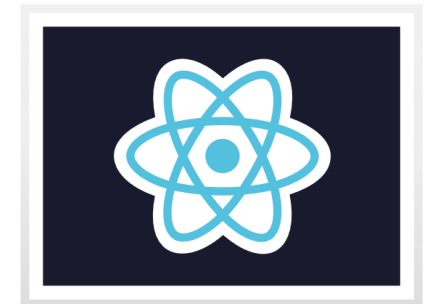
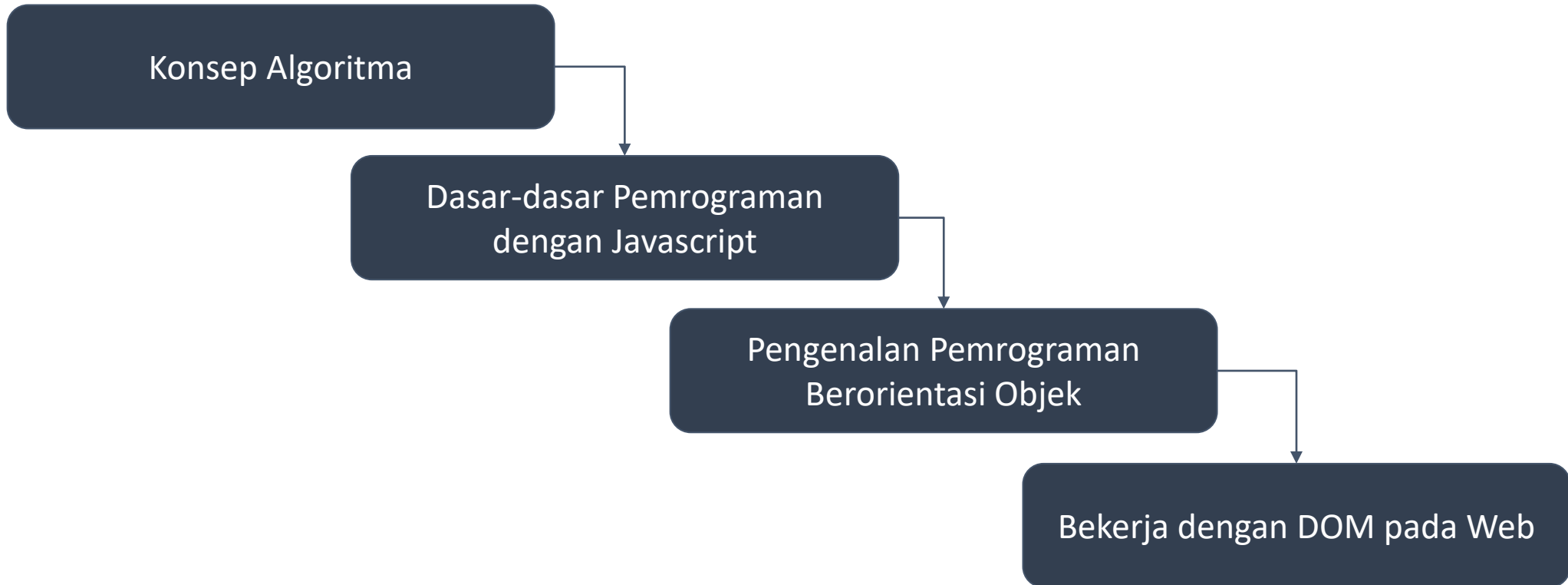


# PEMROGRAMAN JAVASCRIPT



- **KODE** : MK03
- **MATAKULIAH** : PEMROGRAMAN JAVASCRIPT
- **SKS** : 4 SKS
- **SEMESTER** : 1 (SATU)



"Meraih masa depan yang cerah tidak akan didapat dengan mudah, kamu harus mau berkorban untuk mendapatkan hal itu."

**- BJ. Habibie -**

# EVENT JAVASCRIPT

## Event JavaScript

Event adalah sebuah sinyal yang menyatakan bahwa sesuatu terjadi. Semua DOM menghasilkan sinyal tetapi event tidak terbatas pada DOM saja.

- Mouse event (click, contextmenu, mouseover, dll)
- Keyboard event (keydown, keyup, dll)
- Form element event (submit, focus, dll)
- Document event (DOMContentLoaded, dll)
- CSS event (transitionend, dll)

## Event Handler

Event handler adalah sebuah fungsi yang disiapkan untuk bereaksi terhadap suatu event. Jika event tertentu terjadi, maka fungsi ini akan dijalankan.

Event handler dapat dituliskan melalui beberapa cara :

- Dalam atribut HTML on<event> (contoh, onclick)
- Dalam properti DOM on<event>

# EVENT HANDLER

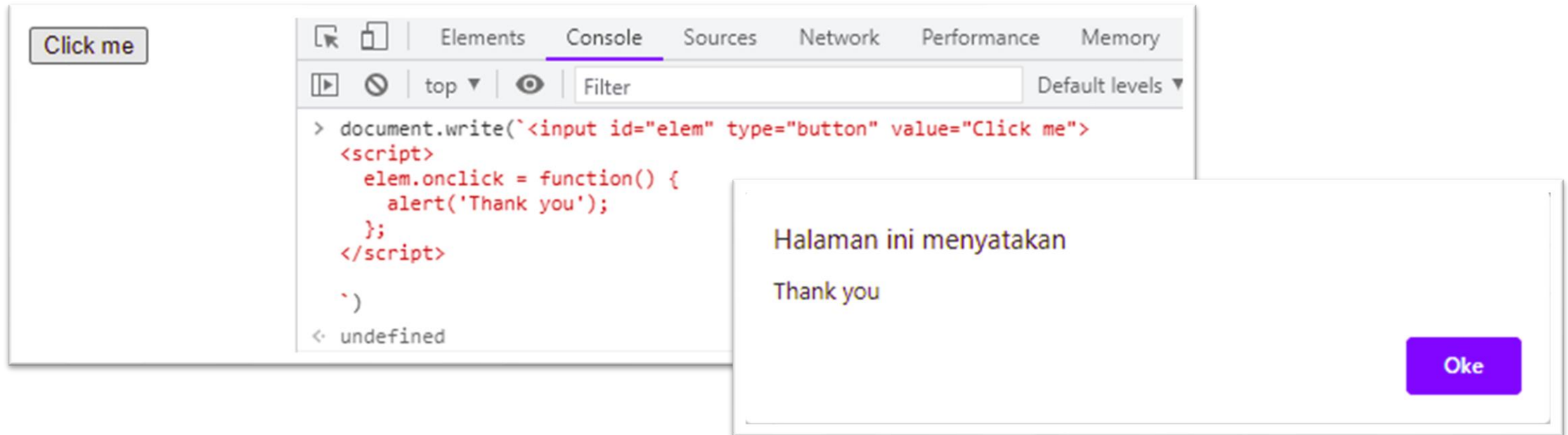
```
<input value="Click me" onclick="alert('Click!')" type="button">
```

```
<input id="elem" type="button" value="Click me">  
<script>  
  elem.onclick = function() {  
    alert('Thank you');  
  };  
</script>
```



# EVENT HANDLER

Berikut ini adalah contoh event handler untuk event onclick. Fungsi akan dieksekusi saat tombol (atau objek yang memiliki event handler tersebut) diklik.



Event handler hanya satu buah saja untuk tiap objek. Kita tidak bisa assign lebih dari satu fungsi. Assignment fungsi yang kedua akan menggantikan fungsi pertamanya. Untuk menangani hal tersebut, pada elemen DOM tersedia metode

- `addEventListener`
- `removeEventListener`

Syntaxnya :

```
element.addEventListener(<event>, <handler>, [options]);
```

## Event Listener

Fungsi `addEventListener` lebih universal dibandingkan memasukkan event handler langsung ke properti DOMnya. Beberapa event ada yang tidak bisa diassign melalui properti DOM, contohnya `DOMContentLoaded`.

```
document.addEventListener = function() {  
    alert("DOM built");  
}; // alertnya tidak akan pernah dijalankan
```

## Event Listener

Oleh karena itu, kita perlu menggunakan `addEventListener`

```
document.addEventListener("DOMContentLoaded", function() {  
    alert("DOM built");  
}); // penggunaan yang benar, dengan addEventListener
```

## Browser Default Action

Beberapa event secara otomatis menjalankan action di browser, misalnya :

- Mengklik link akan menginisiasi navigasi ke url tersebut
- Mengklik tombol submit pada form akan mengirimkan data submissionnya ke server
- Menekan tombol mouse dan menggesernya sepanjang teks akan menyorot dan memilih teks tersebut.

Dalam menghandle event, kadang kita tidak ingin aksi tersebut dijalankan secara otomatis agar tidak mengganggu perilaku fungsi kita.

## Prevent Default

Ada 2 cara agar browser tidak melakukan aksi yang seharusnya (default):

- Menggunakan fungsi `preventDefault` di dalam objek event
- Membuat fungsi mengembalikan nilai `false` (hanya berlaku untuk fungsi yang ditambahkan melalui properti DOM `on<event>`, bukan `addEventListener`)

`<a href="/otherurl/" onclick="return false">Klik link ini</a>`

atau

`<a href="/otherurl/" onclick="event.preventDefault()">link ini</a>`

## Custom Event

Buatlah kode berikut, dengan sedikit CSS bisa tampil seperti button

Tapi sifat dasar dari link tetap muncul (waktu hover akan tampil link tujuannya, waktu diklik akan pindah halaman)

```
<ul id="menu" class="menu">
  <li><a href="/html">HTML</a></li>
  <li><a href="/javascript">JavaScript</a></li>
  <li><a href="/css">CSS</a></li>
</ul>
```

[HTML](#)[JavaScript](#)[CSS](#)

## Custom Event

Kita ganti perilaku link tersebut saat diklik (misalnya menampilkan alert)

```
menu.onclick = function(event) {  
    if (event.target.nodeName != 'A') return;  
    let href = event.target.getAttribute('href');  
    alert( href ); // perilaku baru yang diberikan  
    return false; // mencegah browser menavigasi ke halaman baru  
};
```



## UI Event

- UIEvent
- FocusEvent
- MouseEvent
- WheelEvent
- InputEvent
- KeyboardEvent
- CompositionEvent

## Mouse Event

Selain diklik, berikut ini event yang bisa terjadi karena mouse:

- mouseover (hover) saat pointer berada di atas element;
- mouseenter saat pointer mendekat pada elemen;
- mouseout saat pointer menjauh dari elemen;

## Keyboard Event

Berikut ini beberapa event yang bisa terjadi saat kita menekan keyboard:

- keypress: action saat kita menekan dan menahan tombol tertentu;
- keyup: action saat kita berhenti menekan (melepas) tombol tertentu;
- keydown: action saat kita mulai menekan tombol tertentu;

## Event Change

Event change biasanya terjadi pada elemen input seperti input text, radio, checkbox, select-option, dll. Event change akan terjadi saat nilai pada elemen tersebut berubah.

# TUGAS JAVASCRIPT

---

1. Buatlah program yang menggunakan mouse event
2. Buatlah program yang menggunakan keyboard event
3. Buatlah program yang menggunakan event change
4. Simpan dalam file index.js
5. Upload pada akun github kalian dengan nama repository “Pertemuan-11-JS”



**Terima Kasih  
Syukron Katsiron  
Arigatou Gozaimasu**