# BAB 1

# HTML5

Pada bab ini akan dibahas mengenai perkembangan dalam industri website dan mengapa HTML5 sangat penting digunakana dalam pengembangan *website*. Setelah mempelajari bab in maka pembaca diharapkan dapat:

1. Mengerti sejarah dan pengertian HTML5
2. Memahami struktur HTML5
3. Dapat menyusun struktur dasar syntax HTML5 dalam pengembangan website

## 1.1 Pengenalan HTML5

HTML5 merupakan versi terakhir dari HTML yang menjadi bahasa markup paling banyak digunakan dalam pengembangan suatu website. HTML5 merupakan pengembangan dari HTML4 dan XHTML versi 1.0. HTML5 sendiri memiliki keunggulan seperti dapat berjalan di semua browser, memiliki kemampuan handling error yang lebih baik dan mempunyai tag elemen yang lebih semantik seperti article, section, header dan footer.

Sejarah pengembangan *website* dimulai pada tahun 90an, dimana sebuah *website* yang memiliki gambar dan desain yang menarik menjadi pilihan utama dalam pengembangan suatu *website*. Sekarang ini, prioritas utama dalam pengembangan web sudah berubah. Pengembangan *website* mulai fokus pada performa dan pengembangan fungsi. Sebuah *website* yang memiliki fungsi khusus menjadi standar utama dalam pengembangan *website*. Bahkan, sebuah website kadang merupakan sebuah program yang memiliki fungsi dan tujuan tersendiri (*standalone applications*).

*Hyper Text Markup Language* (HTML) merupakan bahasa markup yang digunakan untuk membuat sebuah halaman web dan menampilkan informasi didalamnya. Pada mulanya bahasa markup dimulai dari HTML4 kemudian berkembang menjadi XHTML. Walaupun pada dasarnya XHTML hanyalah HTML4 yang diberi tambahan bahasa XML. Singkat cerita versi HTML ini kemudian berkembang lagi menjadi HTML5.

HTML5 sendiri dikembangkan oleh dua organisasi yaitu *Web Hypertext Application Technology Working Group* (WHATWG ) dan *World Wide Web Consortium* (W3C) sehingga menjadikan HTML5 memiliki dua spesifikasi. HTML5 yang dikembangkan oleh W3C berperan sebagai host untuk standar web yang lain seperti SVG (*Scalable Vector Graphics*) dan WCAG (*Web Content Accessibility Guidelines*). W3C sendiri merupakan organisasi nirlaba yang melakukan pengawasan terhadap standar kode HTML dan CSS. Versi HTML5 dari W3C dapat diakses melalui [https://dev.w3.org/html5](https://dev.w3.org/html5/) .

Versi HTML5 kedua yang dikembangkan oleh WHATWG, sebuah group yang berisikan mantan karyawan dari Apple, Mozilla dan Opera yang menganggap W3C tidak memahami kebutuhan dari user dengan malah fokus pada standarisasi HTML daripada untuk mengembangkan XHTML 2.0. Para mantan karyawan ini kemudian memilih untuk mengembangkan sendiri spesifikasi yang dibutuhkan untuk pengembangan HTML5. Versi HTML5 dari WHATWG ini dapat diakses melalui [https://html.spec.whatwg.org/multipage](https://html.spec.whatwg.org/multipage/) .

Perbedaan dari kedua HTML5 ini terletak pada penamaan versinya, WHATWG mengenalkan istilah “*living standard”* dimana pengembangan HTML oleh WHATWG dilakukan secara terus menerus tanpa harus menambahkan nomor versi pengembangan. Sedangkan pengembangan oleh W3C nomor versi HTML akan bertambah seiring dengan versi pengembangannya. Selain itu kode pada HTML5 versi WHATWG lebih terlihat informal dan experimental daripada versi W3C. Terlepas dari perbedaan-perbedaan yang ada, kedua kode HTML5 baik versi WHATWG dan W3C lebih banyak memiliki persamaan sehingga tidak akan menyulitkan dalam proses pembelajaran.

## 1.2 HTML5 Herald

Pada sub bab kedua ini kita akan memulai pembahasan tentang syntax pada HTML5, struktur yang digunakan, element-element HTML5 dan bagaimana cara menggunakannya dalam layout sebuah website. Pada akhir bab ini kita akan membuat struktur HTL5 dari sebuah website. Hasil akhir dari halaman web yang akan dibuat seperti pada gambar 1.1



Gambar 1.1 HTML5 Herald

Halaman web yang akan dibuat berbentuk sebuah layout suatu koran yang disebut HTML5 Herald. Pada halaman ini terdapat beberapa media seperti video, gambar, artikel dan iklan. Pembuatan web ini akan difokuskan pada elemen-elemen baru yang ada pada HTML5. Hasil akhir dari web ini dapat diakses melalui [https://www.thehtml5herald.com](https://www.thehtml5herald.com/).

Pengembangan HTML5 dimulai dengan membuat sebuah *boilerplate* yaitu sebuah struktur kode utama yang dapat digunakan untuk memulai proyek dari nol. *Boilerplate* ini akan sangat berguna jika kita memiliki banyak proyek berbasis HTML5, kita dapat menggunakan *boilerplate* sebagai struktur kode dasar dalam pengembangan proyek website.

Struktur dasar kode HTML5 dapat dilihat pada gambar 1.2 :

1. <!DOCTYPE html**>**
2. **<html** lang="en"**>**
3. **<head>**
4. **<meta** charset="utf-8"**>**
5. **<title>**The HTML5 Herald**</title>**
6. **<meta** name="description" content="The HTML5 Herald"**>**
7. **<meta** name="author" content="SitePoint"**>**
8. **<link** rel="stylesheet" href="css/styles.css"**>**
9. **</head>**
10. **<body>**
11. **<script** src="js/scripts.js"**></script>**
12. **</body>**
13. **</html>**

Gambar 1.2 Template dasar HTML5

Selanjutnya kita akan membahas apa yang membedakan HTML5 dengan versi HTML sebelumnya.

## 1.3 Doctype

Pada baris pertama terdapat *Document Type Declaration* atau **doctype**. **Doctype** digunakan untuk memberitahu kepada kepada browser tentang isi dokumen yang ada pada halaman tersebut. **Doctype** ini harus selalu ditulis pada awal baris setiap dokumen HTML. Beberapa tahun yang lalu, deklarasi **doctype** sangat susah untuk dihafal. Pada XHTML versi 1.0 penulisan **doctype** seperti berikut :

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"

Dan berkembang menjadi seperti berikut pada versi HTML4

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"**>**

Penulisan kode yang sulit ini akhirnya diperbaiki pada versi HTML5 menjadi seperti berikut

<!DOCTYPE html>

Versi HTML5 lebih mudah dibaca dan dipahami. Dalam penulisannya, **doctype** dapat ditulis menggunakan huruf kecil, huruf besar atau campuran keduanya. Pada penulisan versi HTML5 ini, tidak ada angka 5 dibelakang tulisan html karena setiap browser yang ada di pasaran diharuskan bisa membaca semua konten dari web maka tidak perlu penulisan versi html lagi pada **doctype**.

## 1.4 Html

Penulisan tag HTML tidak mengalami perubahan pada versi HTML5. Pada contoh di Gambar 1.3 penulisan html dilanjutkan dengan penulisan attribute lang dengan nilai en. Ini berarti isi dari html tersebut menggunakan bahasa Inggris. Pada penulisan XHTML versi terdahulu, diperlukan attribute xmlns. Sedang pada HTML5 syntax ini tidak perlu digunakan lagi

1. <!DOCTYPE html**>**
2. **<html** lang="en"**>**
3. **</html>**

Gambar 1.3 Syntax HTML

## 1.5 Head

Bagian selanjutnya dari penulisan kode html adalah pada bagian head. Baris pertama pada bagian head merupakan baris yang menyatakan character encoding yang digunakan. Pada versi sebelumnya penulisan character encoding seperti berikut

**<meta** http-equiv="Content-Type" content="text/html; charset=utf-8"

Kemudian dipermudah mulai versi HTML5 menjadi

**<meta** charset="utf-8"**>**

Utf-8 merupakan nilai character encoding yang digunakan. Apa itu utf-8 tidak akan dibahas lebih lanjut pada buku ini. Bagian selanjutnya dari isi head adalah title dan meta name seperti yang ditunjukkan pada Gambar 1.3.

1. **<title>**The HTML5 Herald**</title>**
2. **<meta** name="description" content="The HTML5 Herald"**>**
3. **<meta** name="author" content="SitePoint"**>**
4. **<link** rel="stylesheet" href="css/styles.css"**>**

Gambar 1.3 Head HTML5

Baris pertama adalah title dari sebuah halaman web. Penulisan baris in harus ada dalam setiap head pada file HTML. Baris selanjutnya adalah meta element yang menjelaskan tentang deskripsi dan author dari halaman web tersebut. Tidak ada batasan dalam penulisan meta element ini. Baris keempat yaitu penulisan *stylesheet* atau file css yang digunakan dalam proyek ini. Karena menggunakan file css yang terpisah dengan file html maka penulisannya menggunakan syntax link. Rel menunjukkan tipe file yang akan ditunjuk sedangkan href menunjukkan lokasi dari file tersebut.

## 1.5 Body

Setelah penulisan kode pada head, maka kita akan melanjutkan pada penulisan kode selanjutnya yaitu body. Semua kode html yang akan ditampilkan pada layar harus ditulis diantara tag <body> dan </body>. Selain itu penulisan kode *javascript* bisa dilakukan pada bagian ini. Pada contoh kode template diatas penulisan kode javascript menggunakan format HTML5 yang lebih singkat. Penulisan referensi ke kode javacript pada versi HTML sebelumnya seperti :

**<script** src="js/scripts.js" type="text/javascript"**></script>**

Pada versi HTML5 dipersingkat menjadi :

**<script** src="js/scripts.js"**></script>**

Penulisan referensi javascript diletakkan pada bagian bawah sebelum tag penutup </body>. Hal ini dilakukan untuk mempercepat loading suatu website. Jika suatu browser membaca suatu *javascript* pada awal halaman maka browser akan menghentikan proses rendering halaman website dan menunggu file *javascript* tersebut selesai di baca. Itulah kenapa sebagian besar script diletakkan pada bagian paling bawah di struktur dokumen HTML.

## 1.6 Struktur Halaman HTML5

Struktur halaman web yang akan kita buat mencakup :

1. Header dengan logo dan judul
2. Menu navigasi
3. Body yang terdiri dari 3 kolom
4. Artikel dan blok iklan
5. Footer

Sebelum mengaplikasikan semua struktur tersebut maka kita akan belajar tag html yang baru diperkenalkan di HTML5.

* + 1. Header element

Header element dapat digunakan sebagai judul atau alat bantu navigasi yang menunjuk secara spesifik ke halaman html tertentu. Header element biasa diletakkan pada bagian atas halaman website. Walaupun begitu peletakan header dapat diletakkan secara independent sesuai dengan layout pada halaman website tersebut.

* + 1. Section element

Element kedua yang baru diperkenalkan pada HTML5 yaitu *section*. Section memiliki arti yaitu element yang merepresentasikan bagian dari suatu aplikasi atau dokumen. Pada penulisan versi HTML sebelumnya untuk menandai section baru kita menggunakan element div. Hal ini mulai ditinggalkan sejak HTML5 karena tidak memiliki makna. Penulisan section dapat ditulis secara bersarang dengan section yang lain. Sebagai contoh section untuk situs berita online, didalamnya terdapat section untuk masing-masing kategori berita.

Beberapa penggunaan element section yang dianjurkan yaitu untuk:

1. Tab interface
2. Segment untuk halaman tertentu seperti about page
3. Segment untuk artikel, term of service page.
   * 1. Article element

Elemen article memiliki fungsi yang mirip dengan elemen section. Perbedaannya dimana section bisa digunakan untuk semua konten sedangkan article hanya khusus untuk satu konten khusus. Seperti pada element section, penulisan element article juga dapat dilakukan secara bersarang.

Beberapa bagian yang dianjurkan untuk penggunaan article element adalah:

1. Post forum
2. Artikel dalam majalah atau koran
3. Post blog
4. Komentar
   * 1. Nav Element

Sesuai dengan namanya element nav dikhususkan untuk pengaturan navigasi dan link. Walaupun element nav biasa digunakan untuk menunjukkan list dari menu atau link, nav juga dapat digunakan untuk menunjukkan suatu paragraf dalam sebuah teks. Nav dapat digunakan lebih dari sekali dalam suatu halaman web.

Penggunaan nav yang dianjurkan antara lain:

1. Menu utama dalam web
2. Semua link dalam web
3. Link dalam Footer
   * 1. Aside Element

Elemen aside digunakan untuk merepresentasikan bagian dari halaman yang merupakan penunjang dari konten utama. Element aside dapat dianjurkan untuk:

1. Bagian spesifik dari konten utama
2. Sidebar
3. Kumpulan link

Elemen aside tidak boleh digunakan untuk sebagai bagian utama dari web. Elemen aside dapat berdiri sendiri namun harus menjadi bagian dari suatu website tersebut.

* + 1. Footer element

Element terakhir yaitu footer. Seperti halnya dengan header, footer dapat digunakan beberapa kali dalam satu halaman. Pada versi HTML sebelumnya, footer dapat ditulis menggunakan div , namun sejak versi HTML5 dapat langsung ditulis menggunakan footer.

Footer biasanya berisi informasi copyright, link-link yang berkaitan dengan web utama dan informasi penulis. Seperti halnya aside dan header, element footer tidak ditentukan dengan posisi di halaman. Jadi footer tidak harus diletakkan pada bagian bawah halaman. Sebagai contoh pada suatu artikel web, elemen footer dapat diletakkan secara langsung di bawah judul artikel.

## 1.7 Struktur halaman HTML5 Herald

Setelah kita mengetahui dasar struktur halaman website secara umum, sekarang saatnya menerapkan hal tersebut pada web yang akan dibuat.

Pada halaman pertama bagian atas kita akan menggunakan elemen header. Pada elemen header akan berisi judul dan logo dari herald. Pada bagian header dapat ditambahkan elemen nav sebagai navigasi.

Dibawah bagian header, konten utama website kita dibagai menjadi tiga kolom. Untuk itu maka kita akan menggunakan elemen div untuk setiap kolom. Alasan tidak menggunakan section pada halaman ini dikarenakan tiga kolom yang dibuat hanya untuk kepentingan visual saja, jika tiga kolom tersebut memilliki informasi yang berbeda-beda seperti artikel untuk olahraga, artikel untuk entartainment dan artikel untuk news maka kita boleh menggunakan section.

Didalam div tiap kolom tersebut, akan diisi oleh artikel berita. Maka elemen yang paling cocok digunakan adalah article element.

Kolom yang berada paling kanan, berisi iklan-iklan yang berhubungan dengan artikel utama maka element yang paling cocok digunakan adalah aside. Dibawah iklan tersebut terdapat satu bagian halaman yang berisi artikel, maka element yang paling cocok digunakan adalah article.

## 1.8 Main element

Walaupun bukan termasuk dalam elemen HTML5, main element merupakan elemen yang biasa digunakan oleh pengembang untuk menunjukkan konten utama dari suatu website. Beberapa pengembang menulis main element seperti berikut :

1. **<body>**
2. **<header>**
3. ...
4. **</header>**
5. **<div** id="main"**>**
6. ...
7. **</div>**
8. **<footer>**
9. **</footer>**
10. **</body>**

Gambar 1.4 main element

Kelemahan dari cara ini yaitu element div yang digunakan akan sejajar dengan element header dan footer. Untuk mengatasi hal tersebut maka beberapa pengembang menambah atribut role pada pada div tersebut.

1. **<div** id="main" role="main"**>**
2. .............
3. **</div>**

Gambar 1.5 attribut Role

Menurut W3C main element merupakan sebuah elemen yang digunakan untuk menunjukkan konten utama dari suatu body dokumen atau aplikasi. Penulisan elemen main hanya boleh dilakukan satu kali pada halaman HTML tidak seperti element yang lain. Selain itu penggunaan role=”main” sangat disarankan dalam pengembangan website modern.

Gambar 1.6 berikut merupakan struktur akhir dari website herald yang akan kita buat.

1. **<body>**
2. **<header>**
3. ...
4. **</header>**
5. **<main** role="main"**>**
6. **</main>**
7. **<footer>**
8. ...
9. **</footer>**
10. **<script** src="js/scripts.js"**></script>**
11. **</body>**

Gambar 1.6 Struktur kode HTML

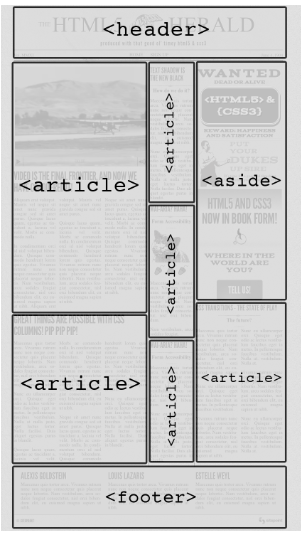
## 1.9 Struktur Section

Bagian terakhir dari layout yang akan kita buat yaitu element footer. Footer yang akan dibuat berada pada bagian paling bawah halaman. Karena footer terdiri dari beberapa bagian dimana setiap bagian memiliki isi yang berbeda maka footer tersebut akan dibagi menjadi beberapa section. Hasil akhir struktur dari website yang akan kita buat menjadi seperti berikut:

1. **<body>**
2. **<header>**
3. **<nav></nav>**
4. **</header>**
5. **<main** role="main"**>**
6. **<div** class="primary"**>**
7. **<article></article>**
8. **</div>**
9. **<div** class="secondary"**>**
10. **<article></article>**
11. **</div>**
12. **<div** class="tertiary"**>**
13. **<aside>**
14. **<article></article>**
15. **</aside>**
16. **<article>**
17. **</article>**
18. **</div>**
19. **</main>**<!-- main -->
20. **<footer>**
21. **<section** id="authors"**>**
22. **<section></section>**
23. **</section>**
24. **<section** id="copyright"**>**
25. **</section>**
26. **</footer>**
27. **<script** src="js/scripts.js"**></script>**
28. **</body>**

Gambar 1.7 Struktur akhir HTML5

Struktur kode pada Gambar 1.7 dapat direpresentasikan dalam Gambar 1.8 sebagai berikut.



Gambar 1.8 Layout website yang akan dibuat