



JUNIOR WEB DEVELOPER



Agenda

- UI
- HTML
- Javascript
- CSS
- Bootstrap
- PHP Fundamental

Pendahuluan

Web Developer adalah seorang yang membuat aplikasi berbasis web dan harus mempunyai kemampuan di server side scripting, client side scripting, dan database.



Aplikasi untuk Web Developer

Text
Editor



Local Development



Laragon

Dependency Manager



UI / UX Design

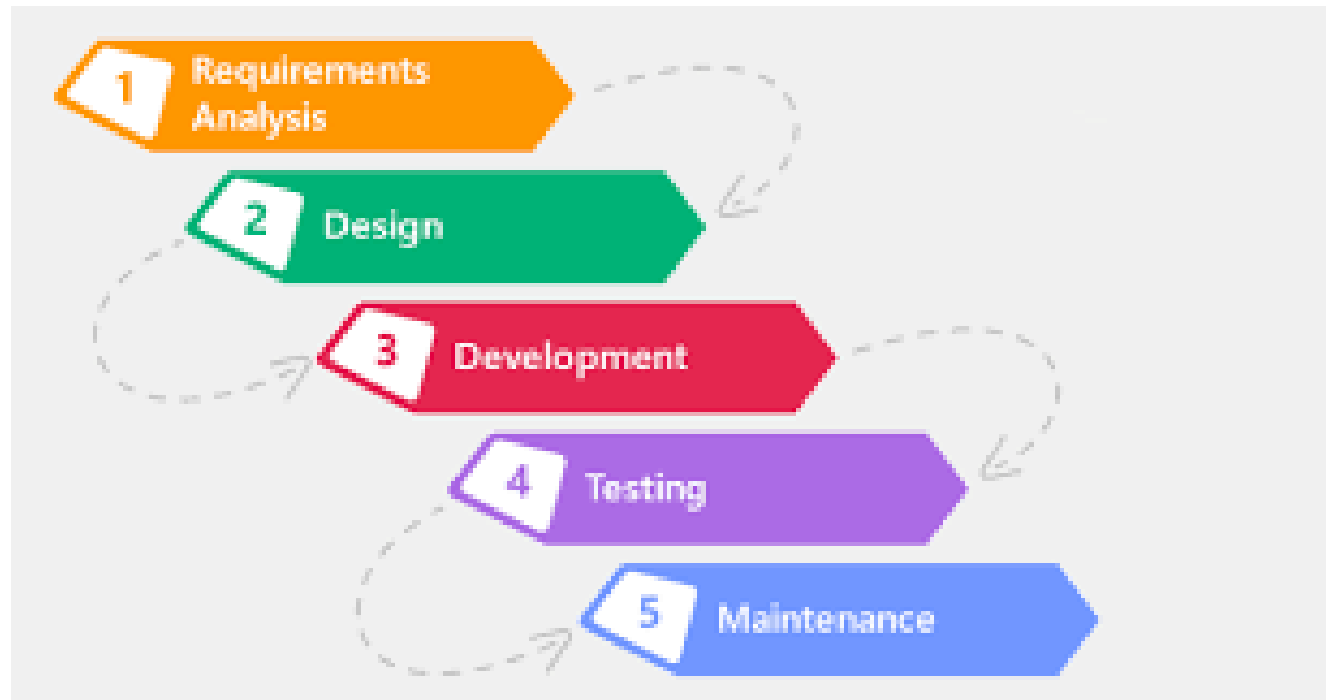


SDLC (Systems development life cycle)

siklus hidup pengembangan sistem.

SDLC adalah tahapan kerja yang bertujuan untuk menghasilkan sistem berkualitas tinggi yang sesuai dengan keinginan pelanggan atau tujuan dibuatnya sistem tersebut. SDLC menjadi kerangka yang berisi langkah-langkah yang harus dilakukan untuk memproses pengembangan suatu perangkat lunak.

LANGKAH SDLC



<https://www.nesabamedia.com/sdlc-systems-development-life-cycle/>



UI Design

Apa itu UI/UX Design?

- UI adalah desain antarmuka untuk produk atau aplikasi dengan fokus pada memaksimalkan pengalaman user
- UX adalah proses mendesain suatu produk atau aplikasi melalui pendekatan user untuk menciptakan produk yang efisien, nyaman, dan mudah digunakan
- UX design ditemukan oleh Don Norman sekitar 1993-1997

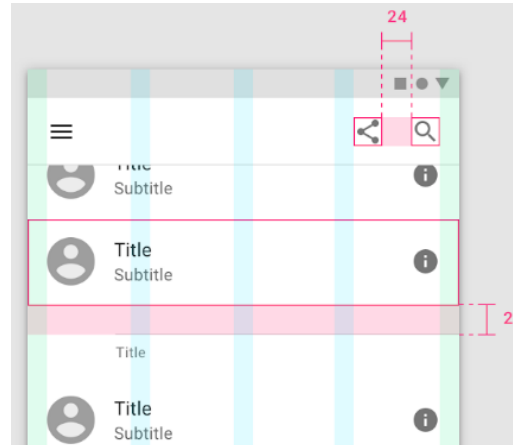
UI vs UX

UI / UX



Aturan Desain UI

- Tidak terlalu banyak menggunakan warna
- Jenis huruf/font yang digunakan maksimal 3 jenis font
- Gunakan warna dan kontras yang benar
- Usahakan desain UI user friendly
- Gunakan jarak/spacing antar komponen sama atau kelipatan



Filter

Export

Add Customer

Jenis UI Desain

WIREFRAME

PROTOTYPE

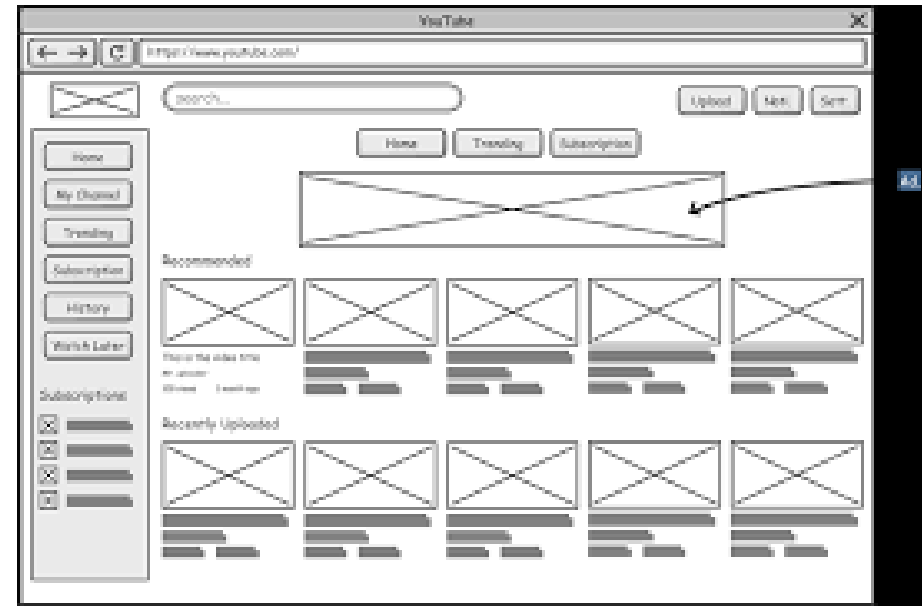
MOCK UP

WIREFRAME

Merupakan *blueprint* dari sebuah arsitektur.

Menyampaikan susunan, layout, navigasi, struktur, dan organisasi konten.

Bergambar hitam putih saja dan menekankan pada isi dari konten yang akan dibuat.



<https://medium.com/fbdevclagos/4-reasons-why-wire-frame-is-important-during-website-or-mobile-app-development-46fabdf47190>

MOCKUP

Mockup adalah konsep yang lebih dalam untuk penyampaian berbagai aspek desain visual, seperti gambar, warna, dan tipografi.

Waktu pengerjaan mockup sama dengan wireframe, yaitu sebelum produk itu dibuat.

Hanya saja gambar yang diberikan mockup lebih deta

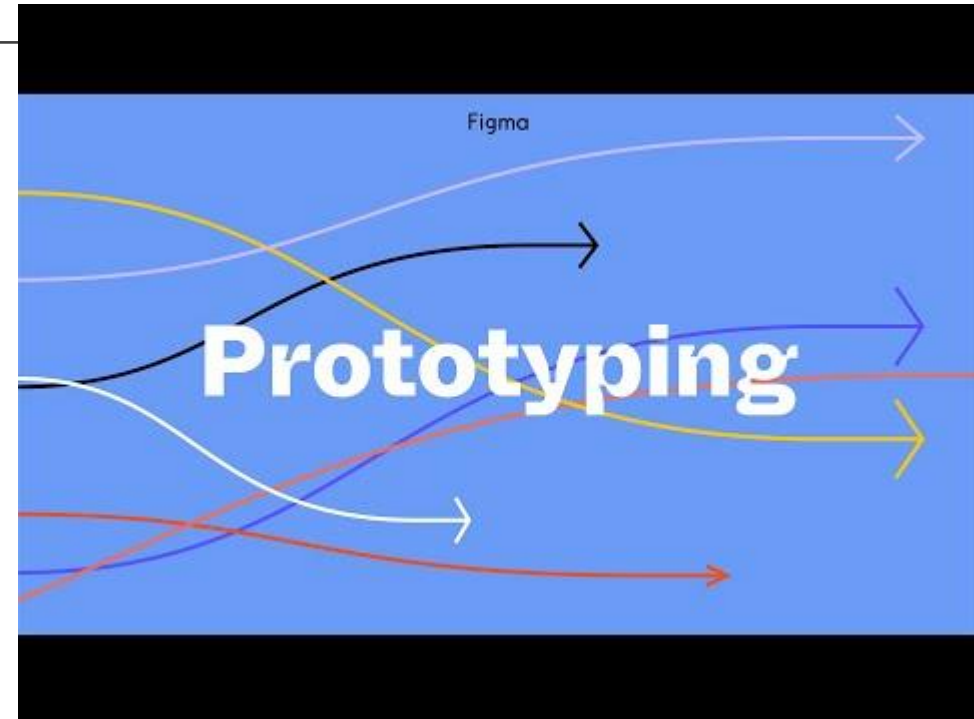


PROTOTYPE

Konsep *clickable* yang sudah dapat merespons berbagai perintah.

Mensimulasikan bagaimana user dapat berinteraksi secara nyata dengan user interface.

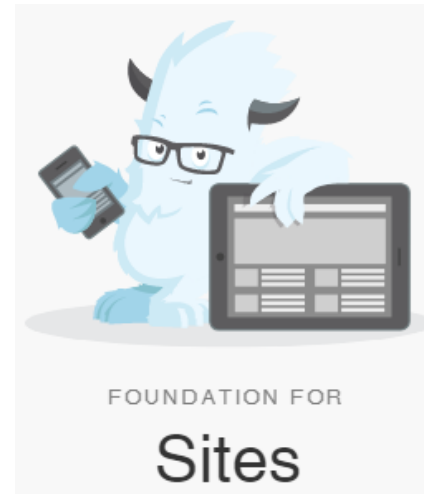
Memungkinkan desainer untuk menguji user journey dan menemukan berbagai masalah yang ada pada produk.



Framework UI



Tailwind CSS



PRAKTEK : Install Aplikasi Pencil

<https://pencil.evolus.vn/>



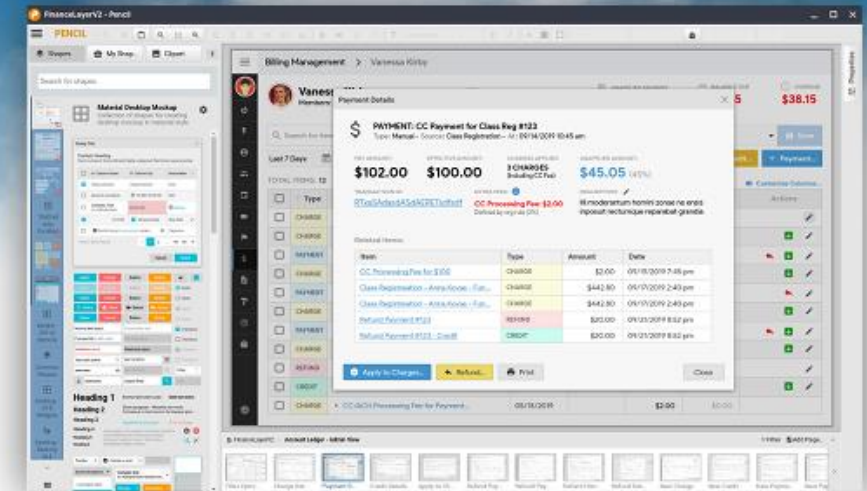
An open-source GUI prototyping tool that's available for ALL platforms.

Pencil is built for the purpose of providing a free and open-source GUI prototyping tool that people can easily install and use to create mockups in popular desktop platforms.

The latest stable version of Pencil is **3.1.0** which contains stability fixes and many new features. More details can be found in the [release notes](#).

Download for Windows x64
Version 3.1.0 ga, x64 .exe, ~83 MB

For other platforms?
[See all downloads »](#)





designed by  freepik

HTML Dasar

YANG AKAN DIBAHAS

1. Pengenalan Dasar HTML untuk Pemula
2. Mengenal Tag, Elemen, dan Atribut dalam HTML
3. Membuat Paragraf di HTML
4. Membuat Heading di HTML
5. Cara Membuat Komentar
6. Text Formatting di HTML
7. Membuat Link pada HTML
8. Menampilkan Gambar di HTML
9. Membuat List pada HTML
10. Membuat Tabel pada HTML
11. Membuat Form pada HTML

<https://www.petanikode.com/tutorial/html/>

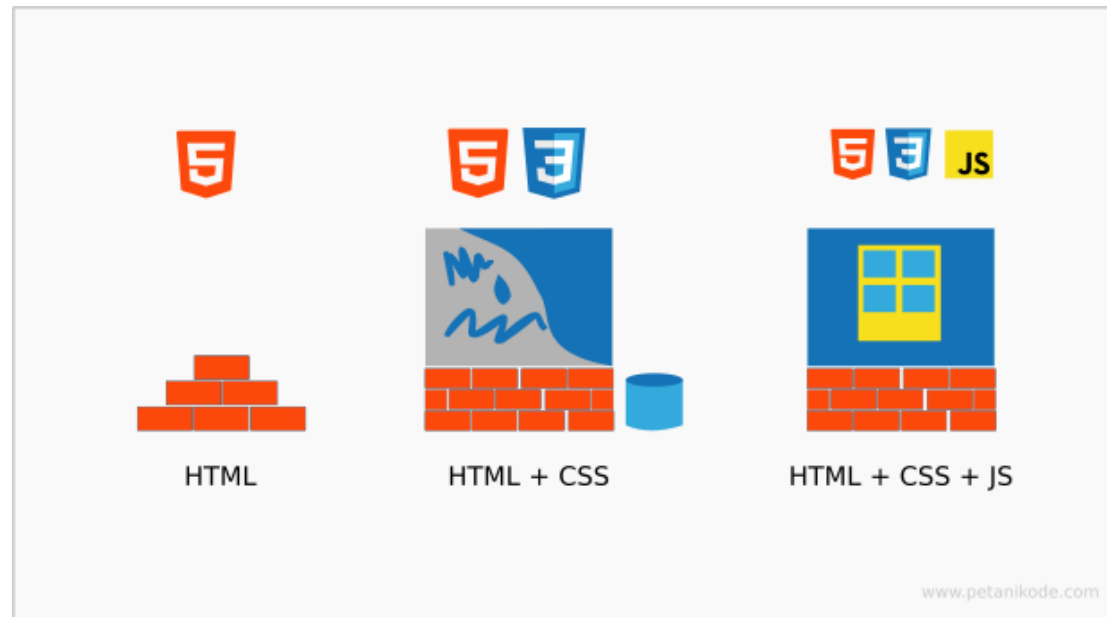
Apa itu HTML ?

- HTML (HyperText Markup Language) adalah suatu bahasa yang menggunakan tanda-tanda tertentu (tag) untuk menyatakan kode-kode yang harus ditafsirkan oleh browser agar halaman tersebut dapat ditampilkan secara benar.
- Secara umum, fungsi HTML adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di Internet melalui layanan web.

Peran HTML

HTML berperan untuk menentukan struktur konten dan tampilan dari sebuah web.

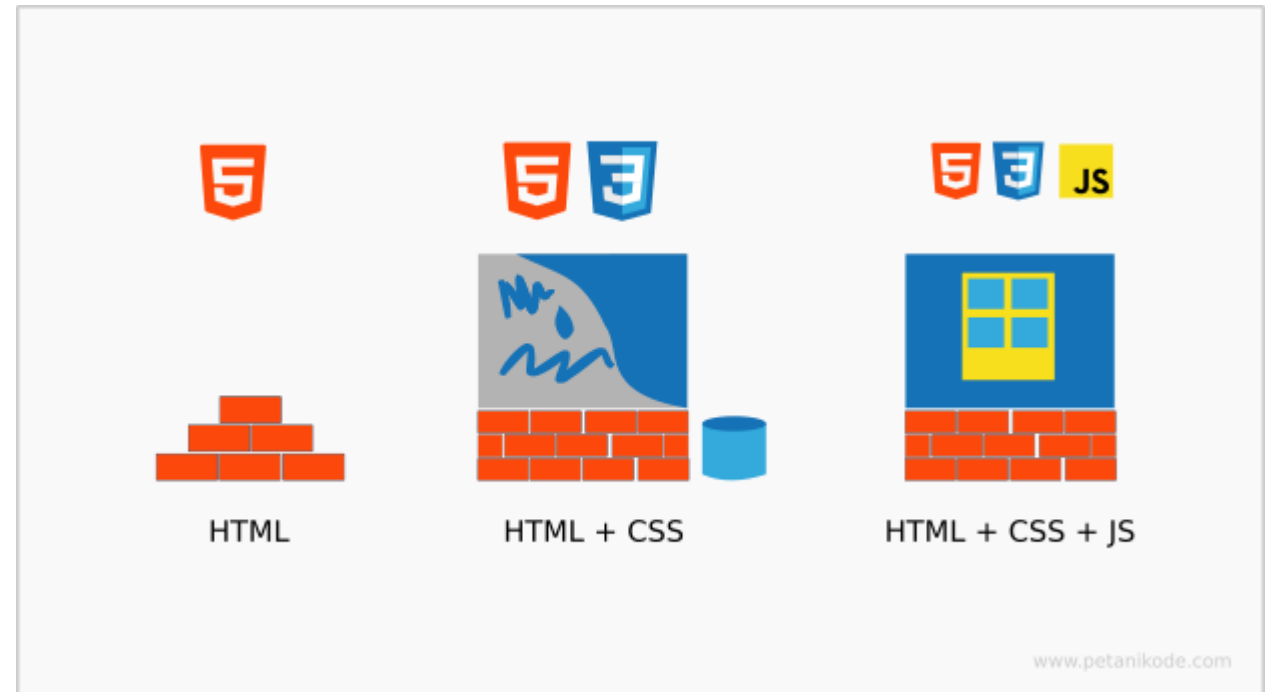
HTML itu seperti batu bata untuk membangun rumah. Batu bata ini dapat disusun, hingga menjadi fondasi dasar.



<https://www.petanikode.com/tutorial/html/>

HTML - CSS - JAVASCRIPT

- CSS adalah bahasa khusus yang digunakan untuk memperindah tampilan web.
- Javascript bertugas untuk membuat halaman web menjadi hidup.
- Karena dengan Javascript, kita dapat menentukan fungsi-fungsi maupun efek yang akan diterapkan di website.



<https://www.petanikode.com/tutorial/html/>

VERSI HTML

- HTML 1.0 (*Juni 1993*) adalah versi HTML pertama, namun tidak resmi dirilis;
- HTML 2.0 (*24 November 1995*) adalah versi HTML kedua yang resmi pertamakali beredar di pasaran dan dirilis oleh IETF;
- HTML 3.0 (*28 Maret 1995*) versi ini gagal beredar, karena banyak perubahan yang memicu perdebatan;
- HTML 3.2 (*14 Januari 1997*) versi resmi yang dirilis W3C pertamakali.
- HTML 4.0 (*24 April 1998*) versi pengembangan dari yang sebelumnya;
- HTML 4.01 (*24 Desember 1999*) versi perbaikan dari HTML 4.0;
- XHTML 1.0 (*26 Januari 2000*) pengembangan dari HTML 4.01 dengan mengadopsi XML;
- XHTML 2.0 (*Augustus 2002—Juli 2006*) versi kedua dari XHTML;
- HTML 5 (*28 Oktober 2014*) versi html saat ini.

PERALATAN MEMULAI HTML

1. Tools Editor untuk Menulis HTML



1. Web Browser untuk Membuka HTML

A screenshot of a Notepad++ window titled '*C:\Users\vmorrow\Documents\new 2.html - Not...'. The window shows a basic HTML document structure with the following code:

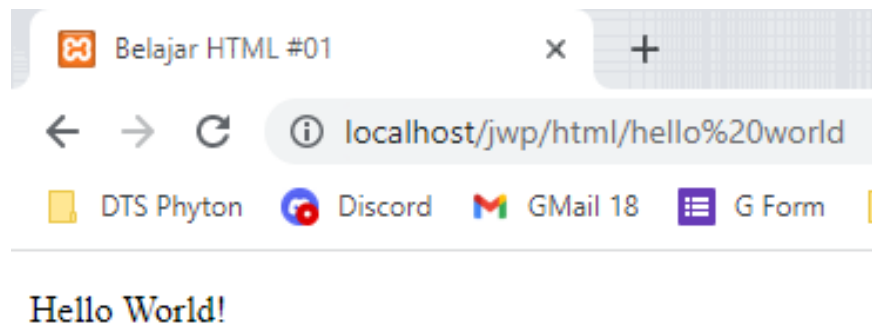
```
1 <!doctype HTML>
2 <html>
3   <head>
4     <title>My web page title</title>
5   </head>
6   <body>
7     <h1 id="hd1">My heading eliplant</h1>
8     <p>This is a paragraph</p>
9     <ol>
10
11   </body>
12 </html>
```

The status bar at the bottom indicates 'Ln: 10 Col: 7 Sel: 0|0', 'Dos\Windows', 'UTF-8 w/o BOM', and 'INS'.

<https://www.petanikode.com/tutorial/html/>

Membuat HTML Pertama

- o Buka Text Editor,
- o Lalu Ketikkan code di samping
- o simpan dengan nama : "hello-world.html"



```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Belajar HTML #01</title>

  </head>

  <body>

    <p>Hello World!</p>

  </body>

</html>
```

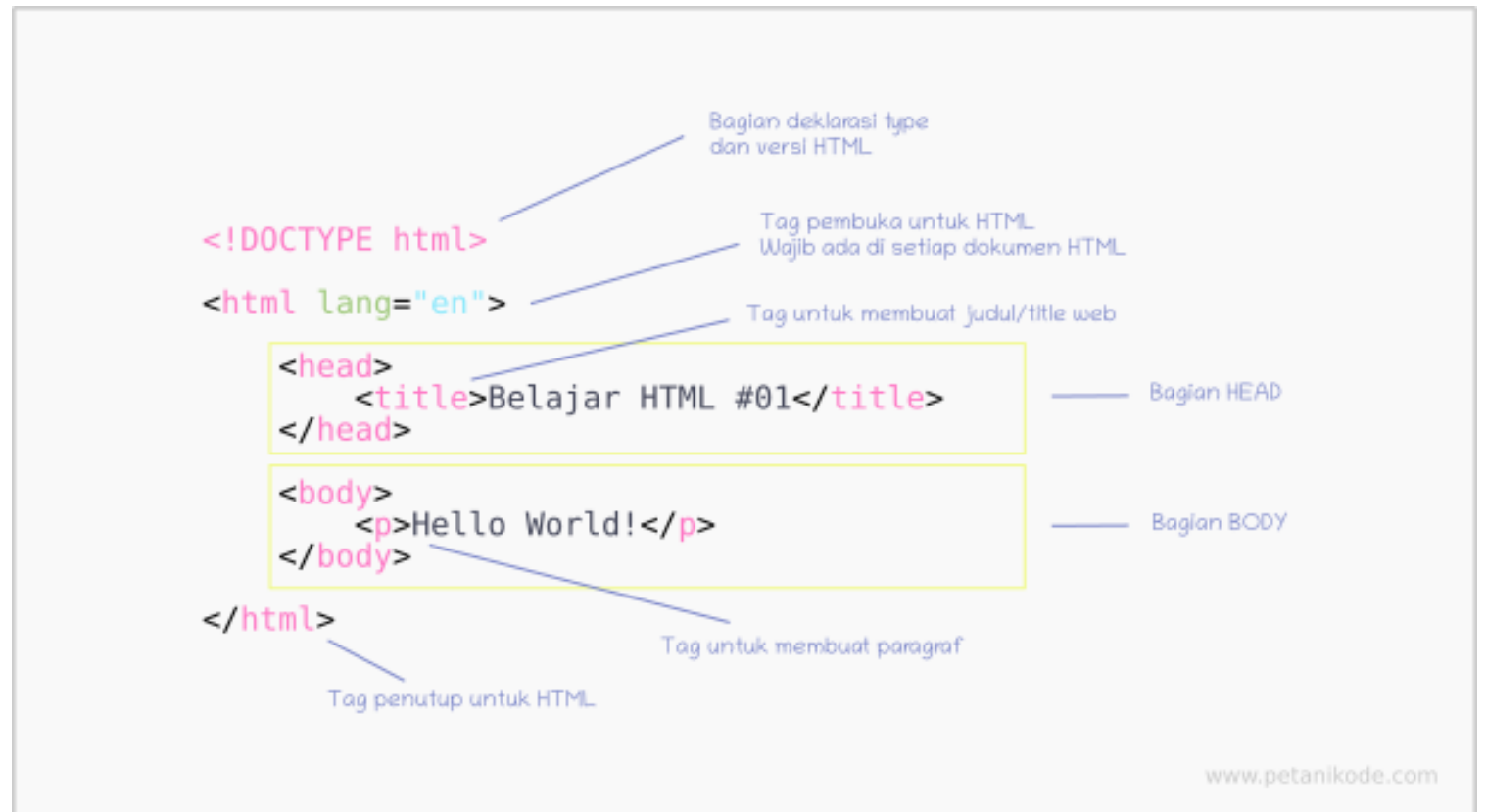
<https://www.petanikode.com/tutorial/html/>

TUGAS HTML

- Buat File HTM dengan Judul namadiri.HTML
- Kemudian Tuliskan/ cetak namamu dalam halaman tersebut

STRUKTUR DASAR HTML

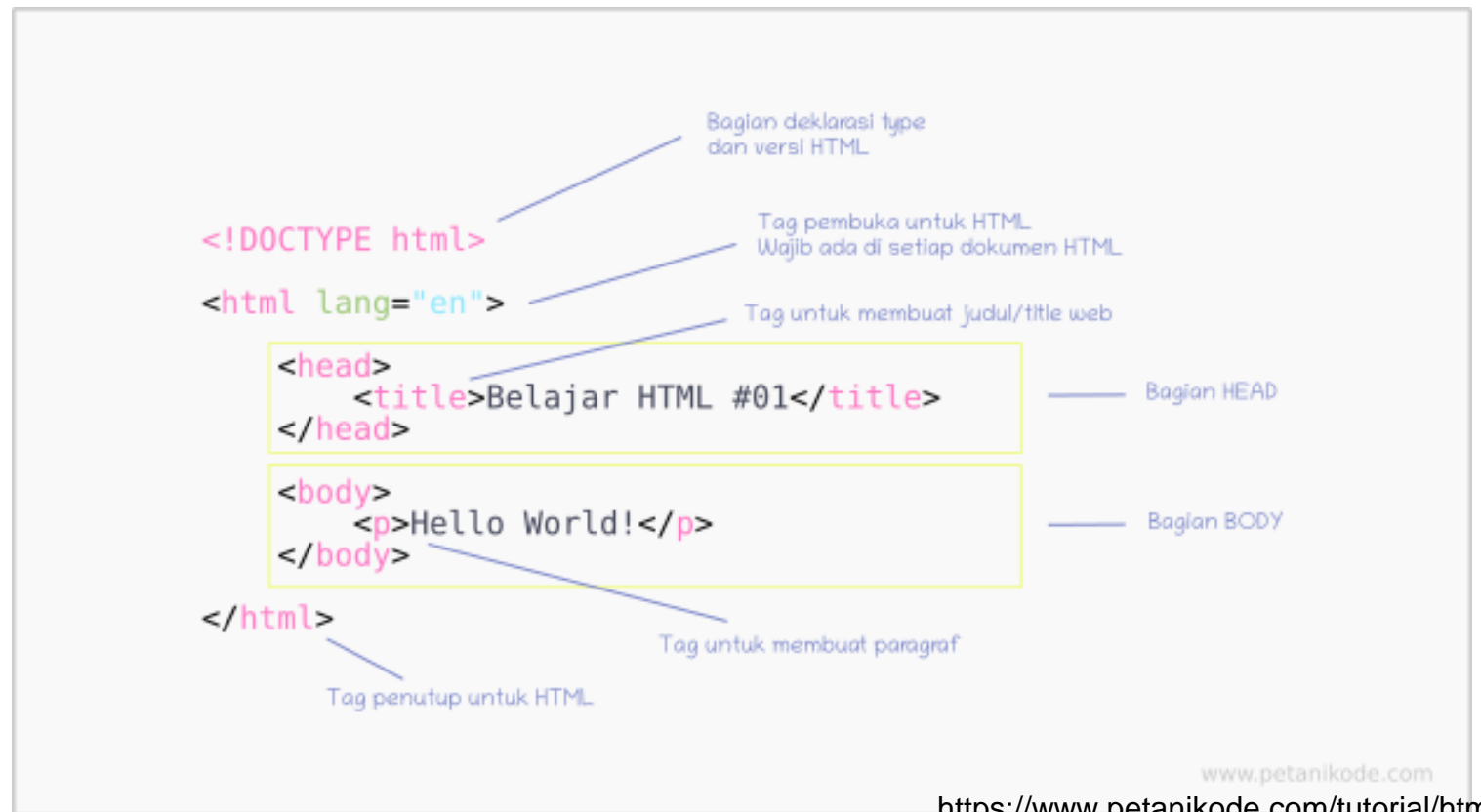
- Penulisan kode HTML selalu dimulai dengan deklarasi DOCTYPE, lalu menuliskan tag `<html>`
- di dalam tag `<html>` terdapat :
 - tag `<head>`
 - dan tag `<body>`.



<https://www.petanikode.com/tutorial/html/>

STRUKTUR DASAR HTML

1. Bagian Deklarasi
2. Bagian Head
3. Bagian Body



<https://www.petanikode.com/tutorial/html/>

1 BAGIAN DEKLARASI

Coba perhatikan kode pada baris pertama: `<!DOCTYPE html>`.

Ini adalah tag deklarasi untuk menyatakan tipe dokumen dan versinya. Pada contoh di atas, kita menyatakan dokumen ini bertipe HTML dan versinya adalah HTML 5.

Lalu apakah boleh kita tidak menulis kode `<!DOCTYPE html>` ini?

Boleh-boleh saja, karena dokumen HTML-nya akan tetap bisa dibuka di web browser.

Tapi, ini akan melanggar aturan standar yang dibuat W3C.

O ya, kita bisa mengecek.. apakah HTML yang ditulis sudah benar atau tidak di <https://validator.w3.org/>.

1 BAGIAN DEKLARASI (2)

```
<html lang="en">
```

Tag `<html>` wajib ada di setiap dokumen HTML..

Pada tag ini, kita memberikan atribut `lang="en"` untuk menyatakan kalau konten dokumen HTML ini akan menggunakan bahasa inggris.

Nah di dalam tag `<html>` ini, terdapat dua tag penting lagi.. yakni: tag `<head>` dan tag `<body>`.

Setelah itu barulah terakhir tag HTML ditutup dengan `</html>`.

2. BAGIAN HEAD

Bagian HEAD adalah bagian kepala dari HTML. Dimulai dari tag `<head>` dan ditutup dengan `</head>`.

Pada bagian HEAD, biasanya digunakan untuk menuliskan tag-tag yang akan dibaca oleh mesin.

Seperti:

- Tag meta untuk SEO;
- Tag `<title>` untuk judul;
- Tempat menulis kode CSS dan Javascript;
- dan lain-lain.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Belajar HTML #01</title>
  </head>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

3. BAGIAN BODY

- o Bagian BODY adalah bagian yang akan ditampilkan pada web browser.
- o Penulisan di mulai dari tag `<body>` dan ditutup dengan `</body>`
- o Di sinilah nanti kita akan banyak menuliskan konten dengan berbagai macam tag.
- o Saat ini kita baru mengisinya dengan tag `<p>`.
- o Tag `<p>` adalah tag yang digunakan untuk membuat paragraf.

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Belajar HTML #01</title>

  </head>

  <body>

    <p>Hello World!</p>

  </body>

</html>
```

<https://www.petanikode.com/tutorial/html/>

TAG - ELEMEN - ATRIBUT

Tag adalah sebuah penanda awalan dan akhiran dari sebuah elemen di HTML.

Tag dibuat dengan kurung siku (< . . . >), lalu di dalamnya berisi nama tag dan kadang juga ditambahkan dengan atribut.

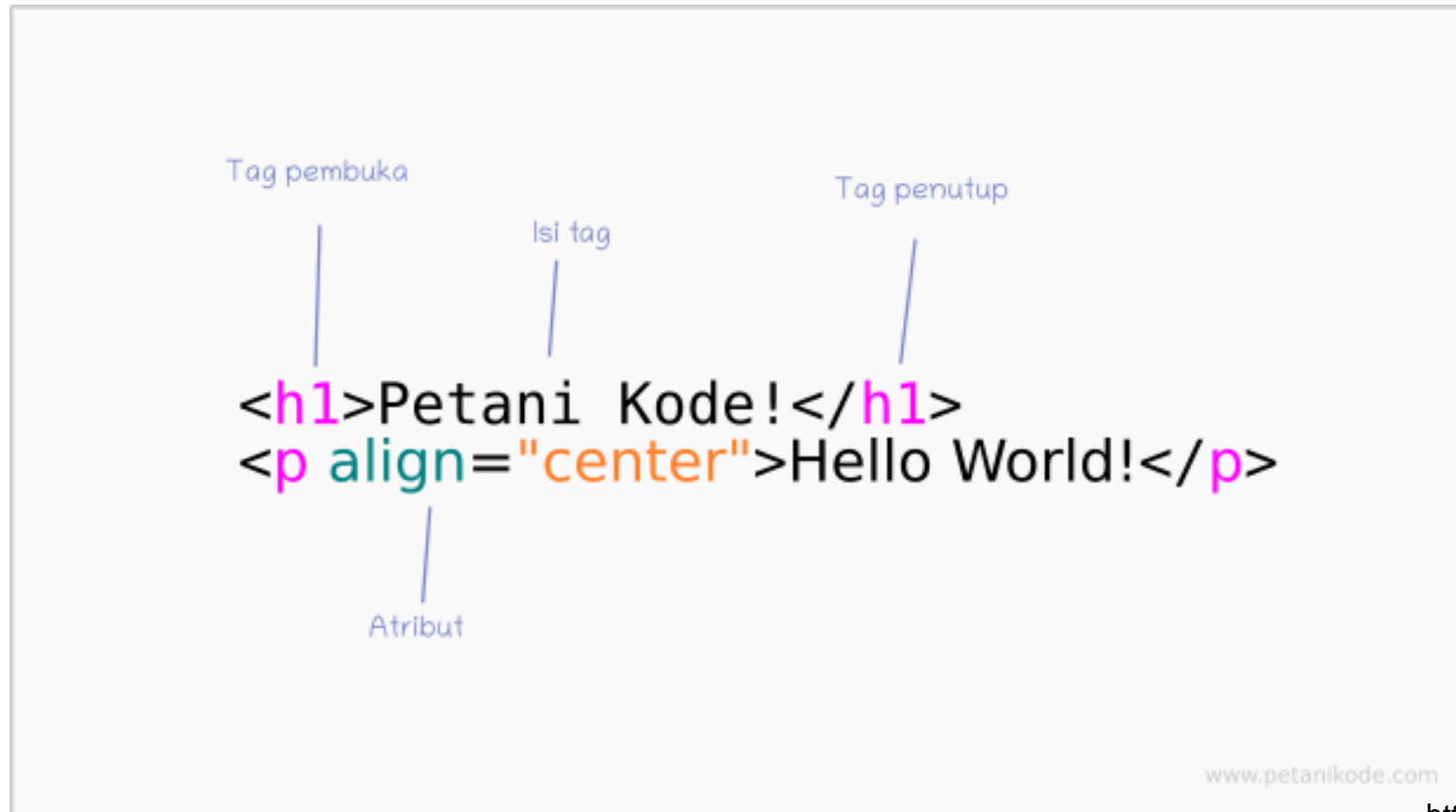
Contoh: <p>, <a>, <body>, <head>, dan sebagainya.

Tag selalu ditulis berpasangan. Ada tag pembuka dan ada tag penutupnya.

Namun, ada juga beberapa tag yang tidak memiliki pasangan penutup.

Tag penutup ditulis dengan menambahkan garis miring (/) di depan nama tag.

BAGIAN TAG



<https://www.petanikode.com/tutorial/html/>

TAG DASAR YANG HARUS DIINGAT

| Tag | Fungsi |
|--|------------------------------------|
| <code><html></code> | untuk memulai dokumen HTML |
| <code><head></code> | untuk membuat bagian head |
| <code><body></code> | untuk membuat bagian body |
| <code><h1></code> sampai <code><h6></code> | untuk membuat heading pada artikel |
| <code><p></code> | untuk membuat paragraf |
| <code><!-- --></code> | untuk membuat komentar |

<https://www.petanikode.com/tutorial/html/>

TAG WAJIB ADA

Berikut ini daftar tag yang wajib ada di HTML:

- `<!DOCTYPE html>` — untuk deklarasi type dokumen;
- `<html>` — tag utama dalam HTML;
- `<head>` — untuk bagian kepala dari dokumen;
- `<title>` — untuk judul web;
- `<body>` — untuk bagian body dari dokumen.

ATURAN PENULISAN TAG

1. Gunakan Huruf Kecil
2. Pastikan Menutup Tag dengan Benar

Penutupan tag HTML

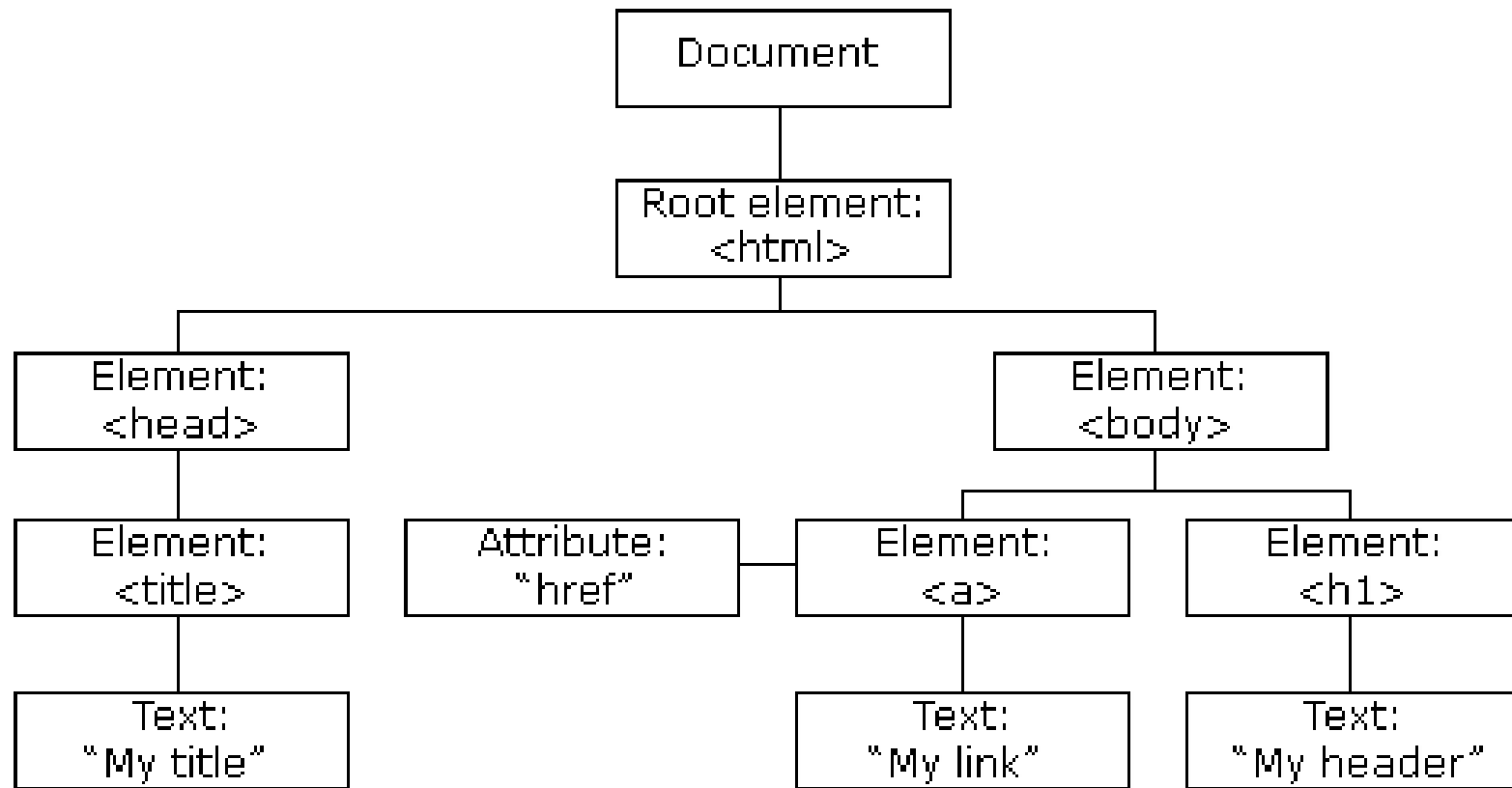
✓ Benar

```
<i><b><u>memasak</u></b></i>
```

✗ Salah

```
<i><b><u>memasak</i></b></u>
```


TAG - ELEMEN - ATRIBUT



<https://www.petanikode.com/tutorial/html/>

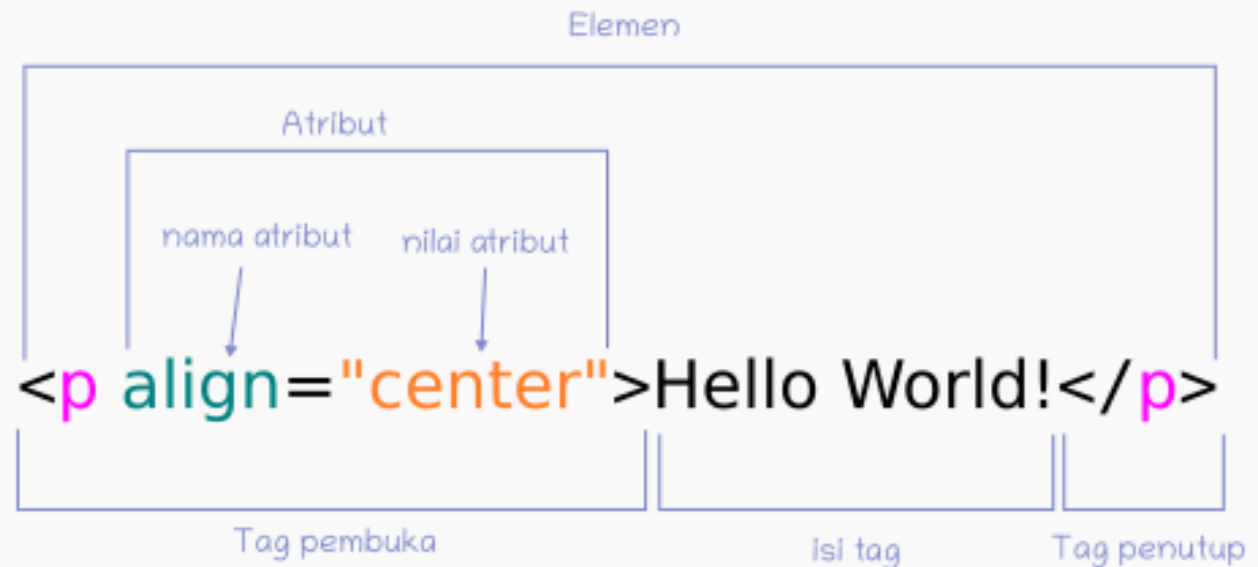
DEFINISI ELEMEN

- o Elemen dalam HTML adalah sebuah komponen yang menyusun dokumen HTML.
- o Elemen kadang juga disebut sebagai *node*, karena ia merupakan salah satu jenis *node* yang menyusun dokumen HTML dalam diagram HTML tree.
- o pada diagram tersebut, terdapat tiga macam *node*.. yakni: Node elemen, Node atribut, dan Node teks.
- o Elemen dibentuk dari tag pembuka, isi tag, dan tag penutup. Kadang juga ditambahkan beberapa atribut.

BAGIAN ELEMEN

Terdapat :

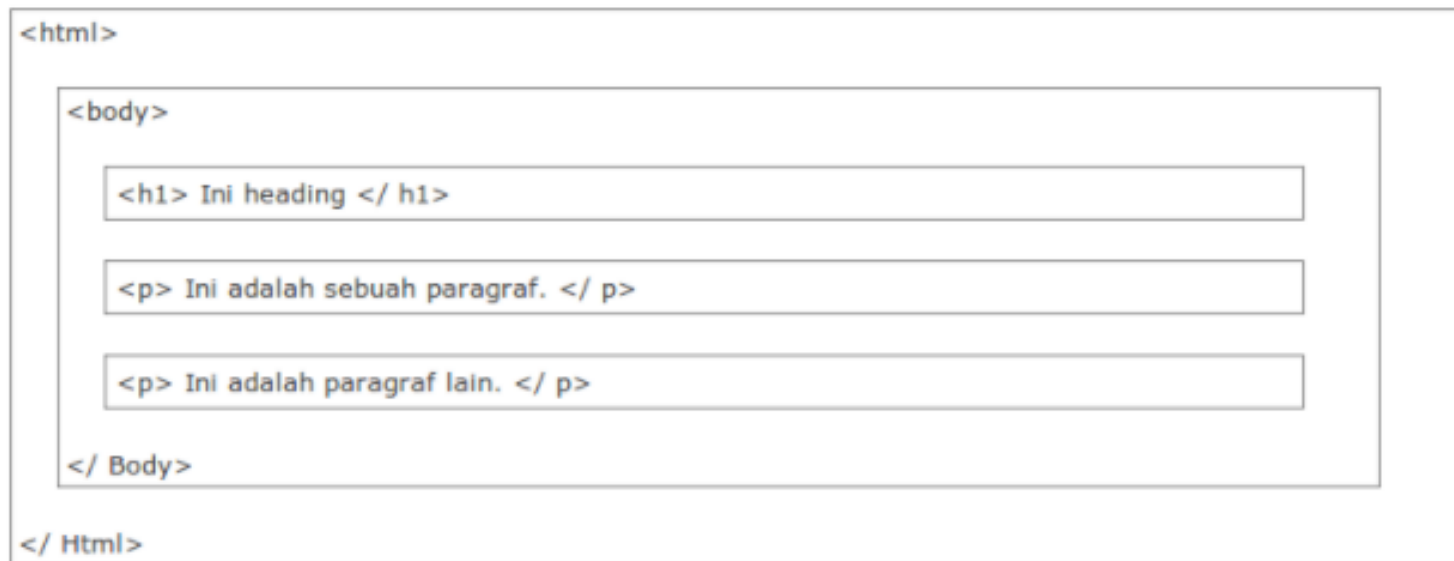
- satu elemen `<p>`
- dengan atribut `align="center"`
- memiliki isi berupa teks, yakni `Hello World!.`



NESTED ELEMEN

Elemen tidak selalu berisi teks, kadang ia juga akan berisi elemen lain. Ini biasanya kita sebut dengan *nested element* atau elemen di dalam elemen.

Bila digambarkan dalam bentuk kotak persegi, maka akan terlihat seperti ini:



<https://www.petanikode.com/tutorial/html/>

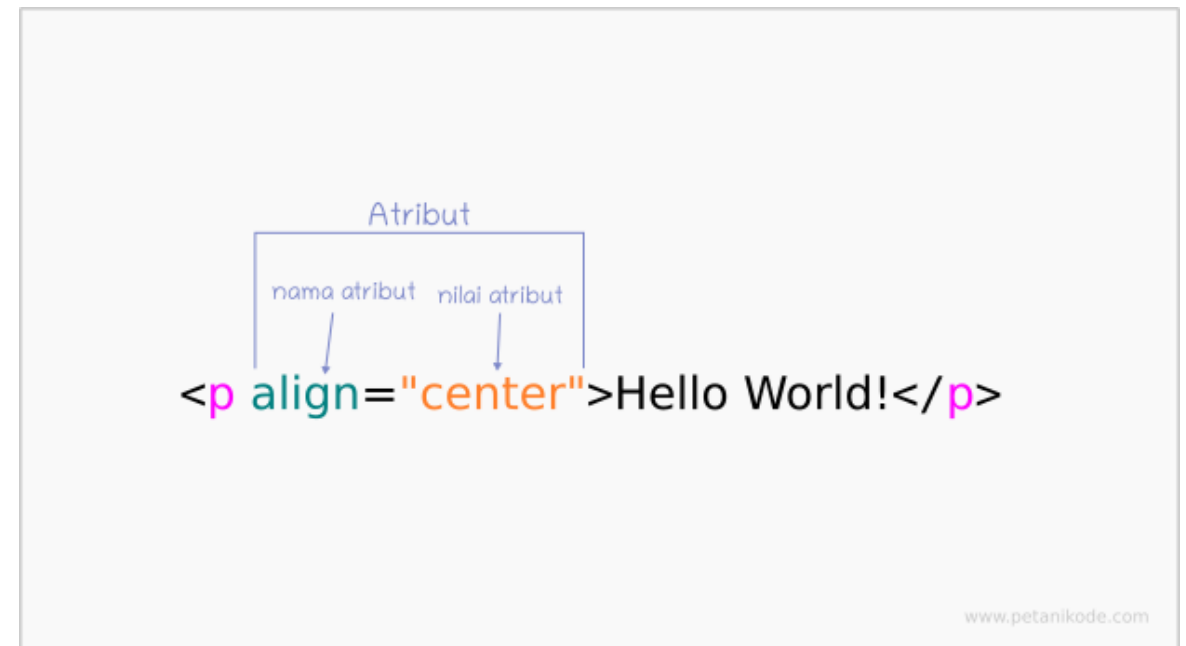
TAG - ELEMEN - ATRIBUT

Atribut adalah kata khusus yang berada di dalam tag pembuka.

Atribut juga disebut sebagai *modifier* yang akan menentukan perilaku dari elemen.

Atribut dapat ditambahkan pada elemen manapun.

Ada juga elemen yang mewajibkan menggunakan atribut seperti elemen `<a>`, ``, `<video>`, dll.



Contoh Atribut

```
<a href="https://petanikode.com">Petanikode.com</a>
```

Tag `<a>` adalah tag untuk membuat link. Tag ini mewajibkan menambahkan atribut `href` untuk menyatakan halaman tujuan dari link.

Jumlah atribut pada elemen bisa lebih dari satu.

```

```

Atribut `src` adalah atribut khusus untuk tag `` yang fungsinya untuk menentukan gambar yang akan ditampilkan. Lalu atribut `width` dan `height` adalah atribut yang mengatur ukurannya.

<https://www.petanikode.com/tutorial/html/>

JENIS ATRIBUT

1. Atribut Global

Atribut Global adalah atribut yang bisa ditambahkan pada semua elemen HTML.

2. Atribut Event

Atribut event adalah atribut yang digunakan untuk menentukan aksi yang akan dilakukan saat terjadi sesuatu pada elemen. Atribut ini nanti akan banyak digunakan pada [pemrograman Javascript](#).

3. Atribut Khusus

Atribut khusus adalah atribut yang hanya bisa dipakai pada elemen tertentu dan kadang atribut ini tidak bisa digunakan pada elemen yang lain.

<https://www.petanikode.com/tutorial/html/>

CARA PENULISAN ATRIBUT

1. Gunakan Huruf Kecil

2. Guanakan Tanda Petik

3. Penggunaan Spasi

Jika ada atribut yang memiliki lebih dari satu nama, maka ia ditulis dengan tanda min (–), bukan spasi.

CONTOH HEADING

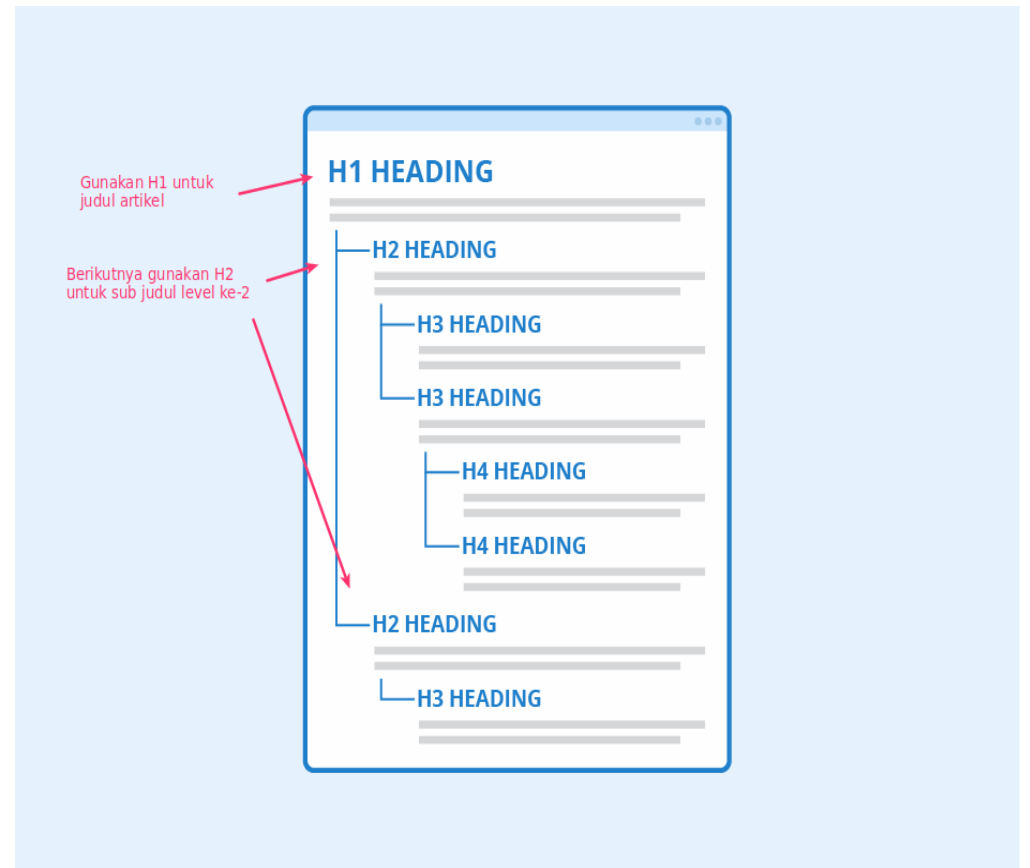
- Judul pada HTML dapat kita buat dengan tag `<h1>` sampai `<h6>`.
- Tag `<h1>` digunakan pada judul level pertama.
- Lalu tag lainnya digunakan pada sub heading atau level berikutnya.
- Masing-masing judul akan ditampilkan dengan ukuran teks yang berbeda.
- Tag `<h1>` adalah yang paling besar, dan tag `<h6>` paling kecil

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Heading di HTML</title>
  </head>
  <body>
    <h1>Ini Judul Level 1</h1>
    <h2>Ini Judul Level 2</h2>
    <h3>Ini Judul Level 3</h3>
    <h4>Ini Judul Level 4</h4>
    <h5>Ini Judul Level 5</h5>
    <h6>Ini Judul Level 6</h6>
  </body>
</html>
```

html/

CARA MEMBUAT HEADING

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Heading di HTML</title>
  </head>
  <body>
    <h1>Belajar Heading di HTML</h1>
    <p>
      Heading di HTML ada enam, yakni H1, H2, H3, H4, H5, dan H6.
      Heading berfungsi untuk membuat judul untuk artikel dan juga
      sub judul pada bagian artikel.
    </p>
    <h2>Membuat Sub Judul</h2>
    <p>
      Sub judul atau sub heading dimulai dengan tag H2. Lalu diikuti
      dengan tag selanjutnya hingga sampai H6. Sementara itu, tag H1
      hanya digunakan untuk judul artikel saja.
    </p>
  </body>
</html>
```



<https://www.petanikode.com/tutorial/html/>

ATRIBUT HEADING

- `id` untuk memberikan nama id unik. Biasanya ini akan digunakan oleh link, CSS, dan Javascript;
- `class` untuk memberikan nama class yang akan dipakai oleh CSS;
- `style` untuk memberikan kode css secara langsung;
- `title` untuk menambahkan informasi tambahan;
- dll.

HEADING VS TITLE VS HEADER

- Heading adalah judul untuk artikel dan bagian artikel yang dibuat dengan tag `<h1>` sampai `<h6>`
- Title adalah judul dari web yang dibuat dengan tag `<title>`
- Header adalah bagian kepala (kop) pada web yang dibuat dengan tag `<header>`



KOMENTAR

Komentar adalah elemen yang akan diabaikan oleh browser. Ia tidak akan ditampilkan di dalam web.

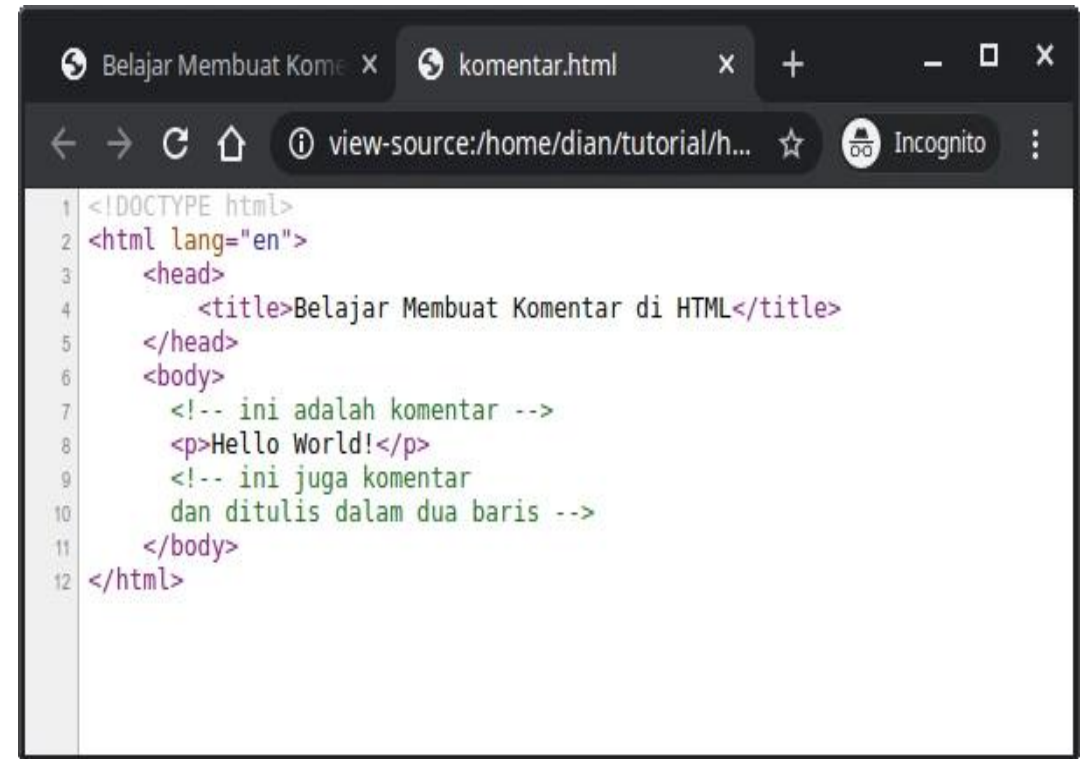
Komentar biasanya digunakan untuk memberikan informasi tambahan pada kode HTML dan kadang juga digunakan untuk menon-aktifkan beberapa kode HTML.



<https://www.petanikode.com/tutorial/html/>

MENULIS KOMENTAR

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Belajar Membuat Komentar di
HTML</title>
  </head>
  <body>
    <!-- ini adalah komentar -->
    <p>Hello World!</p>
    <!-- ini juga komentar
    dan ditulis dalam dua baris -->
  </body>
</html>
```

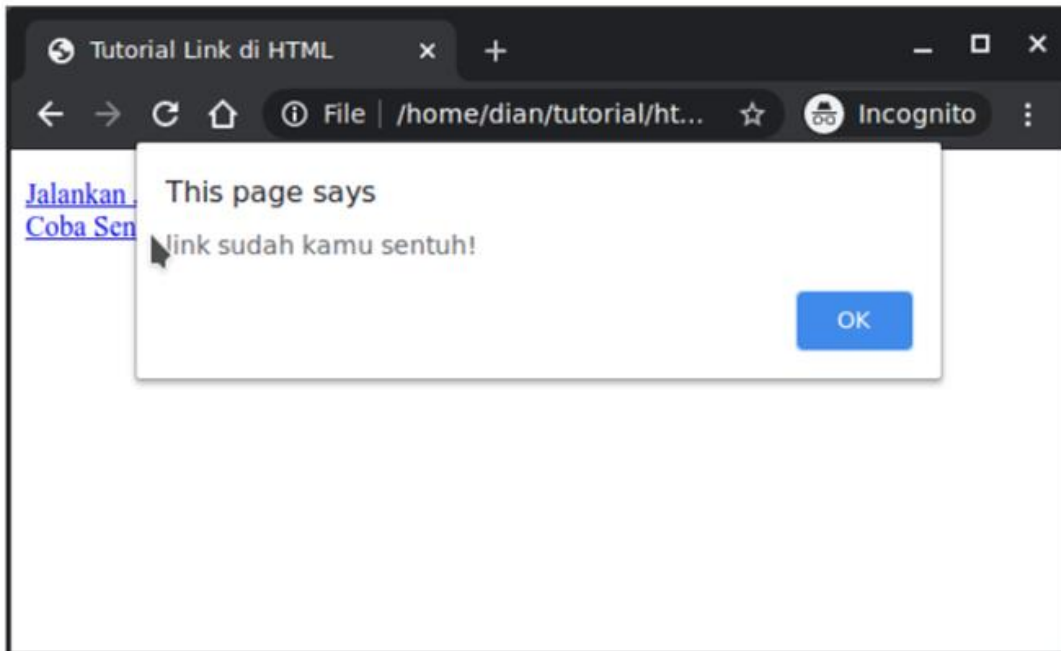
A screenshot of a web browser window. The address bar shows 'view-source:/home/dian/tutorial/h...'. The page content is the same HTML code as shown in the left panel, with line numbers 1 through 12 on the left side of the code editor. The code includes a DOCTYPE declaration, an HTML tag with lang="en", a head section with a title 'Belajar Membuat Komentar di HTML', and a body section containing a single-line comment, a paragraph 'Hello World!', and a multi-line comment.

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <title>Belajar Membuat Komentar di HTML</title>
5   </head>
6   <body>
7     <!-- ini adalah komentar -->
8     <p>Hello World!</p>
9     <!-- ini juga komentar
10    dan ditulis dalam dua baris -->
11   </body>
12 </html>
```

<https://www.petanikode.com/tutorial/html/>

FUNGSI JAVASCRIPT

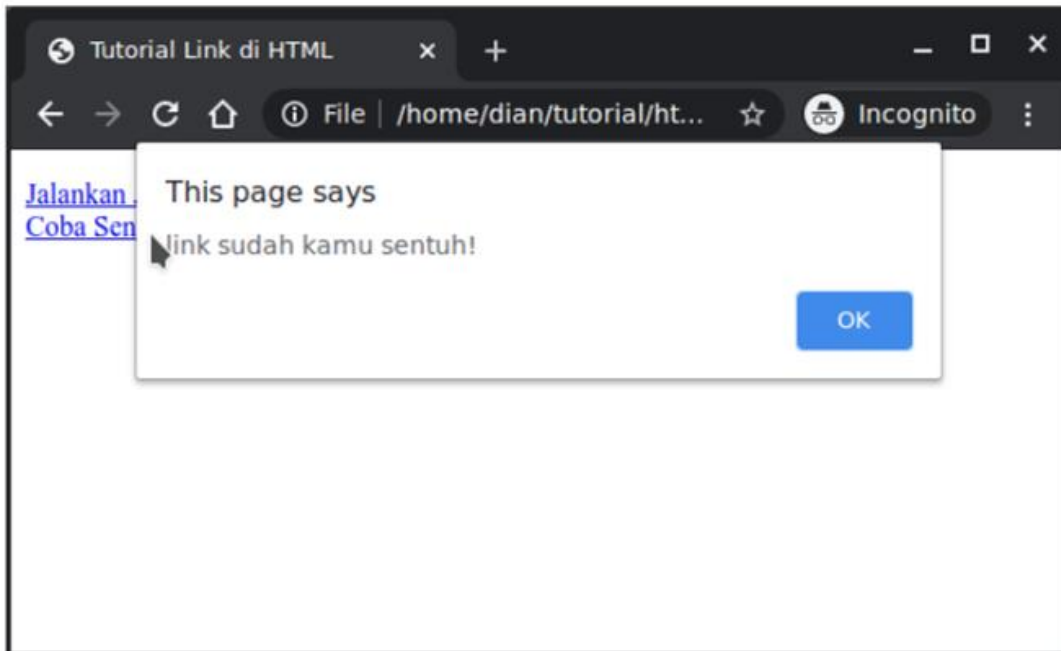
Pemanggilan fungsi Javascript biasanya dilakukan dengan atribut even seperti `onclick`, `onmouseover`, `onmouseout`, dan sebagainya.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>
      <a href="#">Link Buntu</a>
      <a href="#!">Link Buntu 2</a>
    </p>
  </body>
</html>
```

LINK MEMANGGIL FUNGSI JAVASCRIPT

Pemanggilan fungsi Javascript biasanya dilakukan dengan atribut even seperti `onclick`, `onmouseover`, `onmouseout`, dan sebagainya.



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Tutorial Link di HTML</title>
  </head>
  <body>
    <p>
      <a href="javascript:alert('Hello
World!')">Jalankan JS</a>
    </p>
  </body>
</html>
```


MEMBUAT TABEL

Tabel terdiri dari 4 unsur utama:

1. Baris
2. Kolom
3. Sel
4. Garis

A diagram of a table with 4 columns and 7 rows. The first row is highlighted in light green. The second column is highlighted in light orange. A label 'baris' with a horizontal line points to the first row. A label 'Kolom' with a vertical line points to the second column. A label 'Sel' with a diagonal line points to the cell at the intersection of the first row and third column. The URL 'www.petanikode.com' is visible at the bottom right of the table.

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

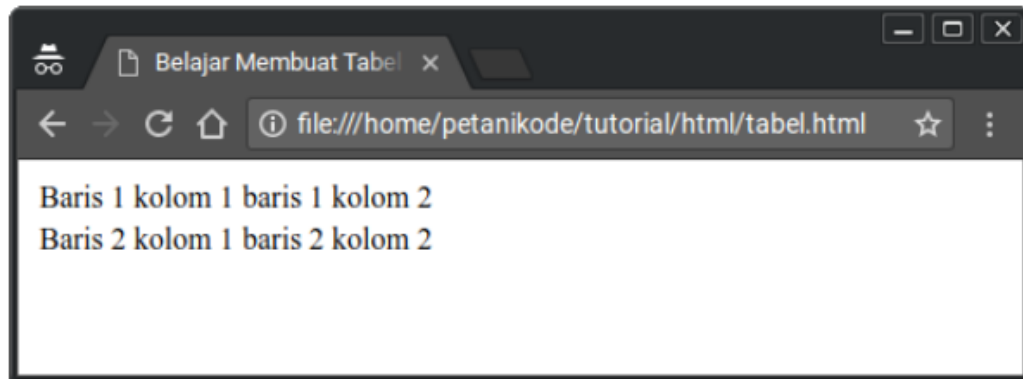
www.petanikode.com

Tag Membuat Table

1. Tag `<table>` untuk membungkus tabelnya
2. Tag `<thead>` untuk membungkus bagian kepala tabel
3. Tag `<tbody>` untuk membungkus bagian body dari tabel
4. Tag `<tr>` (tabel row) untuk membuat baris
5. Tag `<td>` (table data) untuk membuat sel
6. Tag `<th>` (table head) untuk membuat judul pada header

Praktek Membuat Tabel

Coba, kemudian tmbahkan border = 1



```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Membuat Tabel HTML</title>
</head>
<body>

  <table>
    <tr>
      <td>Baris 1 kolom 1</td>
      <td>baris 1 kolom 2</td>
    </tr>
    <tr>
      <td>Baris 2 kolom 1</td>
      <td>baris 2 kolom 2</td>
    </tr>
  </table>

</body>
</html>
```

MEMBUAT FORM

```
<form action="proses.php" method="GET">
<!-- form field di sini -->
</form>
```

Form di HTML dapat kita buat dengan tag `<form>`.

Tag ini memiliki beberapa atribut yang harus diberikan, seperti:

- `action` untuk menentukan aksi yang akan dilakukan saat data dikirim;
- `method` metode pengiriman data.

Untuk atribut `action`, kita dapat mengisinya dengan alaman URL dari *endpoint* yang akan memproses form.

Secara sederhana,—pada contoh di atas— kita akan menyuruh file `proses.php` untuk memproses data form.

FIELD

```
<input type="text" name="info" />
```

Field adalah ruas yang dapat diisi dengan data.

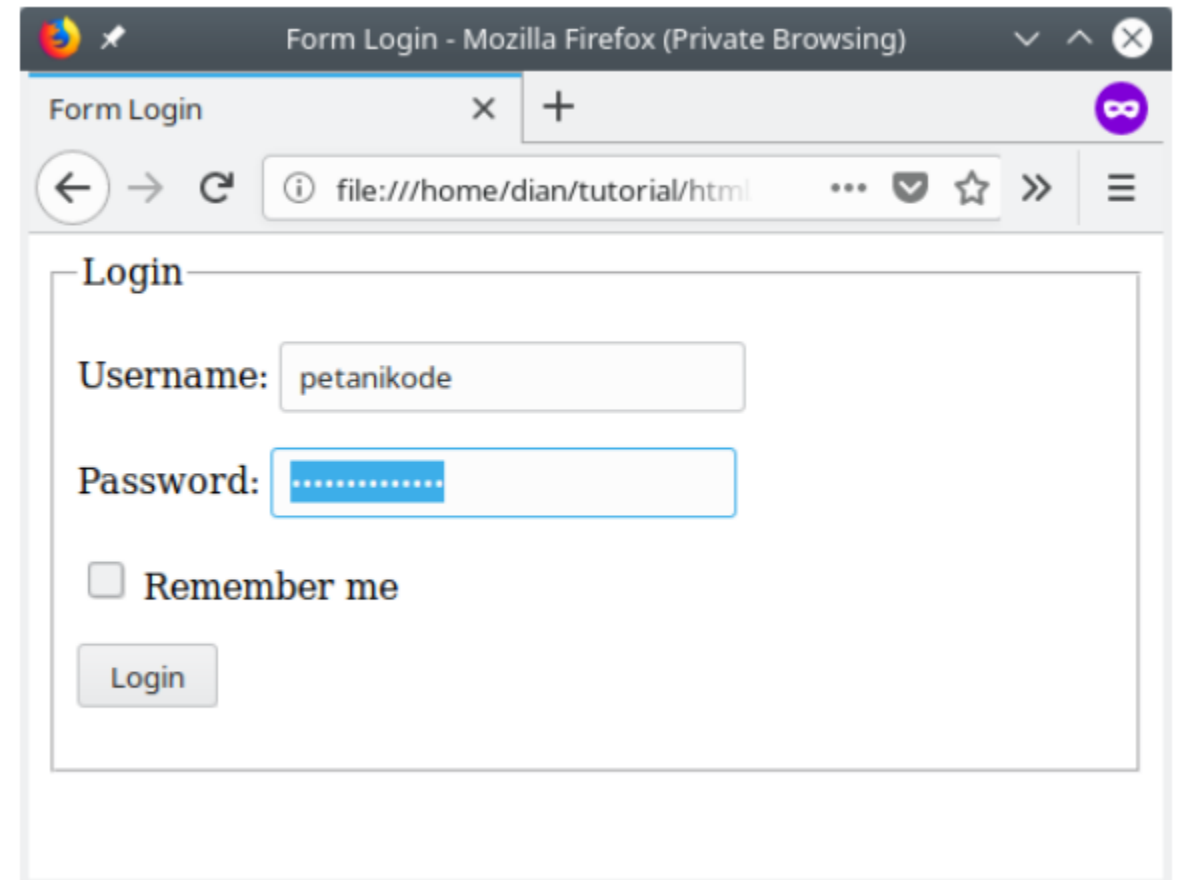
Field memiliki beberapa atribut yang harus diberikan:

1. `type` merupakan type dari field.
2. `name` merupakan nama dari field yang akan menjadi kunci dan variabel di dalam program.

PRAKTEK : FORM LOG IN

Pada form login, terdapat beberapa field dan elemen:

1. Field untuk input username atau email;
2. Field untuk input password;
3. Checkbok untuk remember me;
4. Tombol untuk login.



The screenshot shows a Mozilla Firefox browser window in Private Browsing mode. The title bar reads "Form Login - Mozilla Firefox (Private Browsing)". The address bar shows the file path "file:///home/dian/tutorial/html". The page content is a login form titled "Login". It contains the following elements:

- A "Username:" label followed by a text input field containing the text "petanikode".
- A "Password:" label followed by a password input field with blue dots.
- A checkbox labeled "Remember me" which is currently unchecked.
- A "Login" button at the bottom left of the form.



CSS

TENTANG CSS

CSS adalah bahasa kedua setelah **HTML** yang harus dipelajari seorang *web developer*.

Pengetahuan tentang CSS sangatlah penting bagi seorang web developer, karena dengan CSS kita bisa:

- Halaman *landing page* yang menarik;
- Template atau tema blog;
- Mengubah PSD (Desain web) menjadi HTML; dll



<https://9gag.com/gag/amz44B4>

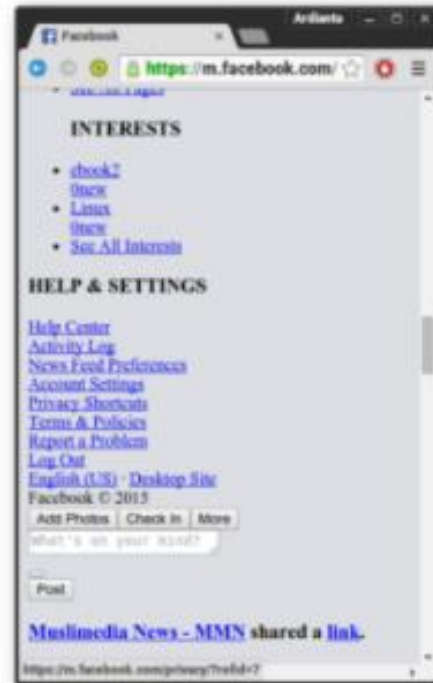


Jordy Gabriel

<https://me.me/i/html-html-css-jordy-gabriel-23022503>

DENGAN DAN TANPA CSS

Tampa CSS



Dengan CSS



<https://www.petanikode.com/tutorial/html/>

CSS

CSS (*Cascade Style Sheet*) merupakan sebuah bahasa untuk mengatur tampilan web sehingga terlihat lebih menarik dan indah.

Dengan CSS, kita dapat mengatur *layout* (tata letak), warna, font, garis, dan lain-lain.

CSS pertama kali diperkenalkan oleh **Håkon Wium Lie** pada tahun 1994.

VARIASI CSS

1. CSS 1: adalah versi pertama (17 Desember 1996)
2. CSS 2: adalah versi ke-2 (Mei 1998)
3. CSS 2.1: (7 juni 2011)
4. CSS 3: (2012)
5. CSS 4: masih dalam pengembangan.

STRUKTUR CSS

1. Selektor;
2. Blok Deklarasi;
3. Properti dan nilainya.



<https://www.petanikode.com/>

SELEKTOR

Selektor adalah kata kunci untuk memilih elemen HTML yang akan kita atur.

```
h1 {  
    color: red;  
}
```

Artinya: Kita memilih semua elemen `<h1>`, lalu diberikan warna teks `red` (merah).

Selektor dapat berupa nama tag, class, id, dan atribut.

```
/* Selektor dengan nama tag */  
h2 {  
    color: blue  
}
```

```
/* Selektor dengan class */  
.bg-yellow {  
    background-color: yellow;  
}
```

```
/* selektor dengan ID elemen */  
#header {  
    background: grey;  
}
```

```
/* Selektor dengan Atribut */  
input[type=text] {  
    background: yellow;  
}
```

BLOK DEKLARASI

Blok deklarasi adalah tempat kita menuliskan atribut-attribut CSS yang akan diberikan ke pada selektor.

```
p {  
    font-size: 18px;  
}
```

Artinya, kita akan mengatur ukuran font dari tag `<p>` sebesar `18px`.

Blok deklarasi dimulai atau dibuka dengan tanda kurung `{` lalu ditutup dengan `}`.

PROPERTI DAN NILAINYA

Properti merupakan atribut atau sekumpulan aturan yang akan diberikan kepada elemen yang dipilih.

```
property: "nilai";
```

Setiap properti harus diakhiri dengan titik koma (;).

Apabila hanya terdapat satu properti, boleh tidak menggunakan titik koma.

Properti harus ditulis di dalam blok deklarasi.

CARA MENULIS CSS DALAM HTML

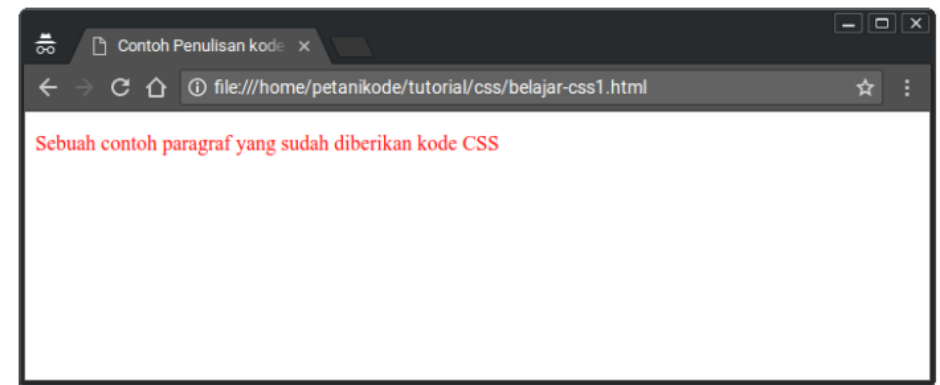
penulisan kode CSS di HTML dapat dilakukan di dalam tag `<style>`. Tag tersebut dapat ditulis di dalam tag `<head>` atau `<body>`. Kebanyakan orang menulisnya di dalam tag `<head>`.

```
<!DOCTYPE html>
<html>
<head>
<title>Contoh Penulisan kode CSS</title>

<style type="text/css">
  p { color: red }
</style>

</head>

<body>
  <p>Sebuah contoh paragraf yang sudah diberikan oleh kode
  CSS</p>
</body>
</html>
```



2 CARA MENULIS CODE

1. Internal/ Embedded / Style CSS
2. Eksternal CSS
3. Inline CSS

I .INTERNAL CSS/ EMBEDDED CSS

kode CSS yang ditulis di dalam tag `<style>`.

Internal CSS juga dikenal Tag `<style>` biasanya ditulis di dalam tag `<head>` atau `<body>`,
Lebih banyak ditulis di dalam `<head>`

HTML

Result

EDIT ON
CODEPEN

Ini judul artikel

Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk percobaan saja. Percobaan untuk mendemokan *internal css*. Seperti namanya, *inline CSS* adalah kode CSS yang ditulis langsung dalam file HTML.

Resources

1x 0.5x 0.25x

Rerun

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Internal CSS</title>
  <!-- penulisan internal css dalam tag head -->
  <style type="text/css">
    p{
      font-family: serif;
      line-height: 1.75em;
      font-size: 18px;
    }
    i {
      font-family: sans;
      color: orange;
    }
  </style>
</head>

<body>
  <!-- penulisan internal css dalam tag body -->
  <style type="text/css">
    h2 {
      font-family: sans;
      color: #333;
    }
  </style>
  <h2>Ini judul artikel</h2>
  <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya untuk
  percobaan saja. Percobaan untuk mendemokan <i>internal css</i>. Seperti
  namanya, <i>inline CSS</i> adalah kode CSS yang ditulis langsung dalam file
  HTML.</p>
</body>
</html>
```

II. EKSTERNAL CSS

Eksternal CSS adalah kode CSS yang ditulis terpisah dengan kode HTML.

Eksternal CSS ditulis disebuah file khusus yang berekstensi `.css`.

PRAKTEK : BUAT FILE CSS

style-ku.css

```
p {  
    font-family: serif;  
    line-height: 1.75em;  
}
```

```
i {  
    font-family: sans;  
    color: orange;  
}
```

```
h2 {  
    font-family: sans;  
    color: #333;  
}
```

2 CARA MEMANGGIL FILE CSS

1. Menggunakan tag `<link>`

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

1. Menggunakan `@import`

```
<style type="text/css">
@import "style-ku.css";
</style>
```

Pemanggilan Dalam HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Eksternal CSS</title>
  <link rel="stylesheet" type="text/css" href="style-ku.css">
</head>

<body>
  <h2>Ini judul artikel</h2>
  <p>Ini adalah paragraf yang memuat isi artikel. Paragraf ini hanya
untuk percobaan saja. Percobaan untuk mendemokan <i>internal css</i>.
Seperti namanya, <i>inline CSS</i> adalah kode CSS yang ditulis
langsung dalam file HTML.</p>
</body>
</html>
```

III. INLINE CSS

Inline CSS adalah kode CSS yang ditulis langsung pada atribut elemen HTML.

Setiap elemen HTML memiliki atribut `style`, di sana lah inline CSS ditulis. Contohnya seperti ini:

```
<h2 style="color:red; font-family: sans;">Ini  
judul artikel</h2>
```

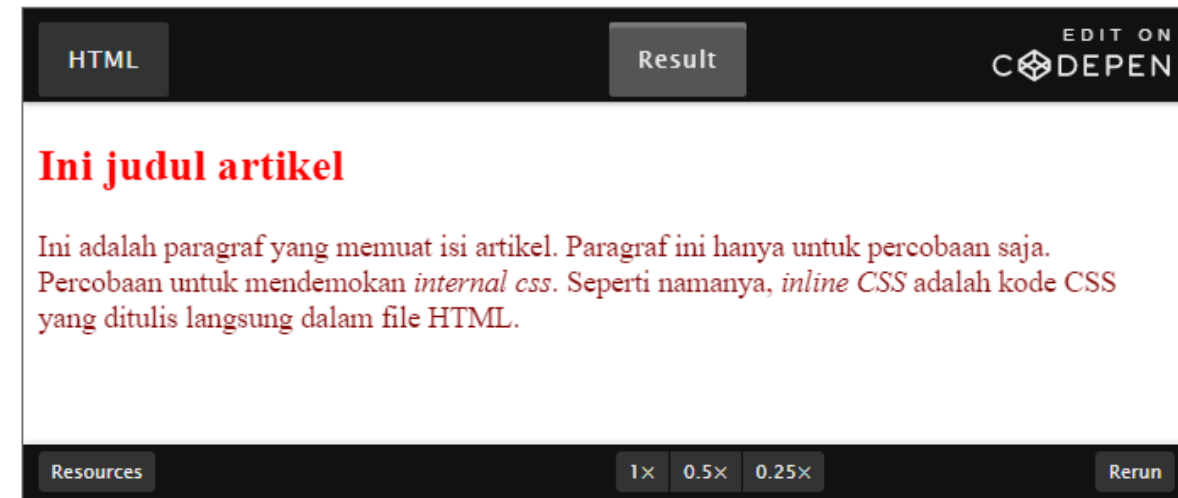

PRAKTER INLINE CSS

```
<!DOCTYPE html>
<html>
<head>
  <title>Contoh Inline CSS</title>
</head>

<body>

  <h2 style="color:red;font-family:sans">Ini
judul artikel</h2>
  <p style="color:maroon">Ini adalah paragraf
yang memuat isi artikel. Paragraf ini hanya
untuk percobaan saja. Percobaan untuk
mendemonikan <i>internal css</i>. Seperti
namanya, <i>inline CSS</i> adalah kode CSS
yang ditulis langsung dalam file HTML.</p>
</body>

</html>
```



TIPS BELAJAR CSS

- 1. Cara Menghapal Kode dan Properti CSS dengan banyak praktek**
- 2. Gunakan Inspect Elemen**
- 3. Gunakan Cheat Sheet**

<https://www.petanikode.com/>

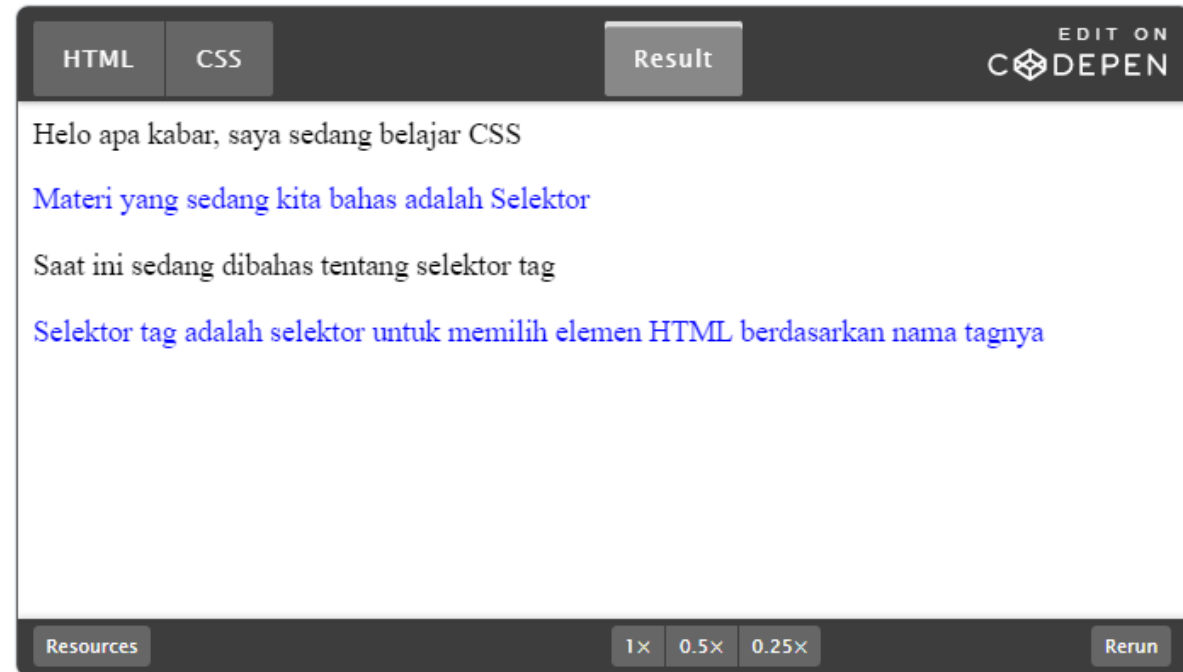
5 macam selektor

1. Selektor Tag
2. Selektor Class
3. Selektor ID
4. Selektor Atribut
5. Selektor Universal

1)Selektor Tag

```
p {  
  color: blue;  
}
```

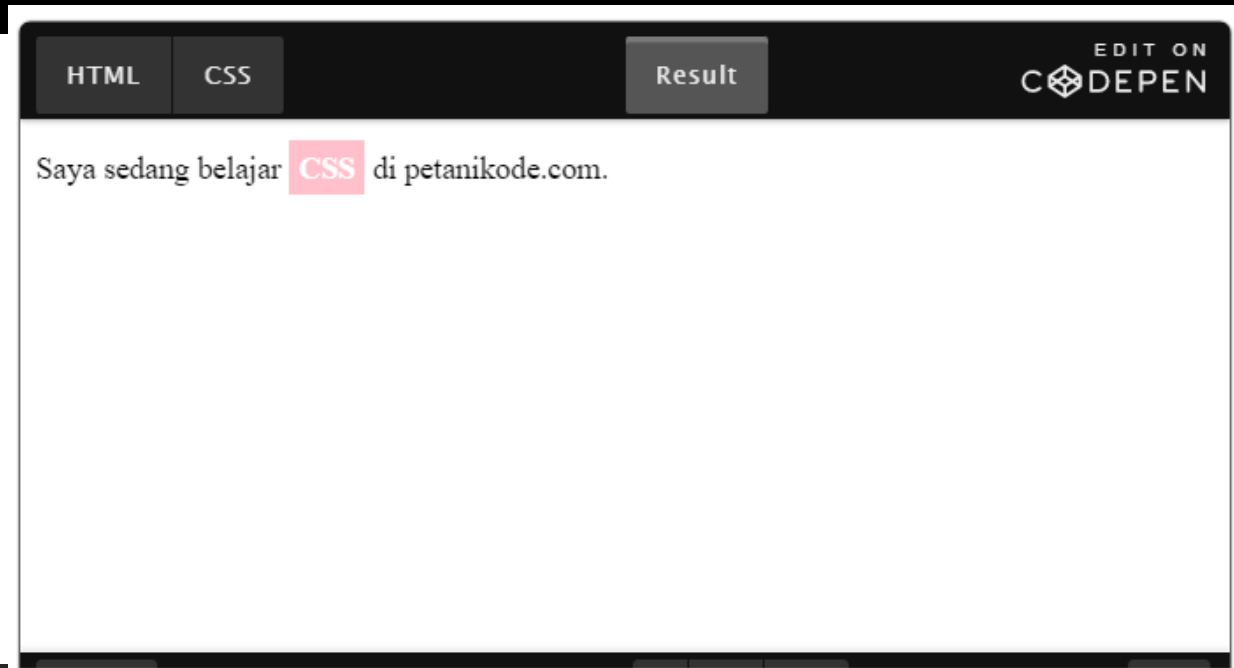
```
<div>Helo apa kabar, saya sedang  
belajar CSS</div>  
<p>Materi yang sedang kita bahas  
adalah Selektor</p>  
<div>Saat ini sedang dibahas  
tentang selektor tag</div>  
<p>Selektor tag adalah selektor  
untuk memilih elemen HTML  
berdasarkan nama tagnya</p>
```



2) Selektor Class

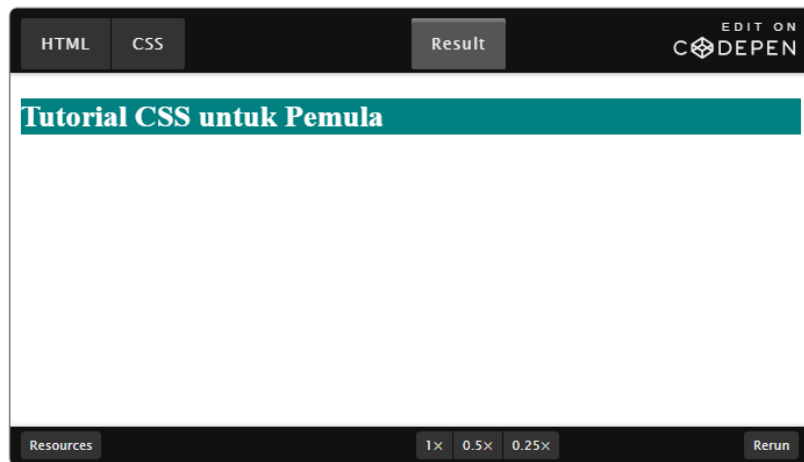
```
<p>Saya sedang belajar <b class="pink">CSS</b> di  
petanikode.com.</p>
```

```
.pink {  
  color: white;  
  background: pink;  
  padding: 5px;  
}
```



2) Selektor Class

Selektor class dapat kita gunakan pada elemen yang kita inginkan. Sebuah elemen HTML dapat menggunakan satu atau lebih class.



```
.text-white {  
  color: white;  
}
```

```
.bg-teal {  
  background: teal;  
}
```

```
<h2 class="text-white bg-teal">Tutorial CSS untuk  
Pemula</h2>
```

3) Selektor ID

Selektor ID hampir sama dengan class. Bedanya, ID bersifat unik. Hanya boleh digunakan oleh satu elemen saja.

Selektor ID ditandai dengan tanda pagar (#) di depannya.

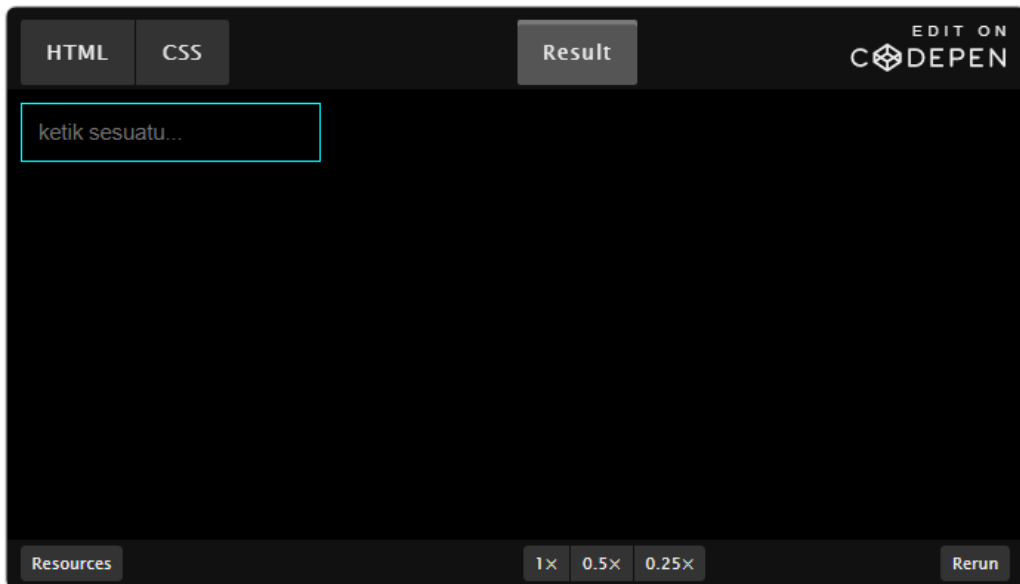


```
#header {  
  background: teal;  
  color: white;  
  height: 100px;  
  padding: 50px;  
}
```

```
<header id="header">  
  <h1>Selamat Datang di Website Saya</h1>  
</header>
```

4) Selektor Atribut

Selektor atribut adalah selektor yang memilih elemen berdasarkan atribut. Selektor ini hampir sama seperti selektor Tag.



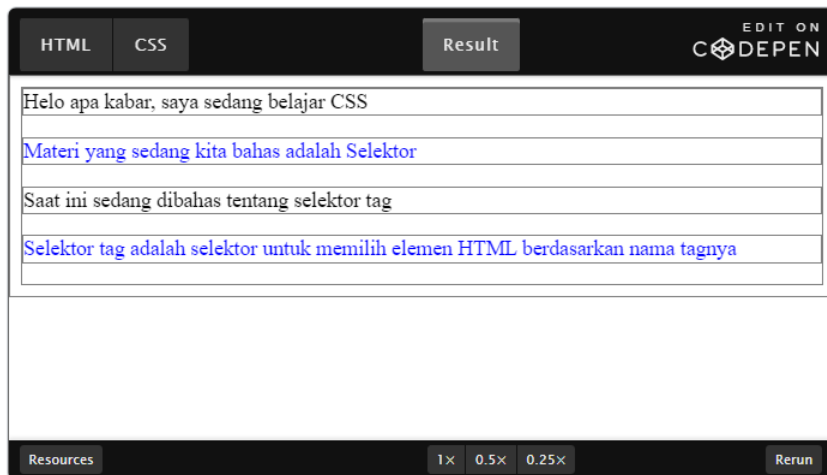
```
input[type=text] {  
  background: none;  
  color: cyan;  
  padding: 10px;  
  border: 1px solid cyan;  
}
```

```
<input type="text" placeholder="ketik sesuatu..." />
```


5) Selektor Atribut

Selektor universal adalah selektor yang digunakan untuk menyeleksi semua elemen pada jangkauan (*scope*) tertentu.

Code di samping, membuat semua elemen akan memiliki garis *solid* dengan ukuran *1px* dan berwarna *grey*.



```
* {  
  border: 1px solid grey;  
}
```



Pengenalan Javascript

JAVASCRIPT

Javascript adalah bahasa pemrograman yang wajib kamu pelajari jika ingin mendalami dunia web development.

Saat ini javascript tidak hanya digunakan di sisi *client (browser)* saja. Javascript juga digunakan pada server, console, program desktop, mobile, IoT, game, dan lain-lain.

Hal ini membuat javascript semakin populer dan menjadi bahasa yang paling banyak digunakan di Github.

<https://www.petanikode.com/>



<https://www.petanikode.com/>

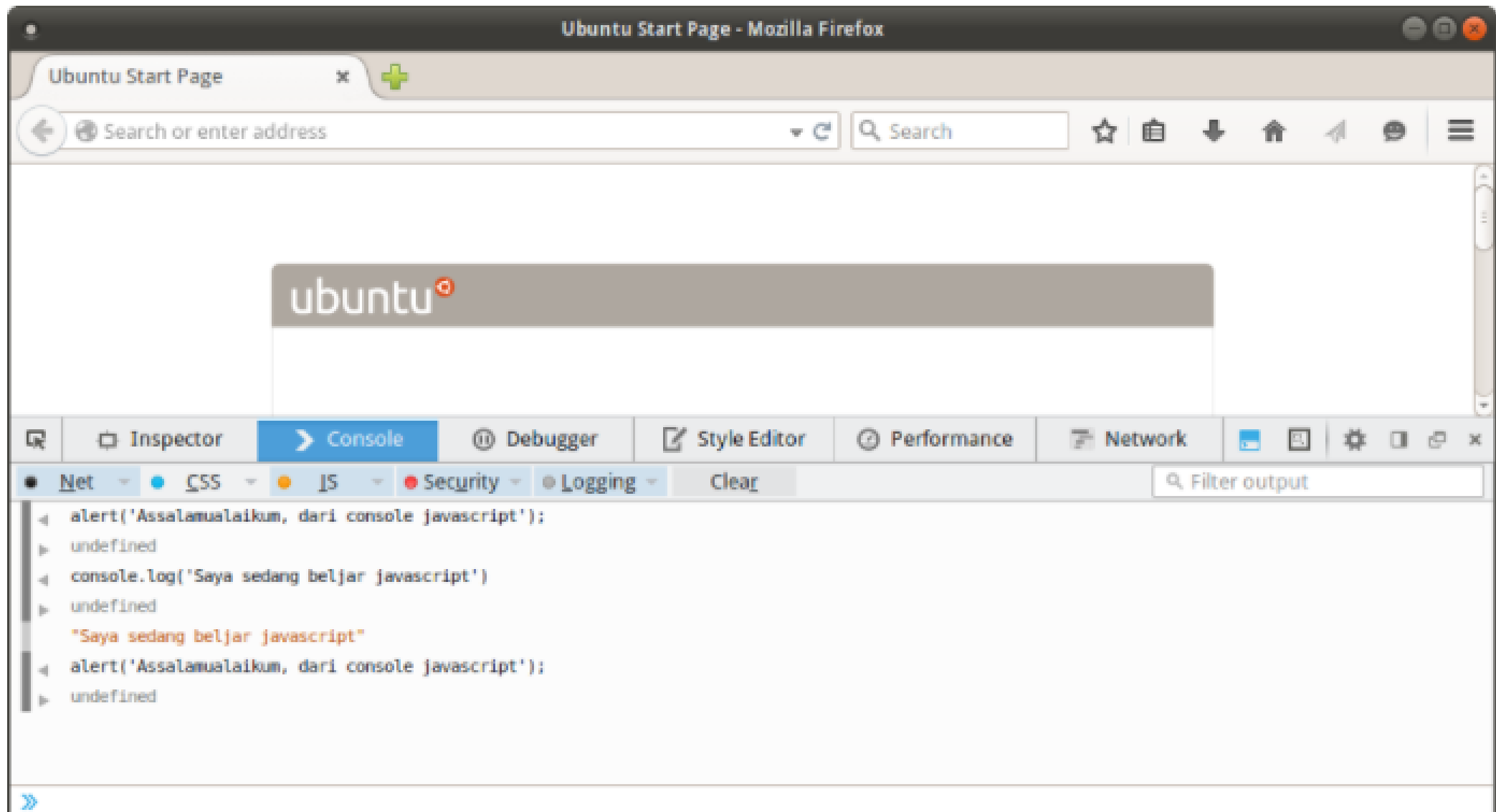
CONSOLE JAVASCRIPT

Ada yang mengatakan, belajar javascript itu susah, karena saat melihat hasilnya di web browser, pesan *error*-nya tidak tampil. Pendapat ini tidak benar.

Karena kita bisa melihatnya melalui *console*.

Console Javascript dapat kita buka melalui Inspect Element->Console.

Di dalam *console*, kita bisa menulis fungsi atau kode-kode javascript dan hasilnya akan langsung ditampilkan.



PENULISAN JAVASCRIPT

1. *Embed* (Kode Javascript ditempel langsung pada HTML. Contoh: yang tadi)
2. *Inline* (kode Javascript ditulis pada atribut HTML)
3. Eksternal (Kode Javascript ditulis terpisah dengan file HTML)

1. Pemanggilan Dalam HTML

Pada cara ini, kita menggunakan tag `<script>` untuk menempelkan (*embed*) kode Javascript pada HTML. Tag ini dapat ditulis di dalam tag `<head>` dan `<body>`.

Banyak yang merekomendasikan menuliskannya di dalam `<body>`, karena akan membuat web di-load lebih cepat.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Belajar Javascript dari Nol</title>
    <script>
      // ini adalah penulisan kode javascript
      // di dalam tag <head>
      console.log("Hello JS dari Head");
    </script>
  </head>
  <body>
    <p>Tutorial Javascript untuk Pemula</p>
    <script>
      // ini adalah penulisan kode javascript
      // di dalam tag <body>
      console.log("Hello JS dari body");
    </script>
  </body>
</html>
```


2. Inline (pada atribut HTML)

Pada cara ini, kita akan menulis kode javascript di dalam atribut HTML. Cara ini biasanya digunakan untuk memanggil suatu fungsi pada *event* tertentu.

Misal: saat link diklik.

Contoh:

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

atau bisa juga seperti ini:

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

<https://www.petanikode.com/>

Pada atribut `onclick` dan `href` kita menuliskan fungsi javascript di sana. Atribut `onclick` merupakan atribut HTML untuk menyatakan fungsi yang akan dieksekusi saat elemen itu diklik.

Pada contoh di atas, kita menjalankan fungsi `alert()`. Fungsi ini merupakan **fungsi untuk menampilkan dialog**.

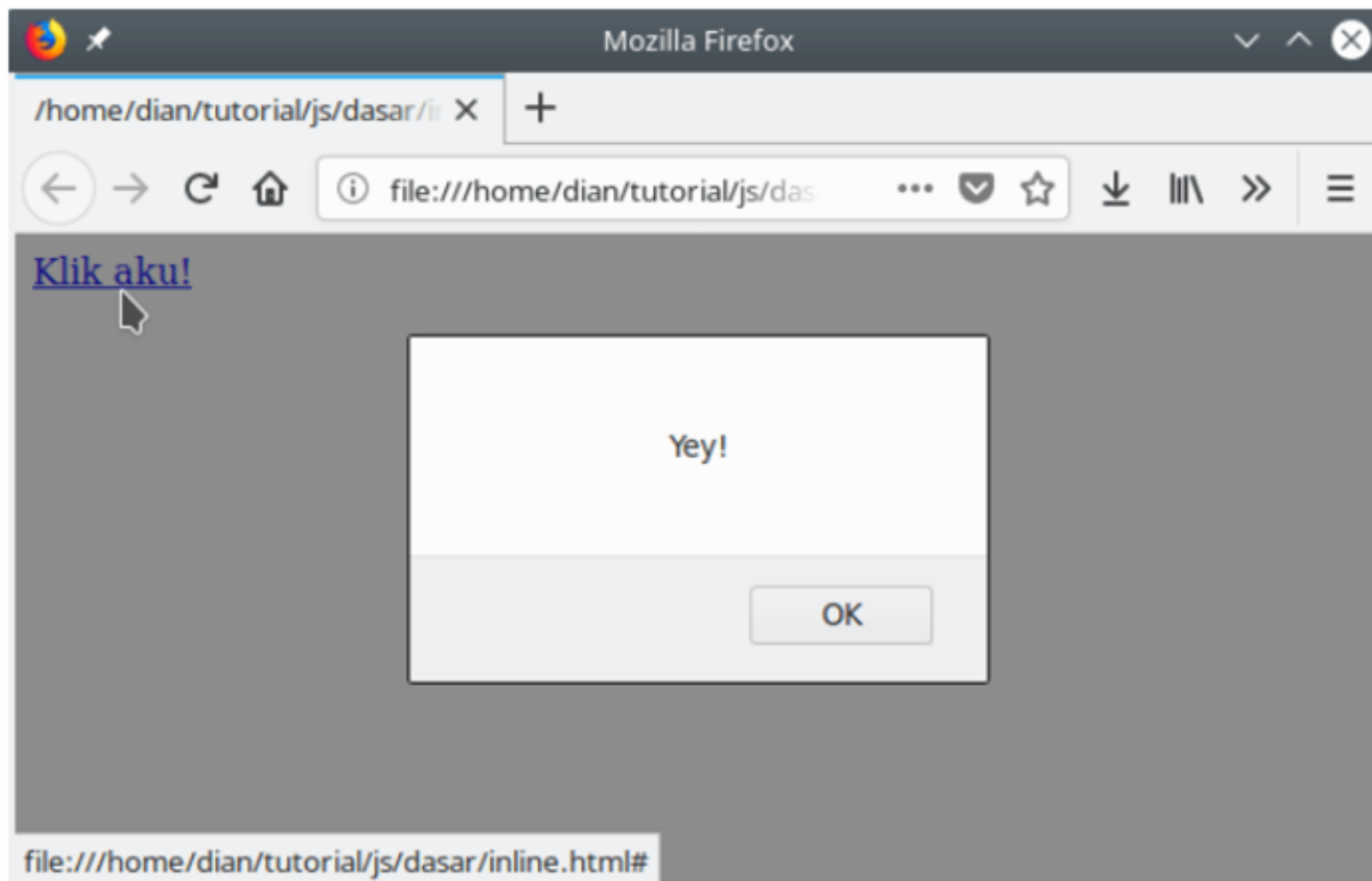
Lalu pada atribut `href`, kita juga memanggil fungsi `alert()` dengan didahului `javascript:`.

Atribut `href` sebenarnya digunakan untuk mengisi alamat link atau URL.

Karena kita ingin memanggil kode javascript di sana, maka alamat link tersebut kita ubah menjadi `javascript:` lalu diikuti dengan fungsi yang akan dipanggil.



Hasilnya:



3. Penulisan Kode javascript Eksternal

ada cara ini, kita akan menulis kode javascript secara terpisah dengan file HTML.

Cara ini biasanya digunakan pada proyek-proyek besar, karena diyakini—dengan cara ini—dapat lebih mudah mengelola kode project.

Mari kita lihat contohnya...

Kita buat dua file, yaitu: file HTML dan Javascript.

```
belajar-js/  
├─ kode-program.js  
└─ index.html
```

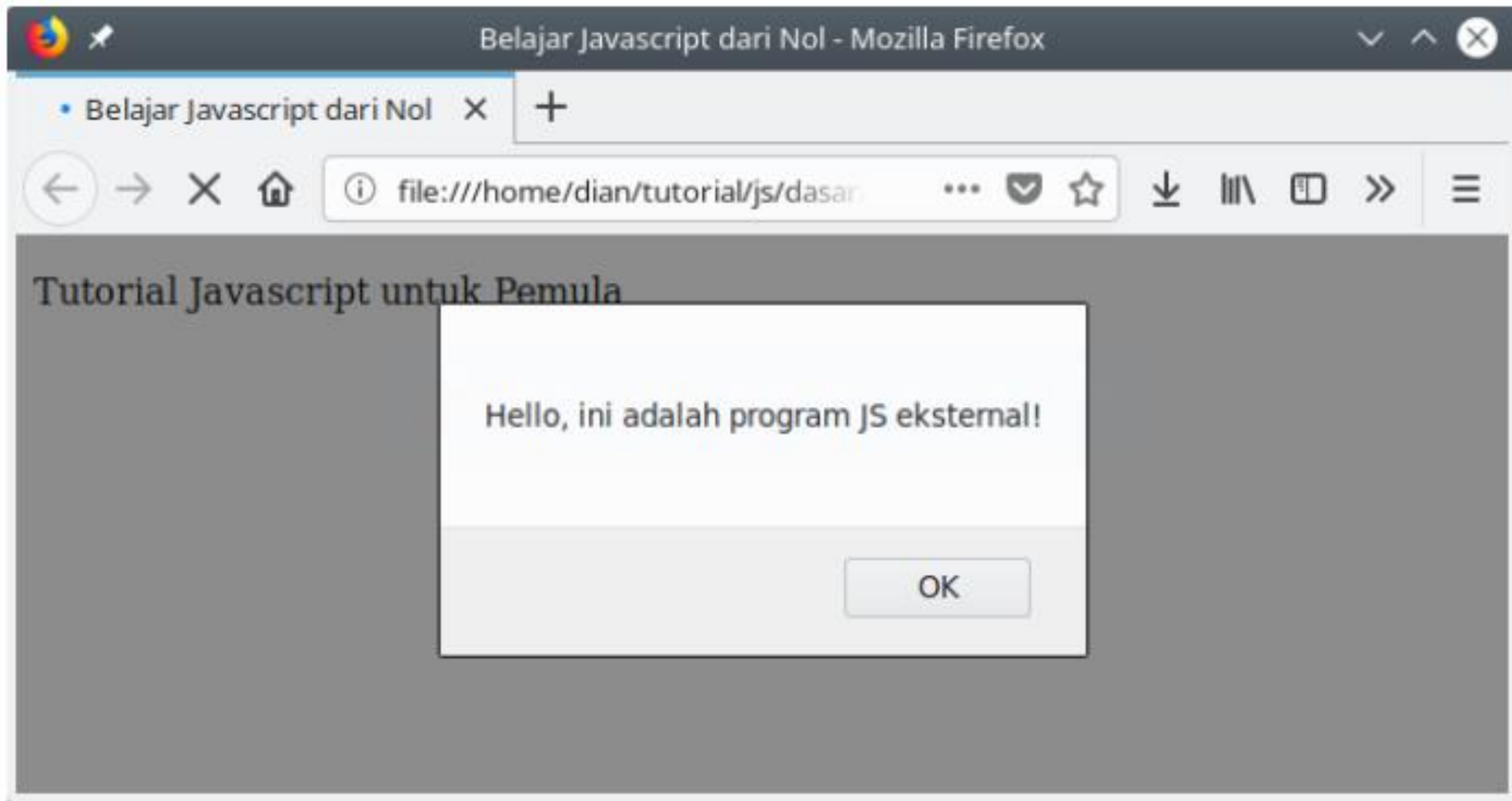
Isi dari file `kode-program.js`:

```
alert("Hello, ini adalah program JS eksternal!");
```

Isi dari file `index.html`:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Belajar Javascript dari Nol</title>
  </head>
  <body>
    <p>Tutorial Javascript untuk Pemula</p>

    <!-- Menyisipkan kode js eksternal -->
    <script src="kode-program.js"></script>
  </body>
</html>
```



<https://www.petanikode.com/>

PENJELASAN

Pada contoh di atas, kita menulis kode javascript terpisah dengan kode HTML.

Pada kode HTML, disisipkan dengan memberikan atribut `src` pada tag `<script>`.

```
<!-- Menyisipkan kode js eksternal -->
```

```
<script src="kode-program.js"></script>
```

Maka, apapun yang ada di dalam file `kode-program.js` akan dapat dibaca dari file `index.html`.



Pengenalan Bootstrap

Bootstrap

- Bootstrap adalah kerangka kerja CSS yang bersifat *open source* dan digunakan untuk kebutuhan pembuatan tampilan desain visual dari aplikasi web atau situs website.
- Kerangka kerja yang digunakan berbentuk *template* desain berbasis HTML dan CSS untuk kebutuhan pengembangan navigasi, tombol, tipografi, formulir, dan komponen antarmuka yang lainnya.
- Selain itu, Bootstrap juga memiliki fitur yang mencakup *library* dari JavaScript.

PENDIRI BOOTSTRAP

Mark Otto



Jacob Thornton



Founder of Bootstrap

3 CARA MENGGUNAKAN BOOTSRAP

1. Menggunakan CDN;
2. Download secara manual;
3. Download dengan *package manager*;

1. Menggunakan CDN

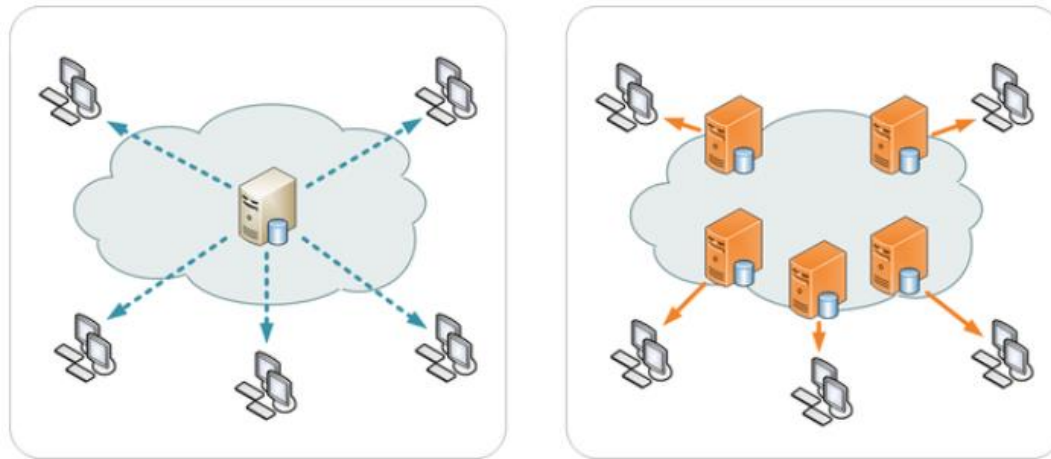
Cara ini termasuk cara yang paling gampang dilakukan untuk pemula yang baru belajar Bootstrap.

Kita cukup copy link CDN Bootstrap, lalu di taruh pada kode HTML.

Tapi syaratnya kita harus terhubung dengan internet, karena Bootstrap akan diambil dari server CDN.

CDN

CDN adalah singkatan dari *Content Delivery Network*, semacam server yang tersebar di seluruh dunia untuk mengantarkan konten secara optimal dan cepat.



Buat file `hello-bootstrap.html`

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hello Bootstrap</title>

  <!-- CSS Bootstrap -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
</head>

<body>
  <header class="bg-primary py-5">
    <div class="container">
      <h1 class="display-4 text-white">Hello Bootstrap!</h1>
    </div>
  </header>

  <!-- JavaScript Bundle with Popper -->

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+lp" crossorigin="anonymous"></script>
</body>
</html>
```

PENJELASAN

Cukup dengan menambahkan ini di dalam tag `<head>`:

```
<!-- CSS Bootstrap -->
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">
```

dan load Javascript di dalam `<body>`:

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"
crossorigin="anonymous"></script>
```

Atribut `integrity` dan `crossorigin` berfungsi untuk mengatur keamanan dan integritas file.

Jadi file `bootstrap.min.css` yang kita terima dari CDN benar file aslinya, bukan hasil modifikasi orang.

2. Download Bootstrap Secara Manual

Download  [Bootstrap-5.0.0.zip](#)

Ekstrak folder `css` dan `js` ke dalam folder `belajar-bootstrap`.

Setelah itu, buat file HTML baru dengan nama `bs-lokal.html`

Kemudian, isi dengan kode berikut

(Kelebihan cara ini adalah kita tidak perlu terhubung ke internet seperti pada CDN untuk dapat menggunakan Bootstrap, karena Bootstrap-nya sudah ada di aset lokal.)

Buat file `bs-lokal.html`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap dari Lokal</title>
  <link rel="stylesheet" href="css/bootstrap.min.css" />

</head>
<body>

  <header class="py-5 bg-success">
    <div class="container">
      <h1 class="display-4 text-white">Hello Bootstrap Lokal!</h1>
    </div>
  </header>

  <script src="js/bootstrap.min.js"></script>

</body>
</html>
```

PENJELASAN

Pada kode di atas, kita menggunakan bootstrap dari asset lokal dengan tag `<link>`.

```
<link rel="stylesheet"
href="css/bootstrap.min.css" />
```

dan tag `<script>` untuk menggunakan script dari Bootstrap.

```
<script src="js/bootstrap.min.js"></script>
```

3. DENGAN PACKAGE MANAGER

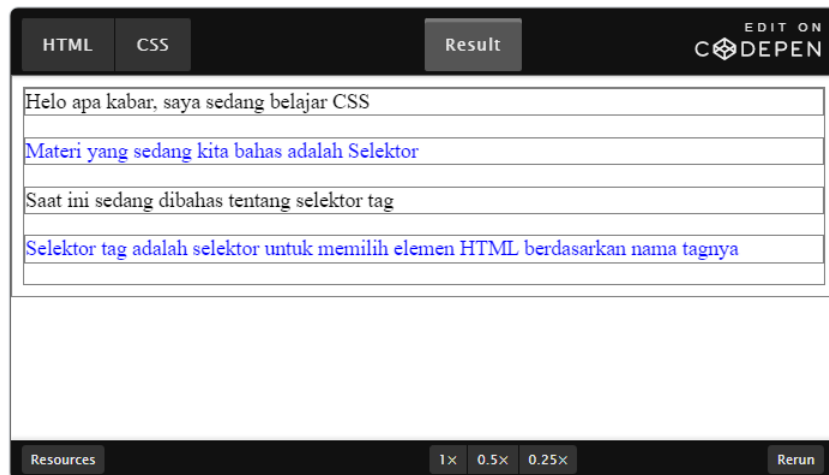
Package manager adalah sebuah program untuk manajemen paket pada proyek web. Contohnya: NPM, Composer, Yarn, PNPM, dan sebagainya.

Menggunakan Bootstrap dengan package manager tidak saya direkomendasikan untuk pemula, karena akan membutuhkan pengetahuan lain seperti cara kerja package manager itu sendiri dan ekosistem yang dipakai.

5) Selektor Atribut

Selektor universal adalah selektor yang digunakan untuk menyeleksi semua elemen pada jangkauan (*scope*) tertentu.

Code di samping, membuat semua elemen akan memiliki garis *solid* dengan ukuran *1px* dan berwarna *grey*.



```
* {  
  border: 1px solid grey;  
}
```



PHP Fundamental

Apa itu PHP?

- Open Source
- Server-side Scripting
- Banyak digunakan untuk web development
- Ekstensi file .php
- Kode PHP diawali dengan `<?php` dan diakhiri `?>`
- Setiap statement diakhiri dengan *semicolon* (;)

PHP

PHP Hypertext Preprocessor

```
<?php  
echo "Hello World";  
?>
```

Variable

- Tempat untuk menyimpan data secara sementara
- Di PHP variable bisa menampung berbagai tipe data dan bisa berubah-ubah tipe data
- Harus diawali dengan \$
- Karakter pertama harus huruf atau *underscore* (_)
- Penamaan variable tidak boleh mengandung spasi
- *Case Sensitive*

```
<?php
    $i;
    $nama;
    $Umur;
    $_lokasi_memori;
    $ANGKA_MAKSIMUM;

?>
```

Komentar

- Baris yang tidak dieksekusi
- Menjelaskan fungsi dari kode kita

```
<?php
// Komentar 1 baris
# Komentar 1 baris juga

/*

Ini adalah beberapa baris
komentar dalam satu blok yang
sama

*/

?>
```


Tipe Data Number

- Di PHP terdapat 2 jenis tipe data number
 - integer: bilangan bulat
 - float: bilangan pecahan

Dapat berupa bilangan positif dan negatif

Tipe Data String

- Tipe data string adalah tipe data yang representasi dari teks
- String bisa mengandung kosong atau banyak karakter
- Untuk membuat string di PHP ada 2 cara:
 - *Single Quoted*
 - *Double Quoted*

Tipe Data Boolean

- Tipe data paling sederhana
- Hanya memiliki nilai **true** dan **false**
- Sifatnya *case insensitive*

OPERATOR

Operator

- PHP membagi operator dalam beberapa kelompok
 - Operator Aritmatika
 - Operator Assignment
 - Operator Increment / Decrement
 - Operator Perbandingan
 - Operator Logika
-

Operator Aritmatika

| Contoh | Nama Operator | Keterangan |
|--------------|---------------|-------------------------------------|
| $+\$x$ | Positif | Nilai positif dari $\$x$ |
| $-\$x$ | Negatif | Nilai negatif dari $\$x$ |
| $\$x + \y | Penambahan | Total dari $\$x$ dan $\$y$ |
| $\$x - \y | Pengurangan | Selisih dari $\$x$ dan $\$y$ |
| $\$x * \y | Perkalian | Hasil kali dari $\$x$ dan $\$y$ |
| $\$x / \y | Pembagian | Hasil bagi dari $\$x$ dan $\$y$ |
| $\$x \% \y | Modulus | Sisa dari pembagian $\$x$ dan $\$y$ |
| $\$x ** \y | Pangkat | Hasil pangkat dari $\$x$ dan $\$y$ |

Operator Assignment

| Assignment | Sama seperti.... | Keterangan |
|----------------|--------------------|---|
| $\$x = \y | $\$x = \y | Operan kiri diatur ke nilai ekspresi di sebelah kanan |
| $\$x += \y | $\$x = \$x + \$y$ | Penjumlahan |
| $\$x -= \y | $\$x = \$x - \$y$ | Pengurangan |
| $\$x *= \y | $\$x = \$x * \$y$ | Perkalian |
| $\$x /= \y | $\$x = \$x / \$y$ | Pembagian |
| $\$x \% = \y | $\$x = \$x \% \$y$ | Modulus |
| $\$x ** = \y | $\$x = \$x ** \$y$ | Pangkat |

Operator Increment / Decrement

| Operator | Keterangan |
|--------------------|---|
| <code>++\$x</code> | Menambahkan satu <code>\$x</code> kemudian return <code>\$x</code> |
| <code>\$x++</code> | return <code>\$x</code> kemudian menambahkan <code>\$x</code> satu |
| <code>--\$x</code> | Mengurangkan satu <code>\$x</code> kemudian return <code>\$x</code> |
| <code>\$x--</code> | return <code>\$x</code> kemudian dikurangkan 1 |

Operator Perbandingan (1)

| Operator | Nama | Contoh | Keterangan |
|----------|-------------------|-------------------------------|---|
| == | sama dengan | <code>\$a == \$b</code> | <i>true</i> apabila <code>\$a</code> sama dengan <code>\$b</code> |
| === | identik | <code>\$a === \$b</code> | <i>true</i> apabila <code>\$a</code> sama dengan <code>\$b</code> dan tipe mereka sama |
| != | tidak sama dengan | <code>\$a != \$b</code> | <i>true</i> apabila <code>\$a</code> tidak sama dengan <code>\$b</code> |
| <> | tidak sama dengan | <code>\$a <> \$b</code> | <i>true</i> apabila <code>\$a</code> tidak sama dengan <code>\$b</code> |
| !== | tidak identik | <code>\$a !== \$b</code> | <i>true</i> apabila <code>\$a</code> tidak identik dengan <code>\$b</code> dan tipe juga tidak sama |
| > | lebih besar dari | <code>\$a > \$b</code> | <i>true</i> apabila <code>\$a</code> lebih besar dari <code>\$b</code> |

Operator Perbandingan (2)

| Operator | Nama | Contoh | Keterangan |
|----------|---------------------------|----------------------------|---|
| < | lebih kecil dari | <code>\$a < \$b</code> | <i>true</i> apabila <code>\$a</code> lebih kecil dari <code>\$b</code> |
| >= | lebih besar / sama dengan | <code>\$a >= \$b</code> | <i>true</i> apabila <code>\$a</code> lebih besar / sama dengan <code>\$b</code> |
| <= | lebih kecil / sama dengan | <code>\$a <= \$b</code> | <i>true</i> apabila <code>\$a</code> lebih kecil / sama dengan <code>\$b</code> |

Operator Logika (1)

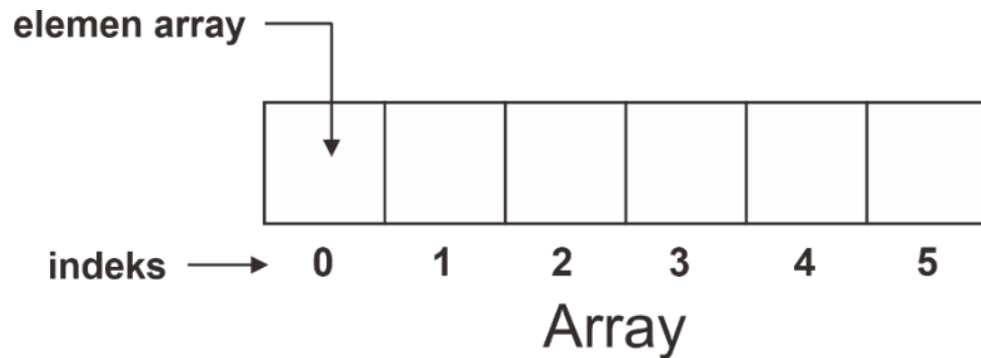
| Operator | Nama | Contoh | Keterangan |
|----------|------|---------------------------------|--|
| and | And | <code>\$a and \$b</code> | true apabila keduanya (<code>\$a</code> dan <code>\$b</code>) bernilai true |
| or | Or | <code>\$a or \$b</code> | true apabila salah satu (<code>\$a</code> atau <code>\$b</code>) bernilai true |
| xor | Xor | <code>\$a xor \$b</code> | true apabila salah satu saja dari <code>\$a</code> atau <code>\$b</code> bernilai true, bukan keduanya |
| && | And | <code>\$a && \$b</code> | true apabila keduanya (<code>\$a</code> dan <code>\$b</code>) bernilai true |

Operator Logika (2)

| Operator | Nama | Contoh | Keterangan |
|----------|------|------------|--|
| | Or | \$a \$b | true apabila salah satu (\$a atau \$b) bernilai true |
| ! | Not | !\$a | true apabila \$a bernilai false |

ARRAY

Overview Array



- Struktur data yang dapat menyimpan sekumpulan data dengan tipe data sama
- Di PHP, tipe data tidak harus sama
- Keyword `[]` atau `array()`
- Tersimpan di memory
- Terdiri elemen array dan index

Array

```
$myArray = ['satu', 2, '3'];  
echo $myArray[0];    // output satu  
echo $myArray[1];    // output 2  
echo $myArray[2];    // output 3
```



Jenis Array

Array Index

Array
Associative

Array
Multidimensi

Array Index

- Untuk mengakses elemennya berdasarkan nomor index
- Index array di mulai dari 0

```
$myArray = ['satu', 2, '3'];  
echo $myArray[1];    // output 2
```

Array Associative

- Array yang memiliki pasangan key dan value
- Untuk mendapatkan nilainya kita menggunakan key bukan nomor index

```
$myArray = [  
    'Bulan 1' => 'Januari',  
    'Bulan 2' => 'Februari',  
    'Bulan 3' => 'Maret'  
];  
  
echo $myArray['Bulan 1'];    // output Januari
```

Array Multidimensi

- Array yang memiliki dimensi lebih dari satu array
- Biasanya digunakan untuk membuat matrik, graph, dan struktur data yang rumit
- Array yang memiliki dimensi lebih dari tiga seringkali sulit dikelola
- Dimensi array dapat diketahui dari jumlah [] atau keyword **array**

Array Multidimensi – 2D

| | | Kolom | | |
|--------|---|-------|---|---|
| indeks | | 0 | 1 | 2 |
| Baris | 0 | A | B | C |
| | 1 | D | E | F |
| | 2 | G | H | I |

```
// array 2 dimensi
$pegawai = [
    ['Andi', 'L', 'Surabaya'],
    ['Cyntia', 'P', 'Jakarta'],
    ['Thomas', 'L', 'Solo']
];

echo $pegawai[1][0];
// output Cyntia
echo $pegawai[1][2];
// output Surabaya
```

Operasi Array

| Operasi | Keterangan |
|-------------------------------------|---|
| <code>\$array[index]</code> | Mengakses data di array pada nomor index |
| <code>\$array[index] = value</code> | Mengubah data di array pada nomor index dengan value baru |
| <code>\$array[] = value</code> | Menambah data di array pada posisi paling belakang |
| <code>unset(\$array[index])</code> | Menghapus data di array, index otomatis hilang dari array |
| <code>count(\$array)</code> | Mengambil total data di array |

Struktur Kontrol

Pernyataan If

- If adalah salah satu perintah yang digunakan untuk percabangan
- Kode di dalam pernyataan if akan dieksekusi ketika kondisi if bernilai **true**
- Sintaks:

```
if(kondisi){  
    // proses yang dilakukan apabila kondisi bernilai benar (true)  
}
```

Pernyataan If (Contoh)

```
$nilai = 70;  
  
if ($nilai >= 65) { // dijalankan ketika kondisinya true  
    echo "Selamat Anda Lulus";  
}
```



Pernyataan If-Else

- Untuk memproses dua kondisi **true** dan **false**
- Kode di dalam pernyataan `if` akan dieksekusi ketika kondisi `if` bernilai **true**
- Kode di dalam pernyataan `else` akan dieksekusi ketika kondisi `if` bernilai **false**
- ```
if(kondisi) {
 // proses yang dilakukan apabila kondisi bernilai benar (true).
} else {
 // proses yang dilakukan apabila kondisi bernilai salah (false).
}
```

# Pernyataan If-Else (Contoh)

---

```
$nilai = 60;

if ($nilai >= 65) { //dijalankan ketika kondisinya true
 echo "Selamat Anda Lulus";
} else { //dijalankan ketika kondisinya false
 echo "Mohon maaf Anda Tidak Lulus";
}
```



# Pernyataan If-Elseif-Else

---

- Untuk memproses lebih dari dua kondisi
- Sintaks:

```
if (kondisi1) {
 // kode dijalankan apabila kondisi1 bernilai benar (true)
} elseif (kondisi2) {
 // kode dijalankan apabila kondisi2 bernilai benar (true)
} else {
 // kode dijalankan apabila semua kondisi bernilai salah (false)
}
```

# Pernyataan If-Elseif-Else (Contoh)

---

```
$nilai = 70;
if ($nilai >= 85) {
 echo "Selamat Anda mendapat Nilai A";
} elseif($nilai>=70) {
 echo "Selamat Anda mendapat Nilai B";
} elseif($nilai>=60) {
 echo "Anda mendapat Nilai C";
} else {
 echo "Anda mendapat Nilai D";
}
```



# Pernyataan Switch

---

- Switch adalah salah satu perintah untuk percabangan dengan kondisi yang lebih sederhana
- Hanya untuk operator perbandingan ==

# Pernyataan Switch (Sintaks)

---

```
switch ($nilai) {
 case label1:
 // Kode dijalankan apabila nilai=label1
 break;
 case label2:
 // Kode dijalankan apabila nilai=label2
 break;
 ...
 default:
 // Kode dijalankan apabila nilai tidak cocok dengan semua label
}

```



# Pernyataan Switch (Contoh)

---

```
$nilai="A";
switch($nilai) {
 case 'A':
 echo "Nilai Anda sangat memuaskan";
 break;
 case 'B':
 echo "Nilai Anda memuaskan";
 break;
 case 'C':
 echo "Nilai Anda cukup";
 break;
 case 'D':
 echo "Nilai Anda kurang";
 break;
```

```
 case 'E':
 echo "Anda tidak lulus";
 break;
 default :
 echo "Nilai tidak terdaftar";
 break;
}
```



# Ternary Operator

---

- Versi sederhana dari pernyataan **if-else**
- Ternary operator menggunakan keyword **?** dan **:**
- Sintaks

```
(condition) ? (statement1) : (statement2)
```



# Ternary Operator (Contoh)

---

```
$nilai=70;
echo ($nilai>=80) ? "Anda Lulus" : "Anda tidak Lulus";
```



# Perulangan For

---

- For adalah salah satu perintah yang digunakan untuk perulangan
- Blok kode di dalam for akan selalu diulangi selama kondisi for terpenuhi
- Sintaks

```
for(initial; condition; increment) {
 // kode yang harus dieksekusi;
}
```

# Perulangan For (contoh)

---

```
for ($i = 1; $i <= 15; $i++) {
 echo "Angka $i
";
}
```



# Perulangan While

---

- Perulangan akan terjadi selama kondisi yang dihasilkan bernilai true
- Sintaks

```
while (kondisi) {
 // kode yang harus dieksekusi;
}
```

# Perulangan While (Contoh)

---

```
$i = 1;

while ($i <= 15) {
 echo "Angka $i
";
 $i++;
}
```



# Perulangan Do-While

---

- Perulangan ini hampir mirip dengan perulangan while
- Perbedaannya hanya pada pengecekan kondisi
- Pengecekan kondisi dilakukan di akhir perulangan
- Sintaks

```
do {
 // kode yang harus dieksekusi;
} while (kondisi);
```

# Perulangan Do-While (Contoh)

---

```
$i = 1;

do {
 echo "Angka $i
";
 $i++;
} while ($i <= 15);
```



# Perulangan Foreach

---

- Perulangan khusus untuk mengakses data array
- Sintaks

```
foreach ($nama_array as $value) {
 statement
}
// atau
foreach ($nama_array as $key => $value) {
 statement
}
```



# Perulangan Tanpa Foreach (Contoh)

---

```
$items = ['Andi', 'Jakarta', 'IT Staf'];

for ($i = 0; $i < count($items); $i++) {
 echo $items[$i] . '</ br>';
}
```



# Perulangan Foreach (Contoh)

---

```
$items = ['Andi', 'Jakarta', 'IT Staf'];

foreach ($items as $item) {
 echo $item . '</ br>';
}
```



# Perulangan Foreach dengan Key (Contoh)

---

```
$items = [
 'nama' => 'Andi',
 'kota' => 'Jakarta',
 'Pekerjaan' => 'IT Staf'
];

foreach ($items as $key => $value) {
 echo $key . ':' . $value . '</ br>';
}
```



# Function

---

# Function

---

- Blok kode program yang akan berjalan saat kita panggil
- Biasanya digunakan untuk membungkus kode program yang akan dipanggil berulang
- Di bahasa pemrograman lain, *function* juga disebut *method*
- Dikelompokkan menjadi 2 jenis:
  - *Function* dengan pengembalian nilai (biasanya disebut *function*)
  - *Function* tanpa pengembalian nilai (biasanya disebut *procedure*)

# Sintaks Function

---

```
function namaFungsi($parameter) {
 // kode program
 [return $result]
}
```



# Function (tanpa return)

---

```
function perkalian($angka1, $angka2) {
 $hasil= $angka1 * $angka2;

 echo 'Hasil perkalian '.$angka1.'x'.$angka2.': '.$hasil;
}

perkalian(4, 5);
```



# Function (dengan return)

---

```
function perkalian($angka1, $angka2) {
 $hasil= $angka1 * $angka2;

 return $hasil;
}
```

```
$hasil = perkalian(4, 5);
echo 'Hasil perkalian 4x5:' . $hasil;
```





# Mengelola File

---

- Pada PHP ada perintah untuk mengelola sebuah file
  - **readfile**: untuk membaca data dari dalam file
  - **fopen**: untuk membaca dan menulis data dari sebuah file
- Sintaks

```
readfile(filename);
```

```
fopen(filename, mode);
```

# Membaca file (contoh)

---

```
<?php
echo readfile("file1y.txt");
?>
```



# Membaca dan menulis file (contoh)

---

```
<?php
$myfile = fopen("testfile.txt", "w")
?>
```



# Membaca dan menulis file (contoh 2)

---

```
<?php

$myfile = fopen("latihanfile.txt", "w") or die("File error");

$txt = "Asipti baris 1\n";

fwrite($myfile, $txt);

$txt = "Asipti baris 2\n";

fwrite($myfile, $txt);

fclose($myfile);

?>
```



# Mengelola File

---

- Pada PHP ada perintah untuk mengelola sebuah file
  - **file\_put\_content**: untuk menulis / mengubah data ke dalam file
  - **file\_get\_content**: untuk membaca data dari sebuah file
- Sintaks

```
file_put_contents(filename, data, opsi);
```

```
file_get_contents(filename);
```

# Menulis file (contoh)

---

```
<?php

$pesan="Ini hasil membuat file melalui PHP";
file_put_contents("test.txt",$pesan);

$tambahPesan = "tambah pesan di file test.txt";
file_put_contents("test.txt", $tambahPesan, FILE_APPEND);

?>
```



# Membaca file (contoh)

---

```
$data = file_get_contents("test.txt");
echo $data;
```



# Penyimpanan Data Array ke File

---

- Ada macam-macam penyimpanan data *array*
  - Serialized (serialize dan unserialize)
  - XML
  - JSON (json\_encode dan json\_decode)



# Serialize (contoh)

---

```
<?php
$karyawan = [
 ['nama'=>'Toni', 'alamat'=>'Bandung'],
 ['nama'=>'Naufal', 'alamat'=>'Bandung'],
 ['nama'=>'Davi', 'alamat'=>'Jakarta']
];

file_put_contents('data.txt', serialize($karyawan));
?>
```



# Unserialize (contoh)

---

```
$getData = file_get_contents('data.txt');
$hasil = unserialize($getData);

echo '<pre>';
print_r($hasil);
echo '</pre>';
```



# json\_encode (Contoh)

---

```
<?php
$karyawan = [
 ['nama'=>'Toni', 'alamat'=>'Bandung'],
 ['nama'=>'Naufal', 'alamat'=>'Bandung'],
 ['nama'=>'Davi', 'alamat'=>'Jakarta']
];

file_put_contents('data.txt', json_encode($karyawan));
?>
```



# json\_decode (contoh1)

---

```
$getData = file_get_contents('data.txt');
$hasil = json_decode($getData);

echo '<pre>';
print_r($hasil);
echo '</pre>';

echo $hasil[0]->nama;
```



# json\_decode (contoh2)

---

```
$getData = file_get_contents('data.txt');
$hasil = json_decode($getData, true);

echo '<pre>';
print_r($hasil);
echo '</pre>';

echo $hasil[0]['nama'];
```



# Form method POST & GET

---

POST : Methode POST untuk pengiriman datanya secara langsung

GET : Pengiriman datanya secara tidak langsung yang artinya saat kita menginput data seperti nama, alamat, dll maka data tersebut akan tampil di URL



# Form method POST & GET

---

```
<form method="POST" action="proses.php">
 <label>Siapa Nama Anda : </label>
 <input type="text" name="nama" placeholder="Masukan Nama
Anda">
 <button type="Submit">Kirim</button>
</form>

<form method="GET" action="proses.php">
 <label>Siapa Nama Anda : </label>
 <input type="text" name="nama" placeholder="Masukan Nama
Anda">
 <button type="Submit">Kirim</button>
</form>
```



# Form method POST & GET

---

```
<?php
$nama = $_POST['nama'];
echo "Nama Anda Adalah $nama";
?>
```

```
<?php
$nama = $_GET['nama'];
echo "Nama Anda Adalah $nama";
?>
```







# Thank you

---