**RANGKUMAN JS**

1. **Pengenalan JS**

* JavaScript adalah Bahasa pemrograman web.
* JS dapat memperbarui dan mengubah HTML dan CSS.
* JS dapat menghitung, memanipulasi, dan memvalidasi data.
* Salah satu dari banyak metode JavaScript HTML adalah getElementById().

**2. JS Where To**

* KodeJS disipkan di antara Tag <script> dan </script>
* Contoh JS lama mungkin menggunakan atribut type: <script type=”text/javascript”>.
* Atribut type tidak diperlukan. #
* JS adalah Bahasa skrip default dalam HTML
* Fungsi JavaScript:

JavaScript function adalah blok kode JavaScript, yang dapat dieksekusi saat "dipanggil".Misalnya, suatu fungsi dapat dipanggil saat suatu **peristiwa** terjadi, seperti saat pengguna mengklik tombol.

* JavaScript di <head>
* JavaScript di <body>
* JavaScript Eksternal:

Berkas eksternal: myScript.js

function myFunction() {  
  document.getElementById("demo").innerHTML = "Paragraph changed.";  
}

* Skrip eksternal praktis ketika kode yang sama digunakan di banyak halaman web yang berbeda.
* Berkas JavaScript memiliki ekstensi berkas **.js** .
* Untuk menggunakan skrip eksternal, masukkan nama file skrip di srcatribut (sumber) dari sebuah <script>tag:
* Contoh

<script src="myScript.js"></script>

* Keuntungan JavaScript Eksternal
* Ini memisahkan HTML dan kode
* Ini membuat HTML dan JavaScript lebih mudah dibaca dan dirawat
* File JavaScript yang di-cache dapat mempercepat pemuatan halaman
* Contoh menambahkan beberapa file skrip ke satu halaman menggunakan beberapa tag skrip:

<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>

* Referensi Eksternal
* Skrip eksternal dapat direferensikan dalam 3 cara berbeda:
* Dengan URL lengkap (alamat web lengkap)
* Dengan jalur file (seperti /js/)
* Tanpa jalan apa pun
* Contoh ini menggunakan **URL lengkap** untuk menautkan ke myScript.js:

<script src="https://www.w3schools.com/js/myScript.js"></script>

* Contoh ini menggunakan **jalur file** untuk menautkan ke myScript.js:

<script src="/js/myScript.js"></script>

* Contoh ini tidak menggunakan jalur untuk menautkan ke myScript.js:

<script src="myScript.js"></script>

**3. JS Output**

* Tampilan JavaScript
* Menulis ke dalam elemen HTML, menggunakan innerHTMLatau innerText.
* Menulis ke dalam keluaran HTML menggunakan document.write().
* Menulis ke dalam kotak peringatan, menggunakan window.alert().
* Menulis ke konsol browser, menggunakan console.log().
* Menulis ke dalam elemen HTML, menggunakan innerHTMLatau innerText.
* Menulis ke dalam keluaran HTML menggunakan document.write().
* Menulis ke dalam kotak peringatan, menggunakan window.alert().
* Menulis ke konsol browser, menggunakan console.log().
* Menggunakan Inner HTML
* Untuk mengakses elemen HTML, Anda dapat menggunakan document.getElementById(id)metode ini.
* Gunakan idatribut untuk mengidentifikasi elemen HTML.
* Kemudian gunakan innerHTMLproperti untuk mengubah konten HTML dari elemen HTML:
* Menggunakan Inner Text
* Untuk mengakses elemen HTML, gunakan document.getElementById(id)metode.
* Kemudian gunakan innerText properti untuk mengubah teks bagian dalam elemen HTML:
* Contoh
* <!DOCTYPE html>  
  <html>  
  <body>  
    
  <h1>My First Web Page</h1>  
  <p>My First Paragraph</p>  
    
  <p id="demo"></p>  
    
  <script>  
  document.getElementById("demo").innerText = "Hello World";  
  </script>  
    
  </body>  
  </html>
* Gunakan inner HTML ketika Anda ingin mengubah elemen HTML.
* Gunakan inner Text ketika Anda hanya ingin mengubah teks biasa.
* Menggunakan Document.Write()
* Menggunakan document.write() setelah dokumen HTML dimuat, akan menghapus semua HTML yang ada
* Metode document.write() hanya boleh digunakan untuk pengujian.
* Menggunakan window.alert()
* Gunakan kotak peringatan untuk menampilkan data:
* Dalam JavaScript, objek window adalah objek dengan cakupan global. Ini berarti bahwa variabel, properti, dan metode secara default merupakan milik objek window. Ini juga berarti bahwa menentukan window kata kunci bersifat opsional:
* Menggunakan console.log()
* Untuk tujuan debugging, dapat memanggil console.log()metode di browser untuk menampilkan data.
* Cetak JavaSript
* JavaScript tidak memiliki objek cetak atau metode cetak.
* Anda tidak dapat mengakses perangkat keluaran dari JavaScript.
* Satu-satunya pengecualian adalah Anda dapat memanggil window.print()metode di browser untuk mencetak konten jendela saat ini.

**4. JS Statements**

* JavaScript Pernyataan JavaScript adalah **proses** yang dijalankan oleh browser satu per satu, dari atas ke bawah.
* Setiap pernyataan biasanya diakhiri dengan **titik koma;** .
* Pernyataan bisa berdiri sendiri atau dalam blok { ... }.
* JavaScript bersifat peka huruf besar-kecil.
* Gunakan indentasi dan pemisahan baris agar kode mudah dibaca.
* Komentar pakai // komentaratau /\* komentar panjang \*/.

**5. JS Syntax**

* **Sintaksis** adalah aturan penulisan adalah **aturan penulisan** mesin dalam JavaScript agar kode dapat dipahami dan dijalankan oleh browser atau mesin JavaScript.
* JavaScript peka huruf besar/kecil → Nama≠nama
* Gunakan titik koma (;) di akhir pernyataan (tidak wajib, tapi disarankan)
* Jangan gunakan spasi pada nama variabel → gunakan camelCase:namaLengkap

**6. JS Comments**

* Komentar adalah bagian dari kode yang **tidak dijalankan** , digunakan untuk memberi **catatan, penjelasan, atau menonaktifkan baris kode sementara .**

| **Jenis Komentar** | **Contoh** | **Keterangan** |
| --- | --- | --- |
| Komentar satu baris | // Ini komentar satu baris | Gunakan // untuk menulis komentar singkat |
| Komentar banyak baris | /\* Ini komentar \n lebih dari satu baris \*/ | Gunakan /\* \*/ untuk penjelasan panjang |

* Komentar membantu orang lain (dan diri sendiri) memahami kode.
* Jangan berlebihan — tulis komentar yang jelas dan berguna .
* Gunakan untuk menonaktifkan kode saat debugging:

// console.log("Ini tidak akan tampil");

**7. Variables**

* Variabel adalah tempat untuk **menyimpan data** yang bisa digunakan dan dimanipulasi dalam program.
* Cara Mendeklarasikan Variabel

| **Kata Kunci** | **Contoh** | **Keterangan** |
| --- | --- | --- |
| var | var nama = "Ali"; | Lama, bisa *re-declare* dan *reassign*, tidak disarankan (rentan error) |
| let | let umur = 20; | Modern, bisa *reassign*, tapi tidak bisa *re-declare* dalam blok yang sama |
| const | const negara = "Indonesia"; | Nilainya tidak bisa diubah setelah dideklarasikan |

* Aturan Umum Penamaan Variabel
* Nama dapat berisi angka, huruf, garis bawah, tanda dolar.
* Nama harus dimulai dengan huruf.
* Nama juga dapat dimulai dengan $ dan \_
* Nama peka huruf besar/kecil (x dan y merupakan variable yang berbeda).
* Kata-kata yang dicandangkan (seperti kata kunci JavaScript) tidak dapat digunakan sebagai nama.
* Objek →{ nama: "Budi", umur: 20 }
* null , tidak terdefinisi