



Fundamental Javascript #6^o

+

Sesi 8

The background is a solid dark blue. It features several abstract geometric elements: a light blue triangle in the top left, a dark blue circle in the top right, a white circle in the bottom left, and a white plus sign in the middle right. A thin white horizontal line is positioned above the word 'Debug'.

Debug

Introduction

Debugging adalah proses mencari dan memperbaiki kesalahan atau 'bug' dalam kode program agar program berfungsi sesuai dengan yang diharapkan. Tujuannya adalah untuk memastikan kode bebas dari kesalahan, lebih efisien, dan mudah dimengerti.

Debug

Jenis 'bug' atau kesalahan

Kesalahan Sintaks (Syntax Errors):

Kesalahan ini muncul saat penulisan kode tidak sesuai dengan aturan bahasa pemrograman yang digunakan.

Contoh: kurangnya tanda kurung tutup atau tanda kutip.

Kesalahan Runtime (Runtime Errors):

Kesalahan ini terjadi saat program sedang dijalankan. Contohnya termasuk pembagian dengan nol atau mengakses elemen array yang tidak ada. Kesalahan ini membuat program crash saat dijalankan.

Kesalahan Logika (Logical Errors):

Ini adalah kesalahan yang membuat program tidak berjalan sesuai harapan, meskipun tidak ada kesalahan sintaks atau runtime. Misalnya, loop yang tidak pernah berakhir (infinite loop) atau kondisi if yang salah.

Debug

Alat debugging

1. Browser Developer Tools

Alat bawaan browser seperti Chrome DevTools atau Firefox Developer Tools digunakan untuk menginspeksi dan debug kode HTML, CSS, dan JavaScript.

Cara buka: tekan F12 atau klik kanan elemen di halaman, lalu pilih "Inspect Element" di browser.

- Elements: Untuk memeriksa dan mengubah HTML dan CSS.
- Console: Menampilkan pesan log, peringatan, dan kesalahan.
- Sources: Manajemen dan debugging kode JavaScript.
- Network: Analisis trafik jaringan. Requests ke API akan muncul disini juga.



Alat debugging

2. Console Logging

Kita bisa gunakan `console.log` di dalam code untuk menampilkan informasi saat program/ web dijalankan.

- `console.log()`: Menampilkan informasi di konsol, berguna untuk pelacakan.
- `console.warn()`: Menampilkan peringatan.
- `console.error()`: Menampilkan pesan kesalahan.



Debug

Alat debugging

3. Debugger

Kita bisa gunakan debugger

Menuliskan 'debugger' di code JavaScript akan menghentikan eksekusi code dan memungkinkan kita untuk memeriksa nilai variabel saat itu.



HACKTIV8

Debugging HTML dan CSS

HTML

- Tag: Pastikan semua tag HTML dibuka dan ditutup dengan benar.
- Validator: Gunakan validator HTML online untuk memeriksa kesalahan.

CSS

DevTools: Gunakan panel "Elements" di DevTools untuk memeriksa dan debug masalah CSS seperti:

- selector yang tidak bekerja atau
- konflik spesifisitas.
- Box model yang tidak sesuai, atau
- masalah layout lainnya.

Debug

Debugging JS

Syntax Errors

Kesalahan Sintaks: Biasanya muncul di konsol; baca pesan kesalahan untuk petunjuk.

Runtime Errors

TypeError, ReferenceError, dan lain-lain. Muncul saat kode dijalankan.

Logical Errors

Gunakan `console.log()` atau debugger untuk melacak alur kode dan temukan kesalahan logika.

Event Handling

DevTools: Gunakan untuk memeriksa apakah event listener terhubung dengan benar atau tidak.



Best Practice Debugging

1. Komentar dan Dokumentasi

Tuliskan komentar kode untuk menjelaskan fungsi dan logika.

2. Testing Code Secara Bertahap

Uji kode dalam potongan kecil untuk mempermudah pelacakan masalah.





Latihan Debugging⁺

Debug

Debug contoh HTML CSS JS berikut

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Debugging Exercise</title>
  <link rel="stylesheet"
href="style.css">
</head>
<body>
  <div id="container">
    <h1>Welcome to My Website</h>
    <button id="changeColor">Change
Background Color</button>
  </div>
  <script src="script.js"></script>
</body>
</html>
```

CSS

```
#container {
  width: 100%;
  height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  background-color: lightgrey;
}

button {
  margin-top: 20px;
  padding: 10px 20px;
  font-size: 16px;
}
```

JS

```
document.addEventListener('DOMContentLoaded', function() {
  const btn =
document.querySelector('#changeColor
');

  btn.addEventListener('click',
function() {

document.body.style.backgroundColor
= 'blue';
  });
});
```





Jawaban Latihan⁺ Debugging

Debug

Debug contoh HTML CSS JS berikut

HTML

id="container" harusnya ditutup dengan tanda kutip.
Tag <h1> tidak ditutup dengan benar.

CSS

justify-conten: center; harusnya justify-content: center;

JavaScript

addEventListener seharusnya addEventListener.
#changeColor seharusnya #changeColor.



HACKTIV8



Local Storage & Session Storage

Local Storage & Session Storage

Introduction

Local Storage dan Session Storage merupakan sebuah fungsi dari javascript yang cara kerjanya hampir sama dengan cookie (data yang tersimpan di dalam browser) tapi memiliki beberapa kelebihan yaitu dapat menyimpan data lebih aman dan lebih besar dibandingkan dengan cookie,

Local Storage dapat menyimpan data lebih dari 5 MB tanpa membebani performa dari browser.

Perbedaan antara local storage dan session storage yaitu, pada local storage data yang tersimpan tidak memiliki waktu expired (tidak ada batasan waktu penyimpanan), namun session storage data yang tersimpan akan hilang ketika kita menutup browser yang akan kita gunakan.



Local Storage & Session Storage

Implementation

Untuk menggunakan local storage dan session storage, kita terlebih dahulu harus mengatur key dan valuenya seperti dibawah ini :

Local Storage

```
1
2 //set localStorage
3 localStorage.setItem("key", "value");
4
5 //value
6 localStorage.getItem("key")
7 |
```

Session Storage

```
1
2 //set sessionStorage
3 sessionStorage.setItem("key", "value");
4
5 //value
6 sessionStorage.getItem("key")
7
```

Local Storage & Session Storage

Implementation

Contoh sederhana:

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <div id="tampil"></div>
6  <script>
7    // Check browser support
8    if (typeof(Storage) !== "undefined") {
9      // set Storage
10     sessionStorage.setItem("key", "Hacktiv8");
11     // menampilkan value
12     document.getElementById("tampil").innerHTML = sessionStorage.getItem("key");
13   } else {
14     document.getElementById("tampil").innerHTML = "Sorry, Sir";
15   }
16 </script>
17 </body>
18 </html>
```





Testing

+

Testing

Software Testing

Apa itu Software Testing ?

Software Testing adalah salah satu step dalam development aplikasi. Pada tahap software testing, dilakukan pengecekan apakah result yang didapatkan dari aplikasi sesuai dengan result yang diharapkan dari aplikasi tersebut. Tidak hanya itu, software testing juga dilakukan untuk memastikan bahwa sistem dari aplikasi tersebut sesempurna mungkin (meskipun tidak mungkin perfect, setidaknya sebisa mungkin mendekati perfect). Software testing juga membantu untuk mengidentifikasi error dan requirements yang kurang sesuai. Software testing dapat dilakukan secara manual ataupun menggunakan automated tools.



HACKTIV8

Testing

Software Testing

Terdapat 3 macam paradigma software testing, yaitu :

1. Unit Testing

Unit Testing mengacu pada test yang dilakukan pada potongan-potongan code atau unit, sehingga kita dapat memastikan bahwa potongan code tersebut dapat berjalan dengan semestinya, sesuai dengan yang kita harapkan.

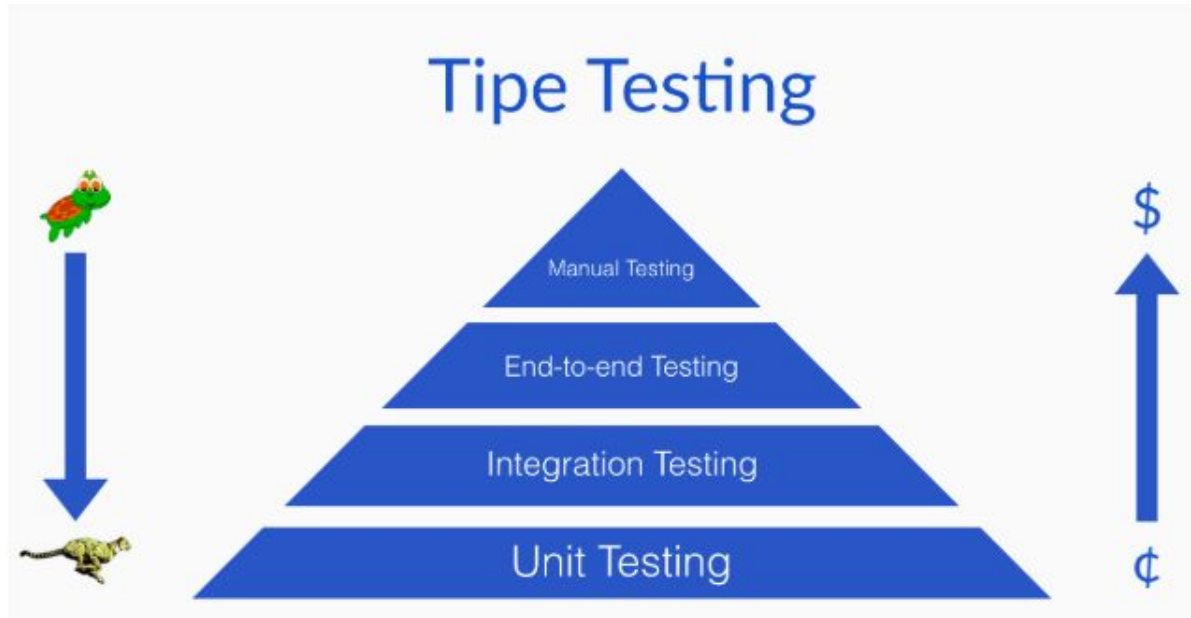
2. Functional Testing

Functional Testing mengacu pada test yang dilakukan untuk melihat behavior dari component.

3. Integration Testing

Integration Testing mengacu pada test yang dilakukan kepada keseluruhan sistem dari aplikasi dengan seperti layaknya end-user menggunakan aplikasi tersebut.





Testing

Software Testing



<https://validator.w3.org/>

HTML Validator

<https://jsonformatter.org/html-validator>



<https://www.10bestdesign.com/dirtymarkup>



https://cdn.tutsplus.com/net/uploads/legacy/562_qunit/qunit.png

QUnit adalah JavaScript unit testing framework yang kuat yang membantu Anda untuk debug kode. Ini ditulis oleh anggota tim jQuery, dan suite resmi uji untuk jQuery. Tapi QUnit cukup umum untuk menguji kode JavaScript biasa, dan bahkan dapat menguji server-side JavaScript melalui beberapa engine JavaScript seperti Rhino atau V8.



Testing

QUnit

Bagaimana menulis Unit Test dengan QUnit

Jadi bagaimana Anda menulis unit test dengan QUnit sebenarnya? Pertama, Anda perlu mengatur lingkungan pengujian:

```
<!DOCTYPE html>
<html>
<head>
  <title>QUnit Test Suite</title>
  <link rel="stylesheet" href="http://github.com/jquery/qunit/raw/master/qunit/qunit.css" type="text/css" media="screen">
  <script type="text/javascript" src="http://github.com/jquery/qunit/raw/master/qunit/qunit.js"></script>
  <!-- Your project file goes here -->
  <script type="text/javascript" src="myProject.js"></script>
  <!-- Your tests file goes here -->
  <script type="text/javascript" src="myTests.js"></script>
</head>
<body>
  <h1 id="qunit-header">QUnit Test Suite</h1>
  <h2 id="qunit-banner"></h2>
  <div id="qunit-testrunner-toolbar"></div>
  <h2 id="qunit-userAgent"></h2>
  <ol id="qunit-tests"></ol>
</body>
</html>
```



Testing

QUnit

Blok bangunan unit test adalah assertion.

Assertion ini adalah pernyataan yang memprediksi hasil kembalian kode Anda. Jika prediksi false, assertion telah gagal, dan Anda tahu bahwa sesuatu yang salah.

```
// Let's test this function
function isEven(val) {
  return val % 2 === 0;
}

test('isEven()', function() {
  ok(isEven(0), 'Zero is an even number');
  ok(isEven(2), 'So is two');
  ok(isEven(-4), 'So is negative four');
  ok(!isEven(1), 'One is not an even number');
  ok(!isEven(-7), 'Neither is negative seven');
})
```



QUnit Test Suite

Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.7)
Gecko/20091221 Firefox/3.5.7

1. isEven() (0, 5, 5)

- 1. Zero is an even number
- 2. So is two
- 3. So is negative four
- 4. One is not an even number
- 5. Neither does negative seven

Tests completed in 9 milliseconds.
5 tests of 5 passed, 0 failed.

QUnit Test Suite

☐ Hide passed tests ☐ Hide missing tests (untested code is broken code)

Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.1.7)
Gecko/20091221 Firefox/3.5.7

1. isEven() (1, 5, 6)

- 1. Zero is an even number
- 2. So is two
- 3. So is negative four
- 4. One is not an even number
- 5. Neither does negative seven
- 6. Three is an even number

Tests completed in 13 milliseconds.
5 tests of 6 passed, 1 failed.





Final Project

+

Final Project

Requirements

Buat Sebuah Company Profile Web-Based

Syarat

1. Company Nama_Kalian bergerak dibidang Jasa.
2. Halaman Wajib Utama terdiri dari Header, Body, Footer
3. Memiliki 5 Menu Wajib :
 - A. Home
Perhatikan pada banner ada button action : order now yang jika diklik maka akan menampilkan pop up berisi list/table.
 - B. Contact Us
 - C. About Us
 - D. Portfolio
 - E. Blog

Final Project

Requirements

F. Todos

****Special Requirement for Menu : Todos**

- Seluruh Fungsi dibuat dengan Javascript
- Pastikan ketika ada action add , data disimpan di Local Storage.

(Cara cek tersimpan di local storage atau tidak :

1. Inspect Element
2. Pilih Application -> Local Storage (Setelah inspect Element)

Final Project

Requirements

Point Highlights Penggunaan Javascript :

i. Symbol == & ===

ii. Ternary Operator

iii. Functions

iv. Javascript Object

v. Filter Function

vi. ForEach Loop

4. Buat Semenarik Mungkin untuk desain setiap Pages.

5. Hasil Akhir untuk Source code di letakkan pada repository Github/Bitbucket, dan harus di lakukan Deployment Online Via Heroku/Netlify

Final Project

Requirements

Contoh Tampilan dapat dilihat disini :

<https://romantic-bassi-f50129.netlify.app/>

Hasil Akhir untuk Source code di letakkan pada repository Github/Bitbucket, dan harus di lakukan Deployment Online Via Heroku/Netlify.

Upload link Heroku/Netlify dan Github/Bitbucket pada Google Classroom.



HACKTIV8

Final Project

Requirements

1

click button order-now, then ..

show on pop up list/table

2

Plan Features	Basic \$5/month	Essential \$10/month
1GB Space	✓	✓
Backup	✓	✓
24/7 Support	✓	✓
Phone Support	✓	✓
Advanced Security	✓	✓
Custom	✓	✓
Responsive Layout	✓	✓
24/7 Support	✓	✓
Refund	✓	✓
Instant Chat	✓	✓