

PROGRAM STUDI INFORMATIKA
SEMESTER GASAL 2023/2024

PEMROGRAMAN WEB I

3 SKS – MINGGU 7

BENISIUS ANU – ehba@ukrida.ac.id

WENDYANTO PANGGALO – panggalowendyanto@ukipaulus.ac.id

informatika.ukrida.ac.id

ukipaulus.ac.id/page/fakultas/3/informatika-dan-komputer



UKRIDA
Universitas Kristen Krida Wacana



SILABUS

PEMROGRAMAN WEB

- *JavaScript Events*
- *JavaScript Arrays*
- *JavaScript Objects*
- *JavaScript HTML DOM*
- *AJAX*

EVENTS

JAVASCRIPT



- *JavaScript* memungkinkan kode dieksekusi ketika *events* terdeteksi.
- HTML memungkinkan atribut *event handler*, dengan kode *JavaScript*, ditambahkan ke elemen HTML.

`<element event='some JavaScript'>`

or

`<element event="some JavaScript">`

EVENTS

JAVASCRIPT



Contoh:

```
<button onclick="document.getElementById('demo').innerHTML =  
Date()">The time is?</button>
```

```
<button onclick="this.innerHTML=Date()">The time is?</button>
```

```
<button onclick="displayDate()">The time is?</button>
```

EVENTS

JAVASCRIPT



Berikut adalah daftar *HTML events*:

<i>Event</i>	<i>Description</i>
onchange	<i>An HTML element has been changed</i>
onclick	<i>The user clicks an HTML element</i>
onmouseover	<i>The user moves the mouse over an HTML element</i>
onmouseout	<i>The user moves the mouse away from an HTML element</i>
onkeydown	<i>The user pushes a keyboard key</i>
onload	<i>The browser has finished loading the page</i>

EVENTS

AKTIVITAS LAB 1



Buatlah kode program sebagai bentuk implementasi dari *JavaScript events*:

1. *Onchange*
2. *Onmouseover*
3. *Onmouseout*
4. *Onkeydown*
5. *Onload*

ARRAYS

JAVASCRIPT



- *Arrays* adalah semacam wadah yang dapat menampung banyak nilai. Nilai-nilai di dalam suatu *array* disebut *elements*.

```
var breakfast = ["roti", "sereal"];
```

- *Array elements* tidak harus memiliki tipe data yang sama. Nilai *elements* dapat bertipe data apa saja — bahkan juga *array*.

```
var gadogado = [100, "telur", [200, "sayur"], false];
```

ACCESSING ELEMENTS

JAVASCRIPT ARRAYS



- Untuk mengakses nilai *element* di dalam suatu *array*, dapat dilakukan dengan menggunakan tanda kurung dan angka seperti berikut:

```
myArray[3];
```

- *JavaScript Array* dimulai dari 0, sehingga *element* pertama akan selalu berada pada indeks [0].

```
var brothers = ["Upin", "Ipin"];
```

```
brothers[0];
```

```
brothers[1];
```


ACCESSING ELEMENTS

JAVASCRIPT ARRAYS



- Untuk mendapatkan *element* terakhir, bisa menggunakan tanda kurung dan panjang *array* - 1.

```
var actors = ["Felicia", "Nathan", "Neil"];
```

```
actors[actors.length - 1];
```

ACCESSING ELEMENTS

JAVASCRIPT ARRAYS



- Cara ini juga berfungsi untuk mengeset nilai *element*.

```
var colors = ["red", "yellow", "blue"];
```

```
colors[1] = "yellow";
```

- Pada *JavaScript*, isi suatu *array* dapat diakses dengan merujuk pada nama *array* itu:

```
colors;
```

ARRAYS METHODS

JAVASCRIPT ARRAYS



Array memiliki variabel dan fungsi bawaannya sendiri yang dikenal sebagai *property* dan *method*. Berikut adalah beberapa yang paling umum:

- *length*
- *concat*
- *pop*
- *push*
- *reverse*

ARRAYS METHODS

AKTIVITAS LAB 2



Buatlah kode program sebagai bentuk implementasi dari *arrays methods* berikut:

1. *length*
2. *concat*
3. *pop*
4. *push*
5. *reverse*

OBJECTS

JAVASCRIPT



Objects adalah nilai yang bisa mengandung nilai lainnya. *Object* menggunakan *keys* untuk mengakses nilai.

Contoh:

```
var student = {  
  id: "412017001",  
  name: "James Bono",  
  age: 20  
};
```

GETTING KEYS

JAVASCRIPT OBJECTS



Untuk mengakses *object's key*, ada dua cara:

- Menggunakan notasi titik yang diikuti nama *key* setelahnya.

student.name;

- Menggunakan notasi kurung siku dengan nama *key* dituliskan dalam bentuk string [""].

student["name"];

SETTING KEYS

JAVASCRIPT OBJECTS



- Untuk menambahkan keys ke dalam sebuah *object* atau mengganti nilai *key*, caranya sama dengan cara mengakses *key*.
- Gunakan notasi titik dengan nama *key* setelah notasi titik dan tanda sama dengan =.

```
var character = { name: "Donna", hair: "red" };
```

```
character.hair = "blonde";
```

SETTING KEYS

JAVASCRIPT OBJECTS



- Atau dapat juga menggunakan notasi kurung siku dengan nama key di dalam string di dalam tanda kurung siku [""] =.

```
var character = {  
    name: "Ricardo",  
    age: 32,  
    hair: "red"  
};  
character["hair"] = "red";
```

DISPLAYING OBJECT PROPERTIES

AKTIVITAS LAB 3



Dengan menggunakan *object* character yang didefinisikan di halaman sebelumnya, cetak semua properties yang dimiliki *object* tersebut di halaman web. Hint: gunakan perulangan.

HTML DOM

JAVASCRIPT



- Ketika suatu halaman web dimuat, *browser* membuat **Document Object Model (DOM)** halaman.
- HTML DOM adalah *object model* standar dan antarmuka pemrograman untuk HTML. Ini mendefinisikan:
 - *HTML elements* sebagai **objects**
 - **Properties** dari semua *HTML elements*
 - **Methods** untuk mengakses semua *HTML elements*
 - **Events** untuk semua *HTML elements*

HTML DOM

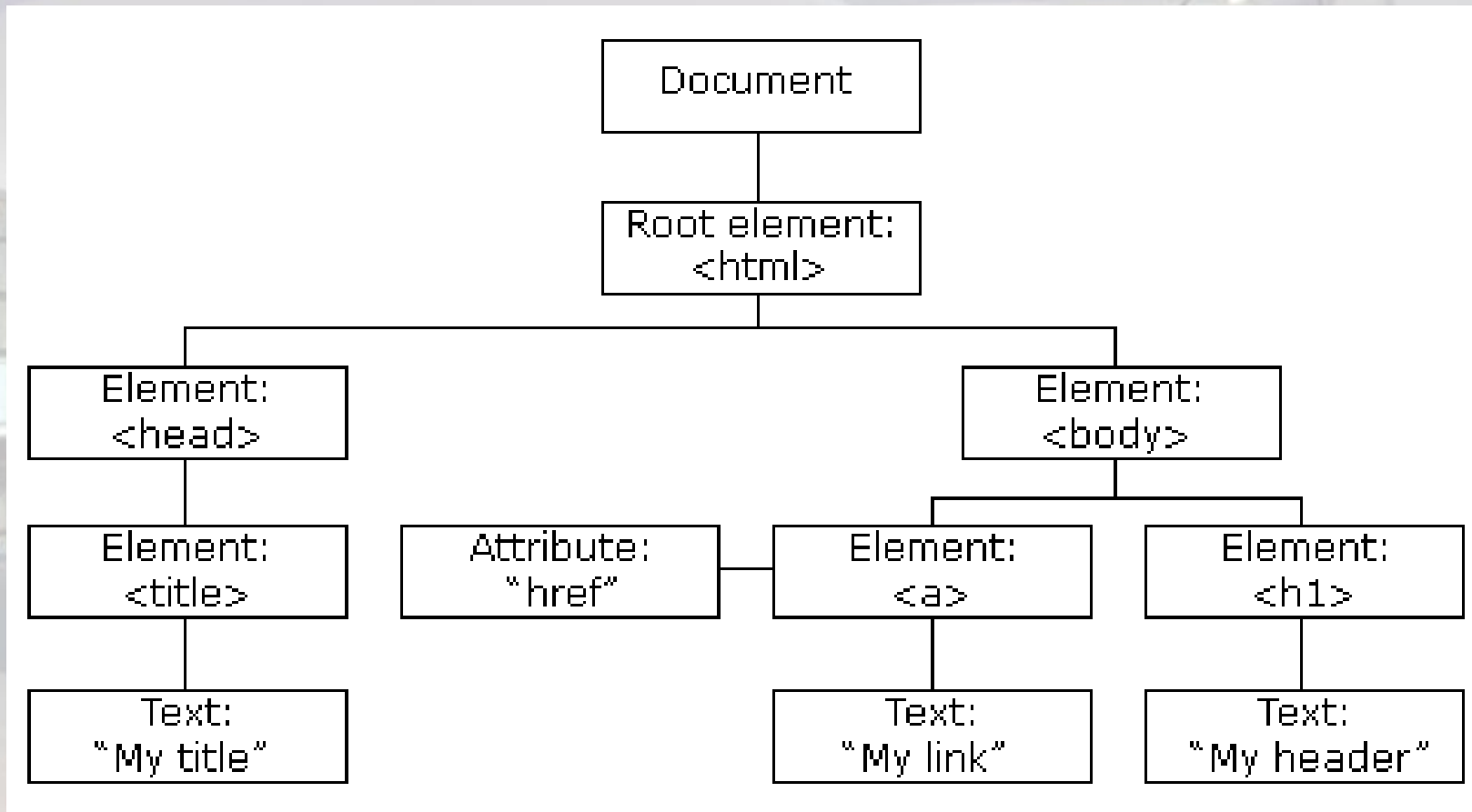
JAVASCRIPT



- *DOM* adalah standar W3C (*World Wide Web Consortium*).
- *HTML DOM* adalah standar untuk mendapatkan, mengubah, menambah, atau menghapus elemen HTML.
- *HTML DOM model* dibangun sebagai sebuah *tree of Objects*:

HTML DOM

JAVASCRIPT



*The HTML
DOM tree of
objects*

HTML DOM

JAVASCRIPT



Dengan *object model*, *JavaScript* dapat membuat HTML dinamis:

- *JavaScript* dapat mengubah semua HTML elements
- *JavaScript* dapat mengubah semua HTML attributes
- *JavaScript* dapat mengubah semua CSS styles
- *JavaScript* dapat menghapus HTML elements dan attributes yang ada
- *JavaScript* dapat menambahkan HTML elements dan attributes baru
- *JavaScript* dapat bereaksi terhadap semua HTML events yang ada
- *JavaScript* dapat membuat HTML events baru

HTML DOM

JAVASCRIPT



- HTML DOM dapat diakses dengan *JavaScript*.
- Pada DOM, semua *HTML elements* didefinisikan sebagai ***objects***.
- Antarmuka pemrograman adalah *property* dan *method* dari masing-masing *object*.
- *Property* adalah nilai yang bisa didapatkan atau diset (seperti mengubah konten *HTML element*).
- *Method* adalah aksi yang bisa dilakukan (seperti *add* atau *deleting* suatu *HTML element*).

HTML DOM

JAVASCRIPT



Contoh berikut, `getElementById` adalah suatu *method*, sedangkan `innerHTML` adalah suatu *property*:

```
<html>
<body>
  <p id="demo"></p>

  <script>
    document.getElementById("demo").innerHTML = "Hello World!";
  </script>

</body>
</html>
```


HTML DOM

JAVASCRIPT



- Cara paling umum untuk mengakses *HTML element* adalah dengan menggunakan *id element* itu.
- Pada contoh sebelumnya *getElementById method* menggunakan `id="demo"` untuk menemukan *element*.
- Cara termudah untuk mendapatkan konten suatu *element* adalah dengan menggunakan *property innerHTML*.
- *Property innerHTML* berguna untuk mendapatkan atau mengganti konten *element HTML*.
- *Property innerHTML* dapat digunakan untuk mendapatkan atau mengubah elemen HTML apa pun, termasuk `<html>` dan `<body>`.

FINDING HTML ELEMENTS

JAVASCRIPT HTML DOM



- Mencari *element* dengan *element id*:

document.getElementById(id)

- Mencari *elements* dengan *tag name*:

document.getElementsByTagName(name)

- Mencari *elements* dengan *class name*:

document.getElementsByClassName(name)

CHANGING HTML ELEMENTS

JAVASCRIPT HTML DOM



- Mengubah *inner HTML* dari suatu *element*:
element.innerHTML = new html content
- Mengubah nilai *attribute* dari suatu *HTML element*:
element.setAttribute(attribute, value)
- Mengubah *style* dari suatu *HTML element*:
element.style.property = new style

CHANGING HTML ELEMENTS

AKTIVITAS LAB 4



Buatlah kode program sebagai bentuk implementasi dari mengubah html element *berikut*:

1. Mengubah *inner HTML* dari suatu *element*.
2. Mengubah nilai *attribute* dari suatu *HTML element*
3. Mengubah *style* dari suatu *HTML element*.

ADDING AND DELETING ELEMENTS

JAVASCRIPT HTML DOM



- Membuat suatu *HTML element*:

document.createElement(element)

- Menghapus suatu *HTML element*:

document.removeChild(element)

- Menambahkan suatu *HTML element*:

document.appendChild(element)

ADDING AND DELETING ELEMENTS

JAVASCRIPT HTML DOM



- Mengganti suatu *HTML element*:

document.replaceChild(element)

- Menulis ke dalam *HTML output stream*:

document.write(text)

ADDING AND DELETING ELEMENTS

AKTIVITAS LAB 5



Buatlah kode program sebagai bentuk implementasi dari mengubah html element *berikut*:

1. Membuat suatu *HTML element*.
2. Menghapus suatu *HTML element*.
3. Menambahkan suatu *HTML element*.
4. Mengganti suatu *HTML element*.
5. Menulis ke dalam *HTML output stream*.

ADDING EVENTS HANDLERS

JAVASCRIPT HTML DOM



- Menambahkan kode event *handler* ke suatu ***onclick*** event:

document.getElementById(id).onclick = function(){code}

- Menambahkan kode event *handler* ke suatu ***onchange*** event:

document.getElementById(id).onchange = function(){code}

ADDING EVENTS LISTENER

JAVASCRIPT HTML DOM



- `addEventListener()` *method* menambahkan suatu *event handler* kepada *element* tertentu.

`element.addEventListener(event, function, useCapture);`

- Parameter pertama adalah jenis *event* (seperti "*click*" atau "*mousedown*").
 - Parameter kedua adalah *function* yang dipanggil ketika *event* terjadi.
 - Parameter ketiga adalah suatu nilai *boolean* yang menentukan apakah *event bubbling* atau *event capturing*.
- Perhatikan bahwa Anda tidak menggunakan awalan "on" untuk *event* tersebut; gunakan "click" daripada "onclick".

ADDING EVENTS LISTENER

JAVASCRIPT HTML DOM



- Tambahkan *event listener* yang dijalankan saat pengguna mengklik sebuah tombol:

```
document.getElementById("myBtn").addEventListener("click",  
displayDate);
```

or

```
document.getElementById("myBtn").addEventListener("click",  
function(){  
    alert("Hello World!");  
});
```

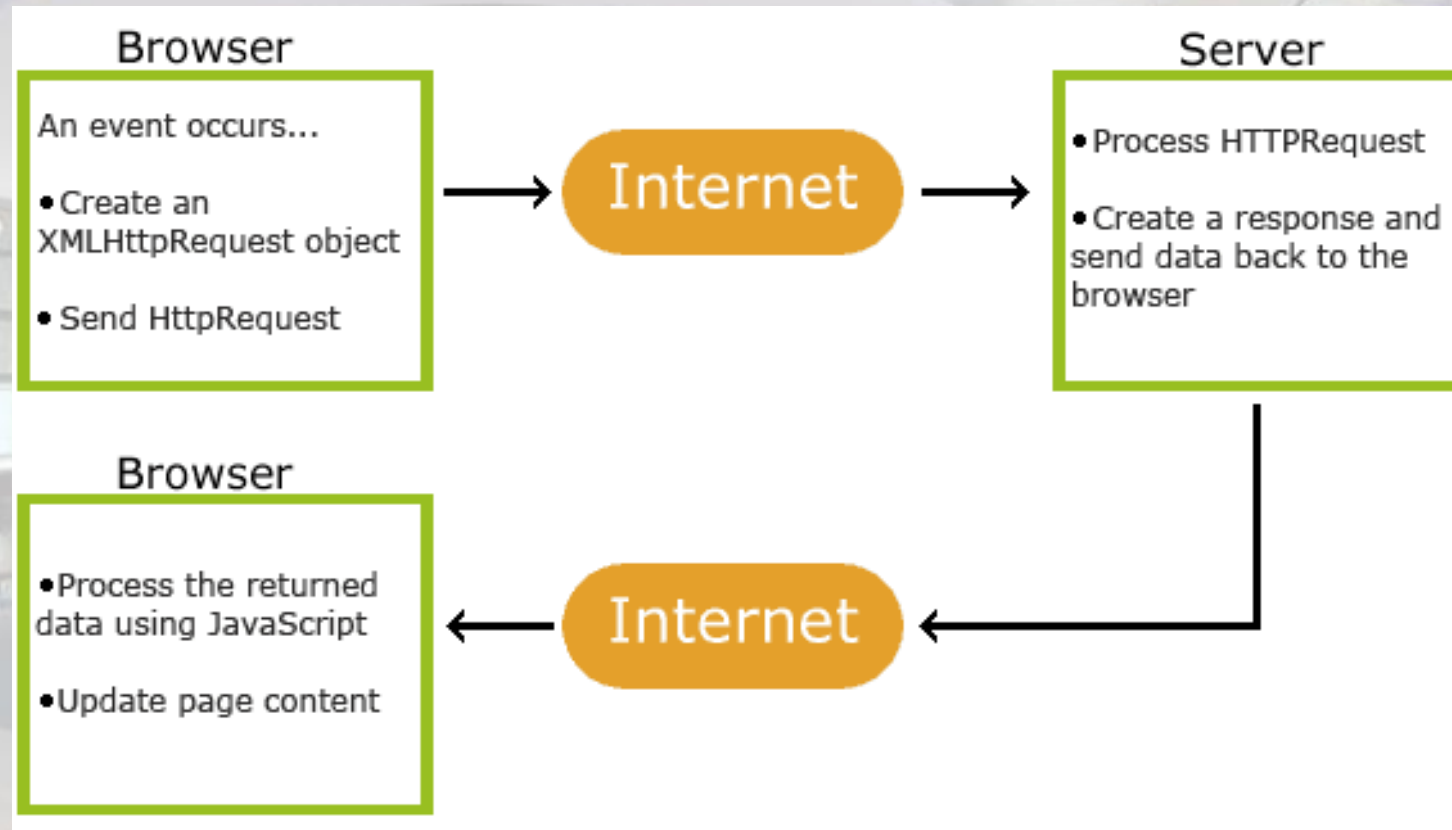
AJAX

PEMROGRAMAN WEB II

- *AJAX is about updating parts of a web page, without reloading the whole page.*
- ***AJAX = Asynchronous JavaScript and XML.***
- *AJAX is a technique for creating fast and dynamic web pages.*
- *AJAX allows web pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. This means that it is possible to update parts of a web page, without reloading the whole page.*
- *Classic web pages, (which do not use AJAX) must reload the entire page if the content should change.*

AJAX

PEMROGRAMAN WEB II



1. An event occurs in a web page (the page is loaded, a button is clicked)
2. An XMLHttpRequest object is created by JavaScript
3. The XMLHttpRequest object sends a request to a web server
4. The server processes the request
5. The server sends a response back to the web page
6. The response is read by JavaScript
7. Proper action (like page update) is performed by JavaScript

AJAX

PEMROGRAMAN WEB II

AJAX is based on internet standards, and uses a combination of:

- ***XMLHttpRequest object** (to exchange data asynchronously with a server)*
- ***JavaScript/DOM** (to display/interact with the information)*
- ***CSS** (to style the data)*
- ***XML** (often used as the format for transferring data)*

AJAX

AKTIVITAS LAB 6

1. Buat sebuah tombol “Change Content” yang akan menjalankan fungsi `loadContent()`.

```
<!DOCTYPE html>
<html>
<body>

<div id="demo">
  <h2>Let AJAX change this text</h2>
  <button type="button" onclick="loadContent()">Change Content</button>
</div>

</body>
</html>
```

AJAX

AKTIVITAS LAB 6

2. Buat fungsi **loadcontent()** yang akan mengisi element dengan id="demo" dengan konten dari file ajax_test.txt.

```
function loadContent() {  
    var xhttp = new XMLHttpRequest();  
    xhttp.onreadystatechange = function() {  
        if (this.readyState == 4 && this.status == 200) {  
            document.getElementById("demo").innerHTML = this.responseText;  
        }  
    };  
    xhttp.open("GET", "ajax_test.txt", true);  
    xhttp.send();  
}
```