



PERTEMUAN IX

# PEMROGRAMAN WEB

# PENGENALAN

# JAVASCRIPT

Izmy Alwiah Musdar  
[izmyalwiah@gmail.com](mailto:izmyalwiah@gmail.com)

# CONTENT

- 01 APA JAVASCRIPT
- 02 CARA PENULISAN JAVASCRIPT
- 03 MENAMPILKAN HASIL PROGRAM
- 04 DASAR JAVASCRIPT
- 05 EVENT
- 06 WINDOW DAN DOKUMEN
- 07 TANGGAL DAN WAKTU
- 08 OBJEK JAVASCRIPT
- 09 *EVENT TIMING*

# TUJUAN PEMBELAJARAN

- Mahasiswa mampu menjelaskan kegunaan JavaScript dalam pembuatan website.
- Mahasiswa mampu menggunakan JavaScript untuk manipulasi HTML dan CSS.
- Mahasiswa mampu mengimplementasikan JavaScript untuk pada halaman website.

# APA JAVASCRIPT

**JavaScript**



- Pemrosesan dilakukan di sisi client
- JavaScript dijalankan ketika ada event tertentu yang terjadi pada halaman web.
- Pesan error dapat dilihat pada Inspect (inspeksi) browser.
- Komentar (//) atau (/\*..\*/).

# CARA PENULISAN JAVASCRIPT

## Inline

- Menggunakan *event handler*.
- Digunakan jika kode hanya sedikit seperti hanya memanggil function.
- Meletakkan kode JavaScript setelah attribut event pada tag HTML.

## Internal

- Digunakan ketika JavaScript terdiri dari beberapa baris.
- Meletakkan kode JavaScript di dalam tag `<script type="text/javascript">` ... `</script>` yang diletakkan di dalam tag `<head>` atau `<body>`.

## Eksternal

- Digunakan apabila JavaScript memiliki kode yang panjang.
- Kode javascript disimpan dalam file tersendiri dengan format .js.
- Menggunakan tag `<script src="file_javascript.js">` diletakkan di dalam tag `<head>`.

```

<!DOCTYPE html>
<html>
<head>
    <script src="js/plotly-2.27.0.min.js"></script>
</head>
<body>
<div class="input-data">
    Perempuan : <input type="number" id="p"></input>
    Laki-laki : <input type="number" id="l"></input>
    <button onclick="pieChart()">Pie Chart</button>
</div>
<div id="myPlot" style="width:100%;max-width:700px"></div>

<script>
    function pieChart(){
        var p = document.getElementById("p").value;
        var l = document.getElementById("l").value;

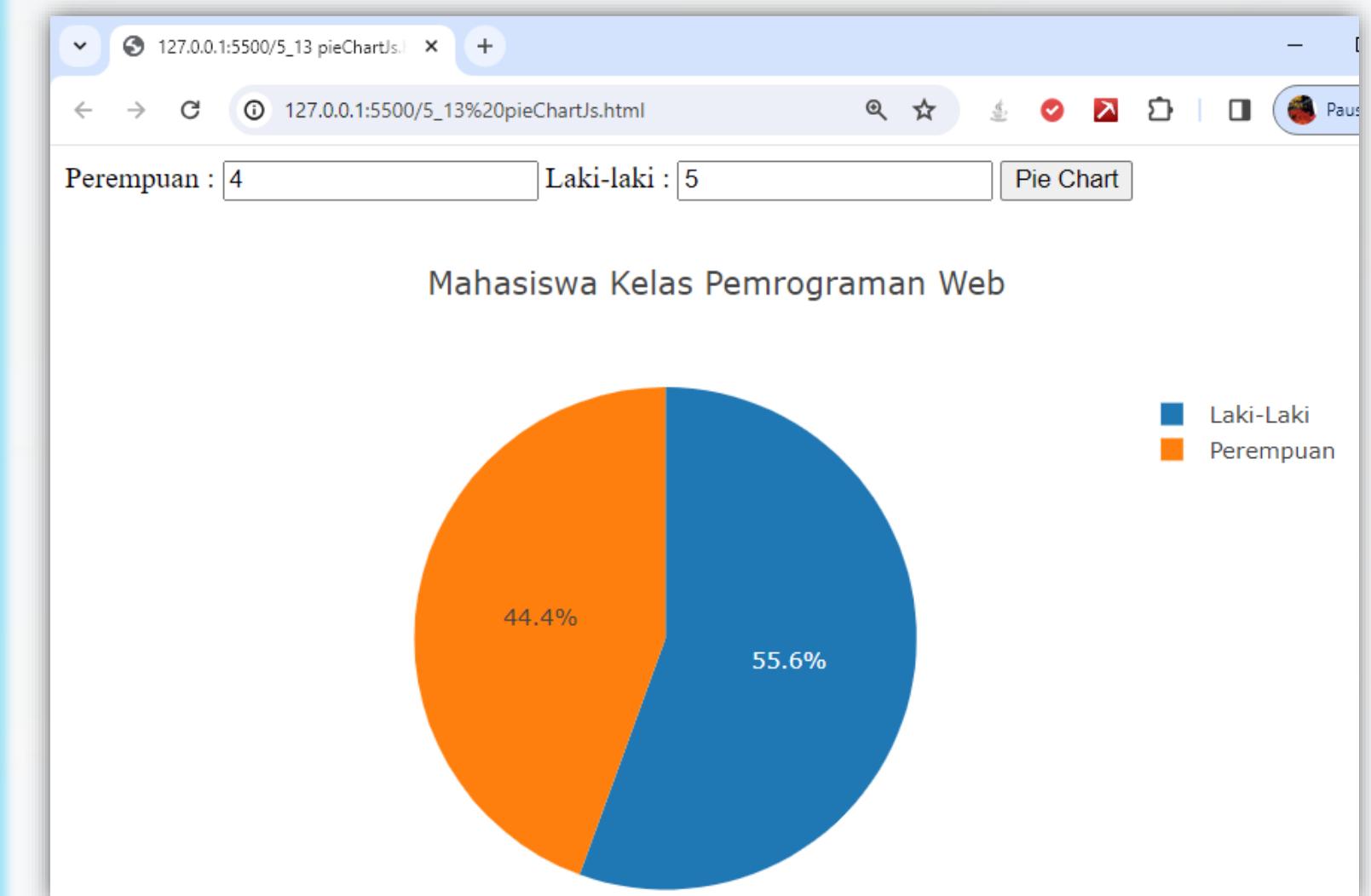
        const xArray = ["Perempuan", "Laki-Laki" ];
        const yArray = [p, l];

        const layout = {title:"Mahasiswa Kelas Pemrograman Web"};
        const data = [{labels:xArray, values:yArray, type:"pie"}];

        Plotly.newPlot("myPlot", data, layout);
    }
</script>

</body>
</html>

```

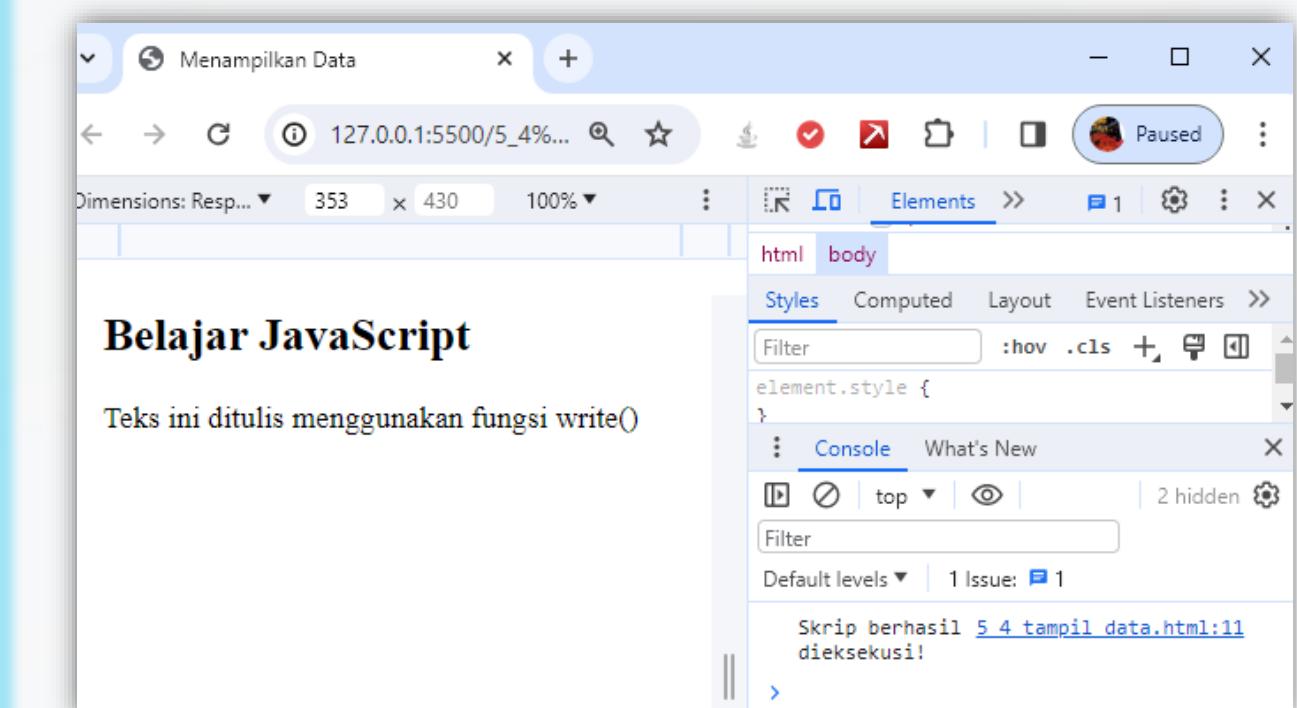
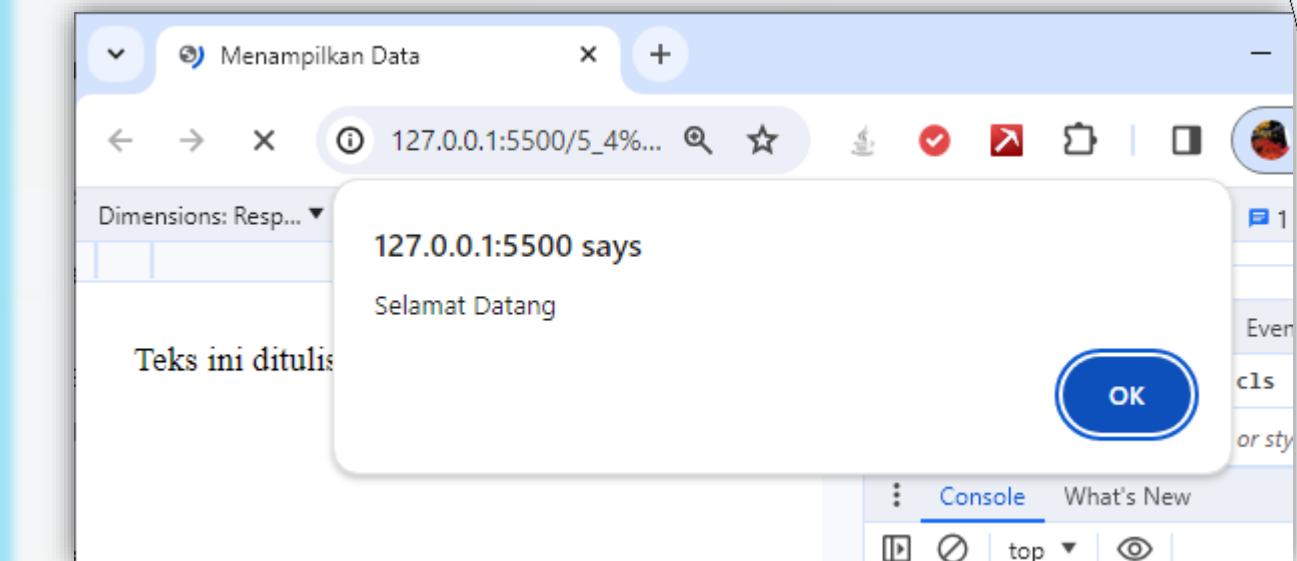


# MENAMPILKAN HASIL PROGRAM

- alert(), menampilkan hasil program melalui kotak pesan.
- write(), menampilkan hasil program pada halaman web.
- getElementById('id\_tag').innerHTML, menampilkan hasil program di dalam tag HTML.
- console.log(), menampilkan hasil program pada tag console (Inspect Browser).

# MENAMPILKAN HASIL PROGRAM

```
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Menampilkan Data</title>
    <script>
        window.onload = function(){
            alert('Selamat Datang');
            document.getElementById('judul').innerHTML=
                "Belajar JavaScript";
            console.log('Skrip berhasil dieksekusi!');
        }
    </script>
</head>
<body>
    <h2 id="judul"></h2>
    <script>
        document.write('Teks ini ditulis menggunakan fungsi
write()')
    </script>
</body>
</html>
```



# DASAR JAVASCRIPT

## VARIABEL

- Case-sensitive dan dideklarasikan dengan var atau let.
- Terdapat variabel global dan variabel lokal.

```
<script>
    var p = 200;
    let kata = "Ini JavaScript";
    const = 21;
</script>
```

# DASAR JAVASCRIPT

## OPERATOR

### Operator Aritmatika

+ , - , \* , / , % , ++ , --

### Operator Perbandingan

== , != , > , < , >= , <=

### Operator Logika

&& , || dan !

### Operator assignment

= , += , -= , \*= , /= , %=

### Operator kondisi

(kondisi)? Nilai true : nilai false

# DASAR JAVASCRIPT

## PERCABANGAN DENGAN IF

### Statement 1 baris

```
if(kondisi) pernyataan;
```

### Statement lebih dari 1 baris

```
if(kondisi) {  
    pernyataan1;  
    pernyataan2;  
}
```

### If ... else ...

```
if(kondisi){  
    Pernyataan jika benar;  
} else {  
    Statement jika salah;  
}
```

### If ... elseif ... else

```
if(kondisi 1) {  
    Statement jika kondisi 1 benar;  
}  
elseif (kondisi2){  
    Statement jika kondisi 2 benar;  
} else{  
    Statement jika semua kondisi salah;  
}
```

# DASAR JAVASCRIPT

## PERCABANGAN DENGAN SWITCH

Switch

```
switch(variabel){  
    case kondisi 1: pernyataan;  
    break;  
    case kondisi 2: pernyataan;  
    break;  
    default: pernyataan;  
}
```

# DASAR JAVASCRIPT

## PERULANGAN

Perulangan dengan while

```
while(kondisi){  
    pernyataan;  
}
```

Perulangan dengan do ... while

```
do{  
    pernyataan;  
} while (kondisi);
```

for

```
for(inisialisasi counter; kondisi; increment/decrement counter){  
    pernyataan;  
}
```

# DASAR JAVASCRIPT

## FUNGSI

Fungsi dengan return

```
function nama_fungsi(parameter1, parameter2){  
    pernyataan;  
    return variabel;  
}
```

Fungsi tanpa return

```
function nama_fungsi(parameter1, parameter2){  
    pernyataan;  
}
```

# DASAR JAVASCRIPT

## ARRAY

### Cara 1

```
const warna = new Array();
warna[0] = "Merah";
warna[1] = "Putih";
warna[2] = "Biru";
```

### Cara 2

```
const color = new Array("Red", "White", "Blue");
```

### Cara 3

```
const hariKerja = ["Senin", "Selasa", "Rabu", "Kamis"];
```

Property/Method	Keterangan
length	Jumlah elemen array.
join(separator)	Menggabungkan nilai array menjadi string yang dipisahkan dengan separator.
concat(array1, array2)	Menggabungkan beberapa array.
pop()	Menghapus elemen terakhir.
push(item1, item2,...)	Menambahkan elemen baru ke bagian akhir array.
reverse()	Membalik urutan array.
shift()	Menghapus elemen pertama.
unshift(item1, item2, ...)	Menambahkan elemen pada awal array.
sort()	Mengurutkan elemen array.
toString()	Mengkonversi array ke string.

## Suatu perlakuan yang dilakukan terhadap suatu elemen HTML.

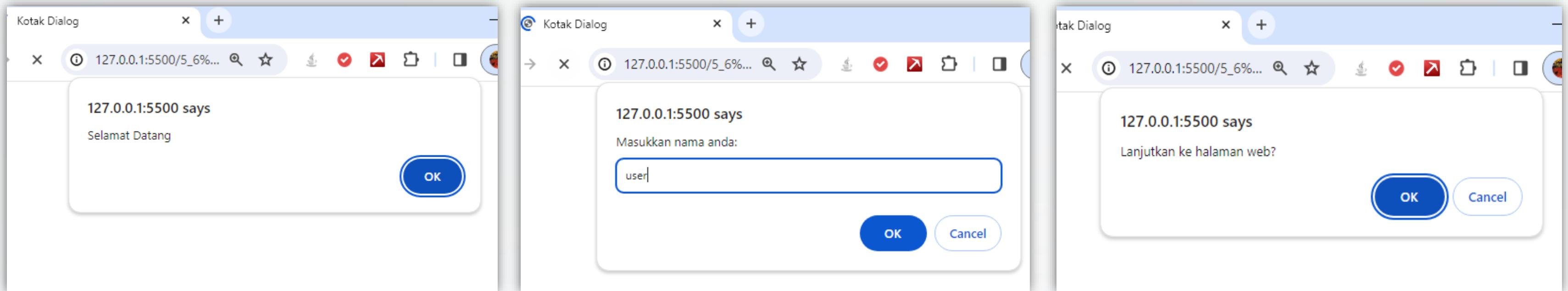
- onblur, ketika suatu elemen tidak fokus
- onchange, ketika nilai suatu elemen diubah
- onclick, ketika suatu elemen diklik.
- ondblclick, ketika suatu elemen di-*doubleclick*.
- onerror, ketika error.
- onfocus, ketika suatu elemen sedang fokus.
- onkeydown, ketika tombol huruf pada keyboard ditekan.
- onkeypress, ketika tombol keyboard ditekan.
- onkeyup, ketika tombol keyboard dilepas.
- onload, ketika dokumen di-load.
- onmousemove, ketika mouse digerakkan.
- onmouseout, ketika mouse keluar dari area suatu elemen.
- onmouseover, ketika mouse melewati area suatu elemen.
- onmouseup, ketika tombol mouse dilepas setelah diklik.
- onreset, ketika form direset.
- onresize, ketika elemen diubah ukurannya.
- onselect, ketika suatu elemen dipilih.
- onsubmit, ketika data form dikirim.
- onunload, ketika keluar dari suatu dokumen.

# KOTAK DIALOG

- alert() : menampilkan pesan tertentu kepada pengguna.
- confirm() : sama dengan alert namun terdapat dua tombol yaitu Oke dan Batal. Berfungsi untuk meminta konfirmasi pada pengguna.
- prompt() : digunakan untuk meminta masukan data dari pengguna.

# KOTAK DIALOG

```
<script>
    alert("Selamat Datang");
    confirm("Lanjutkan ke halaman web?");
    nama = prompt("Masukkan nama anda: ", "nama");
    document.write("Hi " + nama);
</script>
```



# WINDOW DAN DOKUMEN

Window adalah objek browser yang terbuka dan dokumen adalah objek dari dokumen HTML yang dimuat ke browser.

Properti/Method	Keterangan
reload()	Digunakan untuk mereload window window.location.reload()
close()	Digunakan untuk menutup window window.close()
open()	Digunakan untuk membuka window baru dengan aturan yang ditentukan.
print()	Digunakan untuk membuka jendela print window.print()
location	Digunakan untuk pindah ke halaman lain. window.location="url"

Objek document dapat digunakan untuk modifikasi elemen HTML

```
...
<h2>JavaScript HTML DOM</h2>

<br><button onclick='ubah()'>Ubah</button>

<script>
function ubah(){
    document.getElementById("image").src = "landscape.jpg";
}
</script>

<p>Gambar awal adalah smiley.gif, kemudian diganti dengan gambar
landscape.jpg Ketika tombol Ubah dipilih</p>
...
```

# MODIFIKASI CSS

Objek document dapat digunakan untuk modifikasi style CSS

```
...
<script>
    const color = new Array("Red", "White", "Blue");
    const hariKerja = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat"];

    let i = 0;
    let k = 0;
    function setWarnaBgTeks(){
        document.getElementById("div-hari").style.backgroundColor = color[i];
        i = (i < color.length-1) ? i+1 : 0;
        document.getElementById("div-hari").innerHTML = hariKerja[k];
        k = (k < hariKerja.length-1) ? k+1 : 0;
    }

    function run(){
        setInterval('setWarnaBgTeks()',500);
    }
</script>
```

# TANGGAL DAN WAKTU

## Inisialisasi objek Date()

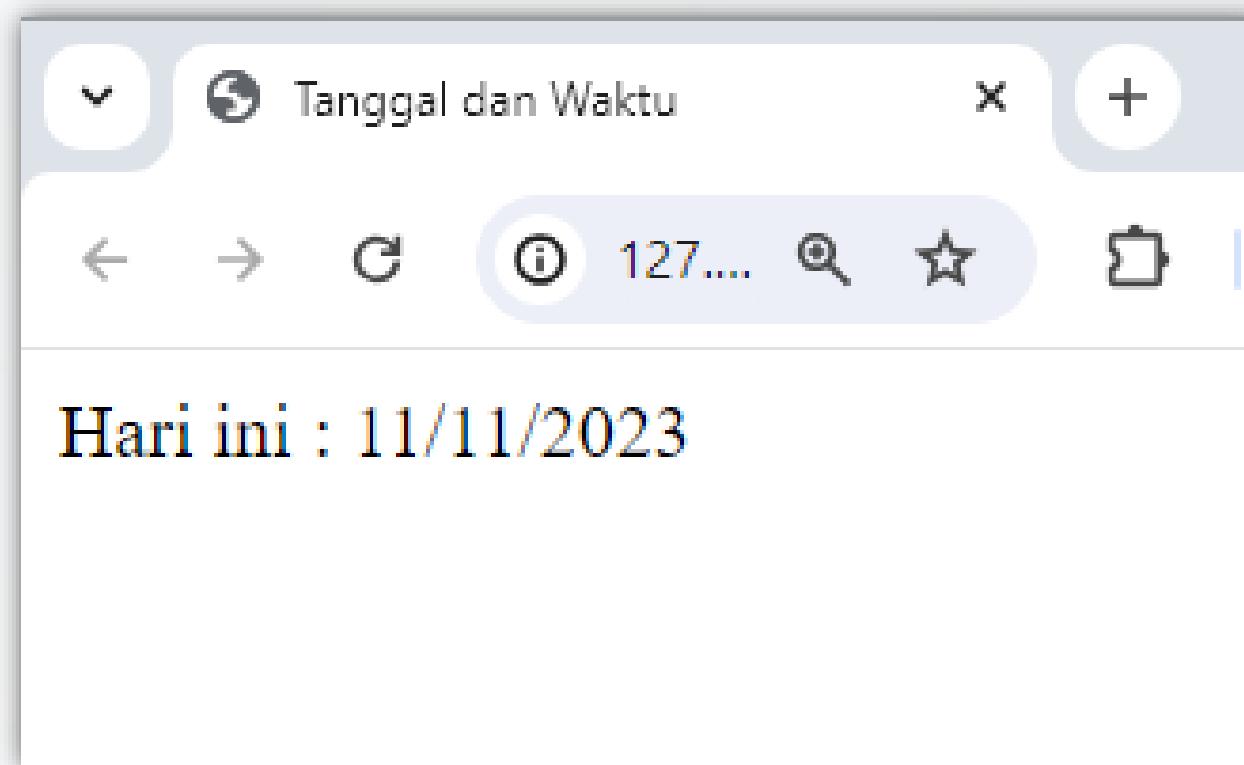
```
var tanggal = new Date();
```

Properti/Method	Keterangan
<code>getDate()</code>	Menghasilkan tanggal dalam satu bulan.
<code>getDay()</code>	Menghasilkan nama hari.
<code>getHours()</code>	Menghasilkan jam dimulai dari 0 hingga 23.
<code>getMinutes()</code>	Menghasilkan menit.
<code>getSeconds()</code>	Menghasilkan detik.
<code>getMonth()</code>	Menghasilkan bulan dimulai dari 0 hingga 11.
<code>getFullYear()</code>	Menghasilkan tahun.

[Property/method lainnya](#)

# TANGGAL DAN WAKTU

```
<script>
    var waktu = new Date();
    document.write('Hari ini : ' + waktu.getDate() + '/' + (waktu.getMonth()+1)
    +'/' + waktu.getFullYear());
</script>
```



# OBJEK JAVASCRIPT

Membuat dan mengakses objek

- Sama dengan membuat variabel tetapi nilainya berada dalam tanda { }.
- Masing-masing properti dipisahkan dengan koma (,) sedangkan antara properti dan nilainya dipisahkan titik dua (:).
- Mengakses objek properi dengan format objek['properti'] atau objek.properti.

Mengubah dan menghapus properti

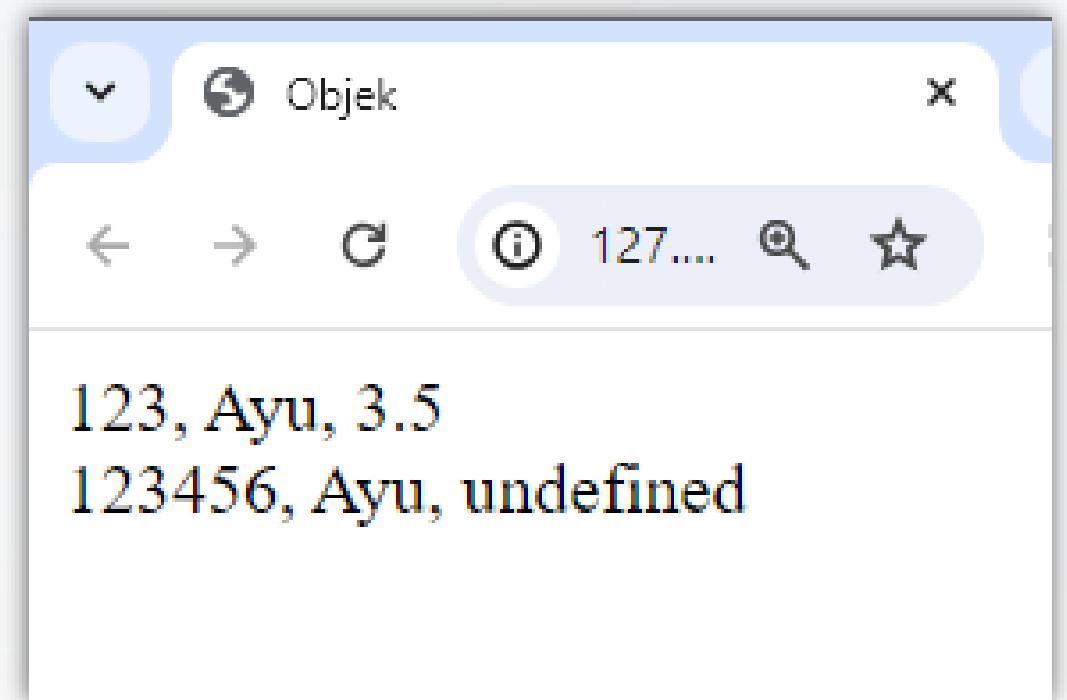
- Memberi dan mengubah nilai pada properti menggunakan (=).
- Menghapus properti menggunakan perintah delete.

# OBJEK JAVASCRIPT

```
<script>
    mahasiswa = {
        nim : '123',
        nama : 'Ayu',
        IPK : 3.5
    }

    document.write(mahasiswa.nim + ' ' + mahasiswa['nama']
    + ', ' + mahasiswa.IPK+<br>')

    mahasiswa.nim='123456';
    delete mahasiswa.IPK;
    document.write(mahasiswa.nim + ' ' + mahasiswa['nama']
    + ', ' + mahasiswa.IPK+<br>)
</script>
```



# EVENT TIMING

- setTimeout() digunakan untuk menjalankan suatu program setelah batas waktu yang ditentukan dan hanya dijalankan 1 kali.
- clearTimeout() program yang seharusnya dijalankan pada batas waktu tertentu dengan setTimeout() tidak jadi dijalankan.

```
variableId = setTimeout(expression, timeout);
```

Expression : fungsi yang akan menjalankan program.

Timeout : batas dijalankannya program dalam milisecond (1000 = 1 detik).

# EVENT TIMING

- `setInterval()` digunakan untuk menjalankan program berkali-kali dalam durasi waktu tertentu. Dapat digunakan untuk membuat animasi pada JavaScript.
- `clearInterval()` disimpan dalam variabel, digunakan untuk menghentikan program.

```
variableId = setInterval(expression, timeout);
```

Expression : fungsi yang akan menjalankan program.

Timeout : batas dijalankannya program dalam milisecond (1000 = 1 detik).

# EVENT TIMING

```
<head>
<script>
    var angka = 0;
    var timer;
    var timeout;

    function tampilkanAngka(){
        document.getElementById('timer').innerHTML = angka;
        angka++;
    }

    function setTimer(){
        timer = setInterval('tampilkanAngka()', 100);
    }

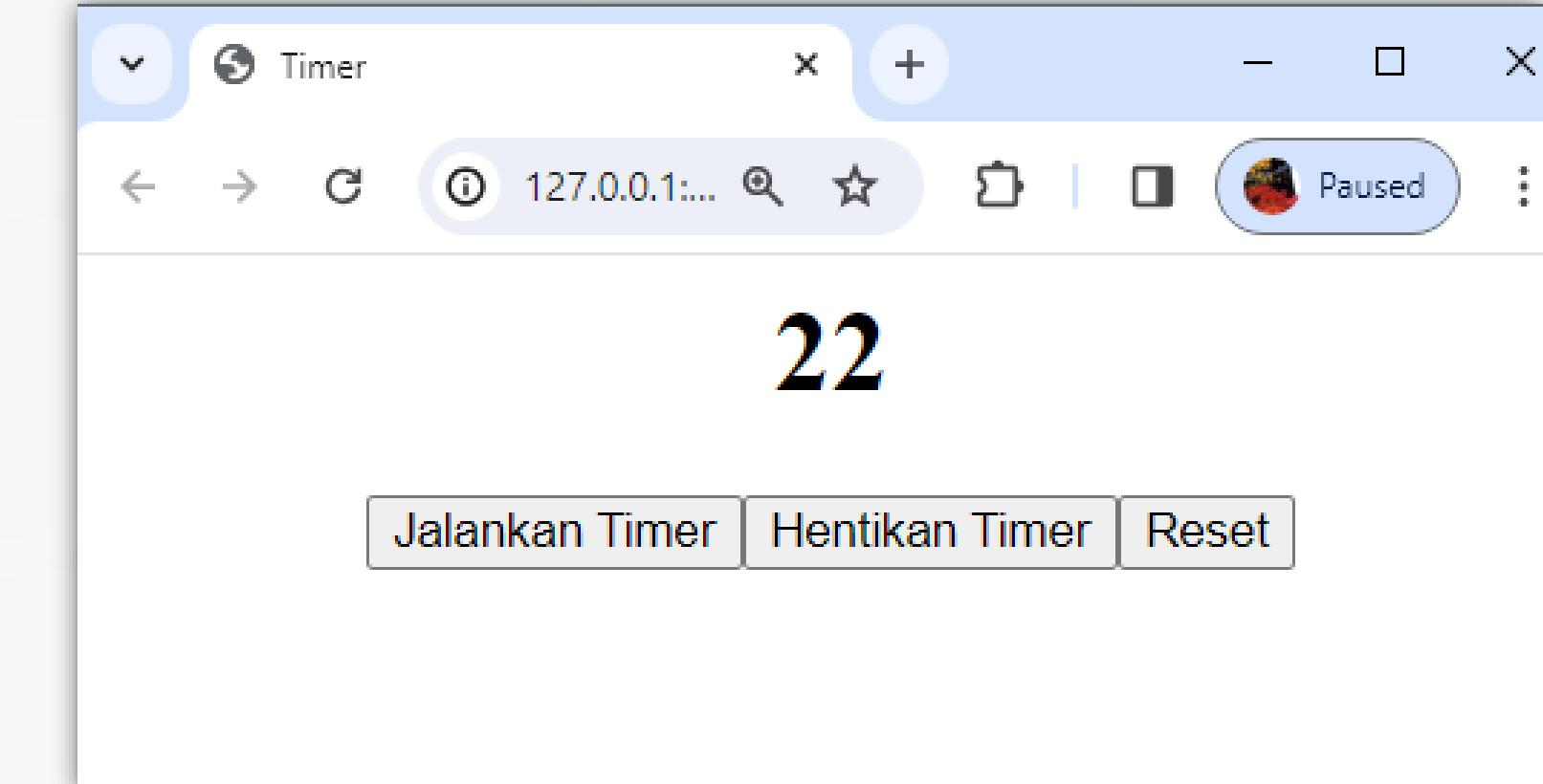
    function jalankanTimer(){
        timeout = setTimeout('setTimer()', 3000)
    }

    function hentikanTimer(){
        clearInterval(timer);
    }

    function hentikanTimeout(){
        clearTimeout(timeout);
        angka = 0;
    }

</script>
</head>
```

```
<body>
    <h1 id="timer" style="text-align: center;">0</h1>
    <div style="display:flex ; justify-content:center;">
        <button onclick="jalankanTimer()">Jalankan
        Timer</button>
        <button onclick="hentikanTimer()">Hentikan
        Timer</button>
        <button onclick="hentikanTimeout()">Reset</button>
    </div>
</body>
```



**TERIMA  
KASIH**

