

NAMA : GAVID RIZAL

NIM : 22650172

Menjelaskan syntax java script

- a) `let list_tugas = [""];`
 - Membuat sebuah array bernama `list_tugas` dengan elemen awal berupa string kosong (`""`). Array ini digunakan untuk menyimpan daftar tugas.
- b) `const output_element = document.querySelector("#output");`
 - Menggunakan `document.querySelector()` untuk memilih elemen HTML dengan `id="output"` dan menyimpannya ke dalam variabel `output_element`. Elemen ini akan digunakan untuk menampilkan daftar tugas.
- c) `function renderTasks() {`
`output_element.innerHTML = "";`
 - Mendefinisikan fungsi `renderTasks()` yang bertujuan untuk memperbarui elemen HTML dengan daftar tugas dari `list_tugas`.
 - `output_element.innerHTML = "";` membersihkan konten elemen `output_element` agar tidak terjadi duplikasi saat daftar diperbarui.
- d) `list_tugas.forEach((value, index) => {`
 - Melakukan iterasi pada array `list_tugas` menggunakan `forEach`. Setiap elemen dari array diakses melalui parameter `value` (nilai tugas) dan `index` (indeks array).
- e) `const p_elm = document.createElement("p");`
 - Membuat elemen paragraf (`<p>`) secara dinamis menggunakan `document.createElement()` dan menyimpannya di variabel `p_elm`.
- f) `p_elm.textContent = value;`
 - Menetapkan teks dari elemen paragraf (`p_elm`) dengan nilai dari elemen array (`value`).
- g) `p_elm.addEventListener("click", () => {`
`list_tugas.splice(index, 1);`
`renderTasks();`
`});`
 - Menambahkan event listener ke elemen `p_elm`. Ketika elemen diklik

- Menghapus elemen dari array `list_tugas` pada indeks tertentu menggunakan `splice`.
 - Memanggil kembali fungsi `renderTasks()` untuk memperbarui daftar tugas yang ditampilkan. javascript Copy code
- h) `output_element.appendChild(p_elm);`
`});`
- Menambahkan elemen paragraf (`p_elm`) yang telah diatur ke dalam elemen HTML `output_element`.
- i) `renderTasks();`
- Memanggil fungsi `renderTasks()` untuk pertama kali agar daftar tugas (saat ini kosong) ditampilkan.
- j) `const formulir = document.querySelector("form");`
- Memilih elemen HTML `<form>` dan menyimpannya ke dalam variabel `formulir`.
- k) `formulir.addEventListener("submit", (e) => {`
`e.preventDefault();`
- Menambahkan event listener ke form untuk menangani event `submit`.
 - `e.preventDefault()` mencegah perilaku default form (seperti reload halaman) agar data dapat diproses langsung oleh JavaScript.
- l) `const formData = new FormData(e.target);`
`const obj = Object.fromEntries(formData);`
- `new FormData(e.target)` mengambil data dari form sebagai objek `FormData`.
 - `Object.fromEntries(formData)` mengubah `FormData` menjadi objek JavaScript biasa, sehingga dapat diakses dengan properti `key-value`.
- m) `if (obj.tugas) {`
`list_tugas.push(obj.tugas);`
`renderTasks();`
`}`
- Memeriksa apakah properti `tugas` dari objek form memiliki nilai.
 - Jika ada nilai, tugas tersebut ditambahkan ke array `list_tugas` menggunakan `push`.
 - Memanggil kembali `renderTasks()` untuk menampilkan daftar yang diperbarui.
- n) `e.target.reset();`
`});`

- Mengosongkan form setelah data berhasil diproses menggunakan `e.target.reset()`.

Menjelaskan syntax html

1. `<!DOCTYPE html>`

`<html lang="en">`

- `<!DOCTYPE html>`: Mendeklarasikan dokumen sebagai HTML5.
- `<html lang="en">`: Membuka elemen root `<html>` dengan atribut `lang="en"`, yang menandakan bahwa dokumen menggunakan bahasa Inggris.

2. `<head>`

`<meta charset="UTF-8" />`

`<meta name="viewport" content="width=device-width, initial-scale=1.0" />`

`<title>Document</title>`

`</head>`

- `<head>`: Bagian ini berisi informasi metadata tentang dokumen HTML.
- `<meta charset="UTF-8" />`: Menentukan encoding karakter dokumen sebagai UTF-8, yang mendukung berbagai karakter termasuk huruf dan simbol non-Inggris.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0" />`
- `<title>Document</title>`: Menetapkan judul dokumen yang akan muncul di tab browser.

3. `<body>`

`<form action="">`

- `<body>`: Membuka bagian utama dokumen HTML yang akan ditampilkan di browser

- `<form action="">`: Membuka elemen form HTML. Atribut action dibiarkan kosong karena pengelolaan data dilakukan menggunakan JavaScript, bukan mengirimkan data ke server.
4. `<input type="text" name="tugas" placeholder="masukan tugas" />`
`<button type="submit">tambah</button>`
 - a. `<input type="text" name="tugas" placeholder="masukan tugas" />`:
 - Membuat input teks untuk pengguna memasukkan tugas.
 - `type="text"`: Menentukan input sebagai teks.
 - `name="tugas"`: Memberikan nama pada input, yang digunakan untuk mengambil data di JavaScript.
 - `placeholder="masukan tugas"`: Menampilkan teks petunjuk sementara dalam kotak input.
 - b. `<button type="submit">tambah</button>`:
 - Membuat tombol submit untuk mengirimkan data form.
 - `type="submit"`: Menentukan tombol sebagai submit, sehingga form akan memicu event submit.
 5. `<div id="output"></div>`
 - Elemen div dengan `id="output"` sebagai wadah untuk menampilkan daftar tugas yang akan dibuat dinamis oleh JavaScript.
 6. `<script src="aplikasi.js"></script>`
 - Menyisipkan file JavaScript eksternal `aplikasi.js`, yang berisi logika utama untuk mengelola tugas.
 7. `</body>`
`</html>`
 - `</body>`: Menutup elemen body.
 - `</html>`: Menutup elemen root HTML.