

PERTEMUAN 4

PELATIHAN JUNIOR WEB DEVELOPER

**Menerapkan Perintah Eksekusi Bahasa Pemrograman
berbasis Teks, Grafik, dan Multimedia**

Course Definition

KODE UNIT : J.620100.010.02

JUDUL UNIT : Menerapkan Perintah Eksekusi Bahasa Pemrograman Berbasis Teks, Grafik, dan Multimedia

DESKRIPSI UNIT: Unit kompetensi ini berhubungan dengan sikap, pengetahuan, dan keterampilan yang dibutuhkan untuk dapat menjalankan *code* yang dibuat pada lingkungan *tools* pemrograman tertentu.

ELEMEN KOMPETENSI	KRITERIA UNJUK KERJA
1. Mengidentifikasi mekanisme <i>running</i> atau eksekusi <i>source code</i>	1.1 Cara dan <i>tools</i> untuk mengeksekusi <i>source code</i> diidentifikasi. 1.2 Parameter untuk mengeksekusi <i>source code</i> diidentifikasi. 1.3 Peletakan <i>source code</i> sehingga bisa dieksekusi dengan benar diidentifikasi.
2. Mengeksekusi <i>source code</i>	2.1 <i>Source code</i> dieksekusi sesuai dengan mekanisme eksekusi <i>source code</i> dari <i>tools</i> pemrograman yang digunakan. 2.2 Perbedaan antara <i>running</i> , <i>debugging</i> , atau membuat <i>executable file</i> diidentifikasi.
3. Mengidentifikasi hasil eksekusi	3.1 <i>Source code</i> berhasil dieksekusi sesuai skenario yang direncanakan. 3.2 Jika eksekusi <i>source code</i> gagal/tidak berhasil, sumber permasalahan diidentifikasi.

Learning Objectives

Tujuan Umum

Setelah mempelajari bahan tayang ini peserta latih diharapkan mampu mengidentifikasi tools dasar pemrograman, element pada HTML, perbedaan eksekusi kode HTML dan PHP, serta mengidentifikasi jenis kesalahan pada program.

Tujuan Khusus

Adapun tujuan mempelajari unit kompetensi ini guna memfasilitasi peserta latih sehingga pada akhir pelatihan diharapkan memiliki kemampuan Menerapkan perintah eksekusi bahasa pemrograman berbasis teks, grafik, dan multimedia, termasuk diantaranya adalah mengidentifikasi mekanisme running atau eksekusi source code.

Materi yang disampaikan

1. Tools untuk mengeksekusi source code
2. Pengenalan Javascript
3. Skema Eksekusi HTML dan PHP
4. Jenis Kesalahan dalam Mengeksekusi Source Code

Pre-Test

1. Berikut ini yang tidak termasuk tools untuk mengeksekusi source code adalah

- a. XAMPP
- b. Visual Studio Code
- c. WhatsApp
- d. Notepad++

2. Jenis kesalahan dalam mengeksekusi source berikut ini yang terjadi ketika variabel yang dipanggil atau dieksekusi belum didefinisikan adalah

- a. Fatal Error
- b. Notice Error
- c. Warning Error
- d. Parse Error

Tools Pemrograman

Tools pemrograman adalah perangkat lunak yang digunakan untuk mengeksekusi source code hasil pemrograman, tools pemrograman bisa berupa compiler atau interpreter.

Contoh tools pemrograman pada Web misalnya HTML dan PHP

Mekanisme eksekusi Source Code dilakukan dengan menggunakan Tools yakni Tag HTML, Tag PHP, Comand Prompt dan XAMPP.

Tools Dasar Pemrograman

- ❖ **XAMPP** adalah perangkat yang menggabungkan tiga aplikasi kedalam satu paket, yaitu Apache, MySQL, dan PHPMyAdmin.
- ❖ **Visual Studio Code** adalah Text editor yang digunakan untuk menuliskan kode program.
- ❖ **Apache** adalah server web yang dapat dijalankan di banyak sistem operasi, yang berguna untuk melayani dan memfungsikan situs web.
- ❖ **MySQL** adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user.
- ❖ **PHP**: Hypertext Preprocessor adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML.



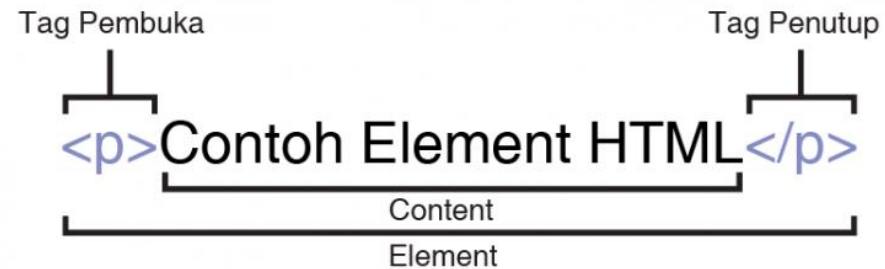
Review Tools Dasar Pemrograman

- ❖ **HTML** (HyperText Markup Language) yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web Internet (Browser).
- ❖ **CSS** (Cascading Style Sheets) berisi rangkaian instruksi yang menentukan bagaimana suatu text akan tertampil di halaman web.
- ❖ **JavaScript** adalah bahasa pemrograman web yang bersifat Client Side Programming Language, yang pemrosesannya dilakukan oleh client.



Element dan Tag pada HTML

Element pada HTML membentuk sebuah struktur dan memberitahu browser untuk halaman website yang akan ditampilkan. Biasanya untuk element terdiri dari tag pembuka, konten, dan tag penutup.



Berikut adalah salah satu contoh element pada HTML:

`<p> Paragraf di HTML </p>`

Attribut pada HTML

Setelah Anda memahami element dan tag, Anda perlu memahami apa itu attribute. Seperti diketahui jika sebuah element berupa struktur pada HTML yang berfungsi untuk memberitahu browser mengenai apa yang akan ditampilkan.

Attribute



```
<p class="contoh-attribut">contoh attribut</p>
```

Berikut adalah contoh element yang disertai sebuah attribute:

```
<p>Berikut adalah <a href="https://www.google.com/" title="Search" target="_blank"> Engine terbaik </a>.</p>
```

Contoh HTML

Langkah Percobaan :

Bukalah sebuah editor teks sebagai contoh menggunakan Visual Studio Code

```
C: > xampp > htdocs > digitalent > <> latihan.html > ...  
1   <html>  
2   |   <b>Tulisan ini dicetak tebal</b>  
3   </html>
```

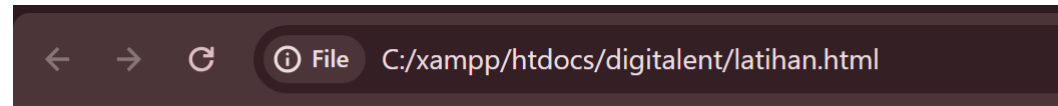
Kemudian lakukan penyimpanan file pada sebuah folder dengan ekstensi .html

Contoh: latihan.html

Contoh HTML

Setelah file tersebut disimpan maka buka kembali file tersebut menggunakan salah satu Browser (Firefox, Chrome, dll)

Tampilan Browser pada saat dieksekusi:



Tulisan ini dicetak tebal

Contoh Membuat Table Menggunakan HTML

```
1 <html>
2   <head>
3     <title>Contoh Membuat Tabel</title>
4   </head>
5   <body>
6     <table border="1">
7       <tr>
8         <td>Kolom 1 baris 1</td>
9         <td>Kolom 2 baris 1</td>
10      </tr>
11     <tr>
12       <td>Barang 1</td>
13       <td>Barang 2</td>
14     </tr>
15   </table>
16 </body>
17 </html>
18
```

Jika kode di sebelah kiri dijalankan pada browser, maka akan menampilkan table seperti di bawah ini:

Kolom 1 baris 1	Kolom 2 baris 1
Barang 1	Barang 2

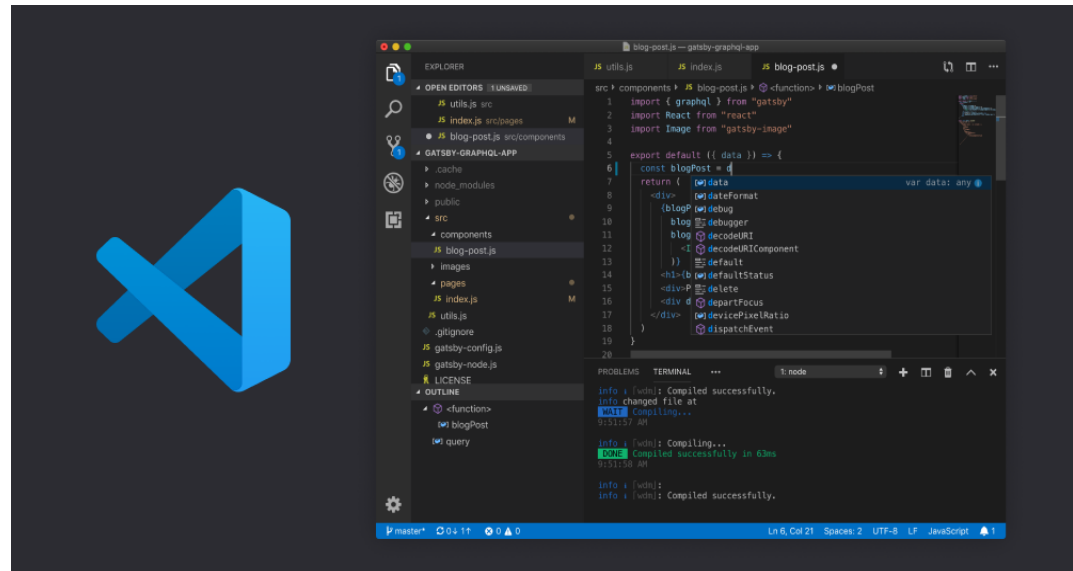
Apa itu javascript?

- ❖ Javascript adalah bahasa pemrograman yang awalnya dirancang untuk berjalan di atas browser.
- ❖ Namun, seiring perkembangan zaman, javascript tidak hanya berjalan di atas browser saja. Javascript juga dapat digunakan pada sisi Server, Game, IoT, Desktop, dan sebagainya.
- ❖ Javascript awalnya bernama Mocha, lalu berubah menjadi LiveScript saat browser Netscape Navigator 2.0 rilis versi beta (September 1995). Namun, setelah itu dinamai ulang menjadi Javascript.

Peralatan untuk Belajar Javascript

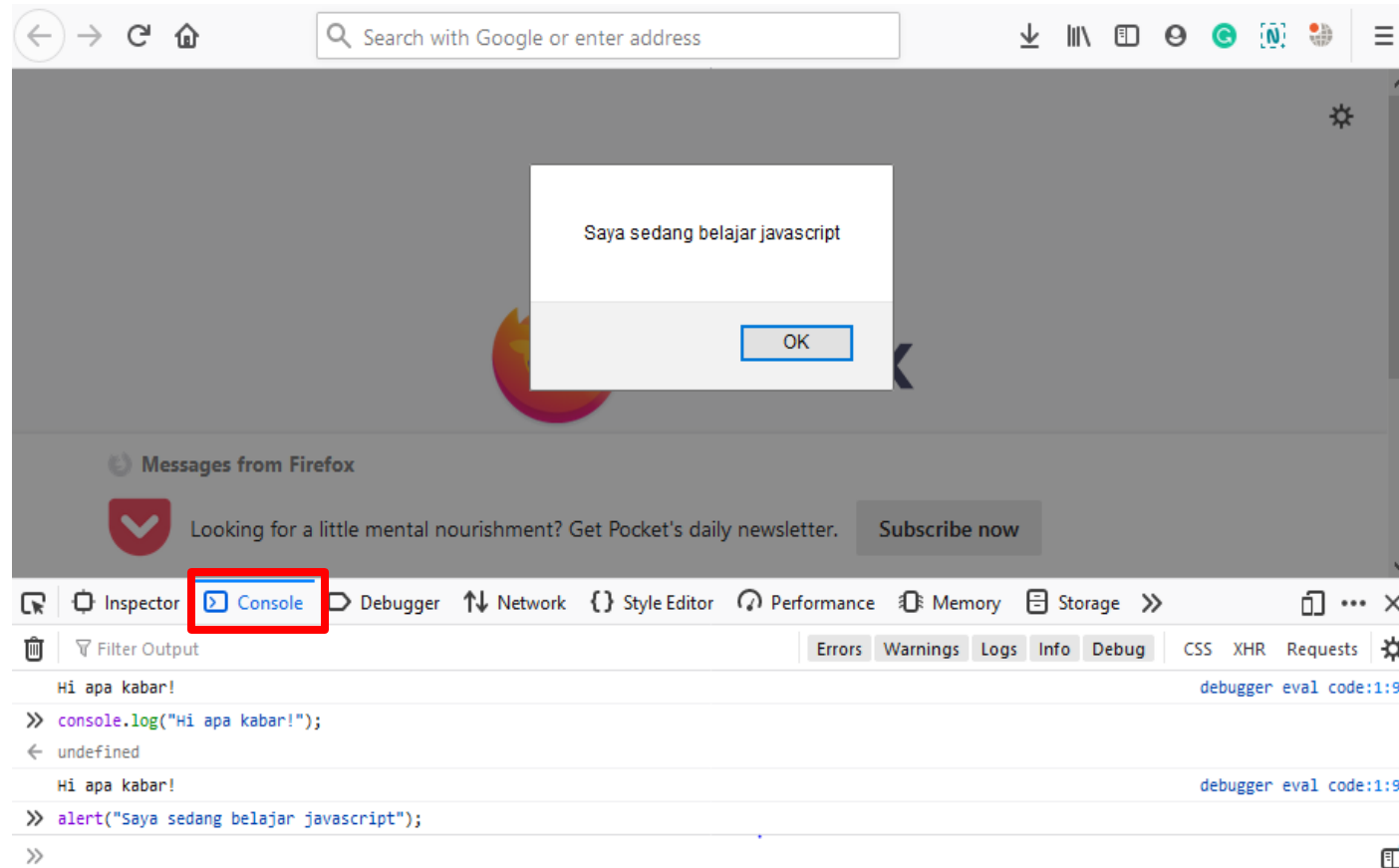
Apa saja peralatan yang harus disiapkan untuk belajar Javascript?

- ❖ Web Browser (Google Chrome, Firefox, Opera, dll)
- ❖ Teks Editor (rekomendasi: VS Code)



Mengenal Console JavaScript

- ❖ Console Javascript dapat di buka melalui Inspect Element > Console.

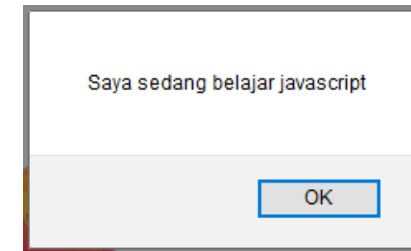


Mengenal Console JavaScript

- ❖ Di dalam console, kita bisa menulis fungsi atau kode-kode javascript dan hasilnya akan langsung ditampilkan.
- ❖ Misalnya, mari kita coba kode berikut:

```
console.log("Hi apa kabar!");  
alert("Saya sedang belajar javascript");
```

- ❖ Maka hasilnya:



Membuat Program Javascript

Buka teks editor, kemudian buat file baru bernama hello_world.html dan isi dengan kode berikut:

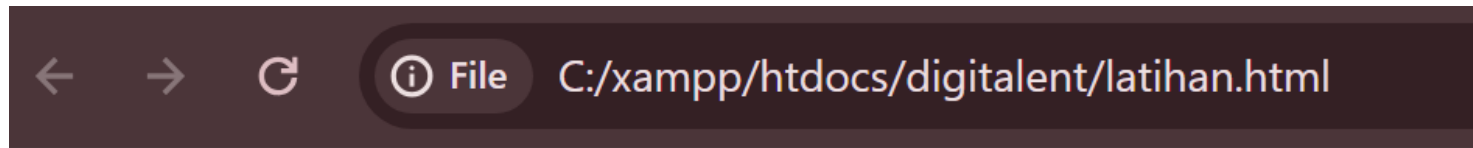
```
1  <html>
2    <head>
3      <title>Contoh Membuat Tabel</title>
4    </head>
5    <body>
6      <script>
7        console.log("Saya ingin berbelanja");
8        document.write("Selamat datang, Selamat berbelanja");
9      </script>
10   </body>
11 </html>
```

berbelanja

Membuat Program Javascript

Simpan dengan nama latihan.html, kemudian buka file tersebut dengan web browser.

Maka hasilnya:



Selamat datang, Selamat berbelanja

Membuat Program Javascript

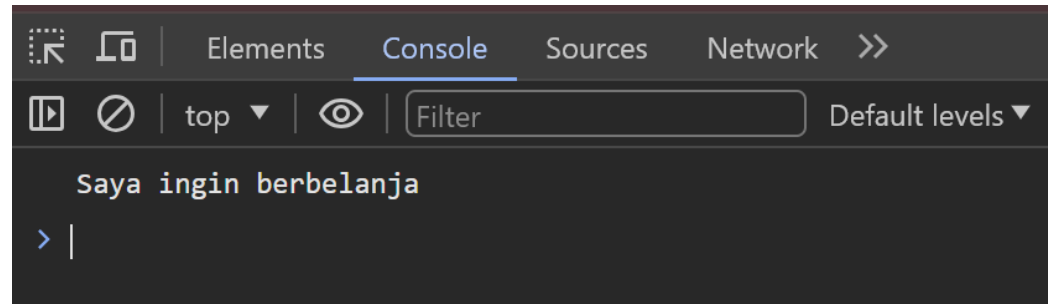
Tunggu sebentar...
Tadi kita menulis perintah:

```
console.log("Saya ingin berbelanja");
```

Mengapa tidak ditampilkan?

Membuat Program Javascript

Karena perintah atau fungsi `console.log()` akan menampilkan pesan ke dalam console javascript. Sedangkan perintah `document.write()` berfungsi untuk menulis ke dokumen HTML.



Buka console javascript.
Maka kita akan melihat pesan:
"Saya ingin berbelanja"

Cara Menulis Kode Javascript di HTML

Pada contoh di atas, kita sudah menulis kode javascript di dalam HTML. Cara tersebut merupakan cara penulisan *embeded* (ditempel). Masih ada beberapa cara lagi yang perlu kita ketahui:

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

Penulisan Kode javascript dengan Embed

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

Contoh:

```
<!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>
```

Penulisan Kode javascript Inline

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:

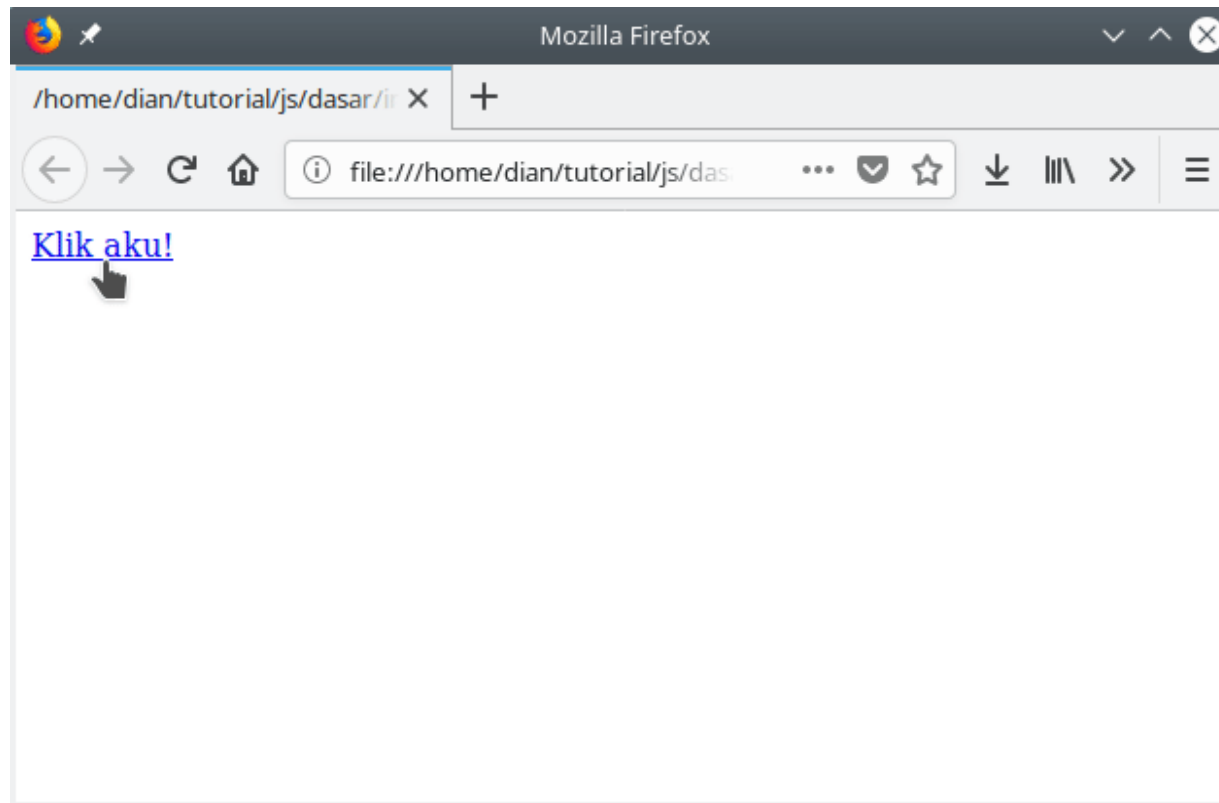
```
<a href="#" onclick="alert('Yey!')">Klik aku!</a>
```

atau bisa juga seperti ini:

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

Penulisan Kode javascript Inline

Hasil:



Penulisan Kode javascript Inline

Perhatikan,

Atribut onclick merupakan atribut HTML untuk menyatakan fungsi yang akan dieksekusi saat elemen itu diklik.

Pada contoh sebelumnya, 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.

Penulisan Kode javascript Eksternal

Pada 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.

buat dua file, yaitu: file HTML dan Javascript :

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

Penulisan Kode javascript Eksternal

Isi file kode-program.js:

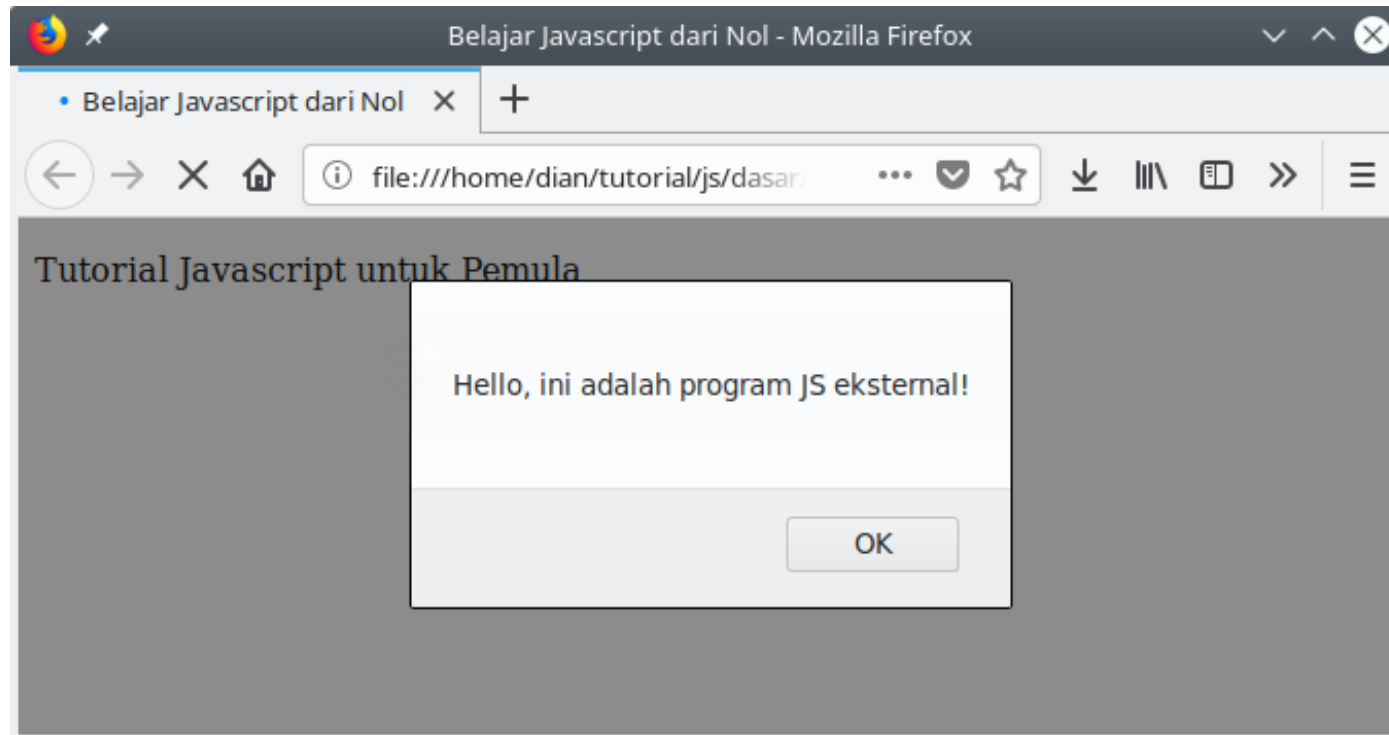
```
alert("Hello, ini adalah program JS eksternal!");
```

Isi 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>
```

Penulisan Kode javascript Eksternal

Hasilnya:



Penulisan Kode javascript Eksternal

Pada contoh tadi, kita menulis kode javascript terpisah dengan kode HTML.

Lalu, pada kode HTML Kita menyisipkannya dengan memberikan atribut src pada tag script.

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

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

Penulisan Kode javascript Eksternal

Bagaimana kalau file javascriptnya berada di folder yang berbeda?

Kita bisa menuliskan alamat lengkap foldernya.

Contoh:

Misal kita punya struktur folder seperti ini:

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

Penulisan Kode javascript Eksternal

Maka untuk menyisipkan file kode-program.js ke dalam HTML, kita bisa menuliskannya seperti ini:

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

Karena file kode-program.js berada di dalam direktori js. Kita juga bisa menyisipkan javascript yang ada di internet dengan memberikan alamat URL lengkapnya. Contoh:

```
<script src="https://www.petanikode.com/js/kode.js"></script>
```

Penulisan Kode javascript HTML

Javascript sebagai bahasa pemrograman yang berjalan di atas browser harus di tulis di dalam HTML.

Ada empat cara penulisan javascript pada HTML.

1. Tag Script

Cara yang umum yang dipakai adalah menuliskannya dalam tag **script**.
Tag **script** bisa dibuat di dalam tag **head**, maupun di dalam tag **body**. Contoh:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Penulisan Javascript</title>
  <script>
    console.log("Hi, ini kode Javascript");
  </script>
</head>
<body>

  <script>
    document.write("Javascript itu keren!");
  </script>
</body>
</html>
```

2. File Eksternal

Bila tidak ingin kode Javascript bercampur aduk dengan HTML, kita bisa menuliskannya pada file terpisah. Caranya, buatlah sebuah file yang berekstensi .js, misalnya isinya sebagai berikut.

```
// file-eksternal.js  
alert("Kode Javascript dari File Eksternal");
```

Penulisan JavaScript pada HTML

Kemudian, kita perlu menghubungkan file eksternal tersebut dengan file HTML. Caranya, gunakan tag **script** dengan atribut `src` untuk menentukan lokasi file Javascriptnya.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Penulisan Javascript</title>

<script src="file-eksternal.js"></script>

</head>
<body>

</body>
</html>
```

3. Atribut Event

Cara yang ketiga ini sering digunakan untuk memanggil fungsi pada event tertentu. Misalnya ketika suatu elemen diklik, maka jalankan fungsi Javascript.

```
<button onclick="alert('Ok Terima kasih!')">Klik donk!</button>
```

Selain *event onclick* ada juga *event-event* yang lain, seperti *onsubmit*, *onload*, *ondblclick*, *onmouseover*, *onmouseout*, dsb.

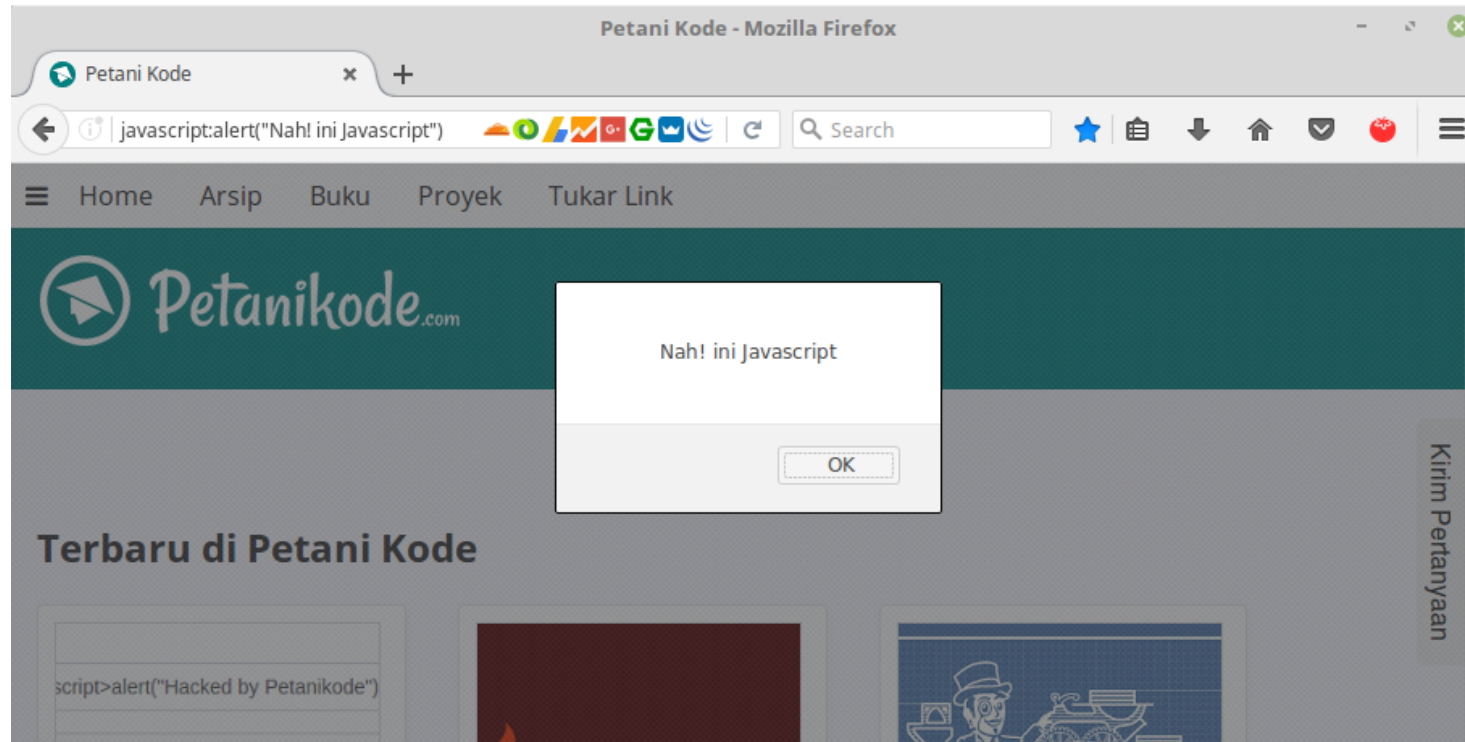
4. URL

Terakhir, penulisan Javascript pada URL. Cara ini jarang digunakan, namun kita juga perlu mengetahuinya. Penulisan Javascript pada URL menggunakan protokol Javascript. Misal, cobalah untuk menuliskan kode seperti ini pada URL browser.

```
javascript:alert("Nah! ini Javascript")
```

Hasilnya, Javascript akan dieksekusi browser.

Output



Lalu, bagaimana kita menggunakan cara ini di HTML?

Penulisan JavaScript pada HTML

Cara ini bisa kita gunakan pada tag **a**, kemudian mengisi kode javascript pada atribut href. Cara ini bisa menggantikan event onclick.

```
<a href="javascript:alert('Wih! hebat bukan?')">Klik Aku</a>
```

Menampilkan Output Javascript

Output adalah sebuah tampilan program yang biasanya digunakan untuk memperlihatkan hasil akhir. Output biasanya ditampilkan dalam bentuk teks dengan fungsi `print()`.

Ada 4 cara menampilkan output pada Javascript:

1. Fungsi `console.log()`
2. Fungsi `alert()`
3. Fungsi `document.write()`
4. `innerHTML`

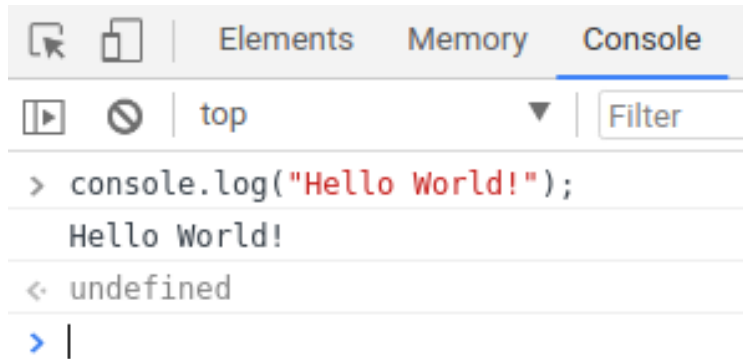
Fungsi `console.log()`

Fungsi `console.log()` adalah fungsi untuk menampilkan teks ke console Javascript.

Contoh penggunaan:

```
console.log("Hello World!");
```

Hasil:



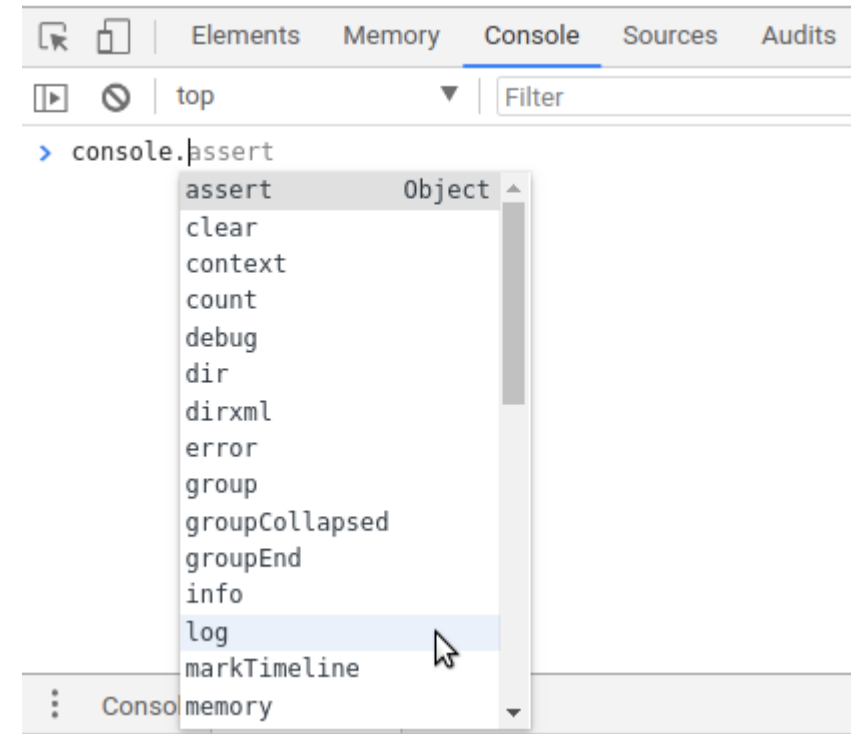
The screenshot shows a browser's developer console with the 'Console' tab selected. The console displays the following output:

```
> console.log("Hello World!");  
Hello World!  
< undefined  
> |
```

Menampilkan Output Javascript

Fungsi `console.log()` biasanya digunakan untuk debugging. Karena setiap pesan error di Javascript selalu ditampilkan di dalam Console.

Selain `console.log()`, terdapat juga beberapa fungsi untuk debugging seperti `console.debug()`, `console.info()`, `console.error()`, `console.dir()`, dsb.



Menampilkan Output Javascript

Fungsi `alert()` adalah fungsi untuk menampilkan jendela dialog. Fungsi sebenarnya berada pada objek `window`.

Secara lengkap bisa ditulis seperti ini:

```
window.alert("Hello World!");
```

Bisa juga ditulis `alert()` saja seperti ini:

```
alert("Hello World!");
```

Fungsi `alert()`, hanya bisa digunakan di dalam browser saja. Sedangkan pada Nodejs fungsi ini tidak ada.

Menampilkan Output Javascript

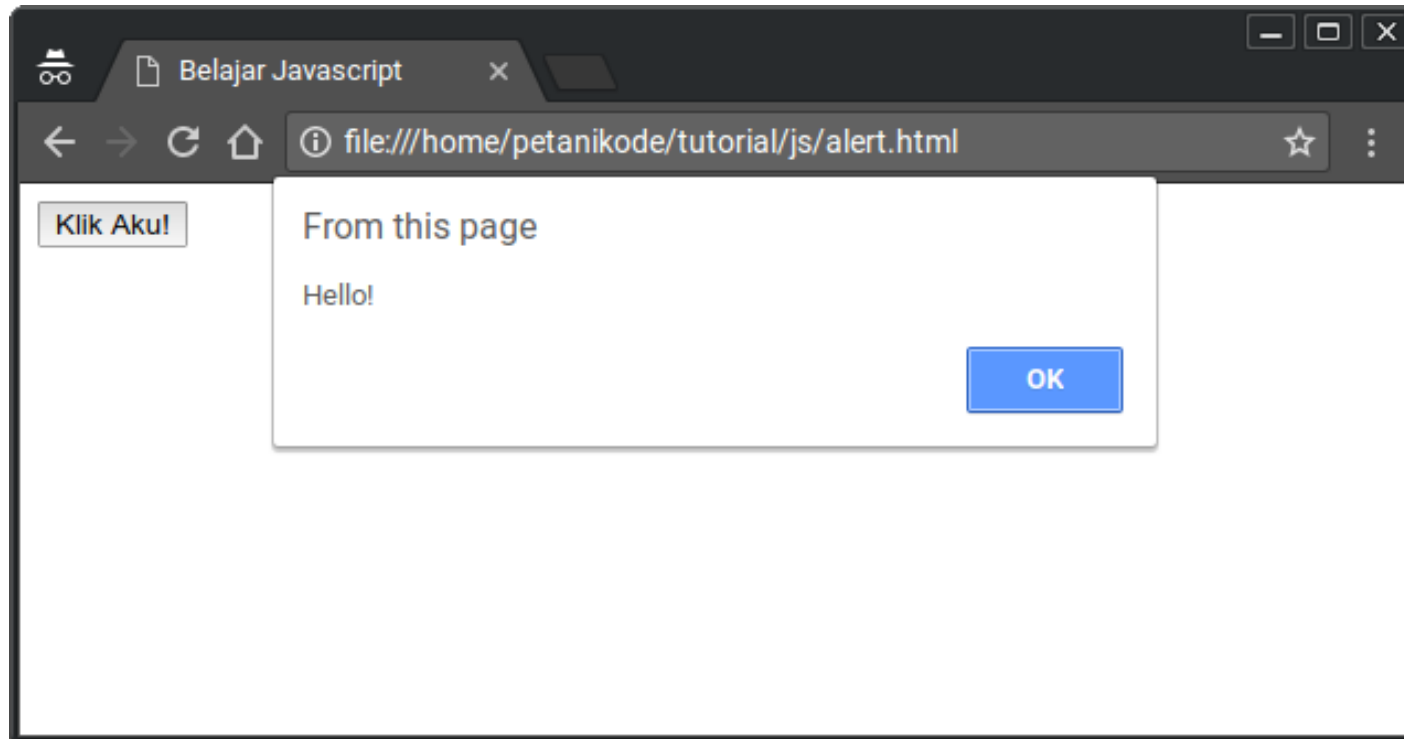
Contoh penggunaan fungsi `alert()`:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Belajar Javascript</title>
  <script>
    alert("Selamat datang di tutorial belajar Javascript");

    function sayHello(){
      alert("Hello!");
    }
  </script>
</head>
<body>
  <button onclick="sayHello()">Klik Aku!</button>
</body>
</html>
```

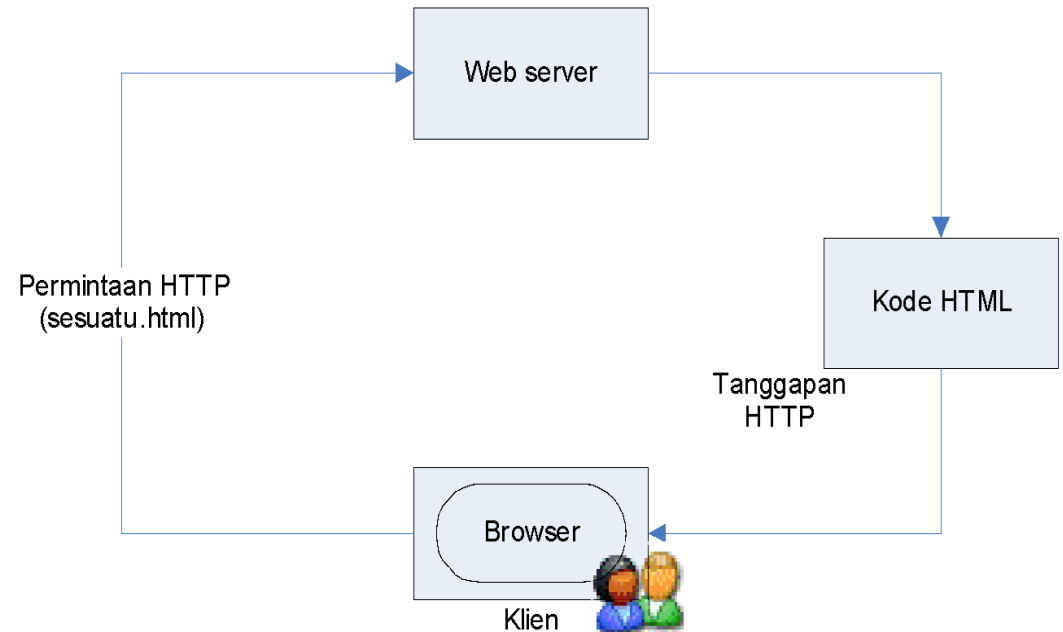
Menampilkan Output Javascript

Hasil:



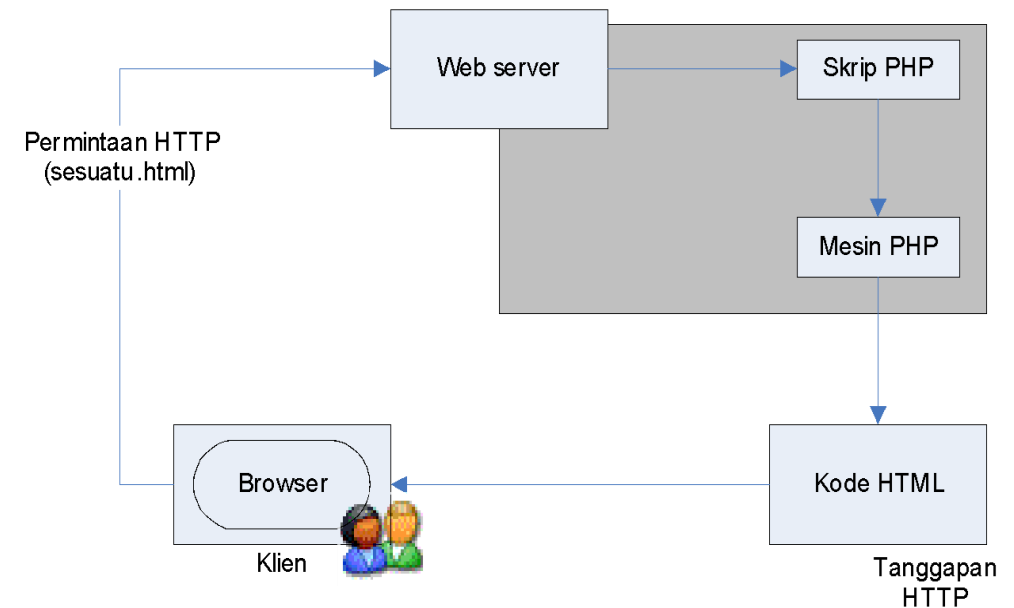
Perbandingan Skema Eksekusi HTML dan PHP

Model kerja HTML diawali dengan permintaan suatu halaman web oleh browser. Browser mendapatkan alamat dari web server, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh web server. Selanjutnya, web server akan mencari berkas yang diminta dan memberikan isinya ke browser. Browser yang mendapatkan isinya segera melakukan proses penerjemahan kode HTML dan menampilkannya ke layar pengguna.



Perbandingan Skema Eksekusi HTML dan PHP

Jika yang diminta adalah sebuah halaman PHP, prinsipnya adalah ketika berkas PHP yang diminta didapatkan oleh web server, isinya segera dikirimkan ke mesin PHP yang memproses dan memberikan hasilnya (berupa kode HTML) ke web server. Selanjutnya, web server menyampaikan ke klien.



Perbedaan Running, Debugging, dan Executable File

Running

Running merupakan proses eksekusi dari kode yang telah ditulis ke dalam Bahasa pemrograman tertentu. Program dapat secara langsung dijalankan pada server web atau dapat juga dijalankan di lingkungan pengembangan lokal. Hal ini bertujuan untuk melihat bagaimana program berperilaku di bawah kondisi normal dan memastikan bahwa semua fungsionalitasnya berjalan seperti yang diharapkan.

Contoh program PHP:

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

Kode di atas jika dijalankan pada halaman web, akan menghasilkan teks **Hello World!** sebagai respons.

Debugging

Debugging merupakan aktivitas yang melibatkan identifikasi, analisis, serta perbaikan kesalahan atau bug yang mungkin muncul dalam struktur dan eksekusi suatu kode program.

Debugging diperlukan ketika terjadi kesalahan dalam program yang mengakibatkan perilaku yang tidak diinginkan atau ketidakmampuan program untuk berjalan dengan benar.

Contoh:

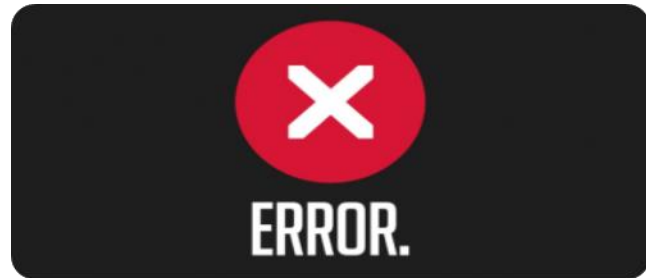
Program PHP yang mengalami masalah dengan manipulasi data dari formulir. Dalam debugging, pengembang dapat menggunakan pernyataan **echo \$variable** atau **var_dump(\$variabel)** untuk melihat nilai variable atau menggunakan alat debugging seperti Xdebug.

Executable File

Proses pembuatan executable file merupakan tahapan yang melibatkan konversi kode sumber suatu perangkat lunak ke dalam bentuk yang dapat dijalankan secara mandiri, tanpa memerlukan interpretasi atau terjemahan langsung oleh Bahasa pemrograman yang digunakan.

Contohnya adalah ketika anda mengunduh dan menginstal suatu aplikasi dari internet. File installer tersebut merupakan contoh dari executable file. Setelah diinstal, anda dapat menjalankan aplikasi tersebut tanpa harus membuka atau mengeksekusi kode sumber asli karena instruksi yang dibutuhkan sudah terkandung dalam executable file tersebut.

Identifikasi Kesalahan / Error Saat Eksekusi Web



Jenis Error pada Software

- ❖ Internal Error : error pada kode program
- ❖ Eksternal Error : error pada interaksi dengan hal lain diluar kode program

Identifikasi Kesalahan Eksekusi Web

Jenis Kesalahan pada saat eksekusi Web :

- ❖ *Error* : Kesalahan tersebut dapat berasal dari kesalahan kode pada script
- ❖ *Http Error* : Kesalahan komunikasi script dengan server php

Kesalahan pada Kode PHP (Error)

Error merupakan salah satu jenis kesalahan yang terjadi ketika melakukan eksekusi pada suatu halaman web, yang menyebabkan halaman web tidak menampilkan hasil sesuai dengan yang diinginkan.

Pada PHP terdapat empat jenis *kesalahan/error* , yaitu :

- 1.Parse Error/ Syntax Error*
- 2.Fatal Error*
- 3.Warning Error*
- 4.Notice*

Parse Errors

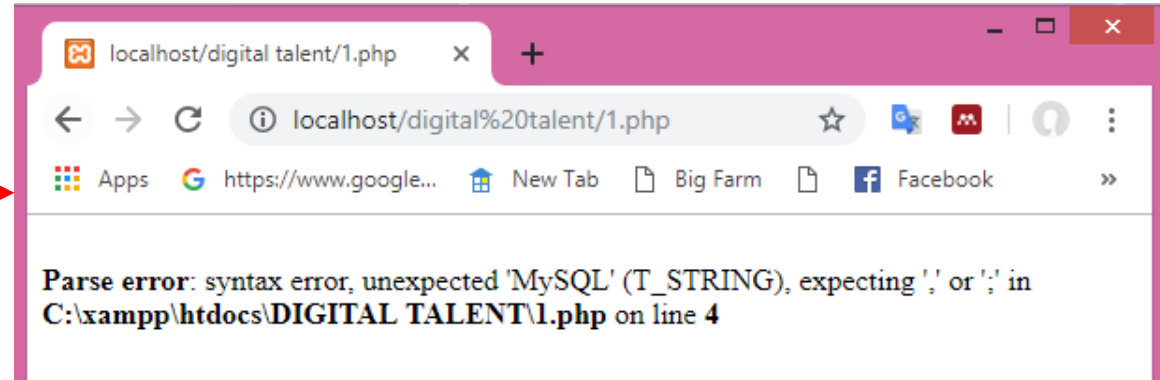
Penyebab :

- ❖ Kutipan yang tidak ditutup, tanda petik tidak sesuai, petik satu atau dua
- ❖ Kelebihan atau kekurangan tanda kurung (“ () “)
- ❖ Kurung kurawal yang tidak ditutup ({ })
- ❖ Kurang tanda titik koma (;)
- ❖ Kurang tanda titik sebagai penggabungan beberapa string (.)
- ❖ Kesalahan dalam penulisan nama variabel

Contoh Parse Error

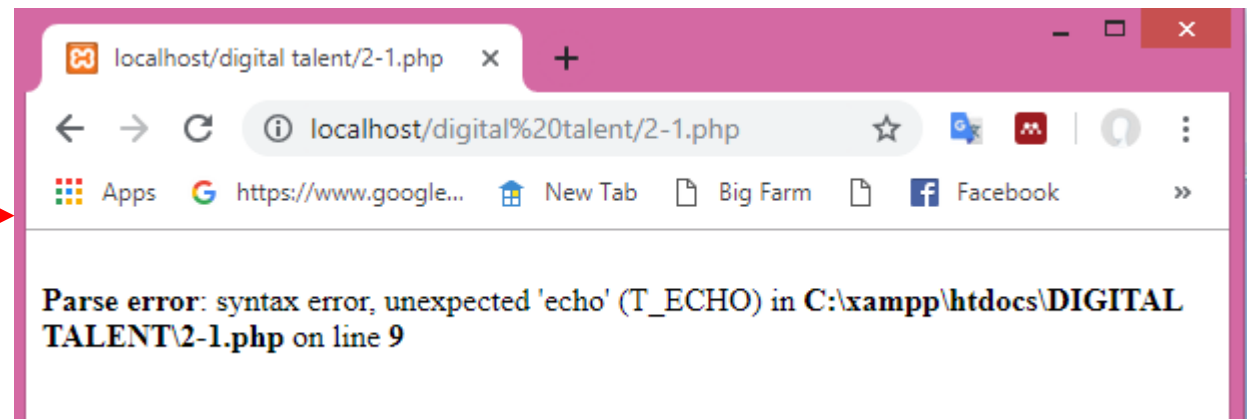
```

1 <?php
2     echo "PHP";
3     echo "HTML";
4     echo "MySQL";
5 ?>
    
```



```

1 <?php
2 function test_error1()
3 {
4     echo "ini fungsi
5     test_error1";
6 }
7 test_error2()
8 {
9     echo "fatal error!";
10 }
11
12 ?>
    
```



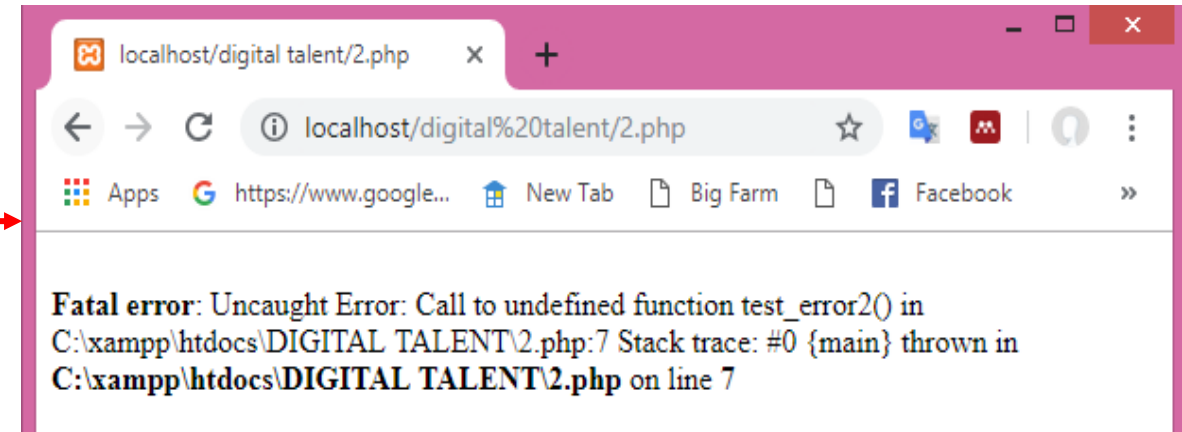
Fatal Errors

Penyebab :

- Apa yang diminta pada kode script tidak dapat dieksekusi.
- Menghentikan proses eksekusi dari sebuah script, sehingga outputnya hanya berupa pesan kesalahan.

Contoh Fatal Errors

```
1 <?php
2 function test_error1()
3 {
4     echo "ini fungsi
5     test_error1";
6 }
7 test_error2();
8 echo "fatal error!";
9
10
11 ?>
```



Warning Errors

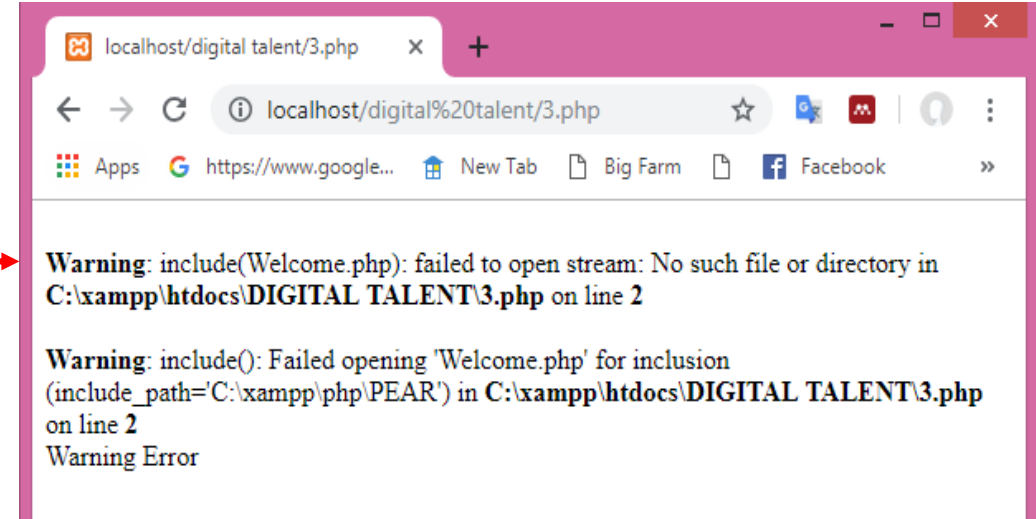
Penyebab :

- Menggunakan fungsi include pada sebuah *missing file*
- Kesalahan parameter yang terdapat pada sebuah fungsi.
- *Warning error* tidak akan menghentikan proses eksekusi dari sebuah script.

Contoh Warning Errors

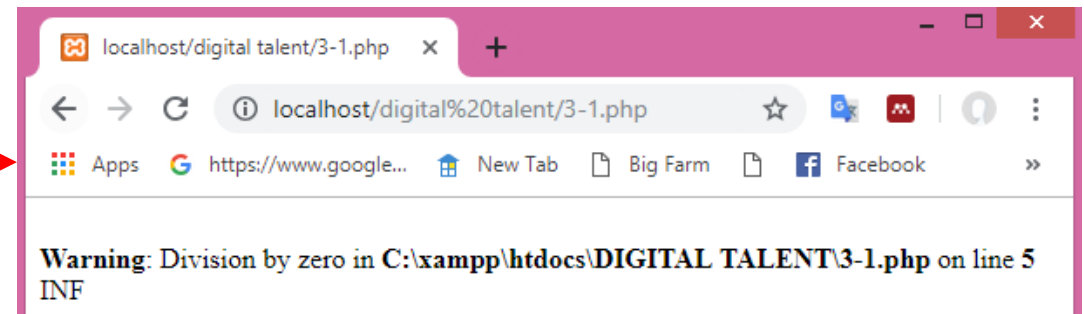
```

1 <?php
2     include ("Welcome.php");
3     echo "Warning Error";
4
5 ?>
    
```



```

1 <?php
2     $a = 10;
3     $b = 0;
4
5     echo $a / $b;
6 ?>
    
```



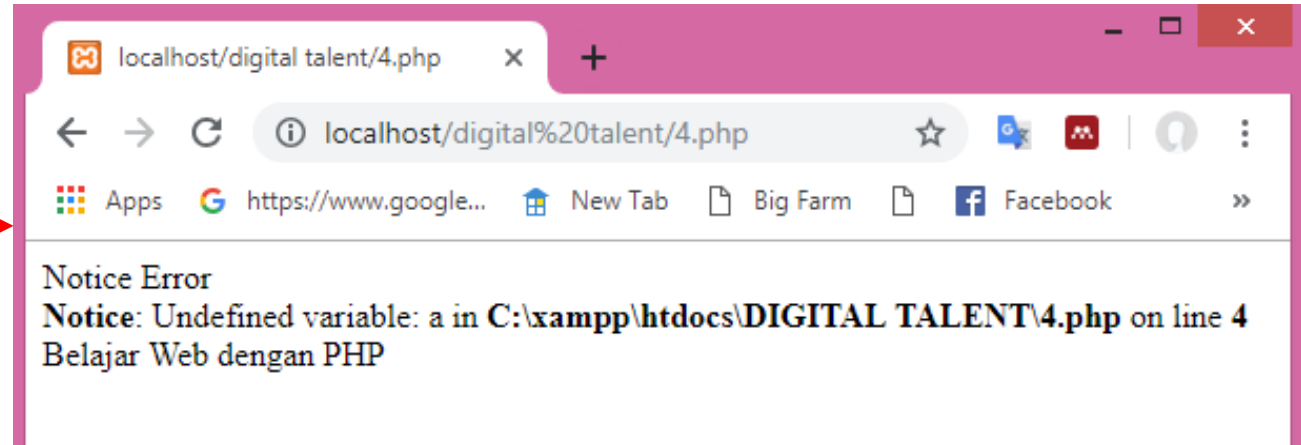
Notice Errors

Penyebab :

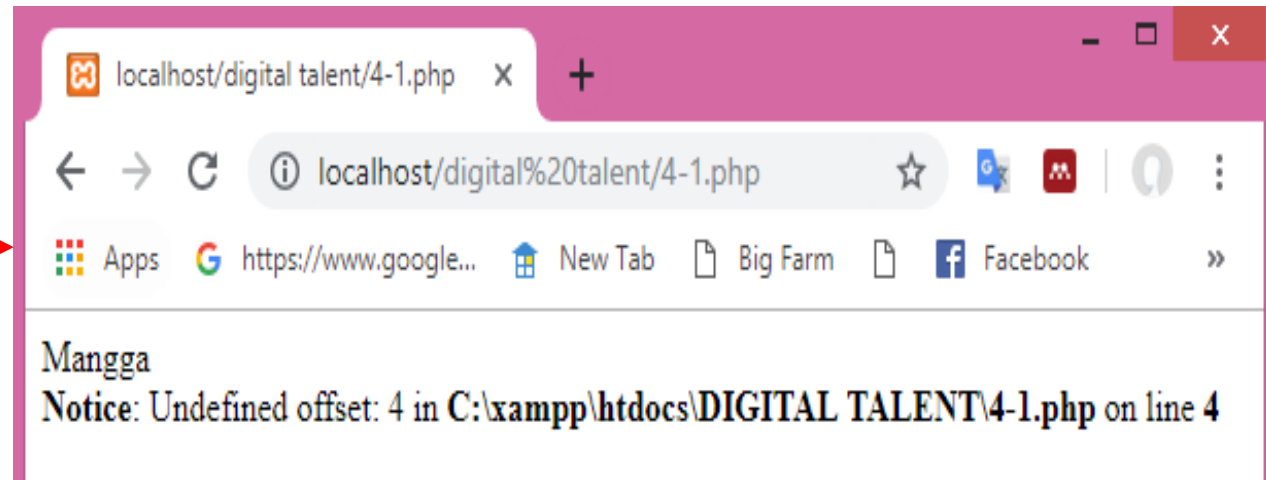
- *Undefined* variabel (variabel yang dipanggil atau dieksekusi, tidak atau belum didefinisikan)
- *Notice error* tidak menghentikan proses eksekusi script.

Contoh Notice Errors

```
1 <?php
2     $x = "Belajar Web
3     dengan PHP";
4     echo "Notice Error";
5     echo $a;
6     echo $x;
```



```
1 <?php
2 $arrBuah = array ("Mangga", "Apel",
3 "Pisang", "Jeruk");
4 echo $arrBuah[0]; //Mangga
5 echo $arrBuah[4]; //Jeruk
6 ?>
```



Referensi

...

...

...

Tools

1. Text Editor:

- Visual Studio Code
- Sublime
- Notepad++

2. Web Server:

- XAMPP

3. Browser:

- Chrome
- Firefox

Summary

1. Tools untuk mengeksekusi source code yang dapat digunakan berupa text Editor (Visual Studio Code/Sublime/dll), web server (XAMPP), browser
2. Pengujian Source Code bertujuan untuk memastikan software dapat berjalan 100% sesuai dengan yang di harapkan baik secara Logika dan secara fungsional
3. Pengujian di lakukan dengan metode Black Box dan White Box
4. Error atau kesalahan eksekusi web dapat terjadi pada sisi internal (error source code) dan eksternal (http error)
5. Pada PHP terdapat empat jenis *kesalahan/error* , yaitu *Parse Error/ Syntax Error, Fatal Error, Warning Error, Notice*
6. Beberapa kesalahan yang umum dan sering terjadi antara lain : Error 404 (Not Found), Error 403 (Forbidden), Error 500 (Internal Server Error), Error 503 (Service Unavailable), Error 504 (Gateway Time-out)
7. Selain itu terdapat beberapa kesalahan yang harus diperhatikan setelah web dihosting ke server



Terima Kasih

