah Warna Kuning</font><br>
  <fontcolor="purple">Ini Adalah Warna Ungu</font><br><hr>
  <fontcolor="#ffffff">Ini Adalah Warna Putih</font><br>
  <fontcolor="#000000">Ini Adalah Warna Hitam</font><br>
  <fontcolor="#f0f0g88f">Ini Adalah Warna Pink</font>
</body>
</html>
```

Selanjutnya buka *file* `text_color.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.5. Dari *output* tersebut dapat dilihat *text* menjadi berwarna sesuai dengan nilai atribut yang dituliskan pada *tag font*.



Gambar 2.5 Output Text Color

2.6. *Border Color*

Border merupakan sebuah garis yang memberikan batasan dari sebuah kata atau kalimat untuk paragraf. Selain itu, penggunaan *border* juga bisa untuk hal yang lain seperti *border* untuk gambar atau sebuah *link*. Pada dasarnya tujuan utama pembuatan *border* adalah untuk menandai suatu yang penting atau sebagai tanda pemisah (Robbins, 2012). Bagi yang ingin menggunakan *border* dapat dilihat pada Tabel 2.4 jenis-jenis *border*.

Tabel 2.4 Jenis-Jenis *Border*

Nama Border
Dotted
Dashed
Ridge
Groove
Double

Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan nama `border_color.html`.

```
<!DOCTYPEhtml>
<html>
<body>
<h3style="border: 2px solid Tomato;">Border Color
Tomato</h3>
<h3style="border: 2px solid DodgerBlue;">Border Color
DodgerBlue</h3>
<h3style="border: 2px solid Violet;">Border Color
Violet</h3>
</body>
</html>
```

Kode 4p digunakan untuk mengganti ketebalan *border*. Kode *padding* 10px untuk mengganti jarak teks dengan *border*. Untuk mengganti tinggi dan lebar *border* dengan cara mengganti nilai kode *height*: 60px (tinggi) dan *width*: 260px (lebar).

Selanjutnya buka *file* `border_color.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.6. Dari *output* tersebut dapat dilihat *text* memiliki *border* sesuai dengan warna yang dituliskan pada kode HTML. Juga terlihat adanya jarak setiap *border* dengan *text* karena telah ditambahkan kode HTML *padding* 10px.



Gambar 2.6 *Output Border Color*

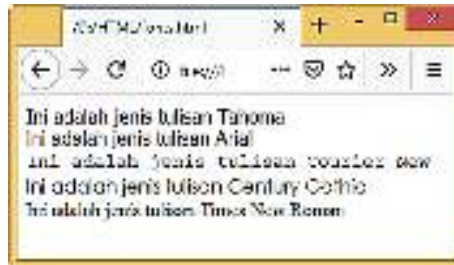
2.7. *Fonts*

Fonts atau huruf digunakan untuk mengganti jenis *font* pada halaman *web* menjadi lebih interaktif. Untuk mengatur huruf dengan menggunakan *tag* `` dan *tag* penutup ``. *Tag* ini mempunyai beberapa atribut untuk mengatur *formatting style* dari suatu teks. Untuk mengganti jenis *font* dengan menambahkan atribut “*face*” dan mengganti nilai atribut sesuai dengan jenis font yang diinginkan sebagai contoh atribut “*face*” dengan nilai “*arial*”. Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan nama `fonts.html`.

```
<!DOCTYPEhtml>
<html>
<body>
<fontface="Tahoma">Ini adalah jenis tulisan
Tahoma</font><br>
<fontface="Arial">Ini adalah jenis tulisan
Arial</font><br>
<fontface="Courier New">Ini adalah jenis tulisan Courier
```

```
New</font><br>
<fontface="Century Gothic">Ini adalah jenis tulisan
Century Gothic</font><br>
<fontface="Times New Roman">Ini adalah jenis tulisan
Times New Roman</font>
</body>
</html>
```

Selanjutnya buka *file* fonts.html dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.7. Dari *output* tersebut dapat dilihat jenis *font* yang berbeda setiap kalimat. Perubahan jenis *font* mengikut dengan kode HTML yang dituliskan.



Gambar 2.7 Output Fonts

2.8. Text Size

Text size digunakan untuk mengatur besar kecil ukuran *font* dengan menambahkan atribut *size* pada *tag fonts* dan mengganti nilai atribut *size*="n" atau angka satu (1) sampai angka tujuh (7). Nilai *n* terdiri dari angka satu menjadi ukuran terkecil sementara angka tujuh menjadi ukuran terbesar (Robbins, 2012). Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan nama text_size.html.

```
<!DOCTYPEhtml>
<html>
<body>
  <fontsize="1">ini adalah ukuran Font 1</font><br>
```

PEMROGRAMAN WEB DENGAN HTML

```
<fontsize="2">ini adalah ukuran Font 2</font><br>
<fontsize="3">ini adalah ukuran Font 3</font><br>
<fontsize="4">ini adalah ukuran Font 4</font><br>
<fontsize="5">ini adalah ukuran Font 5</font><br>
<fontsize="6">ini adalah ukuran Font 6</font><br>
<fontsize="7">ini adalah ukuran Font 7</font><br>
</body>
</html>
```

Selanjutnya buka *file* `text_size.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 2.8. Perhatikan *font* satu (1) sampai *font* tujuh (7) dari halaman *web* memiliki ukuran yang paling kecil dari angka satu dan yang paling besar angka tujuh. Tujuan penggunaan atribut *size* pada *tag font* untuk membedakan *font* tertentu pada halaman *web* untuk memudahkan pengguna.



Gambar 2.8 Output Text Size

2.9. Text Alignment

Untuk mengatur tata letak *text* menjadi rata tengah (*center*), rata kiri, (*left*) atau rata kanan (*right*) dengan menggunakan atribut *style* dan menambahkan nilai atribut *text-align*. *Text-align* ini memiliki beberapa nilai misalnya *center* untuk membuat *text* di tengah-tengah, *left* untuk membuat *text* rata kiri dan *right* membuat *text* menjadi rata kanan (Robbins, 2012). Sekarang tuliskan kode HTML di bawah ini menggunakan *text editor* Visual Studio Code dan simpan *file* tersebut dengan nama `text_alignment.html`.

```
<!DOCTYPEhtml>
<html>
<body>
  <h4style="text-align:left;">Heading Rata Kiri</h4>
  <h4style="text-align:center;">Heading Rata Tengah</h4>
  <h4style="text-align:right;">Heading Rata Kanan</h4>
</body>
</html>
```

Dari Gambar 2.9 dapat dilihat perbedaan kode HTML *text-align left* yang menghasilkan *text* rata kiri. Sementara *text-align center* menghasilkan *text* rata tengah dan *text-align right* menjadikan rata kanan.



Gambar 2.9 *Output Text Alignment*

2.10. Soal Latihan

1. Jelaskan fungsi *heading* pada halaman *web*?
2. Jelaskan perbedaan *tag* `<p>` dengan *tag* `
` pada kode html?
3. Jelaskan perbedaan teks *formatting* dengan *text alignment*?
4. Pada *tag font* terdapat atribut *face*, jelaskan fungsi atribut tersebut?
5. Tuliskan kode HTML untuk membuat *border color*?

BAB III

LIST & LINK

Capaian Pembelajaran

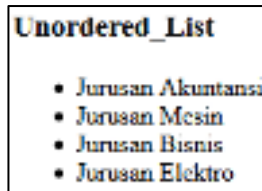
1. Mampu memahami *unordered list*, *ordered list* dan *definition list*.
2. Mampu membuat *text* dan gambar menjadi sebuah *link*.

3.1. Unordered List

Unordered list digunakan untuk membuat daftar yang tidak memiliki urutan atau yang urutannya tidak penting. Pembuatan *unordered list* dilakukan dengan menggunakan tag `` (*unordered list*). Sementara untuk mengisikan daftar *list* pada *unordered list* menggunakan tag `` (*list item*) sebagai pembuka dan tag `` sebagai penutup. *Unordered list* bertujuan untuk memudahkan untuk mengurutkan daftar tidak memiliki urutan (Howe, 2014). Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama `unordered_list.html`.

```
<!DOCTYPEhtml>
<html>
<head>
<title>Unordered_List</title>
</head>
<body>
  <h3> Unordered_List</h3>
  <ul>
    <li> Jurusan Akuntansi </li>
    <li> Jurusan Mesin </li>
    <li> Jurusan Bisnis </li>
    <li> Jurusan Elektro </li>
  </ul>
</body>
</html>
```

Sekarang perhatikan kode HTML yang diapit dengan tag `` yang digunakan untuk membuat *unordered list*. Sedangkan tag `` digunakan untuk membuat daftar *list* dari tag ``. Selanjutnya buka file `unordered_list.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 3.1. Dari *output* gambar tersebut terlihat daftar *list* item yang disusun tidak berurutan.



Gambar 3.1 *Output Unordered List*

3.2. *Ordered List*

Ordered list digunakan untuk membuat daftar yang diurutkan dengan benar seperti angka 1 sampai 9 atau huruf A sampai Z dan lain-lain. Untuk membuat daftar berurut menggunakan tag `` singkatan dari *ordered list*. Sementara untuk membuat daftar urutan tersebut tetap menggunakan tag `` (Powell, 2010).

Secara *default* *ordered list* menggunakan angka untuk membuat *list item*. Apabila ingin mengganti angka tersebut menjadi huruf, yang perlu ditambahkan adalah atribut *type* dan nilai atribut diisi sesuai dengan *list item* yang ingin digunakan. Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama `ordered_list.html`.

```
<!DOCTYPEhtml>
<html>
<head><title>Ordered_List</title></head>
<body>
  <h3> Ordered_List</h3>
  <ol>
    <li> Jurusan Akuntansi </li>
```

```
<li> Jurusan Mesin </li>
<li> Jurusan Bisnis </li>
<li> Jurusan Elektro </li>
</ol>
</body>
</html>
```

Sekarang perhatikan kode HTML yang diapit dengan tag `` digunakan untuk membuat *ordered list*. Sedangkan tag `` digunakan untuk membuat daftar *list item* dari tag ``. Karena pada contoh kode di atas tidak menggunakan atribut *type* maka *list item* dibuat secara *default* menggunakan angka 1.

Selanjutnya buka *file* `unordered_list.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 3.2. Dari *output* gambar tersebut terlihat daftar *list item* yang tersusun berurutan dari angka 1 sampai 4.



Gambar 3.2 *Output Ordered List*

3.3. *Definition List*

Definition list digunakan untuk membuat daftar definisi dari istilah-istilah yang ada dalam dokumen. *Definition list* biasanya digunakan untuk membuat glosarium dan indeks (Howe, 2014). Untuk membuat *definition list* terdapat tiga tag yang perlu diperhatikan sebagai berikut.

1. Untuk membuat daftar definisi menggunakan tag `<dl>` yang berfungsi untuk menyatakan tempat daftar definisi.

PEMROGRAMAN WEB DENGAN HTML

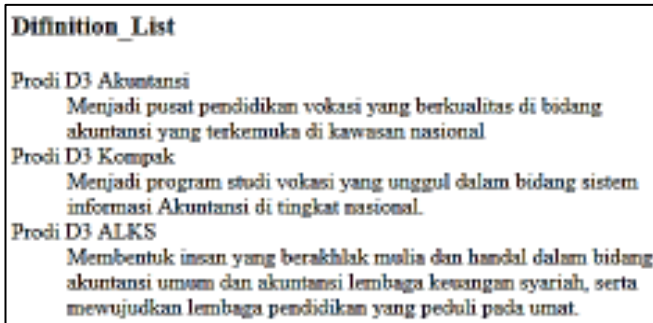
2. Untuk membuat istilah yang memiliki definisi menggunakan *tag* `<dt>` yang berfungsi untuk menjelaskan istilah yang akan didefinisikan.
3. Untuk memberikan penjelasan istilah yang disebutkan menggunakan *tag* `<dd>` yang berfungsi untuk menyatakan tempat definisi istilah.

Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama `definition_list.html`.

```
<!DOCTYPEhtml>
<html>
<head><title>Definition_List</title></head>
<body>
<h3>Difinition_List</h3>
<dl>
<dt> Prodi D3 Akuntansi </dt>
<dd>Menjadi pusat pendidikan vokasi yang
berkualitas di bidang akuntansi yang terkemuka di
kawasan nasional</dd>
<dt> Prodi D3 Kompak </dt>
<dd>Menjadi program studi vokasi yang unggul dalam
bidang sistem informasi Akuntansi di tingkat
nasional.</dd>
<li> Prodi D3 ALKS </li>
<dd>Membentuk insan yang berakhlak mulia dan
handal dalam bidang akuntansi umum dan akuntansi
lembaga keuangan syariah, serta mewujudkan lembaga
pendidikan yang peduli pada umat.</dd>
</dl>
</body>
</html>
```

Sekarang perhatikan *tag* pembuka `<dl>` dan *tag* penutup `</dl>` yang digunakan untuk membuat *definition list*. Sementara *tag* `<dt>` digunakan membuat istilah *definition* dan untuk membuat penjelasan dari istilah *definition list* menggunakan *tag* `<dd>`. Selanjutnya buka *file* `defintion_list.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 3.3. Dari gambar tersebut dapat dilihat Prodi D3 Akuntansi

sebagai *definition list*. Baris selanjutnya menjelaskan Prodi D-3 Akuntansi yang dibuat menggunakan tag <dd>.



Gambar 3.3 Output Definition List

3.4. Text Link

Text link adalah tag html yang berfungsi menghubungkan suatu halaman *web* ke halaman *web* yang lain dengan cara mengklik tulisan yang dijadikan sebagai *link*. *Text link* dibuat menggunakan tag pembuka <a> dan tag penutup. Untuk memasukan *link* yang ingin diklik menggunakan atribut href dan nilai atribut diisi dengan nama *link* (Robbins, 2012). Untuk keterangan yang lebih lengkap terkiat tag *text link* dapat dilihat pada Tabel 3.1 di bawah ini.

Tabel 3.1 Daftar Tag Link

Tag	Atribut	Keterangan
<a>	Href	Link yang akan dituju
	Src	Lokasi gambar yang akan dijadikan <i>link</i>
	Title	Untuk judul gambar yang dijadikan <i>link</i> ketika <i>cursor</i> di arahkan pada gambar

Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama text_link.html.

```
<!DOCTYPEhtml>
<html>
<head>
<title> Membuat Link </title>
</head>
<body>
<h2>Membuat Link pada halaman web</h2>
<p><a href="https://www.google.com/">Visit
Google</a></p>
</body>
</html>
```

Selanjutnya buka *file* text_link.html dengan browser internet yang digunakan maka hasilnya seperti Gambar 3.4. Dari gambar tersebut dapat dilihat apabila Visit Google diklik akan membuka *web google*.



Gambar 3.4 Output Link Text

3.5. Gambar Link

Gambar link adalah sebuah gambar ketika diklik oleh pengguna akan mengarah ke halaman *web*. Untuk membuat gambar menjadi sebuah *link* tidak berbeda dengan membuat *text link* yaitu menggunakan tag `<a>`, tetapi untuk menampilkan gambar perlu ditambahkan menggunakan tag `` dan atribut `src` di dalam tag `<a>` (Howe, 2014). Atribut `src` digunakan untuk menampilkan lokasi gambar tersebut disimpan. Untuk

PEMROGRAMAN WEB DENGAN HTML

mengatur ukuran lebar dan tinggi gambar cukup ditambahkan atribut *width* untuk lebar dan *height* untuk tinggi dengan nilai atribut berupa angka.

Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama *gambar_link.html*. Selanjutnya buat *folder* *img* pada *folder* HTML dan simpan gambar yang akan dijadikan *link* pada *folder* *img*. Contoh ini dibuka ini menggunakan gambar Google untuk dijadikan sebagai gambar *link* untuk membuka halaman *web* www.google.com.

```
<!DOCTYPEhtml>
<html>
<head>
<title>
Membuat Link Menjadi Gambar
</title>
</head>
<body>
<h3>Membuat Gambar Menjadi Link pada web</h3>
<a href="https://www.google.com"></a>
</body>
</html>
```

Selanjutnya buka *file* *gambar_link.html* dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 3.5. Jika gambar Google tersebut diklik maka halaman *web* akan mengarah ke www.google.com.



Gambar 3.5 Output Gambar Menjadi Link

3.6. Soal Latihan

1. Apa yang anda ketahui tentang *list* dan *link*?
2. Tuliskan kode HTML untuk membuat *unordered list*?
3. Tuliskan kode HTML untuk membuat *ordered list*?
4. Apa fungsi *src* pada *link* gambar?
5. Jelaskan atribut apa saja yang digunakan untuk mengatur ukuran gambar?

BAB IV

TABLES

Capaian Pembelajaran

1. Mampu memahami penggunaan *tag* tabel.
2. Mampu membuat *border* tabel.
3. Mampu memahami penggunaan *colspan* dan *rowspan*.

4.1. Tables

Untuk menampilkan data yang terstruktur atau tampilan data dari *database* biasanya disajikan dalam bentuk tabel sehingga data yang disajikan mudah dipahami oleh pengguna. Untuk menampilkan data dalam bentuk tabel pada HTML ada empat *tag* yang digunakan yaitu *tag* `<table>`, *tag* `<th>`, *tag* `<tr>`, dan *tag* `<td>` (Robbins, 2012). Secara *default text* yang dibuat dengan *tag* `<th>` selalu berada di tengah karena digunakan untuk membuat *header* pada tabel. Adapun keempat *tag* tersebut dapat dilihat pada Tabel 4.1 daftar *tag* tabel.

Tabel 4.1 Daftar Tag Tabel

Tag	Keterangan
<code><table></code>	Digunakan untuk memulai tabel
<code><tr></code>	Singkatan dari <i>table row</i> , digunakan untuk membuat baris dari tabel
<code><td></code>	Singkatan dari <i>table data</i> , digunakan untuk menginput data ke tabel
<code><th></code>	Untuk membuat <i>header</i> tabel

Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama `tabel.html`. Perhatikan kode *tag* pembuka `<table>` yang diberikan atribut *style* dengan nilai *width* 100%. Pengaturan *width* 100% ini bertujuan mengatur ukuran tabel mengikuti ukuran *browser* ketika diklik *maximise* atau *minimize*.

```
<!DOCTYPEhtml>
<html>
<body>
<h2>Membuat Table</h2>
<tablestyle="width:100%">
  <tr>
    <th>Nama Depan</th>
    <th>Nama Tengah</th>
    <th>Nama Belakang</th>
  </tr>
  <tr>
    <td>Adzra</td>
    <td>Nadhifa</td>
    <td>Syafa</td>
  </tr>
  <tr>
    <td>Aqmar</td>
    <td>Rafiq</td>
    <td>Syaputra</td>
  </tr>
  <tr>
    <td>Akthar</td>
    <td>Farzan</td>
    <td>Wijaya</td>
  </tr>
</table>
</body>
</html>
```

Selanjutnya buka filetabel.html dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 4.1. Dari gambar tersebut dapat terlihat data tersusun dalam bentuk tabel. *Tag* <tr> digunakan untuk membuat baris pada tabel sedangkan *tag* <th> untuk membuat *header* pada tabel. Sementara *tag* <td> untuk memberikan *input* atau keterangan pada *tag header*.

Membuat Table		
Nama Depan	Nama Tengah	Nama Belakang
Adzra	Nadhifa	Syafa
Aqmar	Rafiq	Syaputra
Akthar	Farzan	Wijaya

Gambar 4.1 *Output* Tabel

4.2. *Border Tabel*

Border digunakan untuk mengatur ukuran garis tepi yang ada pada tabel, satuan yang digunakan adalah satuan *pixel*. Jika tidak memberikan nilai pada bagian *border* maka tabel akan ditampilkan tanpa memiliki garis (Howe, 2014). Ada dua atribut yang perlu diperhatikan dalam membuat *border* tabel yaitu *cellpadding* dan *cellspacing*. *Cellpadding* adalah atribut untuk mengatur jarak teks dengan garis di dalam sel. *Cellspacing* adalah atribut yang digunakan untuk menentukan ruang antar sel.

Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan *name* `border_tabel.html`.

```
<!DOCTYPEhtml>
<html>
<body>
<h2>Membuat Border Pada Table</h2>
<tablestyle="width:100%"border="1"cellpadding="4"cellspacing="0">
  <tr>
    <th>Nama Depan</th>
    <th>Nama Tengah</th>
    <th>Nama Belakang</th>
  </tr>
  <tr>
    <td>Adzra</td>
    <td>Nadhifa</td>
    <td>Syafa</td>
  </tr>
  <tr>
    <td>Aqmar</td>
    <td>Rafiq</td>
    <td>Syaputra</td>
  </tr>
  <tr>
    <td>Akthar</td>
    <td>Farzan</td>
    <td>Wijaya</td>
  </tr>
</table>
</body>
</html>
```

Selanjutnya buka fileborder_tabel.html dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 4.2. Dari gambar tersebut dapat dilihat tabel telah memiliki *border* baik itu baris dan kolom.

Nama Depan	Nama Tengah	Nama Belakang
Adra	Nadifa	Syaifa
Aqman	Rahiq	Syaqman
Alfan	Firrom	Wijaya

Gambar 4.2 Output Border Tabel

4.3. Colspan

Colspan adalah atribut yang digunakan untuk menggabungkan beberapa kolom menjadi satu (Howe, 2014). Atribut *colspan* diletakkan dalam tag `<td>` dengan cara mengatur “*value*” kotak yang ingin digabungkan menjadi satu kolom. Untuk lebih jelas dapat dilihat pada Tabel 4.2 contoh penggunaan *colspan* pada tabel.

Tabel 4.2 Contoh Colspan Pada Tabel

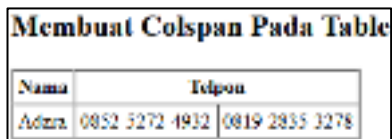
	Colspan =”2”	

Atribut *colspan* dengan *value* 2 bertujuan untuk menggabungkan dua kolom menjadi satu, sehingga dituliskan nilai atribut 2. Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama *colspan_tabel.html*.

```
<!DOCTYPEhtml>
<html>
<body>
<h2>Membuat Colspan Pada Table</h2>
<tableborder="1"cellpadding="4"cellspacing="0">
```

```
<tr>
  <th>Nama</th>
  <th colspan = "2"> Telpon</th>
</tr>
<tr>
  <td>Adzra</td>
  <td>0852-5272-4932</td>
  <td>0819-2835-3278</td>
</tr>
</table>
</body>
</html>
```

Selanjutnya buka filecolspan_tabel.html dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 4.3. Gambar tersebut dapat terlihat kolom 2 dan 3 menjadi satu kolom. Sehingga baris pertama hanya memiliki dua kolom, sementara baris ketiga memiliki tiga kolom.



Nama	Telpon	
Adzra	0852 5272 4932	0819 2835 3278

Gambar 4.3 *Output Colspan*

4.4. *Rowspan*

Rowspan adalah atribut yang digunakan untuk menggabungkan beberapa baris menjadi satu (Howe, 2014). Atribut *rowspan* diletakkan dalam tag `<td>` dengan cara mengatur “value” kotak yang ingin digabungkan menjadi satu baris. Untuk lebih jelas dapat dilihat pada Tabel 4.3 contoh penggunaan *rowspan* pada tabel.

Tabel 4.3 Contoh *Rowspan* pada Tabel

	Colspan = "2"	
Rowspan = "2"		

PEMROGRAMAN WEB DENGAN HTML

Atribut *rowspan* dengan *value* 2 bertujuan untuk menggabungkan dua baris menjadi satu, sehingga dituliskan nilai atribut 2. Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama *rowspan_tabel.html*.

```
<!DOCTYPEhtml>
<html>
<body>
  <h2>Membuat Rowspan Pada Table</h2>
  <tableborder="1"cellpadding="4"cellspacing="0">
    <tr>
      <th rowspan = "2"> Adzra</th>
      <th>Telpon 1</th>
      <th>Telpon 1</th>
    </tr>
    <tr>
      <td>0852-5272-4932</td>
      <td>0819-2835-3278</td>
    </tr>
  </table>
</body>
</html>
```

Selanjutnya buka *file* *rowspan_tabel.html* dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 4.4 dan perhatikan kolom pertama memiliki satu baris sementara kolom 2 dan 3 memiliki dua baris.



Adzra	Telpon 1	Telpon 1
	0852-5272-4932	0819-2835-3278

Gambar 4.4 *Output Rowspan*

4.5. Soal Latihan

1. Jelaskan perbedaan *tag th*, *tr*, dan *td* pada *tag* tabel?
2. Jelaskan fungsi dari *colspan* dan tuliskan kode HTML untuk membuat tabel menjadi *colspan*?
3. Jelaskan fungsi dari *rowspan* dan tuliskan kode HTML untuk membuat tabel menjadi *rowspan*?
4. Apa fungsi *cellpadding* pada *tag* tabel?
5. Apa fungsi *cellspacing* pada *tag* tabel?

BAB V

FORM

Capaian Pembelajaran

1. Mampu membuat *label*, *text input*, dan *text area*.
2. Mampu membuat *combo box*, *check box* dan *radio button*.
3. Mampu membuat tombol *submit button* dan *reset button*.

5.1. Pengertian Form

Form merupakan *interface* yang digunakan untuk menerima inputan yang berasal dari *user* dan mengolah hasil inputan tersebut ke *server* untuk dilakukan proses pengolahan data (Robbins, 2012). Dalam sebuah *website* biasanya terdapat satu atau lebih *form*, seperti *form* pencarian, registrasi dan lain sebagainya. *Form* ini biasa digunakan untuk mengumpulkan data dari pengunjung *website*.

Sebuah *form*, boleh jadi memiliki beragam kontrol, mulai dari *text input*, *combo box*, *button* dan lain sebagainya. Kita akan mempelajari sebagian kontrol yang disebutkan di atas karena kontrol di atas adalah yang paling sering digunakan dalam halaman *web*.

5.2. Label

Label digunakan untuk memberikan keterangan pada setiap *input* yang ada. Jika anda perhatikan formulir pendaftaran situs Yahoo, yang disebut dengan *label* seperti Gambar 5.1.



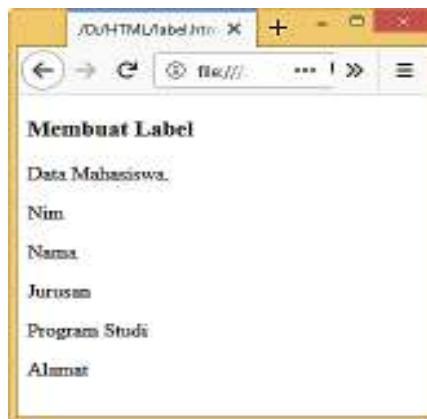
Gambar 5.1 Contoh *Label*

PEMROGRAMAN WEB DENGAN HTML

Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama `label.html`. Perhatikan atribut *for* diisi dengan isi dari atribut *name* pada kontrol yang ingin diberi *label*.

```
<!DOCTYPEhtml>
<html>
<body>
  <h3>Membuat Label</h3>
  <p>Data Mahasiswa.</p>
  <p><labelfor =,nim'> Nim </label>
  <p><labelfor =,nama'> Nama </label>
  <p><labelfor =,jurusan'> Jurusan </label>
  <p><labelfor =,program_studi'> Program Studi
</label>
  <p><labelfor =,alamat'> Alamat </label>
</body>
</html>
```

Selanjutnya buka `filelabel.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.2. Pada gambar tersebut terlihat label dari data mahasiswa sampai dengan alamat. Akan tetapi tidak terdapat inputan yang bisa dilakukan oleh *user*.



Gambar 5.2 *Output Label*

5.3. Text Input

Text input merupakan data yang akan masuk ke *server* dalam bentuk *text* yang diisi oleh *user* bisa berupa angka, huruf dan simbol (Robbins, 2012). *Text input* juga dapat diisi dengan teks yang memiliki kata yang tidak terlalu panjang atau hanya satu baris. Selain itu, *text input* biasanya digunakan dalam *form* sebagai pencarian, nama dan lain sebagainya. Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan *name* `text_input.html`.

```
<!DOCTYPEhtml>
<html>
<body>
  <h3>Membuat Text Input</h3>
  <p>Data Mahasiswa.</p>
  <p>
    <labelfor =,nim'> Nim </label>
    <inputtype=,text'name=,nim'>
  </p>
  <p>
    <labelfor =,nama'> Nama </label>
    <inputtype=,text'name=,nama'>
  </p>
  <p>
    <labelfor =,jurusan'> Jurusan </label>
    <inputtype=,text'name=,jurusan'>
  </p>
  <p>
    <labelfor =,program_studi'> Program Studi </label>
    <inputtype=,text'name=,program_studi'>
  </p>
  <p>
    <labelfor =,alamat'> Alamat </label>
    <inputtype=,text'name=,alamat'>
  </p>
</body>
</html>
```

Perhatikan *tag* `<input>` dengan atribut *type* yang memiliki nilai *text* digunakan untuk inputan yang berupa angka. sementara Selanjutnya buka `filetext_input.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.3.



Gambar 5.3 *Output Text Input*

5.4. Text Area

Textarea sama halnya dengan *text input*, namun *textarea* digunakan untuk dapat diisi yang lebih dari satu baris, cocok digunakan untuk isian yang panjang, seperti alamat, deskripsi, atau biodata. Hal ini dikarenakan *textarea* dapat diberikan inputan dengan jumlah karakter yang tidak terbatas (Howe, 2014).

Textarea berbeda dengan kontrol lainnya yang menggunakan tag `<input>`, *textarea* memiliki tag sendiri yaitu tag `<textarea>`. Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama `text_area.html`.

```
<!DOCTYPEhtml>
<html>
<body>
  <h3>Membuat Text Area</h3>
  <p>
    <labelfor =,alamat`> Alamat </label>
    <textareaname=,alamat`></textarea>
  </p>
</body>
</html>
```

Selanjutnya buka filetext_area.html dengan browser internet yang digunakan maka hasilnya seperti gambar 5.4 yang menampilkan inputan *textarea* lebih dari satu baru.



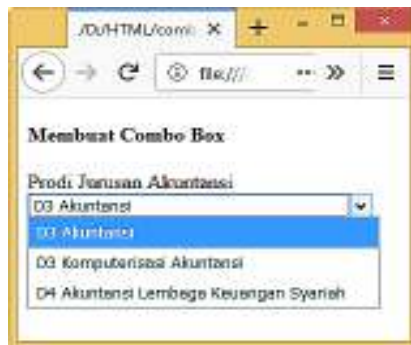
Gambar 5.4 *Output Text Area*

5.5. Combo Box

Combo box adalah kontrol yang memiliki pilihan ketika diklik menampilkan *list* menu ke bawah (*drop down list*). *Combo box* lebih efisien digunakan jika data berupa opsi yang akan dimasukkan terbatas. Proses memilih *combo box* dilakukan dengan cara mengklik *combo box* tersebut, maka akan tampil sebuah daftar teks secara keseluruhan di dalam *combo box* tersebut, lalu klik salah satu teks yang akan pilih dari *box* tersebut (Pratama, 2016). Sekarang tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan *name* *combobox.html*.

```
<!DOCTYPEhtml>
<html>
<body>
  <h4>Membuat Combo Box</h4>
  <labelfor=,prodi'>Prodi Jurusan Akuntansi</label>
  <selectname=,prodi'>
    <option>D3 Akuntansi</option>
    <option>D3 Komputerisasi Akuntansi</option>
    <option>D4 Akuntansi Lembaga Keuangan Syariah</option>
  </select>
</body>
</html>
```

Perhatikan tag `<select>` dengan atribut `name` berisi nilai prodi yang akan menampilkan daftar prodi. Untuk mengisikan daftar prodi menggunakan tag `<option>`. Semua data yang akan ditampilkan pada *list combo box* dituliskan pada tag `<option>`. Berbeda dengan tag `<input>` yang tidak memerlukan tag penutup, pada tag `<select>` ini di haruskan memakai tag penutup. Selanjutnya buka file `combobox.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.5.



Gambar 5.5 *Output Combobox*

Atribut `name` pada tag `select` berfungsi sebagai identitas dari tag `select` tersebut. Identitas ini diperlukan untuk pemberian nama *variable* penampung nilai yang dipilih oleh *user* pada saat pemrosesan di sisi *server* (misalnya menggunakan PHP). Atribut `name` diletakkan di dalam tag `select`, bukan di dalam tag `option`.

5.6. Checkbox

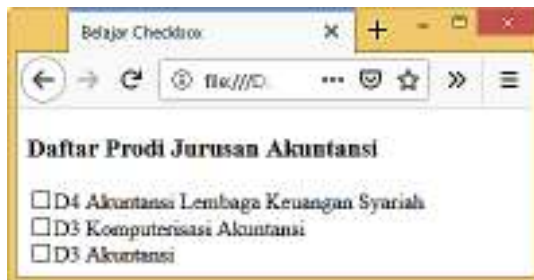
Tag `input type checkbox` berfungsi untuk membuat *checkbox* atau kotak isian yang diisi dengan cara menceklis opsi atau pilihan yang terdapat pada kotak tersebut (Howe, 2014). Dalam penggunaan *checkbox* kita bisa memilih lebih dari satu opsi atau pilihan. Contoh penggunaan input *checkbox* yang sering kita jumpai adalah persetujuan saat melakukan proses pendaftaran atau registrasi pada sebuah *website*. Jika pendaftar

menyetujui persyaratan dari *website* tempat mendaftar maka dinyatakan dengan men-*check* persetujuan.

Untuk membuat *checkbox* di HTML itu tidaklah rumit, kita hanya memerlukan *tag* `<input>` dengan atribut `type="Checkbox"`. Untuk lebih jelas, tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan *name* `checkbox.html`. Pada contoh ini *checkbox* digunakan untuk memilih program studi Jurusan Akuntansi Politeknik Negeri Banjarmasin.

```
<!DOCTYPEhtml>
<html>
<head><title>Belajar Checkbox</title></head>
<body>
  <h3>Daftar Prodi Jurusan Akuntansi </h3>
  <inputtype="checkbox"name="prodi"value="ALKS">D4
  Akuntansi Lembaga Keuangan Syariah<br>
  <inputtype="checkbox"name="prodi"value="Kompak">D3
  Komputerisasi Akuntansi<br>
  <inputtype="checkbox"name="prodi"value="Akuntansi">D3
  Akuntansi <br>
</body>
</html>
```

Selanjutnya buka *file* `checkbox.html` dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.6 yang menampilkan daftar pilihan program studi jurusan akuntansi.



Gambar 5.6 Output Checkbox

5.7. Radio Button

Tag input type radio memiliki fungsi untuk membuat tombol radio atau tombol pilihan. Berbeda dengan *Input type Checkbox*, pada *tag input type radio* ini *user* hanya bisa memilih salah satu dari beberapa opsi atau pilihan yang tersedia dalam satu grup *tag input type radio*. Ketika kita memilih satu opsi (mencentang) dalam satu grup *input type radio*, maka otomatis opsi yang lain akan tidak dipilih (otomatis tidak tercentang). *Input type Radio* ini biasanya digunakan untuk pilihan yang membatasi *user* agar memilih satu dari pilihan yang ada (Hadi, 2014).

Tag input type radio ini merupakan *tag* yang berdiri sendiri dan tidak membutuhkan penutup *tag*. Contoh penggunaan *input type radio* di dalam suatu *website* biasanya digunakan untuk menampilkan pilihan jenis kelamin, karena tidak mungkin *user* memasukkan dua jenis kelamin untuk disimpan ke *database*. Untuk lebih jelas, tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama *radio_button.html*.

```
html>
<head><title>Membuat Radio Button</title></head>
<body>
  <h3>Daftar Prodi Jurusan Akuntansi</h3>
  <inputtype="radio"name="prodi"value="ALKS">D4 Akuntansi
  Lembaga Keuangan Syariah<br>
  <inputtype="radio"name="prodi"value="Kompak">D3
  Komputerisasi Akuntansi<br>
  <inputtype="radio"name="prodi"value="Akuntansi">D3
  Akuntansi <br>
</body>
</html>
```

Selanjutnya buka *file* *radio_button.html* dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.7 yang menampilkan pilihan *radio button* untuk menentukan program studi yang ada di Jurusan Akuntansi.



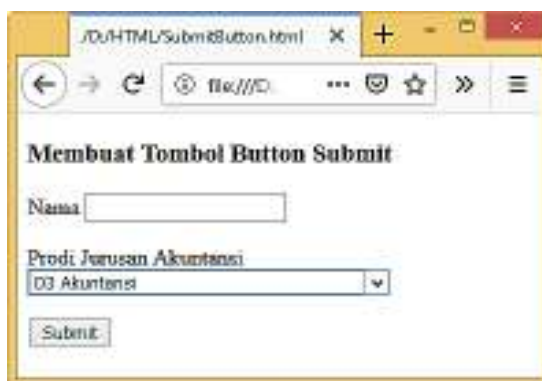
Gambar 5.7 Output Radio Button

5.8. Submit Button

Apa fungsi input tipe *submit* pada *form*? *Input Type Submit* digunakan untuk memulai proses pengolahan data. Pengolahan ini dapat berupa masukan, perubahan, atau penghapusan. Semua aksi tersebut diatur pada *type submit* ini. *Submit button*, berupa tombol yang dapat di klik. Penggunaan atribut *value* pada kontrol ini akan mengganti *text* yang ada di dalamnya (Howe, 2014). Untuk lebih jelas, tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan nama *submit_button.html*.

```
<!DOCTYPEhtml>
<html>
<body>
<h3>Membuat Tombol Button Submit</h3>
<formaction=""method="get">
<p><labelfor =,nama`> Nama
</label><inputtype=,text `name=,nama `/></p>
<labelfor=,prodi `>Prodi Jurusan Akuntansi</label>
<selectname=,prodi `
<option>D3 Akuntansi</option>
<option>D3 Komputerisasi Akuntansi</option>
<option>D4 Akuntansi Lembaga
Keuangan Syariah</option></select>
<p><buttontype="submit">Submit</button>
</p>
</form>
</p>
</body>
</html>
```


Selanjutnya buka *file* submit_button.html dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.8 yang menampilkan *menu submit* yang fungsi untuk menyimpan data ke *database*.



Gambar 5.8 Output Submit Button

5.9. Reset Button

Fungsi *Reset* adalah untuk mengosongkan kembali semua isi *form*. Biasanya tombol ini dipasang dengan tombol *submit* pada sebuah halaman HTML (Howe, 2014). Untuk lebih jelas, tuliskan kode HTML di bawah ini dan simpan *file* tersebut dengan *name* reset_button.html.

```
<!DOCTYPEhtml>
<html>
<body>
  <h3>Membuat Tombol Button Reset</h3>
  <formaction=""method="get">
    <p>
      <labelfor =,nama`> Nama </label>
      <inputtype=,text`name=,nama`/>
    </p>
    <labelfor=,prodi`>Prodi Jurusan Akuntansi</label>
    <selectname=,prodi`>
      <option>D3 Akuntansi</option>
      <option>D3 Komputerisasi Akuntansi</option>
    </select>
  </form>
</body>
</html>
```

```
<option>D4 Akuntansi Lembaga Keuangan  
Syariah</option>  
</select>  
<p><button type="submit">Submit</button>  
<button type="reset">Reset</button></p>  
</form>  
</p>  
</body>  
</html>
```

Selanjutnya buka filereset_button.html dengan *browser* internet yang digunakan maka hasilnya seperti Gambar 5.9 di bawah ini.



Gambar 5.9 Output Reset Button

5.10. Soal Latihan

1. Apa fungsi *form* pada halaman *web*?
2. Jelaskan perbedaan *text input* dengan *text area*?
3. Tuliskan kode HTML untuk membuat *text input*?
4. Tuliskan kode HTML untuk membuat *combobox* dan *checkbox*?
5. Jelaskan perbedaan *submit button* dengan *reset button*?
6. Jelaskan atribut apa saja yang digunakan untuk mengatur ukuran gambar?

DAFTAR PUSTAKA

- Ariona, R. (2013). *Belajar HTML dan CSS: Tutorial Fundamental dalam mempelajari HTML & CSS*.
- Duckett, J. (2011). *HTML & CSS design and build websites*. Canada: John Wiley & Sons.
- Hadi, D. A. (2014). *Belajar HTML & CSS Dasar*.
- Howe, S. (2014). *Learn to Code HTML & CSS: Develop & Style Websites*. Pearson Education.
- Powell, T. A. (2010). *HTML & CSS: The Complete Reference Fifth Edition* (Fifth Edit). New York: Mc Graw Hill.
- Pratama, Ha. (2016). *HTML Uncover: Panduan Belajar HTML untuk Pemula*. Duniaikom.
- Robbins, J. N. (2012). *Learning Web Design: A Beginner's Guide To HTML, CSS, JavaScript And Web Graphics* (Fourth Edi). O'REILLY.
- Wempen, F. (2011). *Step By Step HTML5*. Octal Publishing.

TENTANG PENULIS



Heru K. Candra lahir di Tanjung Pinang tanggal 13 November 1970. Kulian S1 di Institut Teknologi Palapa Malang Jurusan Statistika dan melanjutkan S2 di Institut Teknologi Sepuluh November Surabaya jurusan Teknik Informatika. Tahun 1993 memulai karier dosen di Akademik Manajemen Informatika Komputer Banjarmasin dan STMIK Indonesia Banjarmasin sampai tahun 2008. Mata kuliah yang diajarkan yaitu mata kuliah komputer dan Matematika. Tahun 2008 sampai sekarang mengajar di Politeknik Negeri Banjarmasin pada Jurusan Akuntansi Prodi D3 Komputerisasi Akuntansi.

PEMROGRAMAN WEB dengan HTML

HERU KARTIKA CANDRA

Web *browser* merupakan perangkat utama yang digunakan untuk menampilkan halaman web. Pada dasarnya halaman web dibuat dengan menggunakan HTML dan CSS. Setiap *browser* memiliki perbedaan dalam hal menampilkan halaman web dan fitur-fitur yang didukung dalam HTML dan CSS (Ariona, 2013).

Internet merupakan suatu jaringan global yang berfungsi untuk menghubungkan suatu *network* dengan *network* lainnya di seluruh dunia (Wempen, 2011). TCP/IP (*Transmission Control Protocol/ internet protocol*) menjadi protokol penghubung antara jaringan-jaringan yang beragam di seluruh dunia untuk dapat berkomunikasi. WWW (*World Wide Web*) merupakan bagian dari internet yang paling cepat berkembang dan paling populer.



Penerbit: Politeknik Negeri
Redaksi :
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry
Pangeran, Koms. Kampus ULM, Banjarmasin, Utara
Telp. : 05113305002
Email : info@pnbn.ac.id

ISBN 978-602-786237-2

