

Nama = Riyadi Tri Waluya Budi

Kelas= SIK A3

Matkul = PBO(Pemrograman Berorientasi Objek)

***Pendahuluan JavaScript**

```
<!DOCTYPE html>
<html>
<body>
<h2>Belajar Javascript</h2>
<script>
alert("Saya sedang belajar Javascript")
</script>
</body>
</html>
```

***Operator Pada JavaScript**

Macam Operator

Operator Aritmatika

Operator aritmatika digunakan untuk melakukan aritmatika pada angka.

Operator	Penjelasan	Contoh
+	Penambahan	1+1
-	Pengurangan	1-1
*	Pengalian	1*3
/	Pembagian	10/2
%	Modulus (siswa pembagian)	7/2
++	Increment	variable++
--	Decrement	variable--

Contoh

```
var a = 10;  
var b = 20;  
var ab= a + b;
```

Untuk menguji contoh di atas, kita bisa mengetiknya melalui console kemudian panggil variabel yang kita butuhkan, lalu tekan enter untuk mengetahui output.

Image

Operator Assignment

Operator Assignment memberikan nilai ke variabel Javascript.

Operator	Contoh	Sama Seperti
----------	--------	--------------

=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Contoh

```
var x = 10;  
x += 25;
```

Image

Operator Penggabungan String

Operator "+" juga dapat digunakan untuk menggabungkan string.

Contoh

```
var txt1 = "Saya suka";  
var txt2 = "Nasi";  
var txt3 = "Goreng";  
var txt4 = txt1 + " " + txt2 + " " + txt3;
```

Untuk mengetahui outputnya, kita bisa mengetikkan pada console lalu memanggil variabel yang dibutuhkan dan tekan enter.

```
var x = 5 + 5;
```

```
var y = "5" + 5;
```

```
var z = "Balon Saya" + 5;
```

Image

```
var txt1 = "Saya suka makan ";
```

```
txt1 += "nasi goreng";
```

Image

Operator Pembandingan

Operator	Penjelasan
==	Sama dengan isinya
===	Sama isinya dan sama tipenya
!=	Tidak sama dengan isinya
!==	Tidak sama isinya dan tidak sama tipenya
>	Lebih dari
<	Kurang dari
>=	Lebih dari atau sama dengan
<=	Kurang dari atau sama dengan
?	Operator ternary (if else sederhana)

Operator Logika

Operator	Deskripsi
&&	Logika And. Bernilai TRUE bila kedua kondisi bernilai TRUE.
**`	
!	Logika Not. Logika Pembalik. Bernilai TRUE bila kondisi FALSE, dan bernilai FALSE bila kondisi TRUE.

Demikian sekilas pembahasan tentang operator pada Javascript

***Perulangan Pada JavaScript**

Pengertian Pengulangan

Dalam bahasa pemrograman, pengulangan dapat digunakan untuk "mengulang" kode yang kita inginkan, dengan data yang berbeda.

Pengulangan pada Javascript bisa menggunakan beberapa cara antara lain.

For Loop

While

Do While

For Loop

For loop digunakan untuk pengulangan melalui blok kode beberapa kali.

Sintaks

```
for( Pernyataan1; Pernyataan 2; Pernyataan 3)
```

```
{
```

```
    //blok kode yang akan dijalankan
```

```
}
```

Pernyataan 1 dijalankan (satu kali) sebelum eksekusi blok kode.

Pernyataan 2 mendefinisikan kondisi untuk mengeksekusi blok kode.

Pernyataan 3 dijalankan (setiap saat) setelah blok kode dijalankan.

Contoh

```
for(i=0;i<5;i++)
```

```
{
```

```
    teks += "Ini adalah nomor" + i + "<br>";
```

```
}
```

Pada contoh di atas, i dimulai dari 0, kemudian pada pernyataan 2 cek kondisi apakah i kurang dari 5, apabila bernilai true, maka blok kode dijalankan.

Pada pernyataan 3, i selalu ditambahkan 1 setiap kali usai menjalankan blok kode.

Berikut ini contoh pengulangan dengan output ditampilkan pada console.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
  for (var i = 0; i < 5; i++)
  {
    console.log("Nilai saat ini " + i);
  }
</script>
</body>
</html>
```

Simpan kode di atas misal dengan nama for_js.html, lalu kita bisa buka console pada browser (biasanya terletak pada menu Developer tools di browser). Panggil url sesuai lokasi penyimpanan dan nama filenya.

Image

Namun apabila kita ingin menampilkan output pengulangan melalui browser, kita bisa mengetikkan kode seperti di bawah ini.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
```

```
var text="";  
for (var i = 0; i < 5; i++)  
{  
text += "Nilai saat ini " + i + "<br>";  
document.getElementById("data").innerHTML = text;  
}  
</script>  
</body>  
</html>
```

Output dari kode di atas akan tampil pada halaman browser seperti di bawah ini.

Image

While Loop

While digunakan untuk pengulangan melalui blok kode selama kondisi tertentu benar.

Sintaks

```
while(kondisi)  
{  
    //blok kode yang akan dijalankan  
}
```

Contoh

```
while(i<10)  
{  
    teks += "Ini adalah nomor " + i;  
    i++;  
}
```

Saat kondisi bernilai true atau benar, maka pengulangan akan dijalankan.

Berikut contoh penggunaan while dengan output ditampilkan pada console.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
var j=0;
while(j<5)
{
    console.log("While: Nilai saat ini "+j);
    j++;
}
</script>
</body>
</html>
```

Do While Loop

Do while loop adalah varian dari while loop. Looping ini akan mengeksekusi kode satu kali, sebelum memeriksa apakah kondisinya benar, jika kondisinya benar maka akan mengulang loop.

Sintaks

```
do
{
    //kode yang akan dijalankan
}
while(kondisi);
```

Contoh

```
do
{
    teks += "Ini adalah nomor " + i;
    i++;
}
while(i<10);
```

Do While menjalankan pengulangan, saat kondisi bernilai true atau benar.

Berikut contoh penggunaan do while dengan output ditampilkan pada console.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
var k=0;
do
{
    console.log("Do While: Nilai saat ini "+k);
    k++;
}
while(k<5)
</script>
</body>
</html>
```

Demikian materi tentang perulangan pada Javascript

***Kondisi Pada JavaScript**

Pengertian Kondisi

Pernyataan kondisional digunakan untuk melakukan tindakan yang berbeda berdasarkan pada kondisi yang berbeda. Sangat sering ketika kita menulis kode, kita ingin melakukan tindakan yang berbeda untuk keputusan yang berbeda.

Pada javascript kita bisa menggunakan beberapa sintaks di bawah untuk menjalankan pernyataan kondisional.

If

Else

Else If

Switch

If

Gunakan if untuk menentukan blok kode yang akan dieksekusi jika kondisi yang ditentukan bernilai benar.

if harus diketik dengan huruf kecil, apabila tidak, maka javascript akan menampilkan pesan error.

Sintaks

```
if(kondisi)
```

```
{
```

```
    //kode yang akan dijalankan jika true
```

```
}
```

Contoh

```
if(jam < 10)
```

```
{
```

```
    sambutan = "Selamat Pagi!";
```

```
}
```

Pada contoh di atas, berarti jika jam kurang dari 10, maka sambutan diisi dengan Selamat Pagi!.

Kita coba membuat sebuah contoh lagi dengan menampilkan output melalui console.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
var lampu=true;
if(lampu)
{
console.log("Lampu Menyala");
}
</script>
</body>
</html>
```

Pada contoh di atas, akan menghasilkan output "Lampu Menyala" karena lampu bernilai true.

Image

Sedangkan ketika kita ubah nilai lampu menjadi false, maka output tidak akan mencetak apapun.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
var lampu=false;
if(lampu)
```

```
{  
console.log("Lampu Menyala");  
}  
</script>  
</body>  
</html>
```

Image

Else

Gunakan else untuk menentukan blok kode yang akan dieksekusi jika kondisi yang ditentukan bernilai salah.

SIntaks

```
if(kondisi)  
{  
    //kode yang akan dijalankan jika true  
}  
else  
{  
    //kode yang akan dijalankan jika false  
}
```

Contoh

```
if(jam < 10)  
{  
    sambutan = "Selamat Pagi!";  
}  
else  
{  
    sambutan = "Selamat Malam!";  
}
```

Pada contoh di atas, berarti jika jam kurang dari 10, maka sambutan diisi dengan Selamat Pagi!, jika tidak maka sambutan diisi dengan Selamat Malam!.

Kita coba membuat suatu contoh dengan menambahkan else pada contoh lampu.

```
<!DOCTYPE html>

<html>

<body>

<div id="data"></div>

<script>
var lampu=true;
if(lampu)
{
  console.log("Lampu Menyala");
}
else
{
  console.log("Lampu Padam");
}
</script>
</body>
</html>
```

Pada contoh di atas, akan menghasilkan output "Lampu Menyala" karena lampu bernilai true.

Image

Nah disini perbedaan antara kita hanya menggunakan if dengan kita menambahkan else, pada penambahan else apabila kita isi lampu dengan false, kini akan mencetak "Lampu Padam".

```
<!DOCTYPE html>

<html>

<body>
```

```
<div id="data"></div>

<script>
var lampu=false;
if(lampu)
{
    console.log("Lampu Menyala");
}
else
{
    console.log("Lampu Padam");
}
</script>
</body>
</html>
```

Image

Else If

Gunakan pernyataan else if untuk menentukan kondisi baru jika kondisi pertama salah.

Sintaks

```
if(condition1)
{
    //jalankan jika kondisi 1 true
}
else if(condition2)
{
    //jalankan jika kondisi 1 false dan kondisi 2 true
}
else
{

```

```
    //jalankan jika kondisi 1 false dan kondisi 2 false
}
```

Contoh

```
if(jam < 10)
{
    sambutan = "Selamat Pagi!";
}
else if(jam < 20)
{
    sambutan = "Selamat siang!";
}
else
{
    sambutan = "Selamat Malam!";
}
```

Pada contoh di atas, jika jam kurang dari 10, maka sambutan diisi dengan Selamat Pagi!, jika tidak kurang dari 10 dan jam kurang dari 20 maka sambutan diisi dengan Selamat siang! jika tidak maka sambutan diisi dengan Selamat Malam!.

Kita coba membuat suatu contoh lagi dengan menambahkan else if diantara if dan else.

```
<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
var lampu="merah";
if(lampu=='merah')
{
    console.log("Lampu Merah");
}
```

```

}
else if(lampu=='kuning')
{
    console.log("Lampu Kuning");
}
else if(lampu=='hijau')
{
    console.log("Lampu Hijau");
}
else
{
    console.log("Lampu Padam");
}
</script>
</body>
</html>

```

Output yang dihasilkan akan menyesuaikan dengan nilai / isi dari lampu, yaitu ketika kita isi lampu dengan merah, maka akan mencetak output Lampu Merah, namun jika kita isi dengan kuning maka akan mencetak output Lampu Kuning, jika kita isi dengan hijau maka akan mencetak Lampu Hijau, namun jika selain pilihan di atas maka akan mencetak output Lampu Padam.

Image

Switch

Pernyataan switch digunakan untuk melakukan tindakan yang berbeda berdasarkan pada kondisi yang berbeda. (sama seperti if-else if-else)

Sintaks

```

switch(expression)
{
    case x :

```

```

        //kode
break;
case y:
    //kode
break;
default:
    //kode
}

```

Contoh

```

switch(x)
{
    case 0 :
        text = "Padam";
        break;
    case 1:
        text = "Nyala";
        break;
    default:
        text= "Data Tidak Ada";
}

```

Kita coba membuat suatu contoh dengan menggunakan switch, output yang dihasilkan sama seperti kita menggunakan else if, pada contoh kali ini kita juga akan mencetaknya di browser.

```

<!DOCTYPE html>
<html>
<body>
<div id="data"></div>
<script>
var lampu="merah";
switch(lampu)

```



```
{
  case "merah" :
    console.log("Lampu Merah");
    document.getElementById("data").innerHTML="Lampu Merah";
    break;
  case "kuning":
    console.log("Lampu Kuning");
    document.getElementById("data").innerHTML="Lampu Kuning";
    break;
  case "hijau":
    console.log("Lampu Hijau");
    document.getElementById("data").innerHTML="Lampu Hijau";
    break;
  default:
    text= "Lampu Padam";
}
</script>
</body>
</html>
```

Output akan tercetak pada console dan juga browser sesuai dengan nilai atau isi dari variabel lampu.

Demikian pembahasan mengenai kondisi pada javascript

***Pengertian Array Pada JavaScript**

Pengertian Array

Array merupakan variabel khusus yang dapat menampung lebih dari satu data pada suatu waktu.

Contohnya kita mempunyai daftar item buah-buahan seperti pisang, pepaya, durian dan apel, kita bisa menyimpan semua item tersebut dalam satu variabel, misalnya variabel buah.

Contoh lainnya kita mempunyai daftar item merk mobil seperti Avanza, Agya dan Ayla kita bisa menyimpan daftar item tersebut dalam satu variabel mobil.

Contoh

```
var buah = ["pisang", "pepaya", "durian", "apel"];
```

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

Menggunakan Array di Javascript

Cara Inisialisasi Array

Untuk membuat array pada JavaScript, kita bisa menggunakan sintaks di bawah ini.

Sintaks

```
var nama_array = [item1, item2, item3, ....];
```

atau

```
var nama_array = new Array(item1, item2, item3, ....);
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

```
var mobil = new Array("Avanza", "Agya", "Ayla")
```

Mengakses Data Array (1)

Untuk mengakses data yang ada di dalam array kita bisa menggunakan sintaks di bawah ini.

Sintaks

Untuk menampilkan di console :

```
console.log(nama_array[key]);
```

Untuk menampilkan di browser :

```
document.getElementById('divid').innerHTML=nama_array[key];
```

key merupakan kunci dari elemen array, elemen array pertama memiliki kunci 0, elemen array kedua memiliki kunci 1 begitu seterusnya

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

```
console.log(mobil[0]); // Output : "Avanza"
```

atau

```
document.getElementById('divid').innerHTML=mobil[2];// Output : "Ayla"
```

Untuk mencobanya kita bisa membuat halaman html seperti di bawah.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<div id="divid"></div>
```

```
<script>
```

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

```
console.log(mobil[0]);
```

```
document.getElementById('divid').innerHTML=mobil[2];
```

```
</script>
```

```
</body>
```

```
</html>
```

Dari kode di atas, akan menampilkan output seperti di bawah ini.

Image

Mengubah Data Array

Selain dapat mengakses data, kita juga bisa mengubah data array.

Sintaks

```
nama_array[key]="newItem";
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

Ubah elemen array pertama (key=0) dengan Fortuner.

```
mobil[0]="Fortuner";
```

Maka ketika kita jalankan, outputnya elemen array pertama berubah menjadi Fortuner.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<div id="divid"></div>
```

```
<script>
```

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

```
mobil[0]="Fortuner";
```

```
console.log(mobil[0]);
```

```
</script>
```

```
</body>
```

```
</html>
```

Image

Menghitung Jumlah Array

Kita juga dapat menghitung jumlah array, hal ini sangat berguna ketika kita mempunyai banyak elemen array yang tidak memungkinkan untuk dihitung secara manual.

Sintaks

```
nama_array.length;
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

```
mobil.length; // Output : "3"
```

Untuk menampilkan output dari jumlah array tersebut, kita bisa melihat contoh di bawah ini.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<div id="divid"></div>
```

```
<script>
```

```
var mobil = ["Avanza", "Agya", "Ayla", "Fortuner", "BMW", "Kijang"];
```

```
console.log(mobil.length);
```

```
document.getElementById('divid').innerHTML="Jumlah Elemen Array Mobil  
"+mobil.length;
```

```
</script>
```

```
</body>
```

```
</html>
```

Image

Mengakses Data Array (2)

Sebelumnya kita sudah membahas cara mengakses data dengan key yang sudah ditentukan, kali ini kita akan membahas cara mengakses elemen terakhir dalam array.

Hal ini merupakan aplikasi dari pembahasan tentang menghitung jumlah elemen array sebelumnya.

Untuk mengakses elemen terakhir pada array, kita tinggal memanggil key elemen terakhir (berasal dari jumlah elemen array) dikurangi 1 (karena elemen pertama dimulai dari 0)

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

```
var mobilPertama = mobil[0]; // Output : "Avanza"
```

```
var mobilTerakhir = mobil[mobil.length - 1]; // output : "Ayla"
```

Untuk mencobanya kita bisa menyimak kode sebagai berikut.

```
<!DOCTYPE html>
<html>
<body>
<div id="divid"></div>
<div id="dividx"></div>
<script>
var mobil = ["Avanza", "Agya", "Ayla", "Fortuner", "BMW", "Kijang"];
var mobilPertama = mobil[0];
var mobilTerakhir = mobil[mobil.length - 1];
console.log("Mobil Pertama adalah "+mobilPertama);
console.log("Mobil Terakhir adalah "+mobilTerakhir);
document.getElementById('divid').innerHTML="Mobil Pertama adalah "+mobilPertama;
document.getElementById('dividx').innerHTML="Mobil Terakhir adalah "+mobilTerakhir;
</script>
</body>
</html>
```

Kode di atas akan menampilkan output di console dan di browser sebagai berikut.

Image

Mengubah Array menjadi String (.toString)

Kita dapat mengubah array menjadi string dengan sintaks berikut ini.

```
nama_array.toString();
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
console.log(mobil.toString()); // Output : "Avanza,Agya,Ayla"
```

Untuk lebih jelasnya kita dapat mencobanya dengan menuliskan kode sebagai berikut.

```
<!DOCTYPE html>

<html>

<body>

<div id="divid"></div>

<script>

var mobil = ["Avanza", "Agya", "Ayla"];
var buah = ["pisang", "pepaya", "rambutan"];
console.log("Saya telah memiliki banyak mobil antara lain "+mobil.toString());
document.getElementById('divid').innerHTML="Saya suka buah-buahan "+buah.toString();

</script>

</body>

</html>
```

Mengubah Array menjadi String (.join)

Apabila kita ingin mengganti pemisah dari defaultnya koma dengan yang lainnya, kita bisa menggunakan sintaks berikut ini.

Sintaks

```
nama_array.join(" pemisah_yang_diinginkan ");
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
console.log(mobil.join("*")); // Output : "Avanza * Agya * Ayla"
```

Untuk lebih jelasnya, kita dapat mencobanya dengan menuliskan kode sebagai berikut.

```
<!DOCTYPE html>

<html>

<body>

<div id="divid"></div>
```

```
<script>
var mobil = ["Avanza", "Agya", "Ayla"];
var buah = ["pisang", "pepaya", "rambutan"];
console.log("Saya telah memiliki banyak mobil antara lain "+mobil.join(" * "));
document.getElementById('divid').innerHTML="Saya suka buah-buahan "+buah.join(" dan ");
</script>
</body>
</html>
```

Output dari kode di atas akan tampil seperti di bawah ini.

Image

Mengeluarkan data dari Array (.pop)

Kita bisa mengeluarkan data terakhir dari array dengan menggunakan sintaks di bawah ini.

Sintaks

```
nama_array.pop();
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

Untuk mengeluarkan elemen terakhir dari array mobil, kita bisa menggunakan kode di bawah.

```
mobil.pop(); //mengeluarkan "Ayla" dari array
```

Untuk menyimpan hasil dari pop ke dalam array lain kita bisa menggunakan kode di bawah ini.

```
var mobilku=mobil.pop(); //mengeluarkan "Ayla" dari array ke mobilku
```

Untuk lebih jelasnya, kita dapat mencobanya dengan menuliskan kode sebagai berikut.


```
<!DOCTYPE html>

<html>

<body>

<div id="divid"></div>

<script>
var mobil = ["Avanza", "Agya", "Ayla"];
mobil.pop();
document.getElementById("divid").innerHTML = mobil;
</script>
</body>
</html>
```

Output dari kode di atas akan tampil seperti di bawah ini, yaitu elemen terakhir (Ayla) sudah dikeluarkan.

Image

Memasukkan data ke Array (.push)

Kita bisa memasukkan data ke dalam array dengan menggunakan sintaks di bawah ini.

Sintaks

```
nama_array.push("new_item");
```

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

Contoh kita mempunyai array di atas, kemudian kita hendak memasukkan sebuah item baru ke dalam array tersebut.

```
mobil.push("Fortuner"); //Memasukkan "Fortuner" ke array mobil
```

Untuk lebih jelasnya, kita dapat mencobanya dengan menuliskan kode sebagai berikut.

```
<!DOCTYPE html>
```

```
<html>
<body>
<div id="divid"></div>
<script>
var mobil = ["Avanza", "Agya", "Ayla"];
mobil.push("Fortuner");
console.log("Saya telah memiliki banyak mobil antara lain "+mobil.join(" * "));
</script>
</body>
</html>
```

Output dari kode di atas, item fortuner berhasil dimasukkan ke dalam array mobil.

Image

Menggabungkan dua Array (.concat)

Kita bisa menggabungkan dua array menjadi satu array saja dengan menggunakan sintaks di bawah ini.

Sintaks

```
var Arraygabung=Array1.concat(Array2);
```

Contoh

```
var mobilMahal = ["Fortuner", "Lamborghini"];
```

```
var mobilBudget = ["Avanza", "Agya", "Ayla"];
```

Untuk menggabungkan array mobilMahal dan mobilBudget adalah sebagai berikut.

```
var mobil=mobilBudget.concat(mobilMahal);
```

Untuk lebih jelasnya, kita dapat mencobanya dengan menuliskan kode sebagai berikut.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
<div id="divid"></div>
<script>
var mobilMahal = ["Fortuner", "Lamborghini"];
var mobilBudget = ["Avanza", "Agya", "Ayla"];
var mobil=mobilBudget.concat(mobilMahal);
document.getElementById("divid").innerHTML = mobil;
</script>
</body>
</html>
```

Menggabungkan tiga Array (.concat)

Untuk menggabungkan tiga array, kita bisa melihat sintaks dan contoh di bawah ini.

Sintaks

```
var Arrayabung = array1.concat(array2,array3);
```

Contoh

```
var mobilMahal = ["Fortuner", "Lamborghini"];
var mobilBudget = ["Avanza", "Agya", "Ayla"];
var motor = ["Verza", "CB 150R", "MegaPro"]
```

Untuk menggabungkan ketiga array tersebut kita bisa mengikuti contoh di bawah ini.

```
var kendaraanBermotor = mobilBudget.concat(mobilMahal,motor);
```

Iterasi Data dari Array dengan For

Kita bisa mengakses seluruh array dengan cara looping menggunakan for.

Contoh

```
var mobil = ["Avanza", "Agya", "Ayla"];
```

Untuk looping seluruh elemen dari array mobil, kita bisa menggunakan kode di bawah ini.

```
var teks, i;
teks = "<ul>";
for(i=0; i < mobil.length; i++)
{
    teks += "<li>" + mobil[i] + "</li>";
}
teks += "</ul>";
```

Untuk lebih jelasnya kita bisa membuat kode di bawah ini.

```
<!DOCTYPE html>
<html>
<body>
<div id="divid"></div>
<script>
var mobil = ["Avanza", "Agia", "Ayla"];
var teks, i;
teks = "<ul>";
for(i=0; i < mobil.length; i++)
{
    teks += "<li>" + mobil[i] + "</li>";
}
teks += "</ul>";
document.getElementById("divid").innerHTML = teks;
</script>
</body>
</html>
```

Iterasi Data dari Array dengan .forEach

Kita juga bisa looping menggunakan .forEach seperti berikut ini.

```
var mobil = ["Avanza", "Agya", "Ayla"];
var teks, i;
teks = "<ul>";
mobil.forEach(fungsiMobil)
teks += "</ul>";
function fungsiMobil(value)
{
    teks += "<li>" + value + "</li>";
}
```

Untuk lebih jelasnya kita bisa membuat kode di bawah ini.

```
<!DOCTYPE html>
<html>
<body>
<div id="divid"></div>
<script>
var mobil = ["Avanza", "Agya", "Ayla"];
var teks, i;
teks = "<ul>";
mobil.forEach(fungsiMobil)
teks += "</ul>";
function fungsiMobil(value)
{
    teks += "<li>" + value + "</li>";
}
document.getElementById("divid").innerHTML = teks;
</script>
</body>
```

</html>

Hasilnya sama seperti kita menggunakan for.

Iterasi menggunakan metode .map

Kita bisa menggunakan metode map() untuk membuat array baru dengan melakukan fungsi pada setiap elemen array.

Metode map() tidak menjalankan fungsi untuk elemen array tanpa nilai.

Metode map() tidak mengubah array asli.

Untuk lebih jelasnya kita bisa mencobanya seperti di bawah ini.

<!DOCTYPE html>

<html>

<body>

<div id="divid"></div>

<script>

var nomor1 = [4, 9, 16, 25];

var nomor2 = nomor1.map(kaliDua);

function kaliDua(value,index,array)

{

 return value * 2;

}

console.log(nomor2);

</script>

</body>

</html>

Demikian pembahasan mengenai array pada JavaScript

***Pengertian Object Pada Javascript**

Pengertian Object

Object pada bahasa pemrograman merupakan "abstraksi" dari benda yang ada di dunia nyata. Object dapat digunakan untuk menyimpan data.

Pada JavaScript, object terdiri dari name dan value(variable) atau method(fungsi).

Name dan value disebut sebagai property.

Analogi Object

Kita bisa menganalogikan suatu object mobil mempunyai beberapa name antara lain nama mobil, berat, warna dan tipe.

Masing-masing name tersebut mempunyai value misalnya nama mobil memiliki value Toyota, berat memiliki value 600kg, warna memiliki value putih, tipe memiliki value Agya. Nah name dan value tersebut disebut sebagai property.

Selain dapat menyimpan property, kita juga bisa menyimpan method atau fungsi, misalnya sebuah mobil bisa mempunyai fungsi maju, mundur dan sebagainya.

Inisialisasi Object & Property

Berdasarkan analogi di atas, kita bisa menginisialisasi property sebagai berikut ini.

```
var Mobil = {nama:"Toyota", tipe:"Agya",berat:600, warna:"putih"}
```

Inisialisasi Method

Selain inisialisasi property, kita juga bisa menginisialisasi method di dalam object.

```
var Mobil =
```

```

{
  nama:"Toyota",
  tipe:"Agya",
  berat:600,
  warna:"putih",
  maju : function() {
    return "Mobil Melaju";
  },
  mundur : function()
  {
    return "Mobil Mundur";
  }
}

```

Pada contoh di atas, selain menginisialisasi property kita juga menginisialisasi fungsi maju dan mundur pada object Mobil.

Mengakses Property dan Method pada Object

Untuk mengakses method pada object kita bisa menggunakan contoh sintaks seperti di bawah ini.

```
nama_object.nama_method();
```

Contoh

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p id="data"></p>
```

```
<script>
```

```
var Mobil =
```

```

{
  nama:"Toyota",

```



```
tipe:"Agya",  
berat:600,  
warna:"putih",  
maju : function() {  
    return "Mulai Melaju";  
},  
mundur : function()  
{  
    return "Mulai Mundur";  
}  
}
```

```
document.getElementById("data").innerHTML = Mobil.nama + " " + Mobil.maju();  
</script>  
</body>  
</html>
```

Nah demikian sedikit pembahasan tentang object pada JavaScript.