



[Beranda](#) / [Academy](#) / [Belajar Dasar Pemrograman Web](#) / Arrays dan Objek

**! Pembaruan!** Modul ini dibuat pada tanggal 6 December 2019. Pembaruan terakhir adalah: **Konversi potongan kode menjadi kode interaktif..**

[Lihat riwayat »](#)

## Arrays dan Objek

Kita sudah mengenal tipe data dasar yang ada pada JavaScript. Kali ini kita akan membahas mengenai array dan objek pada JavaScript. Keduanya dapat menampung lebih dari satu tipe data dasar yang digunakan untuk mengelola sebuah data.

### Arrays

Array merupakan tipe data yang dapat mengelompokkan lebih dari satu nilai dari tipe data lain dengan menempatkannya pada satu variabel. Contoh:

main.js +

```
1 let myArray = ["Coklat", 42.5, 22, true, "Programming"];
2 console.log(myArray);
3
4 /* output:
5 [ 'Coklat', 42.5, 22, true, 'Programming' ]
6 */
```

INPUT

RESET

JALANKAN

Output:

Nilai - nilai yang berada pada array disusun dan diakses secara *indexing*. Untuk mengakses nilai di dalam array kita gunakan tanda kurung siku `[]` yang di dalamnya berupa angka yang merupakan posisi nilai yang ingin diakses.

main.js +

```
1 let myArray = ["Coklat", 42.5, 22, true, "Programming"];
2 console.log(myArray[1]);
3
4 /* output:
5 42.5
6 */
```





INPUT

RESET

▶ JALANKAN

Output:

Yang perlu kita ketahui adalah nilai index dimulai dari angka 0. Terlihat pada contoh kode di atas, kita mengakses index ke-1 pada `myArray`. Nilai yang muncul adalah nilai ke-2 dari array tersebut, yakni 42.5. Jika kita mengakses nilai array lebih dari *index*-nya maka hasilnya akan `undefined`. Index terakhir array selalu jumlah nilai array - 1.

main.js +

```
1 let myArray = ["Coklat", 42.5, 22, true, "Programming"];
2 console.log(myArray[0]);
3 console.log(myArray[1]);
4 console.log(myArray[2]);
5 console.log(myArray[3]);
6 console.log(myArray[4]);
7 console.log(myArray[5]);
8 console.log("Panjang nilai myArray adalah " + myArray.length + ".");
9
10 /* output:
11 Coklat
12 42.5
13 22
```

INPUT

RESET

▶ JALANKAN

Output:

## Objek

Objek serupa dengan array yang dapat menampung banyak nilai dengan tipe data yang beragam. Untuk mengelola data menggunakan objek, bedanya objek diakses tidak melalui *indexing*, melainkan menggunakan pendekatan *key-value*. Untuk mengakses nilainya kita gunakan *key*. *Key* juga biasa disebut dengan properti.

Untuk menetapkan objek pada variabel gunakan tanda kurung kurawal { } dalam menginisialisasinya. Kemudian di dalamnya kita tetapkan **key: value**.



```
1. let object = {key1: "value1", key2: "value2", key3: "value3"}
```

Dalam menentukan nama *key*, gunakanlah nama yang dapat mendeskripsikan dari *value*-nya. Pada *value*, kita dapat mengisi nilai dengan tipe data apapun, termasuk array. Contoh:



```
1. let user = {firstName: "Harry", lastName: "Potter", age: 20, isMuggle: false, stuff: ["Wand", "Flying Car", "Owl"]};
```

Dalam menuliskan objek, baris baru tidaklah penting dan tidak akan berpengaruh apa pun. Sehingga lebih baik menetapkan *key-value* buatlah baris baru untuk memisahkan antar nilainya, hal ini akan memudahkan kita dalam struktur data yang berada pada objek.



DIBANTU



```
2.   firstName: "Harry",
3.   lastName: "Potter",
4.   age: 20,
5.   isMuggle: false,
6.   stuff: ["Magic Wind", "Flying Car", "Owl"]
7.  };
```

Kemudian untuk mengakses nilai dari properti objek kita dapat gunakan tanda titik diikuti dengan nama *properti*-nya. Contoh:

main.js +

```
1 let user = {
2   firstName: "Harry",
3   lastName: "Potter",
4   age: 20,
5   isMuggle: false,
6   stuff: ["Magic Wind", "Flying Car", "Owl"]
7 };
8
9 console.log("Hallo, nama saya " + user.firstName + " " + user.lastName);
10 console.log("Umur saya " + user.age + " tahun");
11
12 /* output
13 Hallo, nama saya Harry Potter
```

INPUT

RESET

JALANKAN

Output:

Bahkan dalam properti objek, kita dapat menyimpan nilai objek lainnya. Contohnya properti `firstName` dan `lastName` dapat dikelompokkan kembali dalam sebuah objek baru sebagai berikut:

```
1. let user = {
2.   name: {
3.     first: "Harry",
4.     last: "Potter"
5.   },
6.   age: 20,
7.   isMuggle: false,
8.   stuff: ["Magic Wind", "Flying Car", "Owl"]
9. }
```

Untuk mendapatkan nilainya kita perlu mengakses properti dari objek `user` kemudian `name`. Sehingga penulisannya menjadi seperti berikut:

main.js +

```
1 let user = {
2   name: {
3     first: "Harry",
4     last: "Potter",
5   },
```



DIBANTU



```
9  };
10
11 console.log("Hallo, nama saya " + user.name.first + " " + user.name.last);
12 console.log("Umur saya " + user.age + " tahun");
```

☐ INPUT

RESET

JALANKAN

Output:

Mungkin seperti itulah gambaran mengenai objek di JavaScript, penting untuk Anda garis bawahi bahwa dalam mengelola banyak tipe data pada JavaScript, Anda dapat menggunakan array maupun objek. Pembahasan di atas cukup sebagai landasan pengetahuan mengenai dua hal itu ya.

Banyak hal sebenarnya yang dapat diceritakan tentang dua hal ini, terutama untuk objek. Jika Anda ingin tahu lebih dalam, Anda bisa baca dokumentasinya pada tautan yang disediakan oleh MDN:

- Array : [https://developer.mozilla.org/id/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/id/docs/Web/JavaScript/Reference/Global_Objects/Array)
- Objek : [https://developer.mozilla.org/id/docs/Web/JavaScript/Reference/Global\\_Objects/Object](https://developer.mozilla.org/id/docs/Web/JavaScript/Reference/Global_Objects/Object)

KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI BERIKUTNYA



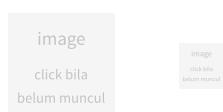
Dicoding Space  
Jl. Batik Kumeli No.50, Sukaluyu,  
Kec. Cibeunying Kaler, Kota Bandung  
Jawa Barat 40123



## Decode Ideas Discover Potential

[Tentang Kami](#)[Blog](#)[Reward](#)[Showcase](#)[Hubungi Kami](#)[FAQ](#)

### Penghargaan



© Copyright Dicoding Indonesia 2021

[Terms](#) • [Privacy](#)

