

# **18 Rahasia** **Menguasai** **Javascript**

**Rasakan sensasi belajar  
javascript yang tidak membuat  
kamu pusing apalagi boring.**

---

**BY RICKY NOVIANSYAH**

Frontend Developer

# SILABUS JAVASCRIPT

Ebook ini akan membantu kamu belajar javascript dari dasar walau belum ada basic sampe kamu paham apa itu javascript.

## PENGENALAN

- Memahami Sejarah Javascript
- Memahami Jenis Framework Javascript
- Memahami Tipe Data
- Memahami HTML DOM

## PRAKTEK

- Membuat HTML DOM
- Membuat Variabel
- Membuat Operator
- Membuat Percabangan
- Membuat Function
- Membuat Arrow Function
- Membuat Event

- Membuat Perulangan
- Membuat Array
- Membuat Array Method
- Membuat Array Sorting
- Membuat Regex
- Membuat Javascript This Keyword
- Membuat Object
- Membuat Object Accessor

# Memahami Sejarah Javascript

## Bagaimana Awal Javascript Diciptakan ?

**JavaScript** ini dibuat dan didesain oleh **Brandan Eich**, seorang karyawan **Netscape**, pada tahun **1995**. Bahasa pemrograman ini awal mulanya disebut **Mocha**, tetapi kemudian diganti menjadi **Mona**, lalu menjadi **LiveScript** sebelum pada akhirnya resmi menyandang nama sekarang yaitu **JavaScript**.

Fungsi dari JavaScript ini sendiri yaitu diantaranya adalah untuk pengembangan dalam sebuah aplikasi web, mobile, membangun web server dan juga aplikasi server, membuat website yang bisa interaktif, bagi pengunjung

Popularitas penggunaan JavaScript yang dimana kian meroket tak akan bisa dicapai tanpa adanya kerja keras serta kerja cerdas dari JavaScript Developer yang handal, dalam pengembangan sehingga tidak perlu diragun

# Memahami Jenis-Jenis Framework Javascript

## Apa Fungsi Framework Javascript ?

Framework ini merupakan sebuah alat pendukung yang dimana menyediakan berbagai jenis struktur, konsep, dan juga panduan bagi seorang developer ketika dalam mengembangkan sebuah aplikasi.

Ada banyak tentunya jenis framework yang bisa dipakai dalam pembuatan aplikasi, tetapi jenis framework ini dibedakan berdasarkan bahasa pemrograman yang digunakan salah satunya yaitu framework JavaScript.

Beberapa framework javascript yang harus kamu ketahui, VueJS, AngluarJS, ReactJS, NextJS, NuxtJS, GatsbyJS, framework tadi itu sering di pakai dalam pengembangan aplikasi

# Memahami Tipe Data

## Apa Itu Tipe Data ?

Tipe data dapat diartikan sebagai bagian dari alat yang biasa digunakan dalam pemrograman untuk menentukan sebuah tipe nilai pada suatu objek maupun operasi, yang disesuaikan oleh penggunaan.

Beberapa tipe data yang harus kamu ketahui :

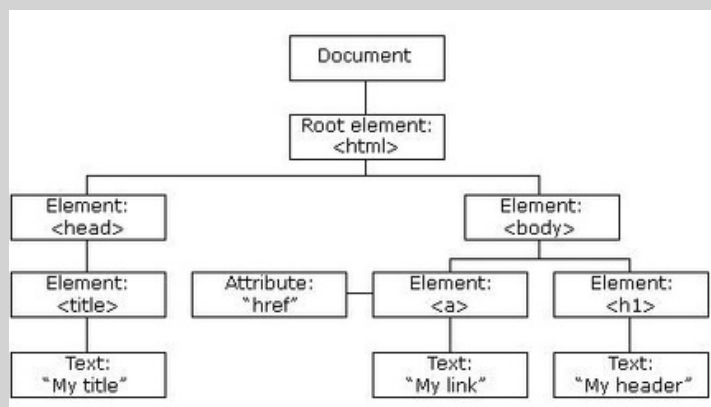
- Null adalah tipe data, null juga sebuah nilai, tipe data null hanya bernilai null meskipun nilai null bisa dikatakan tidak memiliki nilai
- Undefined adalah tipe data spesial yang hanya memiliki nilai undefined.
- boolean biasa digunakan untuk merepresentasikan entitas logika, dan hanya memiliki 2 nilai yaitu True (benar) dan False (salah).
- String adalah tipe data yang dipakai biasa untuk teks,

# Memahami HTML DOM

## Apa Itu HTML DOM ?

DOM yaitu (Document Object Model) adalah sebuah model data standar. DOM adalah cara javascript untuk melihat suatu halaman html, DOM ini adalah sebuah platform dan juga interface yang dimana memperbolehkan pengaksesan serta adanya perubahan pada konten pada dokumen.

HTML DOM ini mengacu kepada sebuah dokumen html, jadi disini ialah konten, struktur, dan juga style pada dokumen html dapat diakses dan dirubah dengan menggunakan sintaks javascript.



# Membuat HTML DOM

## Bagaimana Membuat HTML DOM?

DOM yaitu (Document Object Model) adalah sebuah model data standar. DOM adalah cara javascript untuk melihat suatu halaman html, DOM ini adalah sebuah platform dan juga interface yang dimana memperbolehkan pengaksesan serta adanya perubahan pada konten pada dokumen.

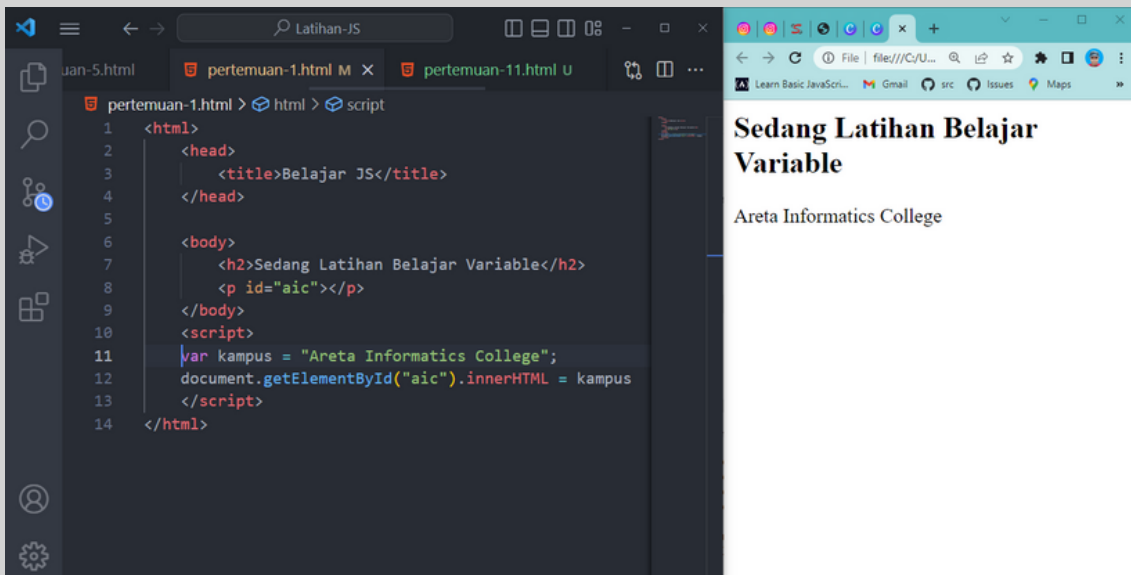
HTML DOM ini mengacu kepada sebuah dokumen html, jadi disini ialah konten, struktur, dan juga style pada dokumen html dapat diakses dan dirubah dengan menggunakan sintaks javascript.



# Membuat Variabel

## Bagaimana Membuat Variabel ?

Variabel dalam JavaScript adalah sebuah container yang digunakan untuk menyimpan sebuah nilai. Variabel dapat berisi berbagai jenis nilai seperti teks, angka, dan objek. Anda dapat menggunakan variabel untuk menyimpan informasi yang diperlukan dalam sebuah program atau skrip, untuk menggunakan variabel dalam JavaScript, pertama-tama anda perlu mengisi variabel tersebut dengan sebuah nilai



# Membuat Operator

## Bagaimana Membuat Operator ?

Operator Javascript merupakan sebuah simbol yang dapat digunakan untuk memberi tahu kepada JavaScript untuk melakukan sebuah tindakan dari operasi,

Ketika melakukan perhitungan matematika, atau logika pada nilai variabel yang dimana dibutuhkan untuk melakukan sebuah operasi disebut operan.

dibawah ini ada beberapa jenis operator javascript yang harus kita ketahui

### **Jenis Operator Javascript :**

- Operator Aritmatika(Arithmetic)
- Operator Perbandingan javascript
- Operator Logika Javascript
- Operator Penugasan(Assignment)
- Operator Ternary Javascript

# Membuat Operator Aritmatika

## • Operator Aritmatika

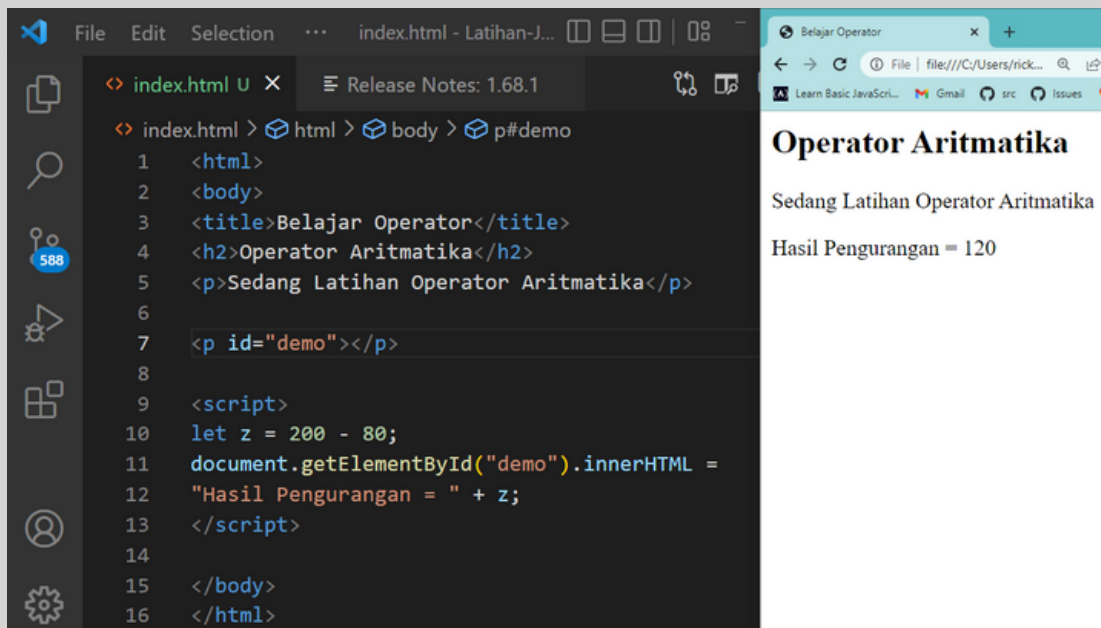
Operator Aritmatika dalam Javascript adalah operator ini banyak digunakan untuk melakukan sebuah perhitungan numerik dalam sebuah sintaksis programming pada bahasa pemograman Javascript.

### **Simbol operator aritmatika :**

- + Penjumlahan
- - Pengurangan
- \*Perkalian
- / Pembagian
- % Modulus (siswa bagi)
- \*\* Eksponensial
- ++ Increment
- -- Decrement

# Membuat Operator Aritmatika

## Contoh Operator Arimatika



The image shows a side-by-side view of a code editor and a web browser. The code editor on the left displays an HTML file named 'index.html' with the following code:

```
1 <html>
2 <body>
3 <title>Belajar Operator</title>
4 <h2>Operator Aritmatika</h2>
5 <p>Sedang Latihan Operator Aritmatika</p>
6
7 <p id="demo"></p>
8
9 <script>
10 let z = 200 - 80;
11 document.getElementById("demo").innerHTML =
12 "Hasil Pengurangan = " + z;
13 </script>
14
15 </body>
16 </html>
```

The web browser on the right shows the rendered page with the title 'Operator Aritmatika', the subtitle 'Sedang Latihan Operator Aritmatika', and the output 'Hasil Pengurangan = 120'.

# Membuat Operator Perbandingan

## • Operator Perbandingan

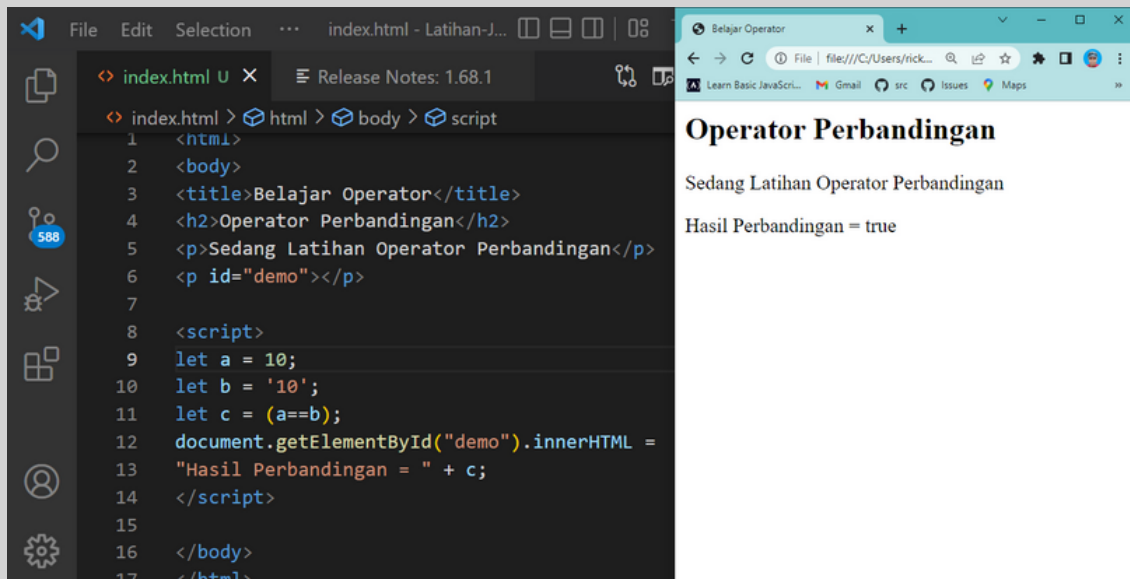
Operator perbandingan atau biasa disebut comparison operator digunakan untuk membandingkan 2 operand atau lebih. yang dimana biasanya Operator perbandingan digunakan penentu dari tingkat kecocokan.

### **Simbol operator perbandingan :**

- == Nilai Sama dengan
- === Nilai dan tipe data sama dengan
- != Tidak sama dengan
- !== Nilai dan tipe data tidak sama dengan
- > Lebih besar dari
- < Lebih kecil dari
- >= Lebih besar dari atau sama dengan
- <= Lebih kecil dari atau sama dengan

# Membuat Operator Perbandingan

## Contoh Operator Perbandingan



The image shows a side-by-side view of a code editor and a web browser. The code editor on the left displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Operator</title>
4 <h2>Operator Perbandingan</h2>
5 <p>Sedang Latihan Operator Perbandingan</p>
6 <p id="demo"></p>
7
8 <script>
9   let a = 10;
10  let b = '10';
11  let c = (a==b);
12  document.getElementById("demo").innerHTML =
13  "Hasil Perbandingan = " + c;
14 </script>
15
16 </body>
17 </html>
```

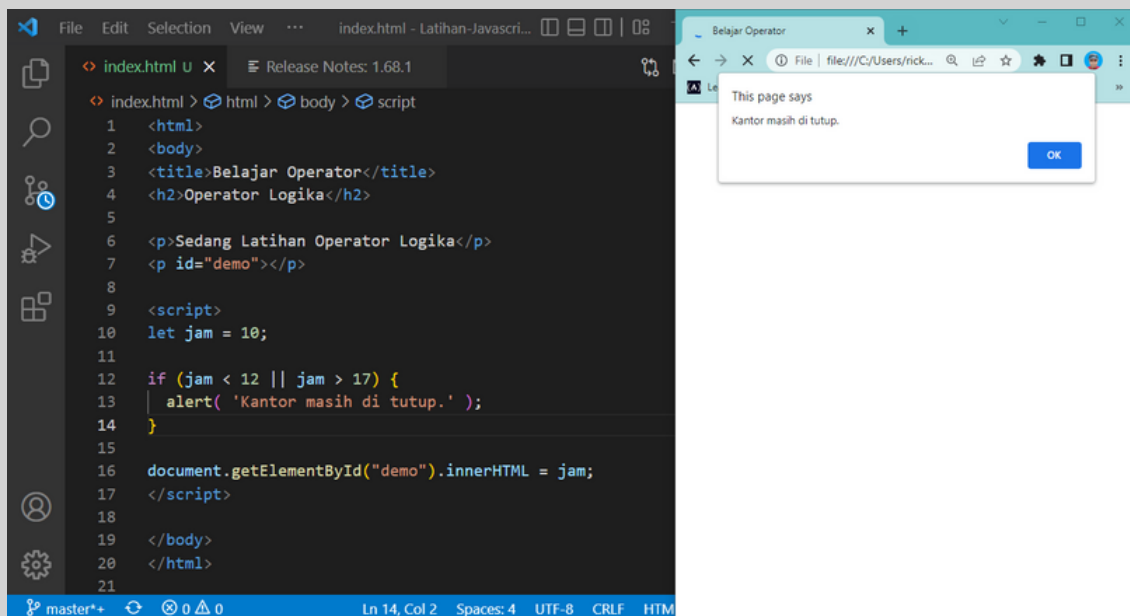
The web browser on the right shows the rendered page with the title 'Operator Perbandingan'. The page content includes the text 'Sedang Latihan Operator Perbandingan' and 'Hasil Perbandingan = true', which is displayed inside the element with id 'demo'.

# Membuat Operator Logika

## Operator Logika

Operator logika biasa dipakai dengan nilai boolean. Walaupun begitu hal tersebut tidak membatasi sebuah tipe data ataupun expression lain digunakan dengan operator logika.

**Simbol operator logika :** &&, (dan) , || Or (atau) ! (tidak)

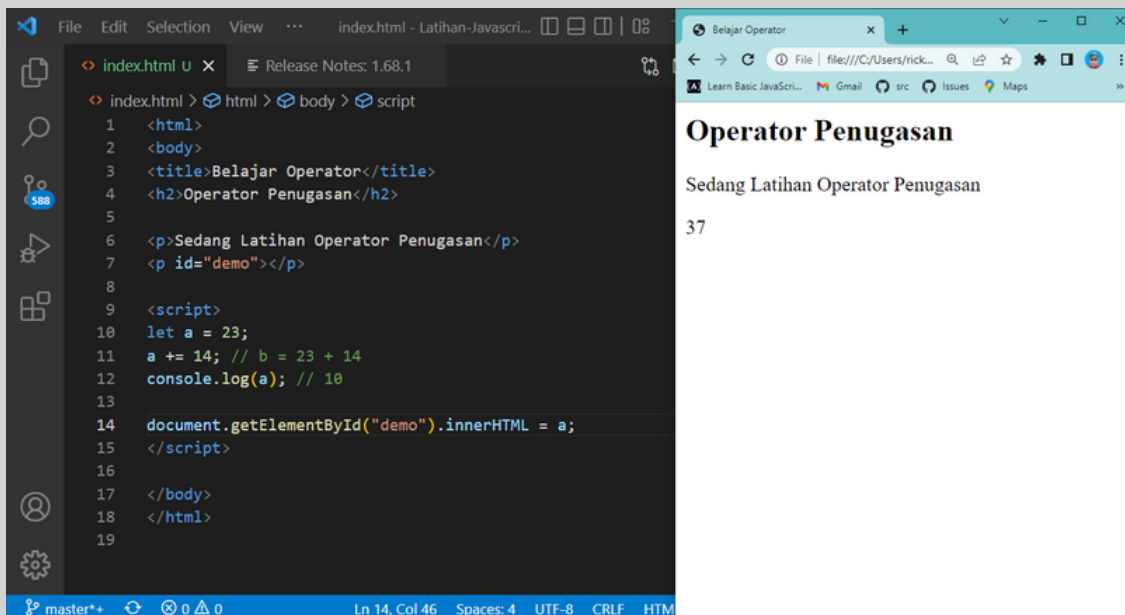


# Membuat Operator Penugasan

- **Membuat Operator Penugasan**

Operator penugasan adalah operator yang dimana menggunakan tanda sama dengan (=) untuk mengisi sebuah nilai pada suatu variabel,

Operator penugasan digunakan untuk menyatakan suatu perintah maupun sebuah fungsi dalam perhitungan untuk memberikan nilai kepada variabel.



The screenshot displays a code editor on the left and a web browser on the right. The code editor shows an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Operator</title>
4 <h2>Operator Penugasan</h2>
5
6 <p>Sedang Latihan Operator Penugasan</p>
7 <p id="demo"></p>
8
9 <script>
10 let a = 23;
11 a += 14; // b = 23 + 14
12 console.log(a); // 10
13
14 document.getElementById("demo").innerHTML = a;
15 </script>
16
17 </body>
18 </html>
19
```

The web browser on the right shows the rendered page with the title 'Operator Penugasan' and the content 'Sedang Latihan Operator Penugasan'. The page number '37' is visible in the bottom right corner of the browser window.

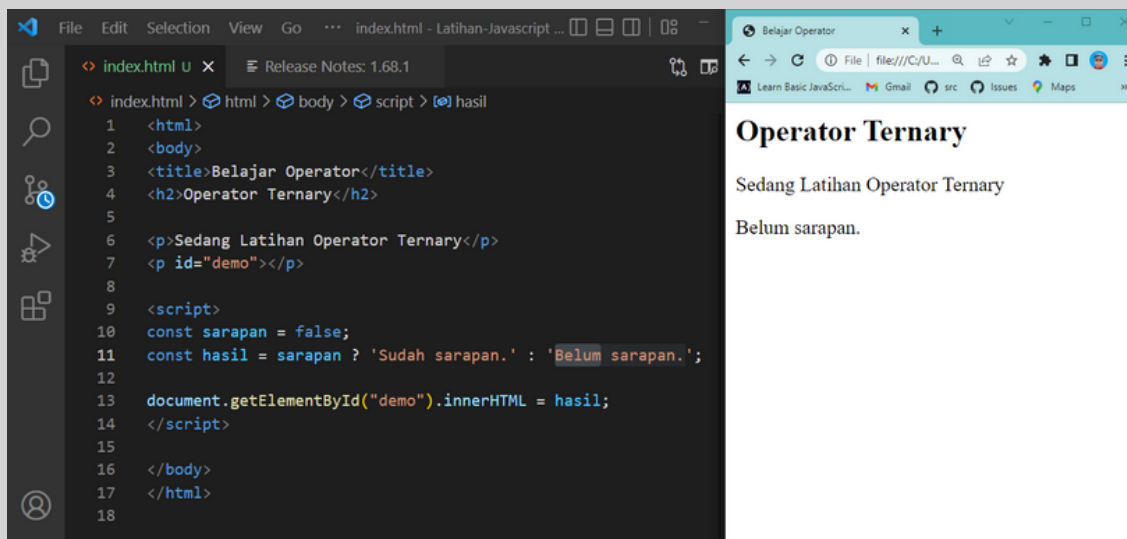


# Membuat Operator Ternary

- **Membuat Operator Ternary**

Operator Ternary merupakan nilai yang disimpan jika Benar akan digunakan jika kondisi bernilai true, tapi jika false akan menggunakan nilai di jika Salah;

**Aturan operator ternary :** kondisi ? jikaBenar : jikaSalah;



# Membuat Percabangan

## Apa Itu Percabangan Javascript ?

Percabangan dalam javascript akan menentukan akan menghasilkan sebuah proses yang akan dilewati oleh program ketika adanya parameter berada pada dalam **pengkondisian benar atau salah**, inilah alasan mengapa di sebut sebagai percabangan.

Biasanya dalam, struktur percabangan pemrograman yang sering digunakan yaitu “if” dan juga “case”, dari percabangan dalam suatu cara yang digunakan dalam perograman untuk mengambil sebuah keputusan adanya suatu pilihan “True” atau “False” dari beberapa kondisi.

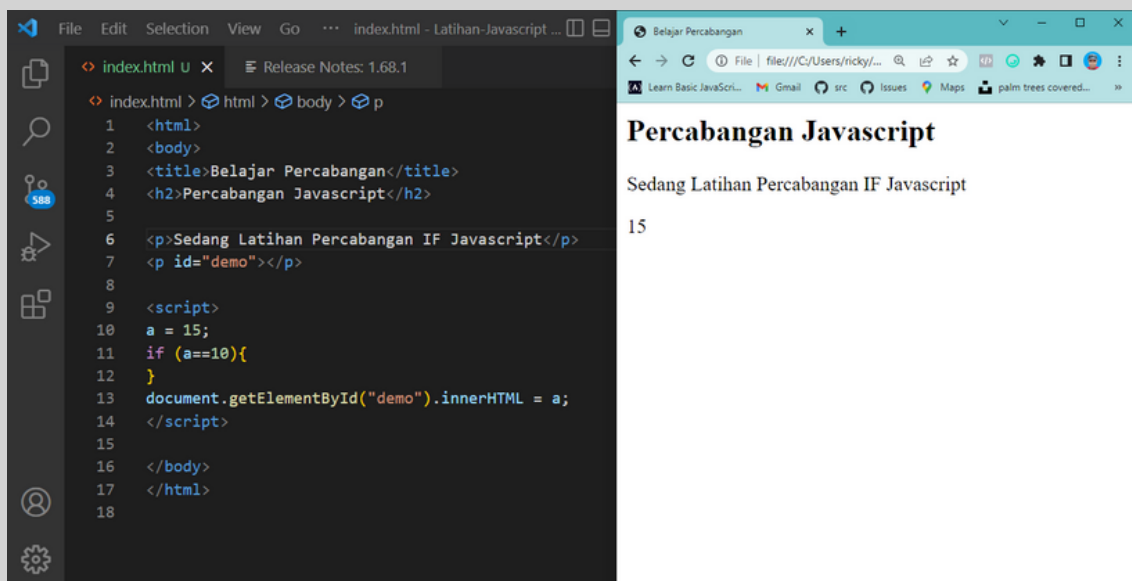
### Jenis-jenis percabangan javascript :

- IF
- IF/ELSE
- SWITCH & CASE

# Membuat Percabangan IF

- **Percabangan IF**

Percabangan if memiliki 1 nilai yaitu true, biasanya digunakan ketika kita hanya memiliki satu kondisi dan satu aksi (pilihan) pada statement.



The screenshot shows a code editor on the left and a web browser on the right. The code editor displays an HTML file named 'index.html' with the following code:

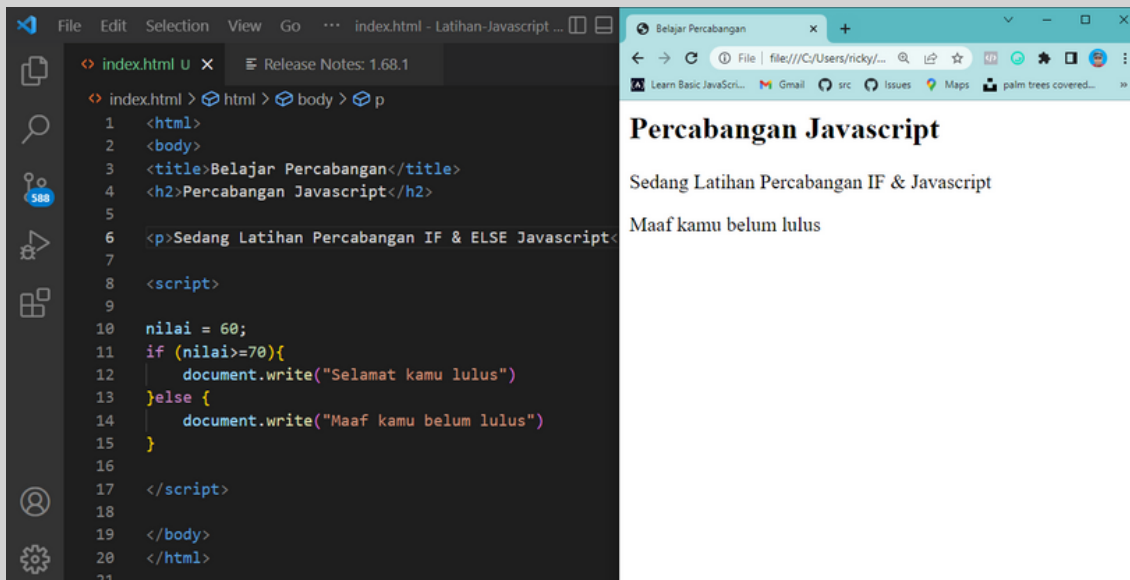
```
1 <html>
2 <body>
3 <title>Belajar Percabangan</title>
4 <h2>Percabangan Javascript</h2>
5
6 <p>Sedang Latihan Percabangan IF Javascript</p>
7 <p id="demo"></p>
8
9 <script>
10 a = 15;
11 if (a==10){
12 }
13 document.getElementById("demo").innerHTML = a;
14 </script>
15
16 </body>
17 </html>
18
```

The web browser on the right shows the rendered page with the title 'Percabangan Javascript' and the content 'Sedang Latihan Percabangan IF Javascript'. The value '15' is displayed below the text, indicating that the if statement condition was not met, and the default content was shown.

# Membuat Percabangan IF & ELSE

- **Percabangan IF & ELSE**

Percabangan if & else menghasilkan 2 nilai yaitu true dan false digunakan ketika kita memiliki dua pilihan atau statement (aksi). Statement yang sudah dieksekusi ketika kondisi bernilai true maupun false, semisal kasus kita login pada social media, "selamat anda sudah login", "anda sudah logout"



The screenshot displays a code editor on the left and a web browser on the right. The code editor shows an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Percabangan</title>
4 <h2>Percabangan Javascript</h2>
5
6 <p>Sedang Latihan Percabangan IF & ELSE Javascript</p>
7
8 <script>
9
10 nilai = 60;
11 if (nilai>=70){
12     document.write("Selamat kamu lulus")
13 }else {
14     document.write("Maaf kamu belum lulus")
15 }
16
17 </script>
18
19 </body>
20 </html>
```

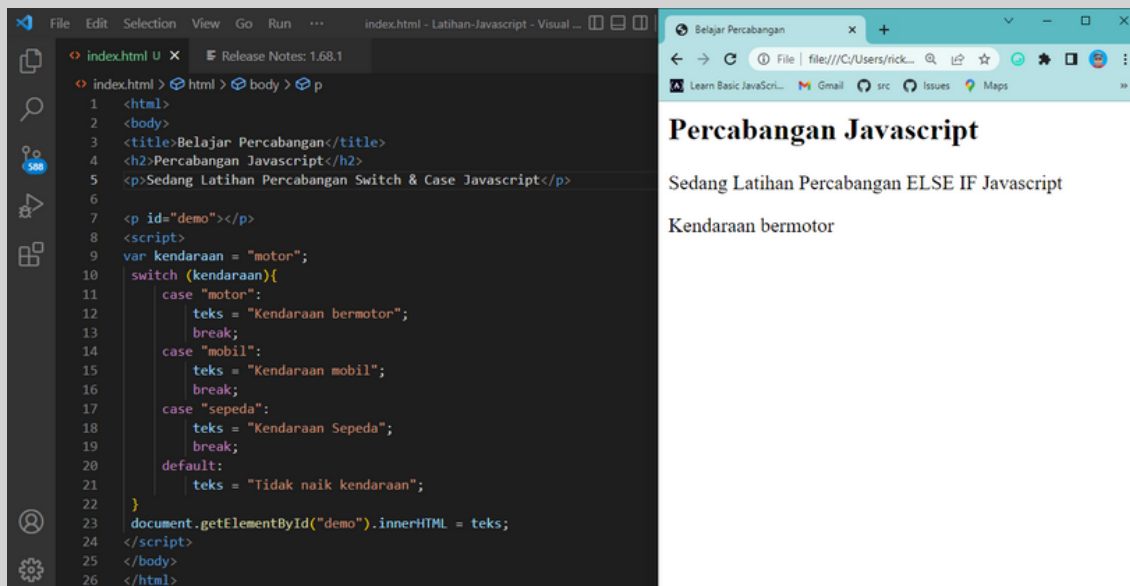
The web browser on the right shows the rendered output of the code. The title is 'Belajar Percabangan' and the content includes the heading 'Percabangan Javascript' and the paragraph 'Sedang Latihan Percabangan IF & Javascript'. Below this, the output of the JavaScript code is displayed: 'Maaf kamu belum lulus'.

# Membuat Percabangan ELSE IF

- **Percabangan ELSE IF**

Aturan dari else if mirip seperti sebelumnya, dimana JavaScript akan memilih setiap kondisi dari atas baru ke bawah.

Jika ditemukan adanya kondisi yang bernilai true maka statement yang ada di dalam kondisi tersebut akan dieksekusi lalu keluar dari percabangan.



The image shows a side-by-side comparison of a code editor and a web browser. The code editor on the left displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Percabangan</title>
4 <h2>Percabangan Javascript</h2>
5 <p>Sedang Latihan Percabangan Switch & Case Javascript</p>
6
7 <p id="demo"></p>
8 <script>
9 var kendaraan = "motor";
10 switch (kendaraan){
11     case "motor":
12         teks = "Kendaraan bermotor";
13         break;
14     case "mobil":
15         teks = "Kendaraan mobil";
16         break;
17     case "sepeda":
18         teks = "Kendaraan Sepeda";
19         break;
20     default:
21         teks = "Tidak naik kendaraan";
22 }
23 document.getElementById("demo").innerHTML = teks;
24 </script>
25 </body>
26 </html>
```

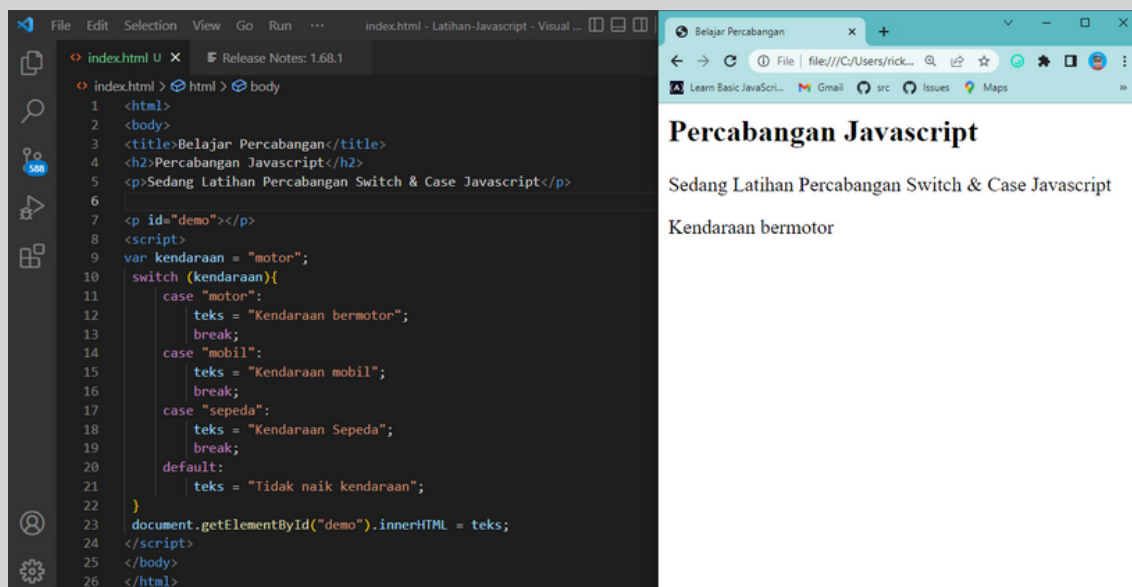
The web browser on the right shows the rendered output of the code. The page title is 'Percabangan Javascript'. The main content area displays the text 'Sedang Latihan Percabangan ELSE IF Javascript' and 'Kendaraan bermotor', which is the result of the switch statement in the code.

# Membuat Percabangan Switch & Case

- **Percabangan Switch & case**

Switch case percabangan ini hampir sama dengan percabangan yang lain yaitu if else, hanya saja dalam penggunaan switch case untuk mengecek sebuah data yang bertipe karakter dan integer.

Sementara dalam if else pengecekan bisa dengan menggunakan (<,>,<=,>=), yang sudah saya bahas komparasi. Itulah yang membedakan keduanya dalam bentuknya percabangan.



The screenshot shows a code editor on the left and a web browser on the right. The code editor displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Percabangan</title>
4 <h2>Percabangan Javascript</h2>
5 <p>Sedang Latihan Percabangan Switch & Case Javascript</p>
6
7 <p id="demo"></p>
8 <script>
9 var kendaraan = "motor";
10 switch (kendaraan){
11     case "motor":
12         teks = "Kendaraan bermotor";
13         break;
14     case "mobil":
15         teks = "Kendaraan mobil";
16         break;
17     case "sepeda":
18         teks = "Kendaraan Sepeda";
19         break;
20     default:
21         teks = "Tidak naik kendaraan";
22 }
23 document.getElementById("demo").innerHTML = teks;
24 </script>
25 </body>
26 </html>
```

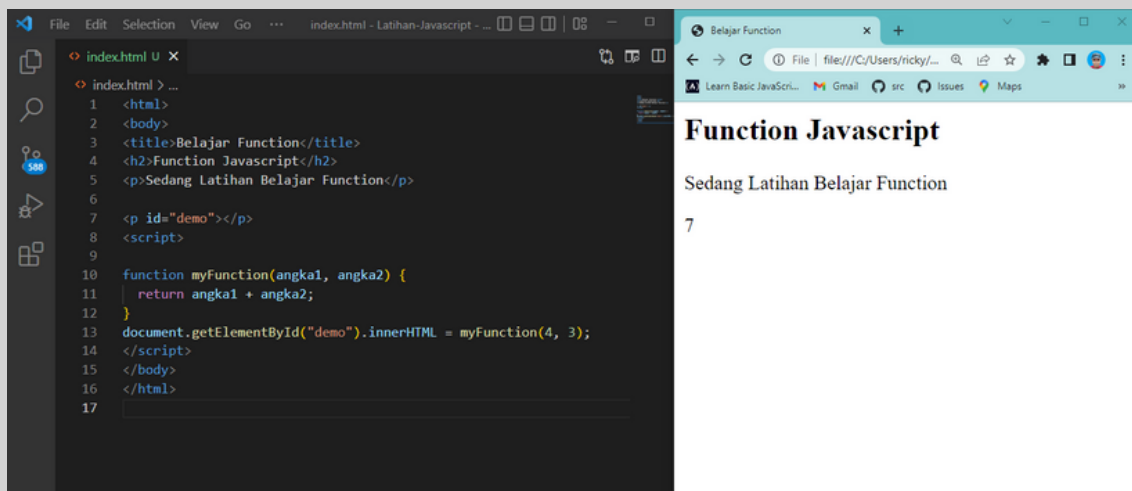
The web browser on the right shows the rendered page with the title 'Percabangan Javascript' and the text 'Sedang Latihan Percabangan Switch & Case Javascript' and 'Kendaraan bermotor' displayed below it.

# Membuat Function

## Bagaimana Membuat Function ?

Function dalam javascript merupakan sebuah blok kode yang digunakan untuk membungkus suatu proses yang akan di eksekusi dengan tujuan dalam penulisan kode atau proses yang sama tidak ditulis secara berulang kali.

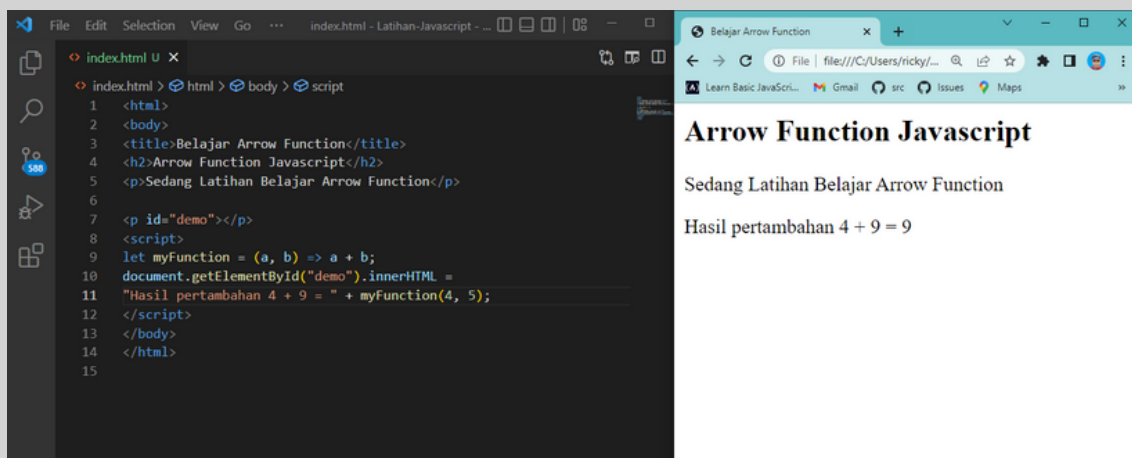
Dalam function javascript terdiri dari parameter, ini digunakan untuk menentukan proses yang akan dijalankan, sesuai dengan kebutuhan program yang kita buat.



# Membuat Arrow Function

## Bagaimana Membuat Arrow Function

Arrow function adalah merupakan sebuah penulisan sintak yang lebih singkat, menggunakan simbol “=>”, fungsi yang dideklarasikan ini menggunakan sebuah panah ini bersifat anonim.





# Membuat Event

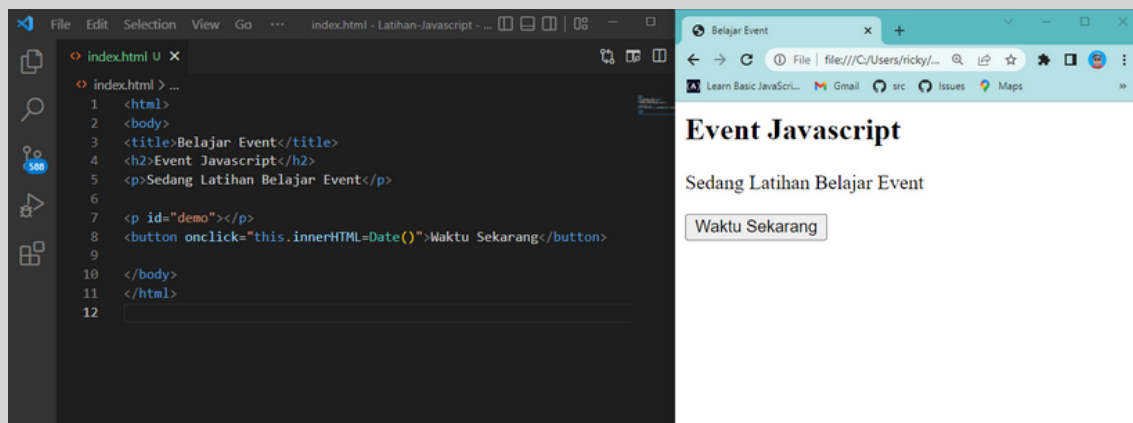
## Bagaimana Membuat Event ?

Event dalam javascript merupakan hal yang terjadi pada element, semisal nya kita memiliki sebuah tombol pada halaman website maupun aplikasi yang sedang di buat, dan kita ingin memberikan suatu efek aksi pada tombol tersebut ketika di klik. adapun beberapa event yang terdapat dalam javascript yang kamu harus ketahui

- onclick = event jika sebuah element html di klik.
- onchange = event jika sebuah element html berubah.
- onmouseover = event jika sebuah element html di letakkan cursor mouse.
- onmouseout = event jika saat cursor mouse meninggalkan element html.
- onkeydown = event jika saat di terjadi pengetikan pada elemnet html.
- onload = event ketika jika saat element atau halaman di buka.

# Membuat Event

## Contoh event



# Membuat Perulangan

## • Membuat Perulangan Loop For

Penggunaan perulangan for di peruntukan ketika sudah ada yang diketahui berapa kali jumlah yang akan dilakukan perulangan yang harus dilakukan berdasarkan kondisi yang sudah ditetapkan.

Membuat perulangan dalam javascript **di bedakan menjadi 2** jenis, ada **Counted Loop**, dan **Uncounted Loop**

Perulangan yang termasuk dalam (Counted Loop)

- Perulangan For
- Perulangan Foreach
- Perulangan Repeat

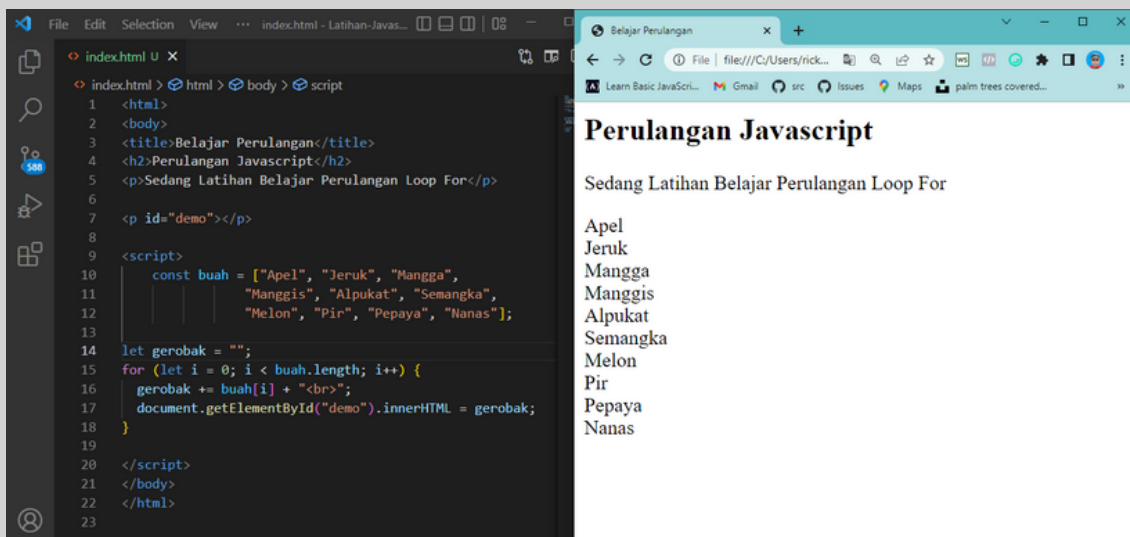
Perulangan yang termasuk dalam (Uncounted Loop)

- Perulangan While
- Perulangan Do/While

# Membuat Perulangan Loop For

## Contoh Perulangan Loop For

Penggunaan perulangan for di peruntukan ketika sudah ada yang diketahui berapa kali jumlah yang akan dilakukan perulangan yang harus dilakukan berdasarkan kondisi yang sudah ditetapkan.



The image shows a side-by-side comparison of a code editor and a web browser. The code editor on the left displays the following JavaScript code:

```
1 <html>
2 <body>
3 <title>Belajar Perulangan</title>
4 <h2>Perulangan Javascript</h2>
5 <p>Sedang Latihan Belajar Perulangan Loop For</p>
6
7 <p id="demo"></p>
8
9 <script>
10   const buah = ["Apel", "Jeruk", "Mangga",
11                 "Manggis", "Alpukat", "Semangka",
12                 "Melon", "Pir", "Pepaya", "Nanas"];
13
14   let gerobak = "";
15   for (let i = 0; i < buah.length; i++) {
16     gerobak += buah[i] + "<br>";
17     document.getElementById("demo").innerHTML = gerobak;
18   }
19 </script>
20 </body>
21 </html>
```

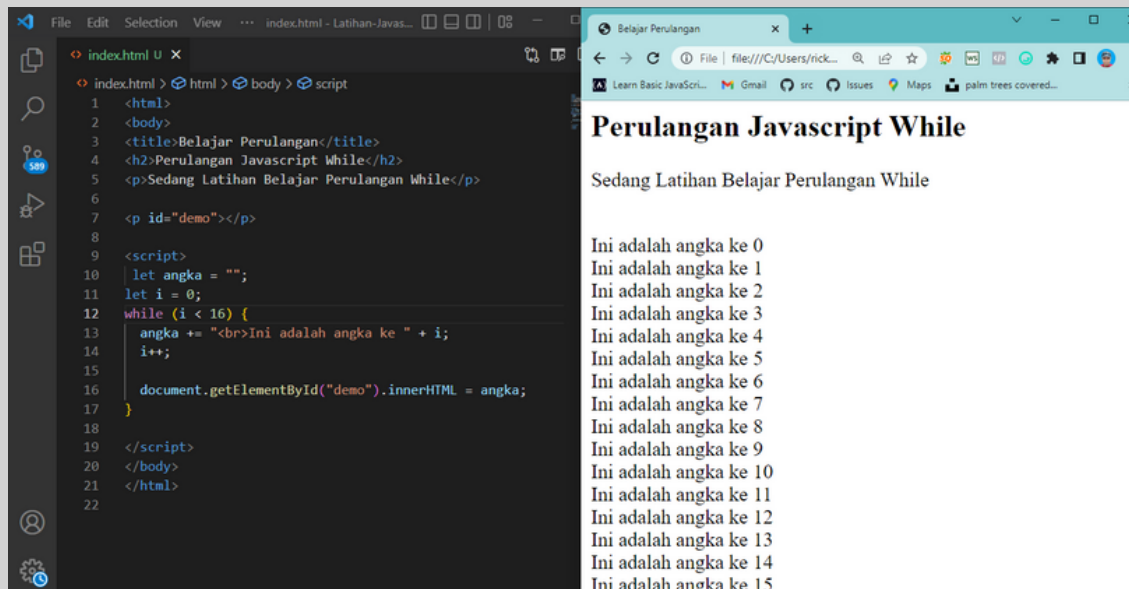
The web browser on the right shows the rendered output of the code. The title is "Perulangan Javascript" and the page content is "Sedang Latihan Belajar Perulangan Loop For". Below this, a list of fruits is displayed, each on a new line:

Apel  
Jeruk  
Mangga  
Manggis  
Alpukat  
Semangka  
Melon  
Pir  
Pepaya  
Nanas

# Membuat Perulangan Loop While

## Contoh Perulangan Loop While

Dalam perulangan loop while, inisialisasi untuk variabel counter harus dispesifikasikan dahulu, sebelum memulai while loop dan increment dari counter harus berada di dalam sebuah while block.



The image shows a side-by-side comparison of a code editor and a web browser. The code editor on the left displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Perulangan</title>
4 <h2>Perulangan Javascript While</h2>
5 <p>Sedang Latihan Belajar Perulangan While</p>
6
7 <p id="demo"></p>
8
9 <script>
10 let angka = "";
11 let i = 0;
12 while (i < 16) {
13   angka += "<br>Ini adalah angka ke " + i;
14   i++;
15   document.getElementById("demo").innerHTML = angka;
16 }
17
18 </script>
19 </body>
20 </html>
```

The web browser on the right shows the rendered output of the code. The title is 'Perulangan Javascript While' and the page content is 'Sedang Latihan Belajar Perulangan While'. Below this, the output of the while loop is displayed as a list of 16 lines, each starting with 'Ini adalah angka ke' followed by the number from 0 to 15.

# Membuat Array

## Bagaimana Membuat Array?

**Array** merupakan sebuah **tipe data** yang dimana berisi sebuah kumpulan dari nilai maupun tipe data dari yang lain.

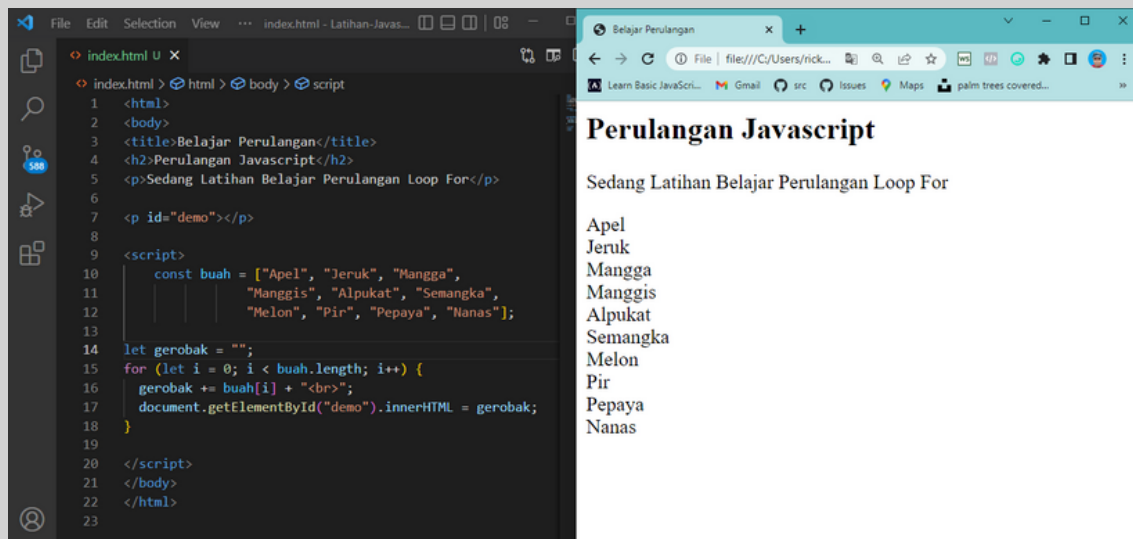
Tetapi nilai di dalam array disebut dengan sebuah **elemen**, dan setiap elemen ini memiliki 'nomor urut' yang dikenal dengan istilah disebut **index**, nah index ini dimulai dari perhitungan angka 0.

Tidak hanya itu array mempunyai sebuah array methods, yang dimana fungsi dari array method ini bisa menambah sebuah Berikut dibawah ini beberapa fungsi dari array method:

- join()
- reserve()
- sort()
- slice()
- splice()
- push dan pop()
- shift() dan unshift()

# Membuat Array

## Contoh Membuat Array



The image shows a side-by-side comparison of a code editor and a web browser. The code editor on the left displays the following code:

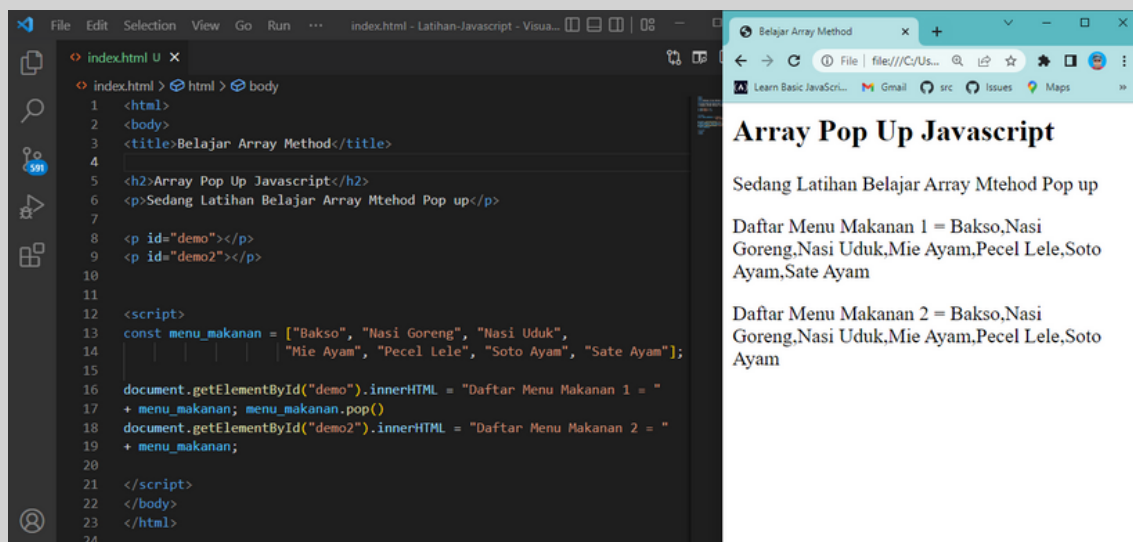
```
1 <html>
2 <body>
3 <title>Belajar Perulangan</title>
4 <h2>Perulangan Javascript</h2>
5 <p>Sedang Latihan Belajar Perulangan Loop For</p>
6
7 <p id="demo"></p>
8
9 <script>
10   const buah = ["Apel", "Jeruk", "Mangga",
11                 "Manggis", "Alpukat", "Semangka",
12                 "Melon", "Pir", "Pepaya", "Nanas"];
13
14   let gerobak = "";
15   for (let i = 0; i < buah.length; i++) {
16     gerobak += buah[i] + "<br>";
17     document.getElementById("demo").innerHTML = gerobak;
18   }
19
20 </script>
21 </body>
22 </html>
23
```

The web browser on the right shows the rendered output of the code. The title is "Perulangan Javascript" and the main heading is "Sedang Latihan Belajar Perulangan Loop For". Below this, a list of fruits is displayed, each on a new line:

Apel  
Jeruk  
Mangga  
Manggis  
Alpukat  
Semangka  
Melon  
Pir  
Pepaya  
Nanas

# Membuat Array Method Pop

## Contoh Method Array pop



The image shows a side-by-side comparison of a code editor and a web browser. The code editor on the left displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3 <title>Belajar Array Method</title>
4
5 <h2>Array Pop Up Javascript</h2>
6 <p>Sedang Latihan Belajar Array Mtehod Pop up</p>
7
8 <p id="demo"></p>
9 <p id="demo2"></p>
10
11
12 <script>
13 const menu_makanan = ["Bakso", "Nasi Goreng", "Nasi Uduk",
14   |   |   |   |   |
15   "Mie Ayam", "Pecel Lele", "Soto Ayam", "Sate Ayam"];
16 document.getElementById("demo").innerHTML = "Daftar Menu Makanan 1 = "
17 + menu_makanan; menu_makanan.pop()
18 document.getElementById("demo2").innerHTML = "Daftar Menu Makanan 2 = "
19 + menu_makanan;
20
21 </script>
22 </body>
23 </html>
24
```

The web browser on the right shows the rendered page titled 'Array Pop Up Javascript'. The page content is as follows:

Array Pop Up Javascript

Sedang Latihan Belajar Array Mtehod Pop up

Daftar Menu Makanan 1 = Bakso,Nasi Goreng,Nasi Uduk,Mie Ayam,Pecel Lele,Soto Ayam,Sate Ayam

Daftar Menu Makanan 2 = Bakso,Nasi Goreng,Nasi Uduk,Mie Ayam,Pecel Lele,Soto Ayam

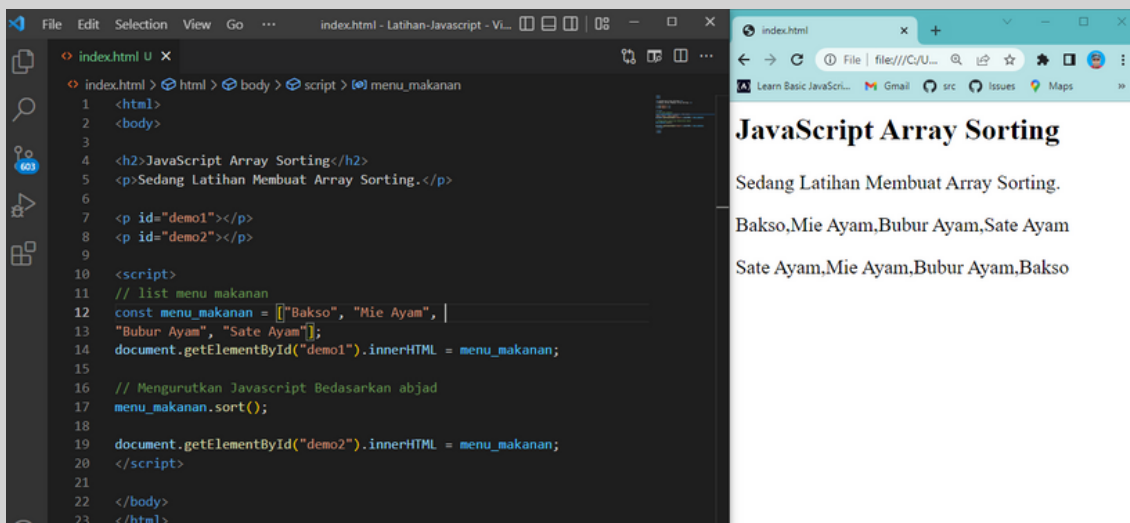


# Membuat Array Sorting

## Bagaimana Membuat Array Sorting ?

Dalam menggunakan metode sort () mengurutkan semua objek pada array berdasarkan abjad, dimulai dengan huruf awal yaitu A, dan diakhiri dengan huruf Z.

Hal ini berguna saat membuat daftar yang dimana memerlukan urutan item berdasarkan sebuah abjad



The screenshot shows a code editor on the left and a web browser on the right. The code editor displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3
4 <h2>JavaScript Array Sorting</h2>
5 <p>Sedang Latihan Membuat Array Sorting.</p>
6
7 <p id="demo1"></p>
8 <p id="demo2"></p>
9
10 <script>
11 // list menu makanan
12 const menu_makanan = ["Bakso", "Mie Ayam", "Bubur Ayam", "Sate Ayam"];
13 document.getElementById("demo1").innerHTML = menu_makanan;
14
15 // Mengurutkan Javascript Berdasarkan abjad
16 menu_makanan.sort();
17
18 document.getElementById("demo2").innerHTML = menu_makanan;
19 </script>
20
21
22 </body>
23 </html>
```

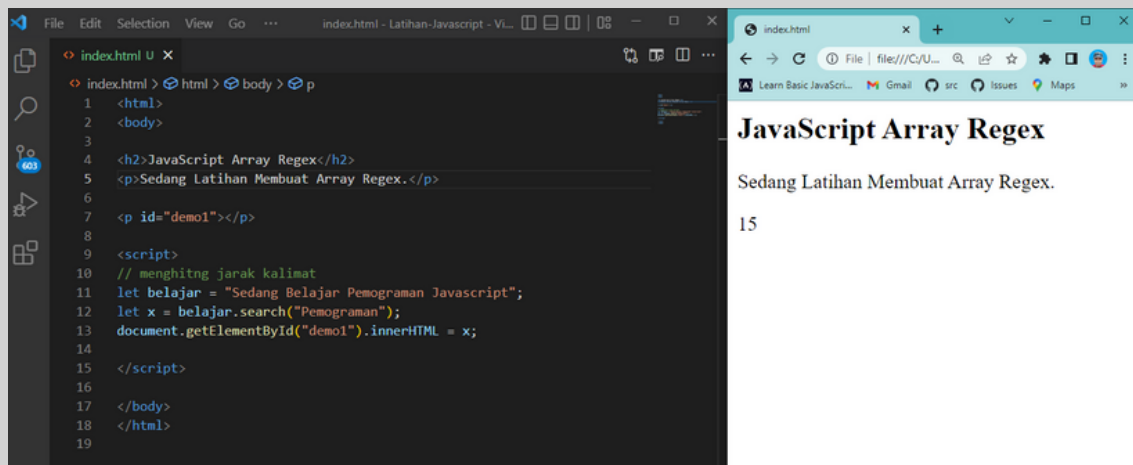
The web browser on the right shows the rendered output of the code. It has a title 'JavaScript Array Sorting' and displays the text 'Sedang Latihan Membuat Array Sorting.' followed by two lines of food items: 'Bakso,Mie Ayam,Bubur Ayam,Sate Ayam' and 'Sate Ayam,Mie Ayam,Bubur Ayam,Bakso'.

# Membuat Array Regex

## Bagaimana Membuat Array Regex?

Dalam javascript Ekspresi reguler merupakan sebuah rangkaian pada karakter yang dimana akan membentuk sebuah pola pencarian.

Saat kita melakukan mencari data dalam kalimat teks, kita dapat menggunakan pola pencarian spesifik ini untuk menjelaskan apa yang ingin dicari.



The image shows a side-by-side view of a code editor and a web browser. The code editor on the left displays an HTML file named 'index.html' with the following content:

```
1 <html>
2 <body>
3
4 <h2>JavaScript Array Regex</h2>
5 <p>Sedang Latihan Membuat Array Regex.</p>
6
7 <p id="demo1"></p>
8
9 <script>
10 // menghitung jarak kalimat
11 let belajar = "Sedang Belajar Pemograman Javascript";
12 let x = belajar.search("Pemograman");
13 document.getElementById("demo1").innerHTML = x;
14
15 </script>
16
17 </body>
18 </html>
19
```

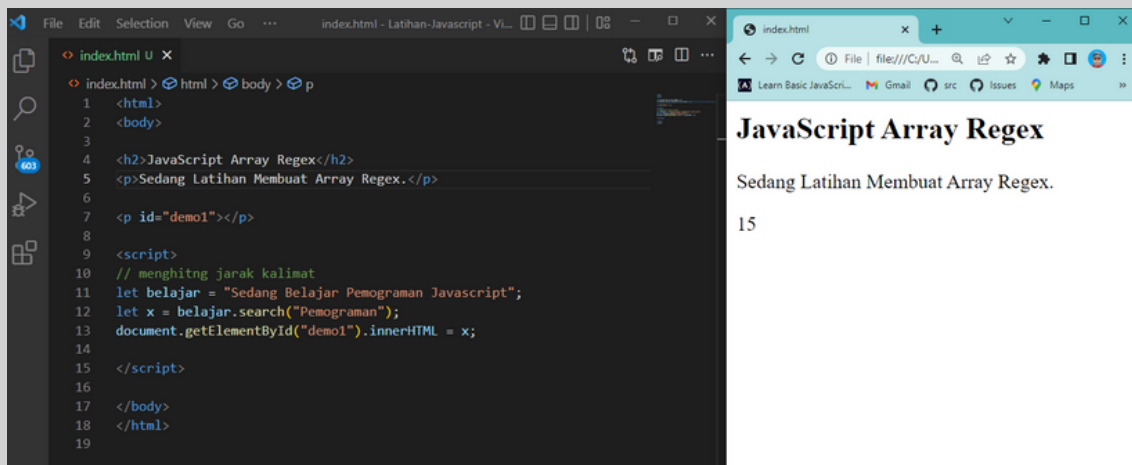
The web browser on the right shows the rendered page with the title 'JavaScript Array Regex' and the text 'Sedang Latihan Membuat Array Regex.' followed by the number '15'.

# Membuat Array Regex

## Bagaimana Membuat Array Regex?

Dalam javascript Ekspresi reguler merupakan sebuah rangkaian pada karakter yang dimana akan membentuk sebuah pola pencarian.

Saat kita melakukan mencari data dalam kalimat teks, kita dapat menggunakan pola pencarian spesifik ini untuk menjelaskan apa yang ingin dicari.



The screenshot shows a web browser window on the right and a code editor on the left. The browser displays the title "JavaScript Array Regex" and the text "Sedang Latihan Membuat Array Regex." with the number "15" below it. The code editor shows the following HTML and JavaScript code:

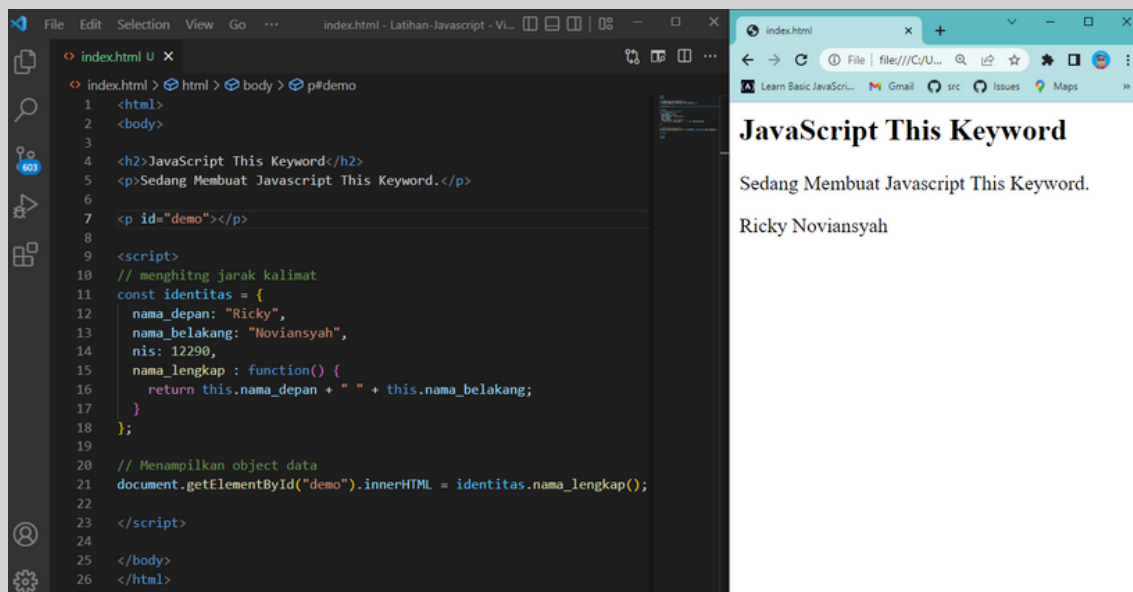
```
1 <html>
2 <body>
3
4 <h2>JavaScript Array Regex</h2>
5 <p>Sedang Latihan Membuat Array Regex.</p>
6
7 <p id="demo1"></p>
8
9 <script>
10 // menghitung jarak kalimat
11 let belajar = "Sedang Belajar Pemograman Javascript";
12 let x = belajar.search("Pemograman");
13 document.getElementById("demo1").innerHTML = x;
14
15 </script>
16
17 </body>
18 </html>
19
```

# Membuat This Keyword

## Bagaimana Membuat This Keyword ?

Pada javascript penggunaan this ini berhubungan dengan object maupun class. Menggunakan this di javascript akan membantu menghindari adanya kesalahan pemanggilan parameter yang ada dalam sebuah class atau object.

JavaScript this ini berada dalam scope object akan mengacu kepada sebuah object. Sementara this yang berdiri sendiri (diluar object) mengacu kepada global object (object window).

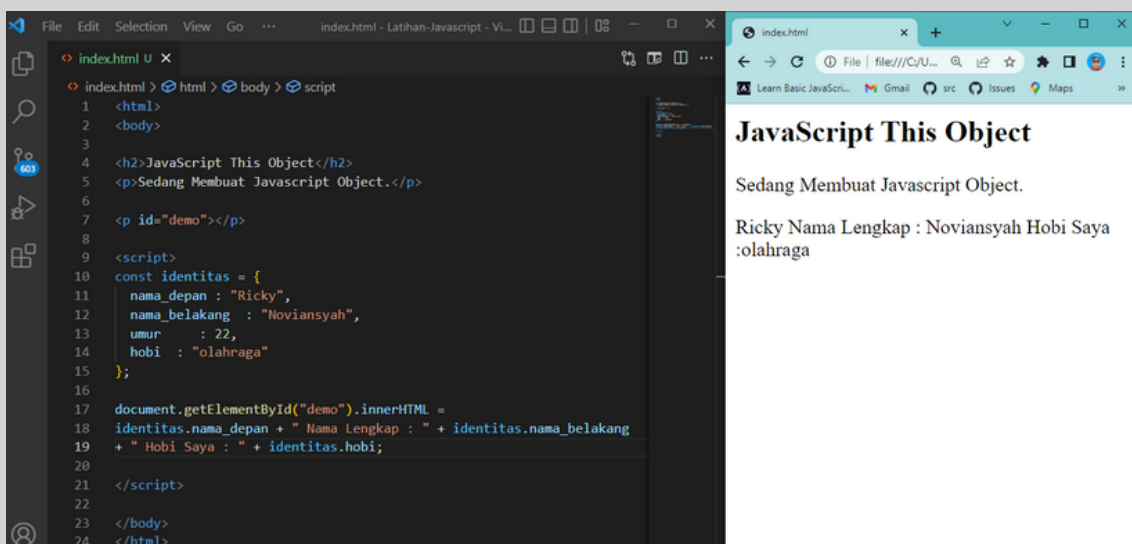


# Membuat Object

## Bagaimana Membuat Object ?

Objek dalam Javascript ini adalah variabel yang dimana berisi 2 nilai yaitu (Properti) dan (Method), dalam Javascript objek.

Hal ini merupakan bagian yang sangat penting, harus memahami object pada javascript dengan baik maka akan dengan mudah kedepan nya dalam mempelajari bahasa pemrograman dengan Javascript.



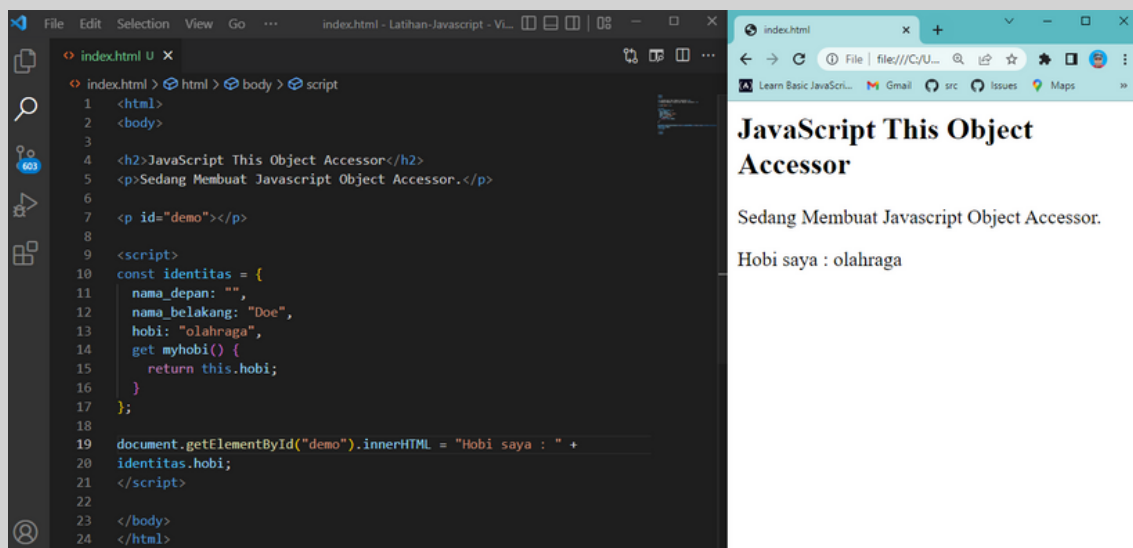
# Membuat Object Accessor

## Bagaimana Membuat Object Accessor ?

JavaScript Accessor adalah cara untuk mengakses nilai properti dalam object dengan menggunakan getter dan setter.

Accessor merupakan suatu properti yang dimana nilainya bisa didapatkan dari sebuah proses pada perhitungan.

Accessor ini terdiri dari 2 jenis, **getter** dan **setter**.



# Terima Kasih

Semoga e-book ini bisa membantu kamu dalam hal belajar membangun web dinamis menggunakan bahasa javascript.

Untuk menguasai javascript jangan terpaku hanya di ebook ini saja, teman teman harus sering searching di internet, semakin banyak masalah error semakin mendapatkan pengalaman baru

Kunci belajar supaya paham, adalah di ulang jangan dihafal sebab kalau di hafal.

## Kamu ada pertanyaan lain ?

Hubungi dibawah ini

✉ EMAIL : [rickynoviansyah.id@gmail.com](mailto:rickynoviansyah.id@gmail.com)

📍 Telegram : [@Rickybrainware](https://t.me/Rickybrainware)