



[Beranda](#) / [Academy](#) / [Belajar Dasar Pemrograman Web](#) / Variabel

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

[Lihat riwayat »](#)

Variabel

Ketika mendengar variabel apakah Anda teringat tentang pelajaran matematika? Yups, variabel di sini memiliki konsep yang sama. Variabel umumnya digunakan untuk menyimpan informasi atau nilai yang akan dikelola di dalam sebuah program.

Sebelum ECMAScript 2015 (ES6) untuk membuat variabel pada JavaScript kita gunakan keyword **var**.

```
1. var firstName = "Harry";
```

Tanda sama dengan (=) digunakan untuk menginisialisasikan nilai pada variabel, sehingga sekarang variabel **firstName** memiliki nilai teks “Harry”.

Kita bisa menggunakan apapun yang kita mau untuk menamai sebuah variabel, tetapi pastikan penamaannya masih masuk akal dengan konteksnya agar berikutnya kode mudah di-*maintenance*.

Sebaiknya hindari penamaan variabel dengan istilah umum seperti “**data**”, Gunakanlah penamaan variabel yang dapat mendeskripsikan nilai dari variabel itu sendiri. Berikut beberapa aturan dalam penamaan variabel yang perlu Anda ketahui:

- Harus dimulai dengan huruf atau underscore (_).
- Dapat terdiri dari huruf, angka, dan underscore (_) dengan berbagai kombinasi.
- Tidak dapat mengandung spasi (*white space*), jika penamaan variabel lebih dari dua kata maka tuliskan secara *camelCase*. Contoh **firstName**, **lastName**, **catName**, dll.
- Tidak dapat mengandung karakter spesial (!, ., / \ + * = dll.)

Nilai variabel yang diinisialisasi menggunakan **var** dapat diubah kembali nilainya, contoh:

main.js +

```
1 var firstName = "Harry";
2 console.log(firstName);
3
4 firstName = "Ron";
5 console.log(firstName);
6
7 /* output:
8 Harry
9 Ron
10 */
```





Output.

Sejak ECMAScript 2015 (ES6) selain **var**, menginisialisasikan variabel dapat menggunakan **let** dan **const**. ES6 melakukan improvisasi pada deklarasi variabel karena menggunakan var terdapat beberapa hal yang kontroversial, salah satunya *hoisting*.

Apa itu *hoisting*? sesuai artinya “Mengangkat” variabel yang dideklarasikan menggunakan var ini dapat diinisialisasi terlebih dahulu sebelum dideklarasikan, Contoh:

main.js +

```
1 x = 100;
2 var x;
3 console.log(x);
4
5 /* output: 100 */
```

☐ INPUT

RESET

JALANKAN

Output:

Ini dikarenakan proses *hoisting*, sebenarnya di belakang layar deklarasi variabel **x** diangkat ke atas sehingga kode yang tampak seperti ini:

main.js +

```
1 var x;
2 x = 100;
3 console.log(x);
4
5 /* output: 100 */
```

☐ INPUT

RESET



DIBANTU

Output:



Hoisting menjadi kontroversial karena tidak sedikit developer yang dibuat bingung akan hal ini. Masalah ini sudah terselesaikan jika kita menggunakan **let** dalam mendeklarasikan variabel.

main.js +

```
1 y = 100;
2 let y;
3 console.log(y);
4
5 /* ReferenceError: Cannot access 'y' before initialization */
```

INPUT

RESET

JALANKAN

Output:

Kemudian **const** digunakan untuk mendeklarasikan sebuah variabel yang sifatnya *immutable* atau tidak perlu diinisialisasi kembali. Jika kita menginisialisasi kembali nilai variabel yang menggunakan **const**, maka akan mendapati error “*TypeError: Assignment to constant variable.*”

main.js +

```
1 const z = 100;
2 console.log(z);
3
4 z = 200;
5 console.log(z)
6
7 /* TypeError: Assignment to constant variable. */
```

INPUT

RESET

JALANKAN

Output:

Dengan begitu sebaiknya kita gunakan **let** atau **const** ketika mendeklarasikan variabel daripada menggunakan **var**.

← KEMBALI KE MATERI SEBELUMNYA

LANJUTKAN KE MATERI SELANJUTNYA



DIBANTU



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

