

LAPORAN RESMI
MODUL IV
JAVASCRIPT
PEMROGRAMAN BERBASIS WEB



NAMA	: MUHAMMAD AKMAL NABIL HIBRIZI
N.R.P	: 220441100068
DOSEN	: SRI HERAWATI S.Kom.,M.Kom
ASISTEN	: ABDUL WAKHID
TGL PRAKTIKUM	: 29 Maret 2023

Disetujui : 04 April 2023
Asisten

ABDUL WAKHID
21.04.411.001.06



LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

HTML (Hypertext Markup Language) adalah sebuah bahasa markup yang digunakan untuk membuat halaman web. HTML memungkinkan pembuatan struktur dan konten sebuah halaman web dengan menggunakan tag dan atribut yang terdiri dari berbagai elemen seperti teks, gambar, audio, video, tabel, dan formulir.

HTML adalah salah satu teknologi dasar dalam pengembangan web, bersama dengan CSS (Cascading Style Sheets). HTML digunakan oleh para pengembang web untuk membuat halaman web yang interaktif, responsif, dan menarik. Oleh karena itu, pengetahuan mengenai HTML sangat penting bagi siapa saja yang ingin terlibat dalam pengembangan web, baik sebagai pengguna, desainer, maupun developer.

JavaScript dirancang untuk bekerja dengan HTML dan CSS, yang digunakan untuk membuat halaman web. Bahasa ini dapat digunakan untuk membuat interaksi pengguna seperti animasi, perubahan tampilan halaman secara dinamis, validasi form, dan banyak lagi. Selain itu, JavaScript juga dapat digunakan untuk membangun aplikasi web berbasis server, seperti aplikasi chatting atau e-commerce.

1.2 Tujuan

- Mampu memahami syntax dasar JavaScript.
- Mampu mengimplementasikan JavaScript dalam pembuatan suatu website.
- Mampu memahami cara penggunaan JavaScript.
- Mampu menjelaskan dan memberi contoh cara menggabungkan JavaScript ke HTML.
- Mampu menjelaskan dan memberi contoh mengenai pendeklarasian variabel dalam JavaScript.

BAB II

DASAR TEORI

2.1 Pengertian

Untuk menyambungkan JavaScript ke dalam dokumen HTML, Anda dapat menggunakan tag script di dalam elemen head atau body dokumen HTML. Berikut adalah contoh kode dasar:

Masukkan kode JavaScript di dalam tag script:

Html Copy

code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Contoh Penyambungan Javascript ke HTML</title>
    <script>
      // Masukkan kode JavaScript di sini
    </script>
  </head>
  <body>
    <!-- Isi dokumen HTML di sini -->
  </body>
</html>
```

Sambungkan file JavaScript eksternal dengan menggunakan atribut src didalam tag script: html Copy code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Contoh Penyambungan Javascript ke HTML</title>
    <script src="script.js"></script>
  </head>
  <body>
    <!-- Isi dokumen HTML di sini -->
  </body>
```

</html>

Pastikan untuk menempatkan file JavaScript eksternal pada direktori yang benar dan menyesuaikan nama file dengan benar. Jika Anda ingin menggunakan framework JavaScript tertentu seperti React atau Vue, Anda dapat menggunakan cara yang lebih spesifik untuk menyambungkan JavaScript ke dalam dokumen HTML.

2.2 Deklarasi Variabel

a. Var

Deklarasi variabel menggunakan kata kunci var pada JavaScript digunakan untuk membuat variabel dan memberi tahu interpreter JavaScript bahwa kita akan menggunakan nama variabel tersebut. Variabel dapat berisi nilai yang berbeda selama program dijalankan.

Berikut adalah contoh penggunaan deklarasi variabel menggunakan var:

```
javascript
```

Copy code

```
var nama = "John";
```

```
var umur = 25;
```

```
var sudahMenikah = false;
```

```
console.log(nama); // output: John
```

```
console.log(umur); // output: 25
```

```
console.log(sudahMenikah); // output: false
```

Dalam contoh di atas, kita membuat tiga variabel: nama, umur, dan sudahMenikah. nama adalah variabel bertipe data string, umur adalah variabel bertipe data number, dan sudahMenikah adalah variabel bertipe data boolean.

Variabel yang dideklarasikan dengan var dapat diakses dan dimodifikasi di mana saja dalam program JavaScript. Variabel yang dideklarasikan tanpa nilai akan memiliki nilai undefined secara default.

javascript

Copy code

```
var a;  
console.log(a); // output: undefined
```

Namun, perlu diingat bahwa variabel yang dideklarasikan dengan `var` memiliki scope fungsi. Artinya, jika variabel dideklarasikan di dalam fungsi, variabel tersebut hanya dapat diakses dari dalam fungsi tersebut.

javascript

Copy code

```
function tes() {  
  var b = 2;  
}  
console.log(b); // output: error
```

Dalam contoh di atas, variabel `b` hanya dapat diakses dari dalam fungsi `tes()`. Jika kita mencoba mengakses variabel `b` dari luar fungsi, maka akan muncul error.

a. **Let**

`Let` adalah kata kunci dalam JavaScript yang digunakan untuk mendeklarasikan variabel yang nilainya dapat diubah atau diinisialisasi ulang di masa depan. Variabel yang dideklarasikan dengan `let` memiliki scope block-level, yang berarti bahwa variabel hanya dapat diakses di dalam blok di mana variabel tersebut dideklarasikan.

Berikut adalah contoh penggunaan `let` untuk deklarasi variabel:

javascript

Copy code

```
let nama = "John";  
nama = "Doe";  
console.log(nama); // output: Doe
```

Dalam contoh di atas, kita membuat variabel nama menggunakan `let`. Setelah itu, kita mengubah nilainya menjadi "Doe". Kita dapat melakukannya karena variabel yang dideklarasikan menggunakan `let` dapat diubah nilainya.

Variabel yang dideklarasikan dengan `let` hanya dapat diakses di dalam blok di mana variabel tersebut dideklarasikan. Blok adalah bagian dari kode yang diapit oleh kurung kurawal `{ }`. Sebagai contoh:

javascript

Copy code

```
let x = 10;
if (x > 5) {
  let y = 5;
  console.log(x); // output: 10
  console.log(y); // output: 5
}
console.log(x); // output: 10
console.log(y); // output: error
```

Dalam contoh di atas, variabel `x` dideklarasikan di luar blok `if`. Variabel `y` dideklarasikan di dalam blok `if`, sehingga hanya dapat diakses di dalam blok `if` tersebut. Jika kita mencoba mengakses variabel `y` di luar blok `if`, maka akan muncul error.

Dalam ringkasan, `let` digunakan untuk mendeklarasikan variabel dengan nilai yang dapat diubah dan scope block-level. Jika kita ingin membuat variabel yang nilainya tetap atau konstan, kita dapat menggunakan `const`.

b. Const

`Const` adalah kata kunci dalam JavaScript yang digunakan untuk mendeklarasikan variabel dengan nilai yang tetap atau konstan. Variabel

yang dideklarasikan dengan const tidak dapat diubah atau diinisialisasi ulang setelah dideklarasikan.

Berikut adalah contoh penggunaan const untuk deklarasi variable:

javascript

Copy code

```
const PI = 3.14;  
console.log(PI); // output: 3.14
```

```
PI = 3.1415; // error
```

Dalam contoh di atas, kita membuat variabel PI menggunakan const dan memberinya nilai 3.14. Kita mencoba mengubah nilai variabel PI menjadi 3.1415, tetapi akan muncul error karena variabel yang dideklarasikan menggunakan const tidak dapat diubah nilainya.

Variabel yang dideklarasikan menggunakan const memiliki scope block-level, yang berarti variabel hanya dapat diakses di dalam blok di mana variabel tersebut dideklarasikan.

javascript

Copy code

```
if (true) {  
  const x = 10; console.log(x); // output: 10  
}  
console.log(x); // output: error
```

Dalam contoh di atas, variabel x dideklarasikan menggunakan const di dalam blok if. Variabel x hanya dapat diakses di dalam blok if tersebut, sehingga jika kita mencoba mengakses variabel x di luar blok if, maka akan muncul error.

Dalam ringkasan, const digunakan untuk mendeklarasikan variabel dengan nilai tetap atau konstan yang tidak dapat diubah setelah dideklarasikan. Variabel yang dideklarasikan menggunakan const memiliki scope block-level. Jika kita ingin membuat variabel yang nilainya dapat diubah, kita dapat menggunakan let.

2.3 Function dan Kondisi

Fungsi di JavaScript

Fungsi adalah blok kode yang dapat digunakan kembali dalam program. Dalam JavaScript, fungsi dapat didefinisikan dengan menggunakan kata kunci `function`. Berikut adalah contoh fungsi sederhana yang menerima dua argumen dan mengembalikan hasil penjumlahannya.

javascript

Copy code

```
function tambah(a, b) {  
  return a + b;  
}
```

Dalam contoh di atas, kita mendefinisikan fungsi `tambah` dengan dua argumen `a` dan `b`. Fungsi ini mengembalikan hasil penjumlahan dari dua argumen tersebut. Kita dapat memanggil fungsi `tambah` dengan memberikan dua argumen:

javascript

Copy code

```
let hasil = tambah(2, 3);  
console.log(hasil); // output: 5
```

Dalam contoh di atas, kita memanggil fungsi `tambah` dengan memberikan dua argumen `2` dan `3`. Hasil penjumlahan dari dua argumen tersebut kemudian disimpan dalam variabel `hasil`, yang kemudian dicetak menggunakan `console.log`.

Kondisi di JavaScript

Kondisi adalah bagian dari program yang menentukan apakah blok kode tertentu akan dijalankan atau tidak, berdasarkan kondisi yang diberikan. Dalam JavaScript, kita dapat menggunakan beberapa kata kunci untuk membuat kondisi:

if

`if` adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika kondisi tertentu terpenuhi. Berikut adalah contoh penggunaan `if`:

javascript

Copy code

```
let x = 10;  
if (x > 5) {  
  console.log("x lebih besar dari 5");  
}
```

Dalam contoh di atas, kita menggunakan if untuk mengecek apakah nilai x lebih besar dari 5. Jika kondisi terpenuhi, maka blok kode di dalam kurung kurawal akan dieksekusi.

else

else adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika kondisi pada if tidak terpenuhi. Berikut adalah contoh penggunaan if-else:

javascript

Copy code

```
let x = 3;  
if (x > 5) {  
  console.log("x lebih besar dari 5");  
} else {  
  console.log("x kurang dari atau sama dengan 5");  
}
```

Dalam contoh di atas, kita menggunakan if-else untuk mengecek apakah nilai x lebih besar dari 5. Jika kondisi pada if tidak terpenuhi, maka blok kode di dalam else akan dieksekusi.

else if

else if adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika beberapa kondisi sebelumnya tidak terpenuhi. Berikut adalah contoh penggunaan if-else if:

javascript

Copy code

```
let x = 3;  
if (x > 5) {  
  console.log("x lebih besar dari 5");  
} else if (x > 2) {  
  console.log("x lebih besar dari 2 dan kurang dari atau sama dengan 5");  
} else {  
  console.log("x kurang dari atau sama dengan 2");  
}
```

BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Sebutkan dan jelaskan cara menggunakan JavaScript ke HTML
2. Sebutkan dan jelaskan beberapa cara untuk mendeklarasikan Variable dalam JavaScript

3.2 Jawaban

1. Berikut adalah beberapa cara untuk menghubungkan JavaScript ke HTML :

- Inline Script

dapat menambahkan kode JavaScript langsung ke dalam tag HTML.

- Internal Script

dapat menempatkan kode JavaScript didalam tag `<script>` pada bagian head atau body dari dokumen HTML

- External Script

dapat membuat file JavaScript terpisah dan menghubungkan ke dalam dokumen HTML menggunakan tag `<script>` dengan atribut `src`.

2. Dalam JavaScript terdapat beberapa cara untuk mendeklarasikan sebuah Variable yaitu :

- Var

kata kunci `var` digunakan untuk mendeklarasikan sebuah Variable yang dapat diakses di seluruh lingkup Fungsi (global) atau lokal pada suatu Fungsi

- let

digunakan untuk mendeklarasikan Variable yang hanya dapat diakses pada blok kode tempat Variable tersebut.

- Const

digunakan untuk mendeklarasikan Variable yang nilainya tetap atau tidak dapat diubah setelah dideklarasikan.


BAB IV

IMPLEMENTASI

4.1 Source Code


- **HTML**

Untuk membuat tampilan jam



```
1 <div class="jam-digital-malasngoding">
2   <div class="kotak">
3     <p id="jam"></p>
4   </div>
5   <div class="kotak">
6     <p id="menit"></p>
7   </div>
8   <div class="kotak">
9     <p id="detik"></p>
10  </div>
11 </div>
```

untuk membuat tampilan scroll



```
1 <div class="button-top" onclick="scrolling()">
2   <i class="bi bi-chevron-double-up"></i>
3   <!-- <i class="bi bi-prescription2"></i> -->
4 </div>
5 <script src="terakhir.js"></script>
```

untuk membuat tampilan navigation



```
1 <nav>
2   <input id="nav-toggle" type="checkbox">
3   <div class="logo">AYOKBelajar</div>
4   <ul class="links">
5     <li><a href="#">Home</a></li>
6     <li><a href="#">About</a></li>
7     <li><a href="#">Pages</a></li>
8     <li><a href="#">Review</a></li>
9     <li><a href="#">Projects</a></li>
10    <li><a href="#">Contact</a></li>
11  </ul>
12  <label for="nav-toggle" class="icon-burger">
13    <div class="line"></div>
14    <div class="line"></div>
15    <div class="line"></div>
16  </label>
17 </nav>
```

- **CSS**

Untuk mengatur tampilan scrolling



```
1  .button-top{
2      width: 50px;
3      height: 50px;
4      background-color: orange;
5      border-radius: 100%;
6      font-size: 39px;
7      color: aliceblue;
8      display: flex;
9      justify-content: center;
10     text-align: center;
11     align-items: center;
12     position: fixed;
13     bottom: 10px;
14     right: 10px;
15     cursor: pointer;
16 }
```

untuk mengatur tampilan jam



```
1  a{
2      font-family: sans-serif;
3      font-weight: normal;
4
5  }
6  .jam-digital-malasngoding{
7      position: relative;
8      overflow: hidden;
9      width: 300px;
10     margin: 120px auto;
11     border: 5px solid #efefef;
12 }
13 .kotak{
14     float: left;
15     width: 96.8px;
16     height: 40px;
17     background-color: #189fff;
18 }
19 .jam-digital-malasngoding p{
20     color: #fff;
21     font-size: 35px;
22     text-align: center;
23     margin-top: 7px;
24 }
```

untuk mengatur tampilan homepage waktu diresponsif

```
1  @media (max-width:972px){
2    .search{
3      width: 40%;
4    }
5  }
6  }
7  @media (max-width:768px){
8    .search{
9      width: 71%;
10   }
11  }
12  }
13  @media(max-width:969px){
14    .content{
15      margin-left: 18%;
16      margin-top: 0;
17    }
18  }
19    .content h1{
20      text-align: center;
21      font-size: 60px;
22      margin-bottom: 290px;
23    }
24  }
25    .content p{
26      text-align: center;
27    }
28  }
29  }
30  @media(max-width:768px){
31    .content{
32      margin-left: 18%;
33      margin-top: 0;
34    }
35  }
36    .content h1{
37      text-align: center;
38      font-size: 60px;
39      margin-bottom: 290px;
40    }
41  }
42    .content p{
43      text-align: center;
44    }
45  }
46  @media(max-width:768px){
47    .image{
48      width: 40%;
49      height: 80%;
50      position: absolute;
51      bottom: 0;
52      right: 100px;
53    }
54  }
55    .image img{
56      height: 70%;
57      left: 60%;
58      top: 430px;
59    }
60  }
```



```
1  .socials{
2    list-style: none;
3    display: flex;
4    align-items: center;
5    justify-content: center;
6    margin: 1rem 0 3rem 0;
7  }
8
9  .socials li{
10   margin: 0 10px;
11 }
12
13 .socials a{
14   text-decoration: none;
15   color: #fff;
16 }
17
18 .socials a i{
19   font-size: 30px;
20   padding-left: 14px;
21   padding-left: 5px;
22   transition: color .4s ease;
23 }
24
25 .socials a:hover i{
26   color: black;
27 }
28
29 .footer-bottom{
30   background: #000;
31   width: 100vw;
32   padding: 20px 0px;
33   text-align: center;
34 }
35
36 .footer-bottom p{
37   font-size: 14px;
38   word-spacing: 2px;
39   text-transform: capitalize;
40 }
41
42 .contents{
43   width: 120vw;
44   background: #ffffff;
45   margin-top: 135px;
46   height: 1400px;
47   border-bottom: 2px solid #000000;
48 }
49 }
50
```

```
1  @media screen and (max-width: 768px) {
2    nav .logo {
3      float: none;
4      width: auto;
5      justify-content: center;
6    }
7    nav .links {
8      float: none;
9      position: fixed;
10     z-index: 9;
11     left: 0;
12     right: 0;
13     top: 100px;
14     bottom: 100%;
15     width: auto;
16     height: auto;
17     flex-direction: column;
18     justify-content: space-evenly;
19     background-color: rgba(142, 140, 140, 0.8);
20     overflow: hidden;
21     box-sizing: border-box;
22     transition: all .5s ease-in-out;
23   }
24   nav .links a {
25     font-size: 20px;
26   }
27   nav :checked ~ .links {
28     top: auto;
29     bottom: auto;
30   }
31   nav .icon-burger {
32     display: block;
33   }
34   nav :checked ~ .icon-burger .line:nth-child(1) {
35     transform: translateY(10px) rotate(225deg);
36   }
37   nav :checked ~ .icon-burger .line:nth-child(3) {
38     transform: translateY(-10px) rotate(-225deg);
39   }
40   nav :checked ~ .icon-burger .line:nth-child(2) {
41     opacity: 0;
42   }
43 }
44 }
```

- JavaScript

hanya ada 2 sourcecode pada javascript yaitu mengatur jam dan mengatur untuk men scroll, yaitu :

Source code yang Anda berikan adalah sebuah JavaScript yang mengatur tampilan waktu pada suatu halaman web dengan menggunakan fungsi `setTimeout()` dan `Date()`.

Penjelasan source code tersebut adalah sebagai berikut:

Baris pertama, `window.setTimeout("waktu()",1000);`, mengatur agar fungsi `waktu()` akan dipanggil setiap 1000 milidetik atau 1 detik.

Fungsi `waktu()` kemudian didefinisikan pada baris kedua hingga baris keenam.

Pada baris ketiga, sebuah objek `Date` dibuat dengan menggunakan fungsi `new Date()` untuk mendapatkan waktu saat ini.

Baris keempat, fungsi `setTimeout()` dipanggil kembali untuk memastikan fungsi `waktu()` terus dipanggil setiap 1 detik.

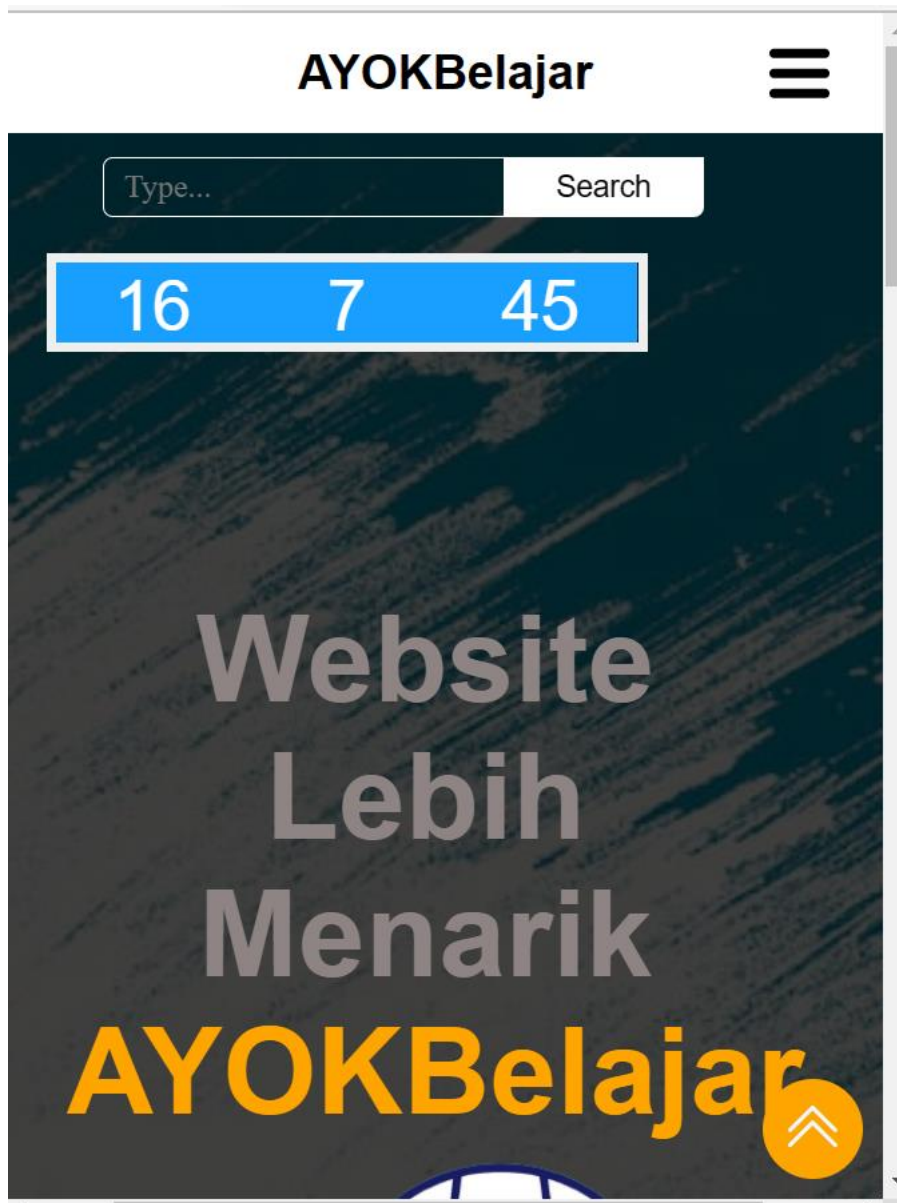
Baris kelima hingga tujuh, mengatur tampilan waktu dengan mengambil nilai jam, menit, dan detik dari objek `Date` yang dibuat pada baris ketiga dan menampilkan nilainya pada elemen HTML dengan menggunakan `document.getElementById()`.

Dengan begitu, setiap 1 detik, fungsi `waktu()` akan dipanggil dan tampilan waktu pada halaman web akan diperbarui sesuai dengan waktu saat ini.



```
1 window.setTimeout("waktu()",1000);
2 function waktu() {
3     var waktu = new Date();
4     setTimeout("waktu()",1000);
5     document.getElementById("jam").innerHTML = waktu.getHours();
6     document.getElementById("menit").innerHTML = waktu.getMinutes();
7     document.getElementById("detik").innerHTML = waktu.getSeconds();
8 }
9
10 // function scrolling(){
11 //     window.scrollTo({
12 //         top: 0,
13 //         behavior: "smooth"
14 //     })
15 // }
16
17 let mybutton = document.getElementById("scroll-Top");
18
19 window.onscroll = function() {scrollFunction()};
20
21
22 function scrollFunction() {
23     if ( document.documentElement.scrollTop > 20) {
24         mybutton.style.display = "block";
25     } else {
26         mybutton.style.display = "none";
27     }
28 }
29 function scrolling() {
30     const scrolloptions = {
31         top: 0,
32         behavior: "smooth"
33     };
34     window.scrollTo(scrolloptions);
35 }
36
37 mybutton.addEventListener("click", scrollToTop);
38
```

4.2 Hasil





INI DIPERUNTUKAN UNTUK ANDA YANG BELAJAR
PEMROGRAMAN
UNTUK MENJADI LEBIH MAHIR DALAM MEMBUAT
WEBSITE



Pelajaran



HTML





CSS





AYOKBelajar

Website ini digunakan untuk belajar bahasa atau pemrograman untuk pemula agar bisa dipahami dasar-dasarnya.



copyright ©2023 AYOKBelajar. designed by AKMAL

BAB V

PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa HTML atau Hypertext Markup Language adalah sebuah bahasa markup yang digunakan untuk membuat halaman web. HTML memungkinkan pembuatan struktur dan konten sebuah halaman web dengan menggunakan tag dan atribut yang terdiri dari berbagai elemen seperti teks, gambar, audio, video, tabel, dan formulir. Sementara JavaScript dirancang untuk bekerja dengan HTML dan CSS, yang digunakan untuk membuat halaman web. Bahasa ini dapat digunakan untuk membuat interaksi pengguna seperti animasi, perubahan tampilan halaman secara dinamis, validasi form, dan banyak lagi. Selain itu, JavaScript juga dapat digunakan untuk membangun aplikasi web berbasis server, seperti aplikasi chatting atau e-commerce.

5.2 Kesimpulan

1. Deklarasi variabel menggunakan kata kunci var pada JavaScript digunakan untuk membuat variabel dan memberi tahu interpreter JavaScript bahwa kita akan menggunakan nama variabel tersebut.
2. Let adalah kata kunci dalam JavaScript yang digunakan untuk mendeklarasikan variabel yang nilainya dapat diubah atau diinisialisasi ulang di masa depan.
3. Const adalah kata kunci dalam JavaScript yang digunakan untuk mendeklarasikan variabel dengan nilai yang tetap atau konstan.
4. Fungsi adalah blok kode yang dapat digunakan kembali dalam program. Dalam JavaScript, fungsi dapat didefinisikan dengan menggunakan kata kunci function.
5. Kondisi adalah bagian dari program yang menentukan apakah blok kode tertentu akan dijalankan atau tidak, berdasarkan kondisi yang diberikan.
6. If adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika kondisi tertentu terpenuhi.
7. Else adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika kondisi pada if tidak terpenuhi.

8. Else if adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika beberapa kondisi sebelumnya tidak terpenuhi.