Pertemuan 03

Pengenalan Javascript

Pendahuluan

Javascript atau disingkat Js merupakan bahasa pemrograman JIT (Just In Time) Compilation, bahasa ini merupakan bahasa High Level, bahasa ini sering digunakan di dalam web technology untuk membuat website lebih interaktif, ada banyak fitur-fitur Javascript ini dan juga project open source yang bisa dipakai kedalam project kita dengan NPM, Javascript ini sendiri sebenarnya bahasa yang dibuat selama 1 minggu pada tahun 1995, tetapi sekarang masih di-lanjutkan dan diawasi oleh standar internasional ECMA, karena sejarah Javascript sangat rumit kita skip aja dulu.

Sebenarnya Javascript ada superset-nya (yang lebih bagus tapi masih sama Javascript) yaitu Typescript sama persis kayak C dengan C++, tapi untuk mempersingkat waktu kita pakai Javascript Dulu.

File javascript berektensi *.js*, kita dapat menggunakan Javascript langsung di HTML dengan menggunakan elemen <script></script>, atau dengan file luar dengan cara :
<script src=”alamat-nya”></script>

Karena Javascript merupakan bahasa High Level dan juga Weakly Type, kita tidak perlu menspesifikan tipe data-nya (walaupun kita nanti akan pusing sendiri kenapa ada tipe data *undefined*), kode di javascript tidak perlu diakhiri dengan *;* tetapi bisa juga diakhiri dengan *;*.

Untuk mendeklarasikan variable kita tinggal memberi

```
var nama_variabel = isi_variabel
let nama_variabel = isi_variabel
const nama_variabel = isi_variabel
```

deklarasi variabel ada 3 jenis, var, let, const

- var = variabel ini diluar function akan bersifat global, kalau didalam function akan hanya ada didalam function itu sendiri, dan scope didalam function itu sendiri.
- let = merupakan variabel yang mengikuti scope (hanya 1 scope).
- const = merupakan variable yang konstant (bisa global).

Scope merupakan istilah pemrograman yang bermaksud umur dari sesuatu.

Misalnya

```
Scope A
{
   Scope B
   {
       Scope C
   }
}

{
   Scope D
}
```

Scope A ini dapat diakses di Scope A, B, C D
Scope B dapat diakses di Scope B, C (Tergantung deklarasi variabel-nya)

Scope C hanya dapat di akses di Scope C
Scope D hanya dapat di akses di Scope D

Nah ini pasti nih kepo kalo bagaimana kita menulis sesuatu lewat javascript, ada banyak cara, cara paling sederhana (dan tidak aman di website umum).

document.write("Hello, World");

Ada yang sedikit aman dengan menggunakan console.log(“isi”), tetapi kita harus mengeluarkan development tool (dev tool) yang disediakan oleh browser (firefox tinggal pencet f12 di keyboard)

console.log(“Hello, World”)
console.warning(“Hello, World”)
console.error(“Hello, World”)
console.info(“Hello, World”)

kenapa console ada banyak jenis-nya, sebenernya console itu sebuah class object global yang disediakan oleh Javascript untuk memberikan akses kepada kita console-nya browser, terus (.) ini bermaksud method/function didalam kelas tersebut, nah ada log, warning, error, info ini dapat kita gunakan untuk mempermudah debugging di Javascript karena kita dapat men-filter-nya lewat console-nya browser

Language Specific Type

Javascript memiliki tipe khusus yang dinamai Javascript Object, ini hampir mirip dengan object class tetapi sedikit berbeda, tipe data ini sebaiknya disimpan dengan const agar tidak sangat mudah ke overwrite.

Contoh :

```
const profile = {
        ‘nama’: ‘Akip’,
        ‘no-absen’: 0,
        ‘kelas’: ‘X PPLG’,
}

console.log(profile.nama) // cara akses-nya
profile.nama = ‘Agus’ // menganti isi “nama”
profile = ‘Agus’ // Tidak bisa, karena const
```

atau bisa

```
const profile = {
        nama: ‘Akip’,
        no_absen: 0,
        kelas: ‘X PPLG’
}
```

Control Flow

Control Flow hampir sama di pertemuan kita yang pertama, bedanya cuman foreach-nya

```
while loop :
while (variable) {
   // code
}

for loop :
for (I = 0; I < 10; i++) {
   // code
}

foreach loop :
const data = [1, 2, 3, 4, 5]

data.forEach(function (value, index) {
   // code
})
```

Function

Function hampir sama dengan PHP

```
const args = 1;

namafungsi (args) {
        return 1 + args;
}

namafungsi(1);
console.log(namafungsi(1))
```

OOP

Class di javascript hampir sama dengan PHP karena Javascript juga OOP

```
class Meja {
        constructor() {
                // code yang akan di eksekusi ketika object dibuat
                // di php __construct
        }
        namaFungsi(){
                // code
        }
}

const Furniture = new Meja(); // membuat object.
```

Arrow Function

di javascript kita bisa menyingkat function kita dengan istilah arrow function misal saat kita forEach data array

```
data.forEach((value, index) => {
   // code
})
```

atau deklarasi function

```
const namaFunction = (args) => {
   // code
}
```

```
const angkaSatu = () => 1; // ini langsung return 1
```

sama saja tetapi lebih pendek

HTML Element Manipulation (DOM Manipulation)

Ini dimana Javascript sangat jaya, DOM Manipulation atau Document Object Manipulation, merupakan istilah ketika mengotak atik HTML Element di halaman website, seperti yang di contohkan jauh diatas

```
document.write("Hello, World");
```

Merupakan salah satu perintah dari DOM karena kita dapat membuat HTML elemen dengan menggunakan perintah ini (Walaupun tidak bagus untuk membuat elemen dengan cara begini)

Ketika kita membuat HTML Element secara normal kita terkadang menginginkan untuk merubah isi-nya, caranya simpel kita tinggal memasang properti id di elemen yang kita inginkan, lalu kita manipulasi-kan lewat Javascript

```
// HTML

<p id='hello-world-p'>Hello, World</p>
```

Elemen ini secara default akan menampilkan Hello, World
Untuk mengganti isi Hello, World tsb kita tinggal menggunakan id yang sudah menempel di elemen tersebut dengan menggunakan perintah

```
const helloWorld = document.getElementById('hello-world-p')
```

karena getElementById ini mengembalikan data dengan format Javascript Object kita secara tradisi menyimpan data tersebut dengan const dikarenakan kita tidak ingin variabel tsb ke overwrite.

kalau kurang yakin bisa di console.log variabel helloWorld, nanti muncul banyak sekali value yang bisa temen-temen ubah, tapi hati-hati jangan asal ubah, kita fokus di textContent karena ini mengatur isi dari elemen tersebut, kita bisa saja pakai innerHTML tetapi itu bisa kena hack dan sangat tidak aman.

```
HelloWorld.textContent = "Hello, Javascript"
```

Harus di-ingat, DOM Manipulation itu sangat mahal (di bidang performa) jadi kalau bisa kurangi animasi yang membutuhkan DOM Manipulation (Pengalaman) kalau mau animasi kalau bisa pakai CSS Animation, kalau tidak bisa ya di trigger lewat Javascript juga bisa, tapi asal animasi-nya berasal dari CSS ok-ok aja.

Membuat Elemen HTML Baru Saat Runtime (ditengah jelajah user)

Kita pasti penasaran kalo kita pengen buat elemen baru yang aman bagaimana, kita menggunakan *document* class lagi bedanya kita pakai method *createElement* method ini akan return sebuah Elemen/Node yang dapat disimpan di variabel (lebih bagus disimpan di const ya kalau sekali pakai, kalau buat elemen-nya loop pakai let saja)

Contoh operasi membuat elemen

```
// HTML

<div id=”root’></div>

<script>
	const root = document.getElementById('root');
	const h1 = document.createElement("h1");
	h1.textContent = "Hello, World"
	appendTarget.appendChild(h1);
</script>
```

Untuk menghapus anak elemen di DOM kita tinggal menggunakan method deleteChild di variabel yang nyimpen Node misal

```
root.deleteChild(nodenya);
```

Ingat, untuk men-nambahkan anak elemen hanya terbatas di elemen tertentu, kalo pasti jelas bisa

Dari contoh di-atas ini mengingatkan kita kalo Javascript Framework ada yang memiliki cara kerja seperti ini tetapi lebih mudah dan cepat.

Note : kita nanti menjumpai elemen/node javascript ketika kita ingin akses isi-nya khusus-nya <input> ini kita memakai *value* bukan textContent

Event Listener

Javascript juga mempunyai fitur yang sangat bermanfaat yaitu event listener, kegunaanya ialah untuk membuat event ketika kita meng-klik, menskrol atau apapun event-nya di elemen, semua elemen dapat dikasih event listener.

Misalnya kita membuat tombol untuk memberikan user yang sedang meng-akses website kita alert

pertama buat button atau elemen apapaun (lebih disarankan button karena yang masuk akal)

```
<button id=’tombol’>Tombol</button>
```

di bagian script tag hampir sama seperti sebelum-nya bedanya kita menambahkan event listener bukan DOM Manipulation

```
<script>
	const tombolPeringatan = document.getElementById(‘tombol’);
	tombolPeringatan.addEventListener(‘click’, (elemen, event) => {
		// kita pakai arrow function
		alert(‘Hey kamu mencet tombol yang terlarang!’);
	})
</script>
```

ketika kita buka dibrowser lalu klik tombol tersebut nanti muncul alert via browser dengan tulisan “Hey kamu mencet tombol yang terlarang!”

Kita juga dapat melakukan sesuatu tanpa eventListener dengan menggunakan attribute HTML yang tersambung langsung dengan javascript

Contoh:

```
<input type="text" id='input'>
<button onsubmit='doSomething'>Submit</button>

<script>
	const input = document.getElementById('input');
	const doSomething = () => {
	   alert(input.value); menampilkan isi <input> dengan cara alert
	   input.value = ''; // mereset isi dari <input>
	};
</script>
```

Tugas : Membuat TodoList yang dapat menambah Todo men-delete Todo bebas mau pakai apa kreatif juga boleh pakai yang sederhana kayak eventListener, createElement juga boleh.

Kalau masih bingung cara-nya boleh tanya, sekalian ada trik-nya

Note : Karena ada banyak fitur Javascript yang belum kita bahas in depth, temen-temen bisa mampir ke https://www.w3schools.com/js buat belajar lebih lanjut