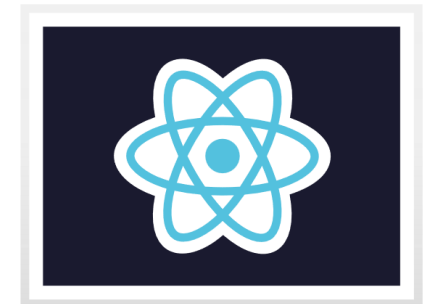
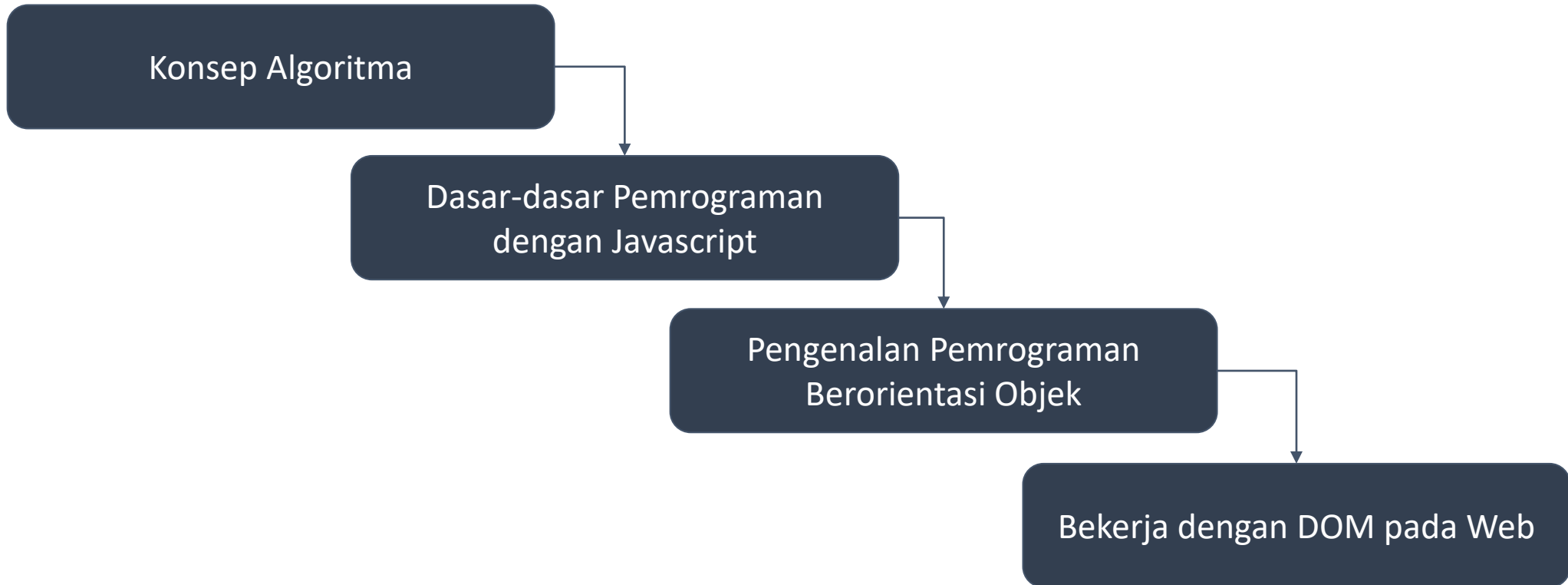


PEMROGRAMAN JAVASCRIPT



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



"Di antara pendosa, yang paling buruk adalah dia yang meluangkan waktunya untuk membahas kesalahan orang lain."

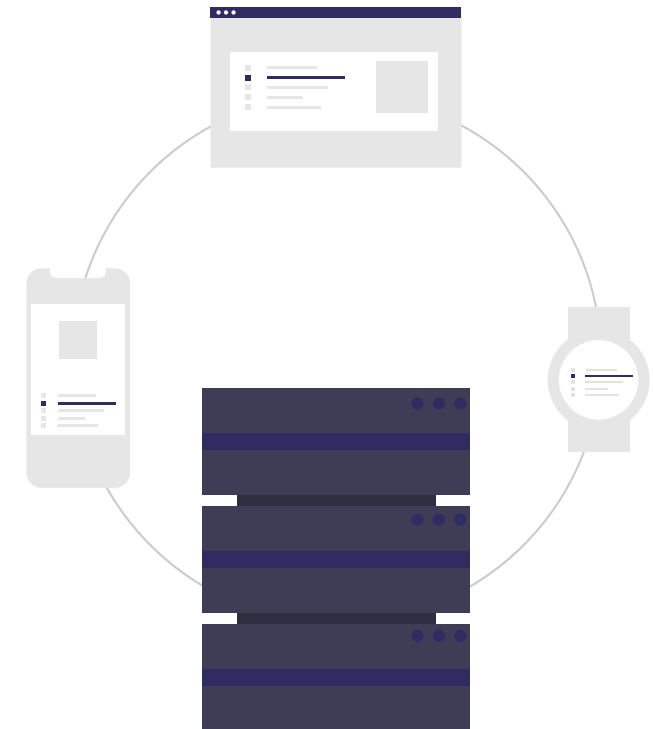
- Utsman bin Affan -

STANDARD OUTPUT JS

OUTPUT JAVASCRIPT

Javascript bisa menampilkan data dalam berbagai cara

- Menuliskan ke sebuah elemen HTML menggunakan `innerHTML`
- Menuliskan ke output HTML dengan `document.write()`
- Menggunakan alert box dengan `window.alert()`
- Menggunakan browser console dengan `console.log()`



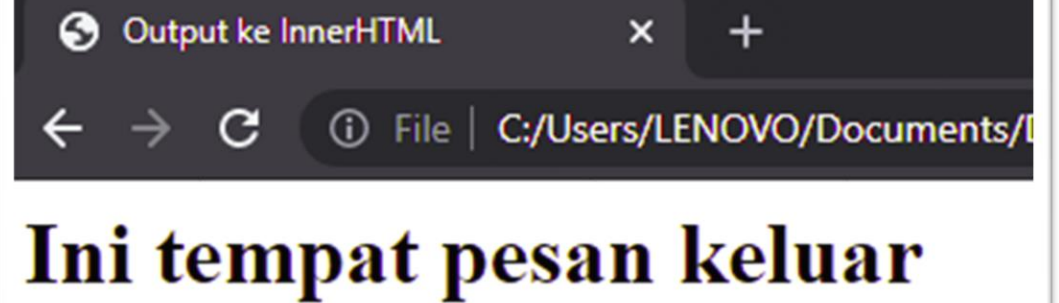
Inner HTML

- Setiap elemen HTML bisa diakses menggunakan
- `document.getElementById(<id elemen html di sini>)`
- Kita bisa tampilkan output dari kode javascript yang kita ketikkan di console ke dalam salah satu elemen html. Misalnya kita punya satu header (tag h1) dengan id pesan

OUTPUT JAVASCRIPT

```
<html>
  <head>
    <title>Output ke InnerHTML</title>
  </head>

  <body>
    <h1 id=pesan>Ini tempat pesan keluar</h1>
  </body>
</html>
```



Output ke InnerHTML

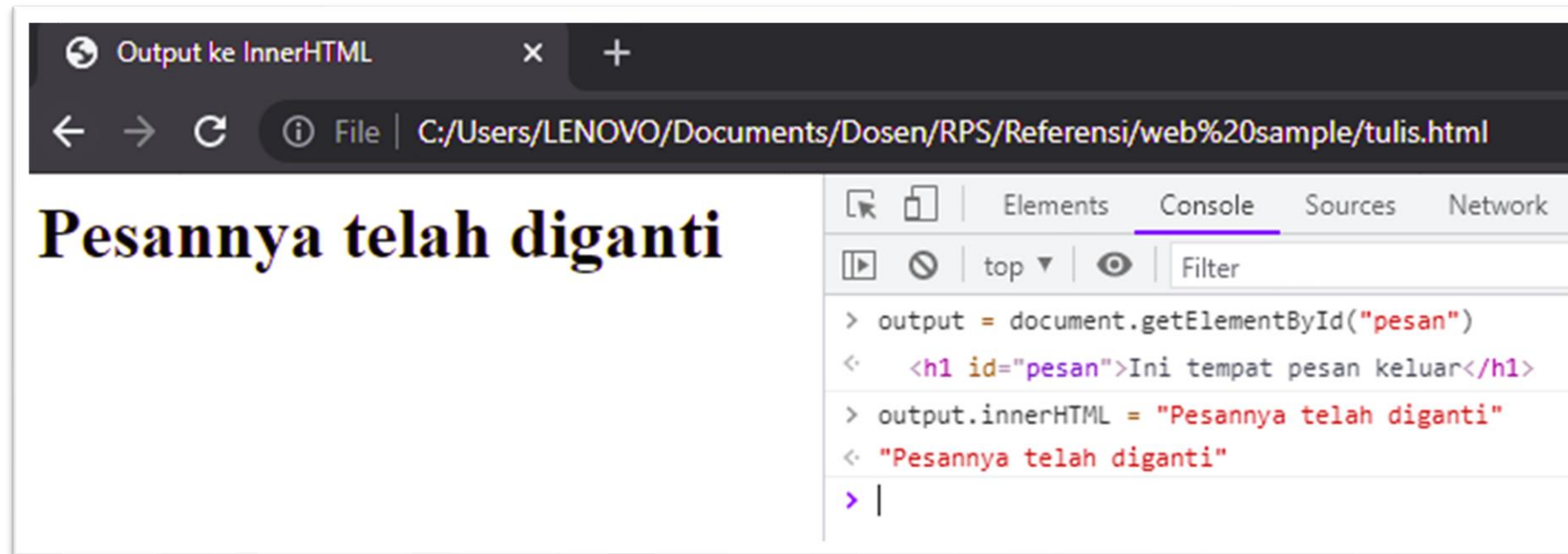
← → ↻ ⓘ File | C:/Users/LENOVO/Documents/

Ini tempat pesan keluar

Inner HTML

- Di console, kita hubungkan header dengan id “pesan” tadi ke sebuah variabel dan mengganti isi innerHTMLnya.
- `let output = document.getElementById("pesan")`
- `output.innerHTML = "Pesannya telah diganti"`

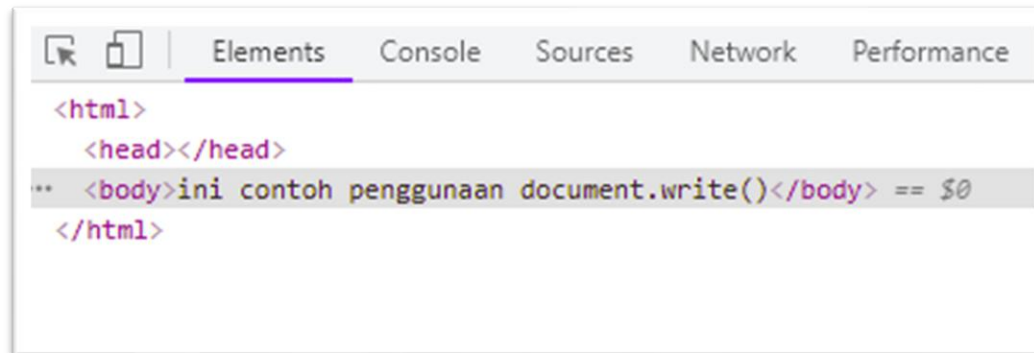
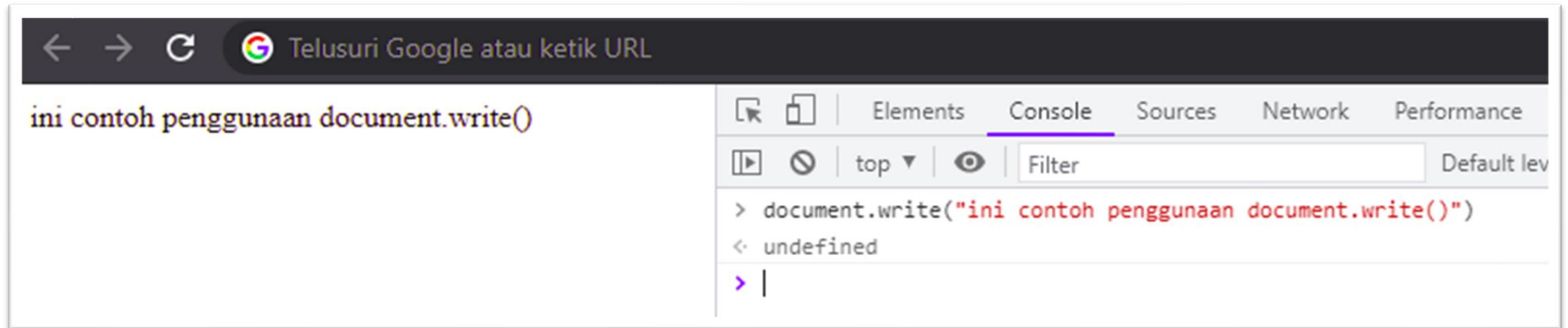
ISTILAH PEMROGRAMAN



Document Write

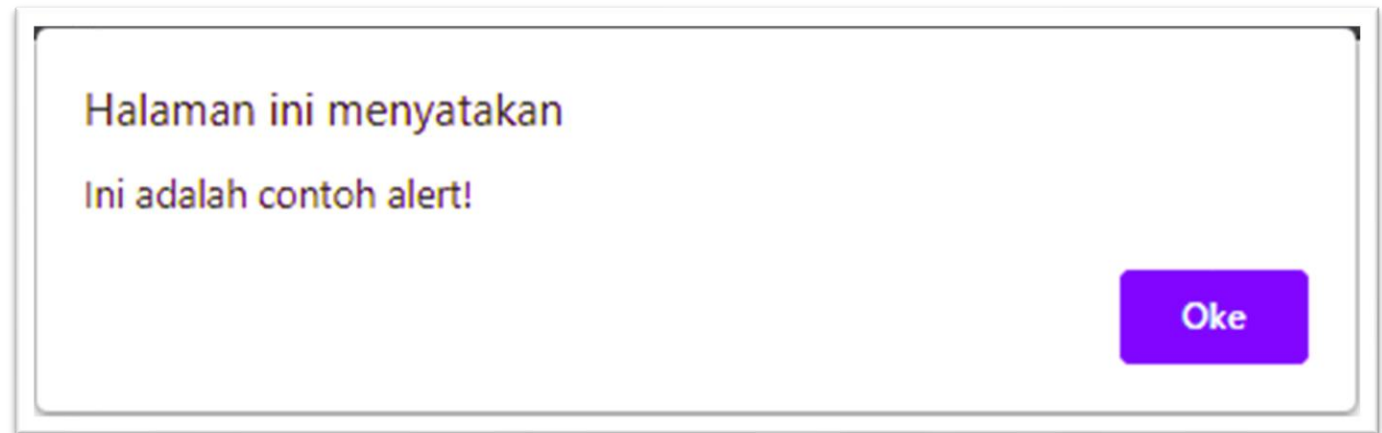
- Selain ke elemen html yang sudah ada, kita juga bisa tuliskan output ke dokumen html baru menggunakan
- `document.write(<pesan>)`
- Browser akan menuliskan pesan tersebut di badan htmlnya.

OUTPUT JAVASCRIPT



Alert Box

- Fungsi untuk menampilkan alert box telah dipelajari sebelumnya. Browser akan memunculkan sebuah modal window yang berisikan pesan.
- `alert(<pesan>);`



Console Log

- Untuk debugging, programmer juga bisa menggunakan metode
- `console.log(<pesan>)`
- untuk menampilkan data di browser, lebih tepatnya menampilkan di console pada browser. Hal ini tidak akan mengganggu kode HTML dan tampilan webnya.

Console Log

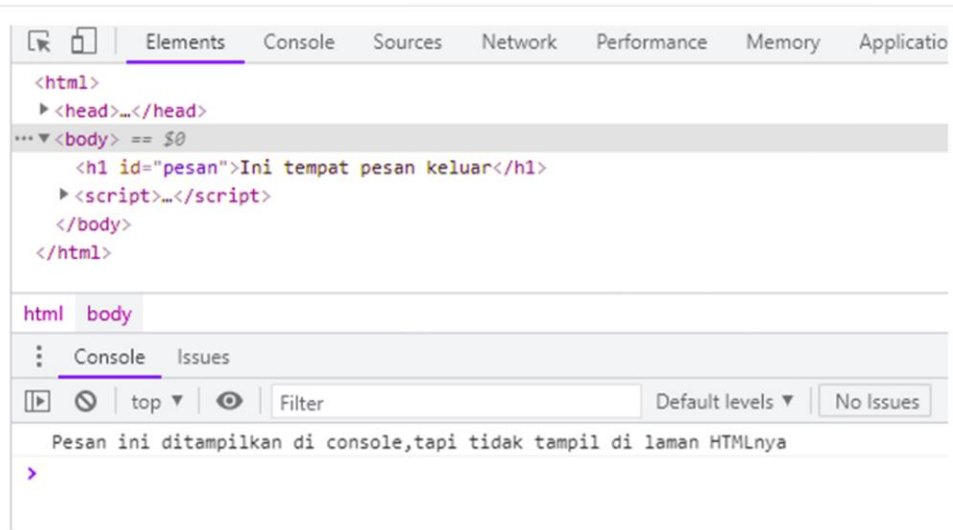
- Untuk debugging, programmer juga bisa menggunakan metode
- `console.log(<pesan>)`
- untuk menampilkan data di browser, lebih tepatnya menampilkan di console pada browser. Hal ini tidak akan mengganggu kode HTML dan tampilan webnya.

VARIABEL JAVASCRIPT

```
<html>
  <head>
    <title>Output ke InnerHTML</title>
  </head>

  <body>
    <h1 id=pesan>Ini tempat pesan keluar</h1>
    <script> console.log("Pesan ini ditampilkan di console,tapi tidak tampil di laman HTMLnya")
    </script>
  </body>
</html>
```

Ini tempat pesan keluar



Debugging dengan console

- Bug = kesalahan pada kode baik berupa syntax error maupun logical error
- Debugging = proses menguji, menemukan, dan mengurangi error pada kode program di komputer
- Debugger = tools yang digunakan untuk melakukan debugging
- Beberapa tipe kesalahan ditunjukkan oleh browser pada saat dijalankan melalui console. Tetapi seringkali pada saat kode mengandung error, tidak terjadi sesuatu apapun. Tidak ada pesan error dan indikasi apapun untuk mencari di bagian mana errornya, namun hasil akhirnya tidak sesuai dengan yang diharapkan.

Debugging dengan console

- Programmer dapat memanfaatkan console untuk debugging program yang dibuat/dijalankan dengan cara menampilkan nilainya pada console dan membandingkannya dengan nilai yang diharapkan.
- `<script>`
- `a = 5;`
- `b = 6;`
- `c = a + b;`
- `console.log(c);`
- `</script>`

STRING JAVASCRIPT

- **String** adalah tipe data yang berupa teks.
- Penulisan dalam javascript dapat menggunakan petik (single quote), petik dua (double quote), maupun backtick.
- ```
let carName1 = "Toyota Avanza"; // Double quotes
```

```
let carName2 = 'Toyota Avanza'; // Single quotes
```

```
let carName3 = `Toyota Avanza`; // Backticks
```

# BACKTICKS JAVASCRIPT

---

- **Backtick** memberikan fungsi tambahan kepada string javascript untuk dapat memasukkan variabel dan ekspresi ke dalam stringnya. Hal itu bisa dilakukan dengan menuliskannya dalam apitan simbol `${...}`.
- **Backtick** juga dapat digunakan untuk menuliskan string dalam beberapa baris.
- `let nama = "John Doe";`  
`alert(`Halo, ${nama}`);` // akan menampilkan Halo, John Doe  
`alert(`Hasil penjumlahannya adalah ${5+10}`);` // akan menampilkan 15

## STRING LENGTH()

- `let abjad = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";`
- `abjad.length;` // akan mengembalikan nilai 26 (panjang string tersebut)
- Fungsi `length` akan menghitung berapa banyak karakter yang terdapat dalam string tersebut (termasuk spasi dan simbol khusus lainnya).

## ESCAPE CHARACTER

- Simbol backslash ( \ ) digunakan sebagai escape character, yaitu karakter khusus untuk mengizinkan simbol tertentu dituliskan dalam string.
- \' = tanda petik
- \" = tanda petik dua
- \\ = tanda backslash
- \b = backspace

## ESCAPE CHARACTER

- `\f` = formfeed
- `\r` = carriage return
- `\n` = baris baru (seperti menekan tombol enter)
- `\t` = tabulasi horizontal (seperti menekan tombol tab)
- `\v` = tabulasi vertikal

## Indexing String

- let str = “Indonesia tanah airku, tanah tumpah darahku”
- Mencari teks dalam teks dapat dilakukan dengan memanggil fungsi yang ada dibawah ini :

**str.indexOf()**

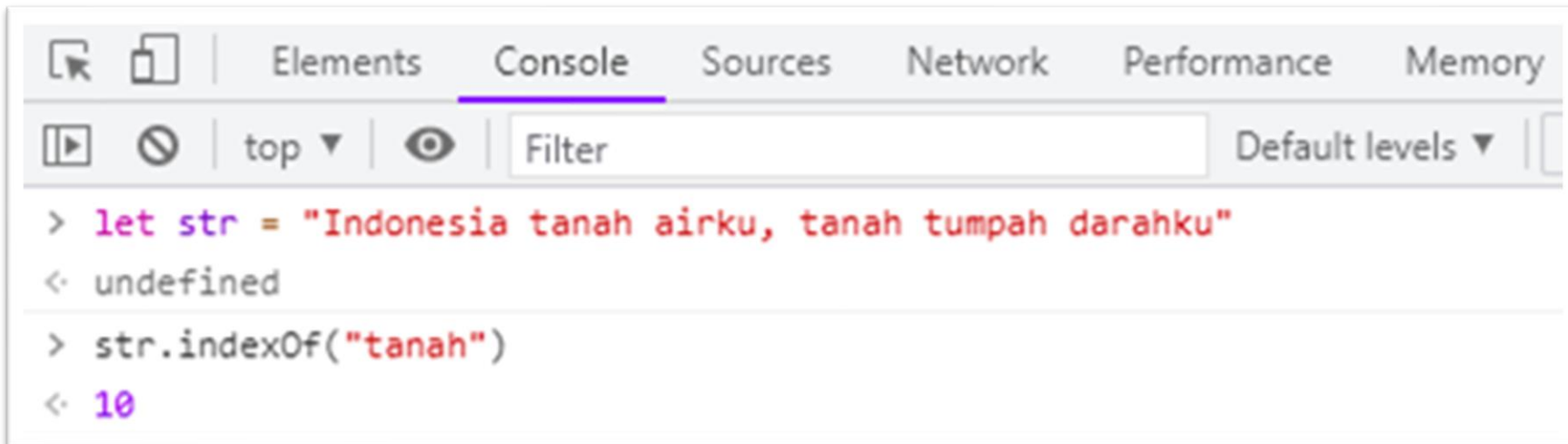
**str.lastIndexOf()**

**str.search()**



## indexOf()

indexOf akan mengembalikan index dari kemunculan pertama kata yang dicari, sedangkan lastIndexOf akan mengembalikan index dari kemunculan terakhirnya.



```
> let str = "Indonesia tanah airku, tanah tumpah darahku"
< undefined

> str.indexOf("tanah")
< 10
```

# STRING METHOD

## search() & lastIndexOf()

- Fungsi search menerima argumen dan mengembalikan nilai yang sama persis dengan indexOf, tetapi keduanya berbeda.
- Fungsi indexOf dan lastIndexOf dapat menerima argumen kedua berupa titik mula pencarian, sedangkan search tidak.

```
> str.lastIndexOf("tanah")
< 23

> str.search("tanah")
< 10

> |
```

# STRING METHOD

## search() & lastIndexOf()

- Pada contoh berikut, kita mencari kata “tanah” dari depan, tapi mulai dari index ke-15, bukan mulai dari awal kalimat.
- lastIndexOf akan mencari dari belakang (argumen kedua menyatakan pencarian katanya dilakukan sebelum index tersebut)

```
> str.indexOf("tanah",15)
< 23
```

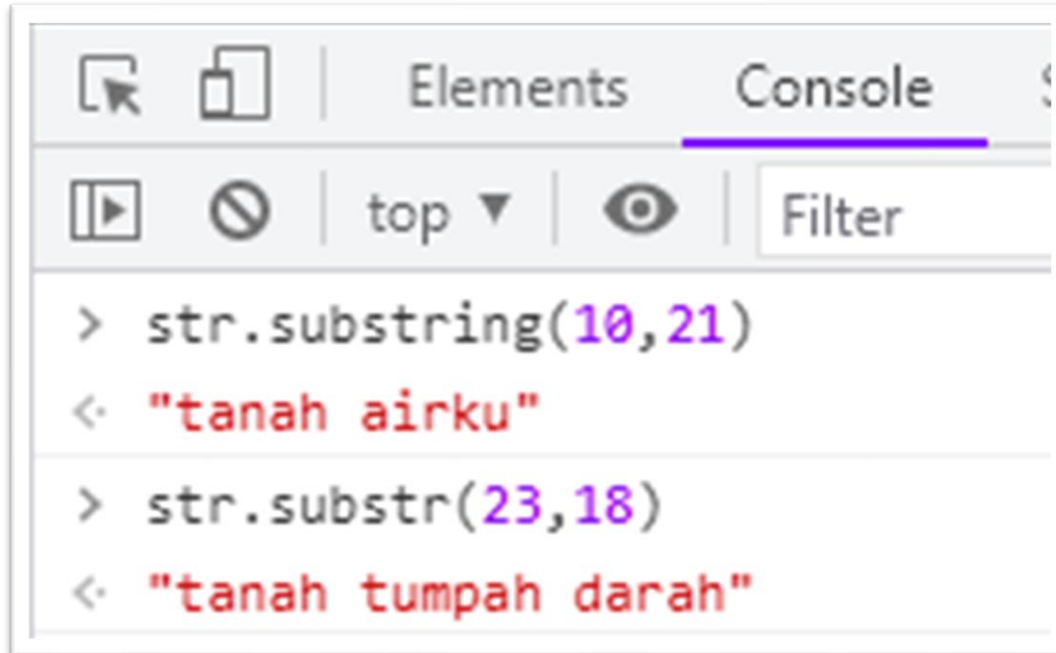
## Substring JavaScript

- Fungsi lain yang dapat digunakan adalah  
**str.includes(<substring>)**  
**str.startsWith(<substring>)**  
**str.endsWith(<substring>)**
- Fungsi tersebut akan mengembalikan nilai true jika substringnya ada di dalam, awal, atau akhir dari stringnya. Tapi tidak bisa tahu ada di index keberapanya.

## Mengambil sebagian String dalam Javascript

- `let str = "Indonesia tanah airku, tanah tumpah darahku"`  
`str.slice(<start>, <end>)`  
`str.substring(<start>, <end>)`  
`str.substr(<start>, <length>)`
- Fungsi slice dan substring mirip, tapi substring tidak dapat menerima index negatif sebagai inputannya.
- Jika inputannya negatif, akan dihitung dari belakang

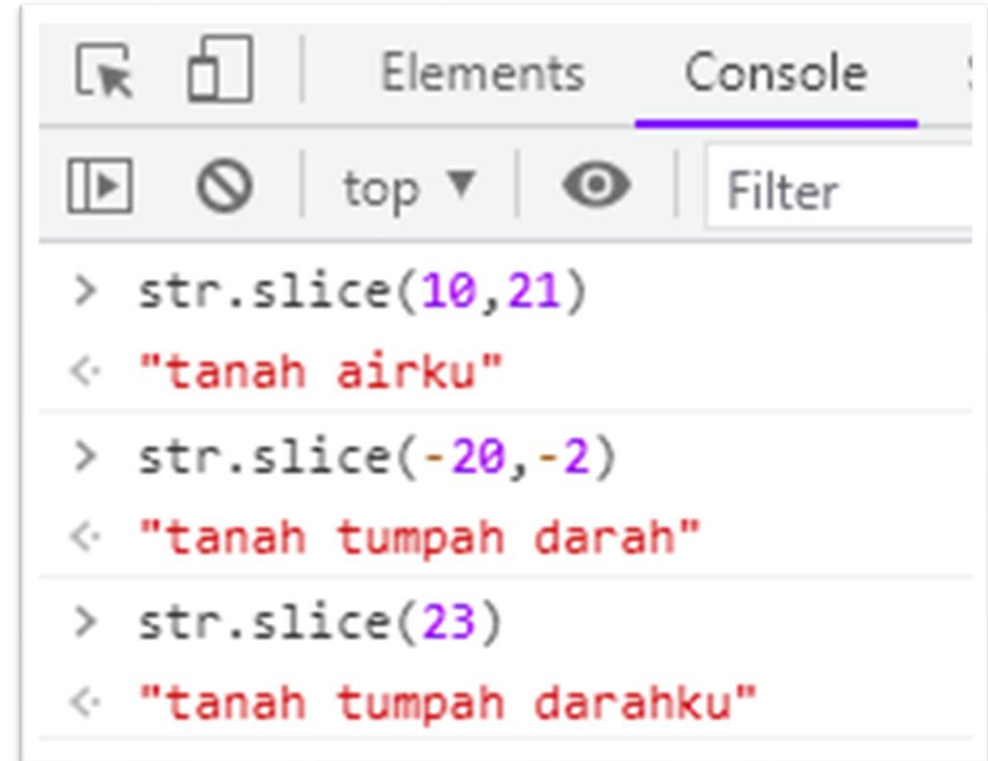
# TIPE DATA JAVASCRIPT



A screenshot of a web browser's developer console. The 'Console' tab is selected. The console shows two JavaScript commands and their outputs. The first command is `str.substring(10,21)` which returns the string `"tanah airku"`. The second command is `str.substr(23,18)` which returns the string `"tanah tumpah darah"`.

```
> str.substring(10,21)
< "tanah airku"

> str.substr(23,18)
< "tanah tumpah darah"
```



A screenshot of a web browser's developer console. The 'Console' tab is selected. The console shows three JavaScript commands using the `slice` method and their outputs. The first command is `str.slice(10,21)` which returns `"tanah airku"`. The second command is `str.slice(-20,-2)` which returns `"tanah tumpah darah"`. The third command is `str.slice(23)` which returns `"tanah tumpah darahku"`.

```
> str.slice(10,21)
< "tanah airku"

> str.slice(-20,-2)
< "tanah tumpah darah"

> str.slice(23)
< "tanah tumpah darahku"
```

## Mengubah Substring

- `let str = "Indonesia tanah airku, tanah tumpah darahku"`  
`str.replace(<kata asal>, <kata pengganti>)`
- Fungsi `replace` hanya akan mengganti kemunculan pertama saja dan inputannya case-sensitive.

# STRING METHOD

```
> let str = "Indonesia tanah airku, tanah tumpah darahku"
< undefined

> str.replace("ku", " kita")
< "Indonesia tanah air kita, tanah tumpah darahku"

> str.replace(/KU/i, " kita")
< "Indonesia tanah air kita, tanah tumpah darahku"

> str.replace(/ku/g, " kita")
< "Indonesia tanah air kita, tanah tumpah darah kita"

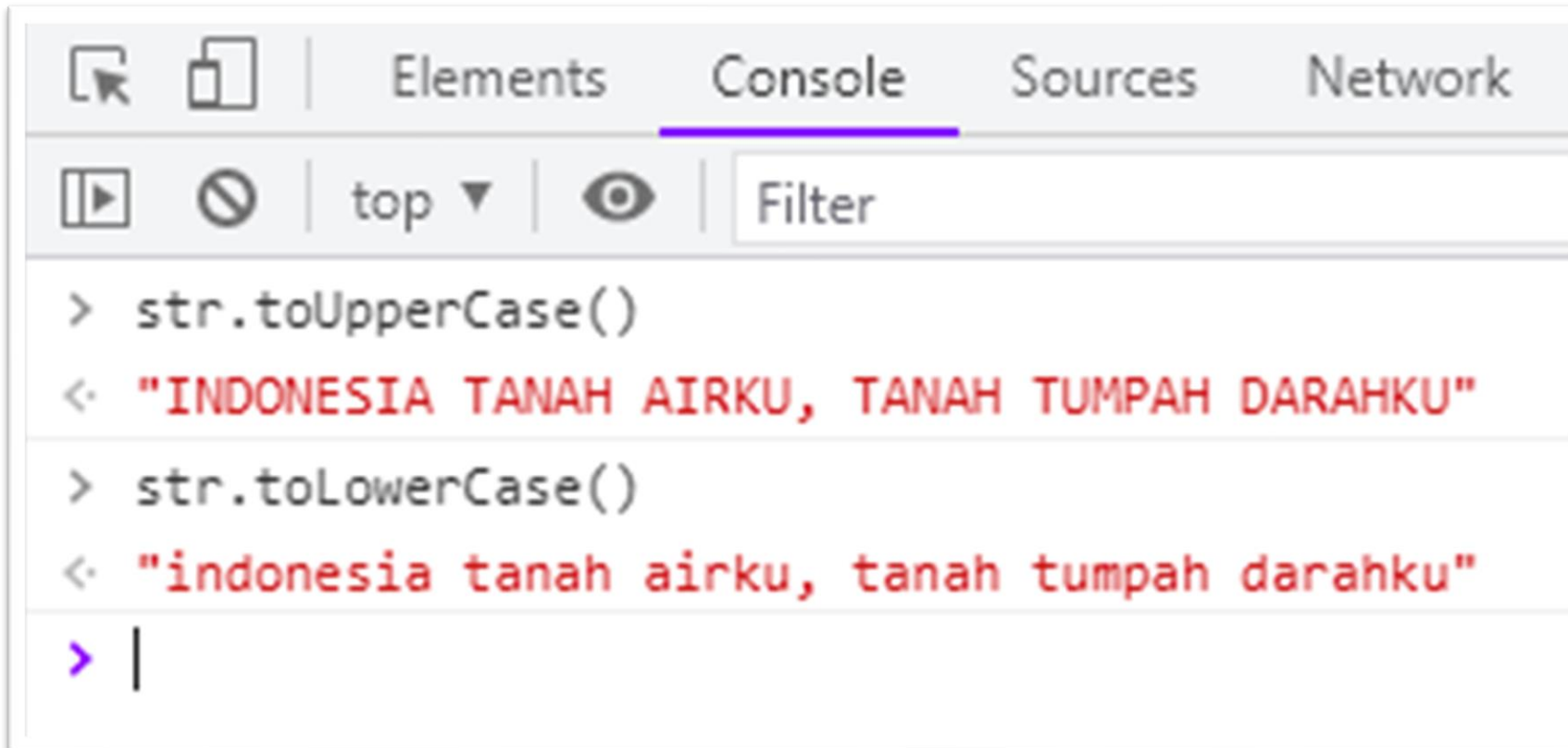
> |
```



## Mengubah Case String

- Uppercase (huruf kapital) dan lowercase (huruf kecil) dianggap berbeda pada javascript (case-sensitive).
- Untuk mengubah ke satu sama lain, dapat digunakan
- `str.toUpperCase()`  
`str.toLowerCase()`

# STRING METHOD



```
> str.toUpperCase()
< "INDONESIA TANAH AIRKU, TANAH TUMPAH DARAHKU"

> str.toLowerCase()
< "indonesia tanah airku, tanah tumpah darahku"

> |
```

## Menggabungkan String

- Fungsi `concat()` akan menambahkan sebuah string atau lebih ke belakang string yang menjalankan fungsinya.  
  
`teks.concat(" ", "World!");` // akan menghasilkan Hello World
- Selain itu dapat juga digunakan operator `(+)`  
  
`teks + " " + "World!"` // akan menghasilkan string yang sama, Hello World

## Menghilangkan Whitespace pada String

- Trimming adalah proses menghilangkan whitespace di ujung kiri dan ujung kanan string.
- `let teks = ' Spasinya banyak ';`
- `teks.trim()` // akan menghilangkan semua spasi di kiri dan kanan

## Padding String

- Kebalikan dari trimming, padding adalah proses menambahkan sesuatu di kiri atau kanan string.

- `let teks = '3'`

`teks.padStart(4,0);` // menambahkan 0 di kiri string hingga 4 digit -> 0003

`teks.padEnd(2,'x');` // menambahkan x di kanan string -> 3x

# STRING METHOD

## Mengetahui tipe data variabel

- Ada 5 tipe data (string, number, boolean, object, function) yang dapat menyimpan nilai, 6 tipe objek (Object, Date, Array, String, Number, Boolean), dan 2 tipe data yang tidak dapat menyimpan nilai (null dan undefined).
- Untuk melihat tipe data apa yang kita miliki, dapat digunakan fungsi **typeof()**

```
Elements Console Sources Network
top Filter
> a = "contoh string"
< "contoh string"
> typeof a
< "string"
> |
```

```
> b = 5
< 5
> typeof b
< "number"
> |
```

## Mengubah tipe data

- Sebuah variabel javascript dapat diubah ke tipe data lainnya
  1. Dilakukan oleh user dengan menggunakan fungsi
  2. Secara otomatis dilakukan oleh javascript
- Contohnya, fungsi berikut akan mengubah nilai yang menjadi inputannya ke tipe data yang ditunjukkan (jika bisa diubah).
  1. `Number()` //diubah ke tipe number, bilangan
  2. `String()` //diubah ke tipe string, teks

# STRING METHOD

```
> a
< "5"

> typeof a
< "string"
```

```
> b = Number(a)
< 5

> typeof b
< "number"
```

```
> c = String(b)
< "5"

> typeof c
< "string"
```



## Mengubah String ke Number

- Ada beberapa cara lain untuk mengubah string ke number :
  1. `Number()`
  2. `parseFloat()`
  3. `parseInt()`

- `let a = "10"`
- `let b = 67`
- Ubahlah variable a kedalam bentuk tipe data double
- Ubahlah variable b kedalam bentuk tipe data string

## Mengubah Number ke String

- Ada beberapa cara lain untuk mengubah number ke string :
  1. `String()`
  2. `x.toString()`

**Let teks = “Merah Putih”**

1. Ubahlah variable teks menjadi uppercase
2. Ubahlah variable teks menjadi lowercase
3. Ubahlah “Putih” menjadi “Muda” dalam variable teks
4. Ukurlah Panjang (length) dari variable teks
5. Simpan dalam file index.js
6. Upload pada akun github kalian dengan nama repository  
“Pertemuan-4-JS”



**Terima Kasih  
Syukron Katsiron  
Arigatou Gozaimasu**