**Pengenalan Kelas**

Selamat datang di kelas Belajar Dasar Pemrograman JavaScript!

Pada kelas ini kita akan belajar hal-hal fundamental yang perlu Anda ketahui tentang bahasa pemrograman JavaScript pada lingkungan Node JS. Untuk penggunaan JavaScript pada lingkungan browser, Anda dapat mempelajarinya pada kelas [Belajar Dasar Pemrograman Web](https://www.dicoding.com/academies/123).

Materi yang tersaji ramah untuk pemula, sehingga tidak ada prasyarat pemrograman yang perlu Anda kuasai untuk belajar di kelas ini.

Beberapa hal yang akan dipelajari pada kelas ini, antara lain:

* **Introduction to JavaScript** : Mengenal apa itu JavaScript, mulai dari sejarah hingga alasan untuk mulai mempelajarinya.
* **JavaScript Fundamentals** : Materi fundamental terkait JavaScript, seperti *statement*, *expressions*, variabel, tipe data, dan *control flow*.
* **Data Structure** : Menjelaskan bagaimana menyimpan dan mengelola data yang lebih kompleks menggunakan Object, Array, Map, Set, dsb.
* **Function** : Mempelajari bagaimana mengelompokkan kode menjadi fungsi yang dapat dipanggil ulang.
* **Object-Oriented Programming** : Memahami konsep OOP pada JavaScript, bagaimana membuat class, properti, dan method dari class.
* **Functional Programming** : Memahami konsep *functional programming* seperti perilaku fungsi, *anonymous function*, dan *higher-order*.
* **Setup Environment** : Proses instalasi lingkungan pengembangan seperti Node JS dan IDE pada perangkat yang Anda miliki.
* **Module** : Mempelajari bagaimana beberapa berkas JavaScript dapat saling berkomunikasi dan berbagi data satu sama lain.
* **Error Handling** : Mempelajari bagaimana menangani kegagalan yang mungkin muncul dalam program, sehingga tidak berdampak kepada pengguna.
* **Concurrency** : Mempelajari bagaimana menangani dan membuat proses *asynchronous* pada JavaScript.
* **Node Package Manager** : Mengetahui beberapa *library* yang penting untuk ditambahkan ke dalam proyek aplikasi JavaScript.
* **JavaScript Testing** : Mempelajari bagaimana melakukan pengujian otomatis pada kode JavaScript.

Setelah mengetahui materi apa saja yang akan dipelajari, mari kita mulai melangkah ke materi pertama yaitu Pengenalan JavaScript. *Let’s go!*

## Mekanisme Belajar

Selamat datang di Dicoding Academy. Sebelum memulai belajar di kelas ini, Anda perlu tahu tahapan dan cara belajar beserta fasilitas yang tersedia agar proses belajar lebih efektif.

### Materi Pembelajaran

* **Materi Bacaan Elektronik**  
  Materi yang ada dalam kelas ini mayoritas berupa teks atau tulisan. Mengapa demikian? Karena kami menemukan bahwa dalam proses belajar di bidang pemrograman, bentuk materi pembelajaran yang paling efektif untuk diingat, dimengerti, dan yang terpenting, dipraktikkan, adalah dalam bentuk teks.
* **Forum Diskusi**  
  Setiap kelas memiliki sebuah forum diskusi yang dapat Anda gunakan untuk bertanya (dan menjawab) mengenai materi kelas. Instruktur kami yang terdiri dari Curriculum Developer dan Academy Code Reviewer siap membantu Anda melalui forum diskusi ini. Tak hanya kami, Anda pun dapat berpartisipasi. Untuk meningkatkan retensi ilmu yang Anda punya, berbagi adalah salah satu kuncinya. Jadi, silakan aktif dan saling membantu di dalam forum.  
    
  Jika Anda menemui kendala perihal situs Dicoding dan administrasi kelas, tanyakan kepada kami melalui tombol chat di kanan bawah layar Anda ketika membuka situs Dicoding. Jangan gunakan forum diskusi ya karena itu salah alamat.

### Evaluasi Pembelajaran

* **Ujian (Kuis, Ujian Akhir Kelas, Knowledge Check)**  
  Terdapat berbagai jenis ujian di kelas guna mengecek pemahaman Anda pada materi pembelajaran. Ketersediaan ujian beserta jenis ujian yang tersedia pada setiap kelas dapat berbeda-beda. Setiap pertanyaan dalam ujian pasti mencakup materi yang telah dibahas. Sehingga jika ada pertanyaan yang tidak dapat Anda jawab, pastikan Anda mengulang kembali pembelajaran.

## Forum Diskusi

### Ekspektasi

Dengan banyaknya jumlah siswa Dicoding Academy, kami tidak mengharapkan siswa untuk membaca semua diskusi atau komentar pada forum diskusi. Sebaliknya, bacalah hal-hal yang menurut Anda menarik dan dapat membantu Anda dalam menyelesaikan kelas. Lebih baik lagi, jika Anda dapat membantu siswa lainnya dengan memberikan jawaban di bidang yang Anda pahami. Berbagi dalam forum diskusi ini dapat pula membantu meningkatkan retensi ilmu Anda.

Peraturan paling penting adalah bersikap sopan dan memperlakukan semua siswa lain dan instruktur, dengan hormat. Pelanggaran etika ini dapat berakibat pada dikeluarkannya Anda dari kelas.

Anda akan memiliki akses ke forum diskusi kelas selama Anda aktif terdaftar ke kelas ini. Saat masa belajar Anda sudah habis, maka forum diskusi tidak akan dapat diakses. Namun, jika Anda sukses menyelesaikan kelas (lulus dan sampai mendapatkan sertifikat kompetensi dari Dicoding), maka Anda tetap dapat mengakses forum diskusi kelas ini walau masa belajar Anda untuk kelas ini telah habis.

### Sebelum Mulai Bertanya

1. Sebelum membuat diskusi baru, gunakan fitur pencarian untuk cek diskusi lama terlebih dahulu. Problem Anda mungkin pernah dibahas dan dijawab solusinya, dalam forum. Dengan begitu, Anda pun tidak perlu menghabiskan waktu untuk bertanya dan untuk menunggu jawaban kembali.
2. Namun jika tidak menemukan diskusi lama dengan topik yang sama dengan masalah Anda, silakan membuat diskusi baru. Simak cara dan langkah berikut ini: <https://help.dicoding.com/academy-dicoding/aturan-membuat-pertanyaan-di-forum-diskusi-kelas/>

### Fitur Bermanfaat

* **Mencari Posting**  
  Anda dapat mencari topik atau diskusi menggunakan tiga cara yang tersedia pada halaman utama forum diskusi.
  1. Gunakan fitur “Cari Judul Diskusi” untuk mencari topik berdasarkan judul atau pertanyaannya.
  2. Gunakan fitur “Pilih Berdasarkan Modul” untuk melihat diskusi berdasarkan modul kelasnya.
  3. Gunakan fitur “Keyword Populer” untuk melihat diskusi berdasarkan Keyword yang paling sering ditanyakan.
* **Upvote dan Downvote Komentar pada Diskusi**  
  Saat membuka sebuah diskusi, Anda akan menemukan komentar beserta jawaban dari siswa dan instruktur di bagian bawah. Pada setiap komentar terdapat tombol panah atas (˄) dan panah bawah (˅). Klik tombol panah atas (˄) jika Anda merasa komentar atau jawaban tersebut bijaksana, menarik, dan bermanfaat. Gunakan tombol panah bawah (˅) jika Anda merasa komentar atau jawaban tersebut tidak sopan, tidak membantu, atau bahkan tidak tepat.  
    
  Gunakan Fitur Upvote dan Downvote ini seperti arahan di atas, sehingga kualitas komentar dan jawaban pada forum diskusi dapat terjaga dengan baik.
* **Approved Answer**  
  Jika terdapat komentar/tanggapan berupa solusi terbaik dari permasalahan yang Anda tanyakan, maka tandai komentar tersebut sebagai jawaban terpilih. Caranya dengan menekan tombol “Jawaban Terpilih” di sisi kanan atas komentar.  
    
  Fitur ini juga akan mempermudah siswa lainnya dengan pertanyaan yang sama untuk menemukan titik terang solusinya.

### Mengenal Teman Sekelas Anda

Berhubungan baik dengan teman sekelas dan instruktur merupakan bagian penting dari kelas ini karena mereka dapat membantu jika Anda mengalami kendala dalam pemahaman materi.

Oleh karena itu, kami ingin Anda meluangkan waktu untuk mencairkan suasana dan saling mengenal. Membangun interaksi dengan siswa lain akan membuat pengalaman belajar Anda jauh lebih menyenangkan dan menarik.

Beberapa hal yang dapat Anda tulis pada perkenalan diri:

* Siapa Anda dan dari mana Anda berasal?
* Apa pekerjaan atau pendidikan Anda saat ini?
* Kenapa Anda mengambil pelatihan ini? Apakah mungkin karena Anda sedang mengejar perubahan dalam karir, atau lainnya?

|  |
| --- |
| Ayo aktif di forum diskusi kelas ini!  Yuk mulai dengan memperkenalkan diri Anda pada thread Perkenalan Diri berikut:  <https://www.dicoding.com/academies/256/discussions/88855> |

## Glosarium

### C

**Compiler**

Program komputer yang menerjemahkan kode yang ditulis dalam satu bahasa pemrograman ke bahasa lain yang dimengerti oleh mesin.

### D

**Debugging**

Proses mengidentifikasi dan menghilangkan error pada aplikasi.

### I

**Interpreter**

Program untuk menerjemahkan setiap baris kode dari bahasa pemrograman menjadi bahasa yang dimengerti oleh mesin secara langsung. Tanpa melalui proses kompilasi.

### L

**Library**

Sekumpulan sumber daya, biasanya berupa kode atau program, yang memiliki fungsionalitas tertentu dan bisa digunakan pada program lain.

### R

**Runtime**

Lingkungan, platform, atau sumber daya yang menjalankan suatu kode/program.

### S

**String Concatenation**

Proses penggabungan dua String atau lebih menggunakan operator penjumlahan (+).

**String Interpolation**

Proses mengganti placeholder di dalam String dengan nilai dari variabel tertentu.

## Daftar Referensi

[1]  Indeed. “Gaji di Indonesia - gaji rata-rata” id.indeed.com. <https://id.indeed.com/career/front-end-developer/salaries?from=top_sb> (diakses pada 2 Juni 2022)

[2]  Indeed. “Gaji di Indonesia - gaji rata-rata” id.indeed.com. <https://id.indeed.com/career/back-end-developer/salaries?from=top_sb> (diakses pada 2 Juni 2022)

[3] D. Flanagan, "The JavaScript Standard Library," dalam JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, Inc., 2020. Available: O'Reilly

[4] D. Flanagan, "Introduction to JavaScript," dalam JavaScript: The Definitive Guide, 7th Edition. O'Reilly Media, Inc., 2020. Available: O'Reilly

[5] S. Amarasinghe. "Reading 17: Concurrency". web.mit.edu. <http://web.mit.edu/6.005/www/fa14/classes/17-concurrency/> (diakses pada 22 Maret 2021)

[6] M. Lee. "What's the difference between a tilde (~) and a caret (^) in a npm package.json file?". michaelsoolee.com. <https://michaelsoolee.com/npm-package-tilde-caret/> (diakses pada 26 Maret 2021)

## Pengenalan JavaScript

Sebelum mulai belajar suatu hal yang baru, penting bagi Anda untuk memahami apa yang akan dipelajari dan apa yang bisa dilakukan dengannya. Dalam hal ini kita akan mulai mengenal JavaScript.

Pada akhir modul ini, Anda diharapkan dapat:

* Mengetahui apa itu JavaScript
* Mengetahui sejarah singkat dan perkembangan JavaScript
* Mengerti beberapa alasan untuk mulai mempelajari JavaScript

## Apa itu JavaScript

JavaScript adalah bahasa pemrograman tingkat tinggi yang pada awalnya dikembangkan untuk membuat website menjadi lebih “hidup”. Bersama dengan HTML dan CSS, JavaScript menjadi bahasa pemrograman paling populer untuk mengembangkan aplikasi berbasis web. Bahasa ini mampu memberikan logic ke dalam website, sehingga website tersebut memiliki fungsionalitas tambahan dan lebih interaktif.

Awalnya JavaScript dibuat supaya dapat berjalan di lingkungan browser dan membuat website menjadi lebih interaktif. Namun, saat ini Anda sebagai developer dapat menggunakan bahasa pemrograman JavaScript di berbagai lingkungan pengembangan. Sehingga, tidak hanya sebatas browser/client, namun JavaScript juga bisa berjalan di server menggunakan Node.js.

JavaScript termasuk ke dalam kategori scripting language. Apa maksudnya? Salah satu ciri-ciri utama dari bahasa scripting adalah kode tidak perlu dikompilasi agar bisa dijalankan. Scripting language menggunakan interpreter untuk menerjemahkan kode atau perintah yang kita tulis supaya dimengerti oleh mesin.

Itulah kenapa bahasa scripting tidak membutuhkan banyak kode yang perlu ditulis agar sebuah program bisa dijalankan. Hanya dengan satu baris kode berikut Anda sudah bisa membuat program yang menampilkan teks “Hello, World!” ke layar.

1. console.log(“Hello, World!”);

Sederhana, bukan? Itulah salah satu ciri utama JavaScript sebagai scripting language. Tentu tidak lengkap rasanya ketika kita sudah mengetahui apa itu JavaScript, tetapi tidak tahu sejarah di baliknya. Sejarah singkat dari JavaScript dapat Anda simak di materi berikutnya.

## Sejarah JavaScript

JavaScript dibuat pada tahun 1995 oleh Brendan Eich, programmer dari Netscape. Bahasa ini awalnya dinamai “Mocha” kemudian berubah menjadi “LiveScript”. Pada saat itu bahasa Java merupakan bahasa pemrograman yang paling populer. Untuk memanfaatkan kepopulerannya, nama LiveScript pun diubah menjadi “JavaScript”. Jadi, meskipun namanya mirip, JavaScript tidak ada hubungannya dengan bahasa pemrograman Java.

Setelah diadopsi di luar Netscape, JavaScript distandarisasi oleh European Computer Manufacturers Association (ECMA). Sejak saat itu JavaScript juga dikenal dengan ECMAScript. Meskipun begitu, masih banyak yang menyebutnya dengan JavaScript hingga saat ini.

Terdapat beberapa versi JavaScript yang sudah distandarisasi oleh ECMAScript. Pada tahun 2000 hingga 2010, ECMAScript 3 merupakan versi yang banyak digunakan ketika JavaScript sedang mendominasi. Selama waktu tersebut, ECMAScript 4 sedang dalam proses pengembangan dengan harapan akan memberikan improvisasi yang cukup signifikan. Namun, ambisi tersebut tidak berjalan mulus sehingga pada tahun 2008 pengembangan ECMAScript dihentikan.

Walaupun begitu, ini bukan akhir dari JavaScript. Pengembangan berganti menjadi ECMAScript 5 dengan mengurangi ambisinya dan hanya melakukan perbaikan pada hal yang tidak jadi kontroversi. Pembaruan tersebut berhasil dan akhirnya ECMAScript 5 rilis pada tahun 2009.

Lalu, pada tahun 2015 ECMAScript 6 rilis dengan membawa perubahan yang cukup besar termasuk ide-ide yang sudah direncanakan untuk versi 4. Sejak saat itu, tiap tahun JavaScript melakukan update bersifat minor.

## Kenapa JavaScript

Jadi, kenapa kita perlu mempelajari JavaScript?

Alasan utamanya karena JavaScript merupakan bahasa yang penting untuk Anda kuasai jika ingin menjadi web developer, baik itu front-end maupun back-end.

Berikut ini adalah beberapa kelebihan dari JavaScript yang dapat Anda pertimbangkan sebelum mulai mempelajari JavaScript:

* **JavaScript bahasa yang versatile**  
  JavaScript bisa berjalan di lingkungan browser, server, bahkan desktop. Artinya, jika Anda bisa menguasai bahasa ini, maka skill Anda bisa digunakan di mana pun.
* **Mudah dipelajari oleh pemula**  
  JavaScript termasuk salah satu bahasa pemrograman yang ramah bagi pemula. Anda tidak perlu menginstal software dan lingkungan pengembangan lain yang rumit untuk memulai membuat program dengan JavaScript. Cukup dengan browser Anda sudah bisa menulis kode JavaScript dan menjalankannya sekaligus.  
    
  Selain itu, sebagai salah satu bahasa pemrograman paling populer, JavaScript memiliki komunitas yang besar pada situs seperti [StackOverflow](https://stackoverflow.com/questions/tagged/javascript" \t "_blank) yang siap membantu Anda jika memiliki pertanyaan atau kesulitan dalam JavaScript.
* **Potensi karir yang meyakinkan**  
  Mengikuti perkembangan teknologi dan banyaknya bisnis yang mulai merambah ke ranah digital, JavaScript menjadi salah satu skill yang paling banyak dicari di industri. Jika Anda mencari kata kunci “JavaScript” pada laman pencarian kerja seperti JobStreet, akan muncul hampir 1.500 lowongan pekerjaan di Indonesia yang bisa Anda lamar.  
    
  Bukan hanya ramai peminat, pekerjaan yang berkaitan dengan JavaScript juga dihargai cukup tinggi. Menurut data yang diambil dari situs id.indeed.com rata-rata gaji seorang Front End Developer adalah Rp 7.300.000 per bulan [[1](https://id.indeed.com/career/front-end-developer/salaries?from=top_sb)] dan untuk Back End Developer adalah Rp 9.200.000 per bulan [[2](https://id.indeed.com/career/back-end-developer/salaries?from=top_sb)].

Bagaimana menurut Anda? Apakah setelah mengetahui beberapa kelebihan dari JavaScript membuat Anda semakin semangat untuk mulai mempelajarinya? Tunggu apalagi, yuk kita mulai belajar JavaScript!

## Rangkuman Materi

Sebelum mempelajari sesuatu tentunya penting untuk mengetahui apa yang akan kita pelajari serta bagaimana sejarahnya dan bagaimana perkembangannya.

Kita telah membahas beberapa hal di modul ini, antara lain:

* JavaScript dibuat dengan tujuan awal agar website menjadi lebih interaktif.
* JavaScript termasuk ke dalam kategori scripting language, sehingga kode tidak perlu dikompilasi untuk bisa dijalankan. Terdapat interpreter untuk menerjemahkan kode kita agar bisa dimengerti oleh mesin.
* Terdapat dua lingkungan umum untuk menjalankan JavaScript, yaitu browser dan Node.js
* JavaScript dikembangkan dengan standar ECMAScript. Update besar terakhir tersaji dalam versi ES6 pada tahun 2015. Sejak saat itu, tiap tahun JavaScript melakukan update bersifat minor.

## Video Pengenalan JavaScript

Untuk memperdalam dan mempermudah pemahaman pada materi ini, Anda dapat menyimak video pembahasannya berikut.

<https://youtu.be/Uwvl-X8wiYc>

### Materi Pendukung

Ulas beberapa sumber di bawah ini untuk mempelajari lebih lanjut tentang JavaScript:

* [JavaScript MDN Web Docs](https://developer.mozilla.org/en-US/docs/Web/JavaScript)
* [An Introduction to JavaScript](https://javascript.info/intro)
* [Introduction to Node.js](https://nodejs.dev/en/learn/)
* [Differences between Node.js and the Browser](https://nodejs.dev/learn/differences-between-nodejs-and-the-browser)

## Pengenalan Fundamentals JavaScript

Setelah kita mengetahui apa itu JavaScript dan sejarah singkat di baliknya, pada modul ini Anda akan mulai menuliskan kode JavaScript pertama. Sudah tidak sabar, bukan? Yuk, kita mulai belajar.

Pada akhir modul ini, Anda diharapkan dapat:

* Menuliskan kode JavaScript dan menampilkannya ke konsol
* Memahami penggunaan komentar pada kode
* Memahami konsep variabel pada JavaScript
* Memahami beberapa jenis tipe data pada JavaScript
* Mengenal operator pada JavaScript dan bagaimana menggunakannya
* Memahami konsep percabangan pada pemrograman
* Memahami konsep perulangan pada pemrograman

## Menulis Kode JavaScript Pertama

Menulis kode dalam suatu bahasa pemrograman sederhananya adalah menuliskan instruksi-instruksi untuk dijalankan oleh komputer. Kode di bawah ini merupakan instruksi bagi terminal atau konsol untuk mencatat (log) kalimat “Hello, World!”.

1. console.log(“Hello World!”);

console.log adalah kode bawaan JavaScript untuk menampilkan pesan ke konsol, bisa berupa web konsol atau konsol dari terminal/command prompt.

[](https://www.dicoding.com/academies/256/tutorials/13863?from=15313)

Kode atau teks yang berada di dalam tanda kurung adalah pesan yang ingin ditampilkan. Pada contoh kode di atas, kita menggunakan tanda kutip (“”) untuk menandakan bahwa pesan yang ingin ditampilkan merupakan sebuah string atau teks. Kita akan membahas mengenai tipe data pada beberapa modul ke depan. Saat ini kita akan fokus untuk menampilkan teks terlebih dulu. Sekarang cobalah untuk menampilkan nama Anda sendiri ke konsol.

* [**main.js**](https://www.dicoding.com/academies/256/tutorials/13863?from=15313#run1-editor1)



1

console.log("Hello World!");

2

​

3

// Tampilkan nama Anda ke layar pada kode di bawah:

 Input  Reset

 Jalankan

JavaScript membagi instruksi secara umum menjadi dua jenis, yaitu expression dan statement.

Expression adalah unit kode yang dapat dievaluasi menjadi suatu nilai. Untuk lebih mudahnya, perhatikan contoh kode berikut:

1. 5; // Baris kode ini merupakan expression karena interpreter akan membaca kode ini dan menghasilkan nilai 5.
2. 2 + 3; // Baris kode ini juga merupakan expression. Interpreter mengevaluasi kode dan juga akan menghasilkan nilai 5.

Sementara itu, statement adalah instruksi untuk melakukan sebuah aksi tertentu. Aksi ini bisa termasuk membuat variabel, melakukan perulangan, pengecekan kondisi, dll.

1. var yourName;
2. let yourAge;
3. const numberOfDays;  // declaration statement

Jika Anda masih bingung untuk membedakan expression dan statement, tidak perlu khawatir. Saat ini belum banyak kode yang bisa kita jadikan contoh. Namun, konsep ini akan banyak kita gunakan pada materi-materi yang akan datang. Yang perlu Anda ingat adalah:

* Expression merupakan kode yang menghasilkan nilai.
* Statement menunjukkan aksi yang dilakukan.

## Comments

Pada materi sebelumnya kita telah membahas bahwa kode yang kita tulis merupakan kumpulan instruksi yang harus dijalankan oleh komputer. Salah satu instruksi yang penting adalah memberi tahu komputer untuk mengabaikan perintah yang kita tulis. Instruksi yang ditulis dalam suatu program tetapi tidak dijalankan oleh komputer disebut “comments”.

Sebuah komentar akan dilewatkan oleh interpreter atau compiler, sehingga tidak akan memengaruhi alur program yang kita tulis. Komentar ini bisa digunakan sebagai dokumentasi atau penjelasan dari kode yang kita tulis.

Terdapat dua metode untuk memberikan komentar. Pertama, untuk memberikan komentar pada satu baris saja, kita bisa gunakan tanda dua garis miring (//) di awal baris.

* [**main.js**](https://www.dicoding.com/academies/256/tutorials/13867?from=13863#run1-editor1)



1

// Ini merupakan komentar satu baris, kode di bawah tidak akan dijalankan

2

​

3

// console.log("Halo!");

 Input  Reset

 Jalankan

Lalu, untuk memberikan komentar lebih dari satu baris kita bisa menggunakan tanda /\* sebagai pembuka komentar dan tanda \*/ untuk penutup komentar. Teks apa pun yang berada di antara tanda tersebut akan dijadikan komentar dan tidak akan dieksekusi.

* [**main.js**](https://www.dicoding.com/academies/256/tutorials/13867?from=13863#run2-editor1)



1

/\* Ini merupakan komentar dengan lebih dari satu baris

2

Teks apapun yang berada disini akan dijadikan komentar.

3

Ketika menggunakan ini, jangan lupa untuk menutupnya.

4

\*/

 Input  Reset

 Jalankan

## Variable

Ketika menulis sebuah program, kita memberi tahu komputer cara memproses informasi seperti mencetak teks ke layar atau melakukan operasi perhitungan. Untuk lebih mudah dalam penggunaan dan pemanggilan data, kita bisa memanfaatkan variabel. Variabel umumnya digunakan untuk menyimpan informasi atau nilai yang akan dikelola dalam sebuah program.

Pada JavaScript setidaknya ada tiga cara untuk mendeklarasikan sebuah variabel, yaitu menggunakan keyword var, let, dan const. Pada versi ECMAScript 2015 (ES6) dikenalkan deklarasi variabel dengan let dan const untuk menggantikan var yang dinilai kontroversial dan rawan menimbulkan bug.

Dalam kelas ini, kita akan banyak menggunakan keyword let dan const. Jika Anda penasaran kenapa var sudah tidak lagi disarankan, silakan simak diskusi [berikut](https://softwareengineering.stackexchange.com/questions/274342/is-there-any-reason-to-use-the-var-keyword-in-es6" \t "_blank).

Sekarang bagaimana caranya membuat sebuah variabel dalam JavaScript? Cukup mudah. Ketikkan keyword let yang diikuti dengan nama variabelnya.

1. let lastName;

Kode untuk mendeklarasikan variabel seperti di atas juga dikenal dengan declaration statement.

Selanjutnya, Anda bisa mengisi nilai variabel di atas menggunakan tanda sama dengan (=).

* [**main.js**](https://www.dicoding.com/academies/256/tutorials/13872?from=13867#run1-editor1)



1

let lastName;

2

lastName = "Skywalker";

3

​

4

console.log(lastName);

5

​

6

/\* output

7

Skywalker

8

\*/

 Input  Reset

 Jalankan

Anda juga bisa langsung menginisialisasi nilai variabel setelah mendeklarasikannya seperti berikut:

* [**main.js**](https://www.dicoding.com/academies/256/tutorials/13872?from=13867#run2-editor1)



1

let lastName = "Skywalker";

2

​

3

console.log(lastName);

4

​

5

/\* output

6

Skywalker

7

\*/

 Input  Reset

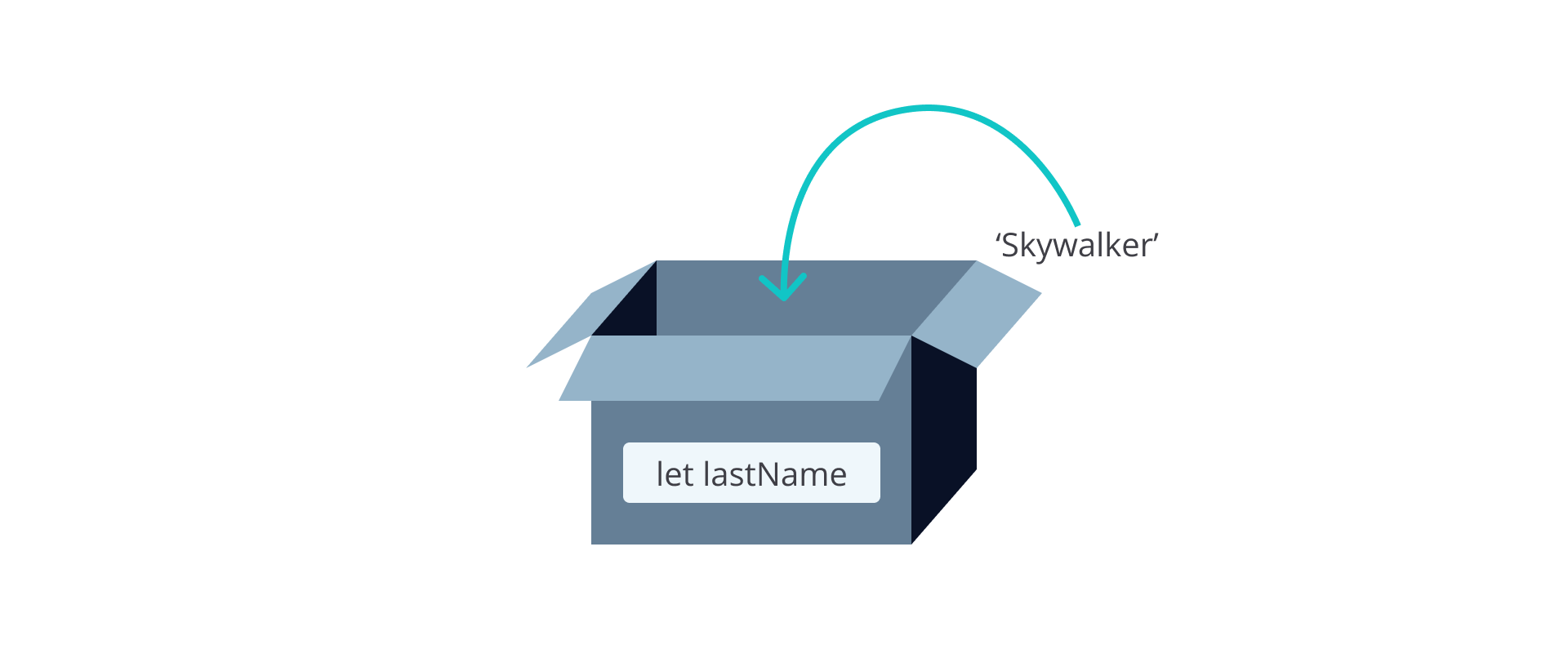
 Jalankan

Kode untuk menginisialisasikan nilai ke dalam sebuah variabel dengan tanda sama dengan (=) ini disebut dengan assignment expression.

Tunggu sebentar. Kita kembali bertemu dengan istilah statement dan expression. Karena expression pasti menghasilkan nilai, sehingga mereka bisa muncul di mana pun dalam program JavaScript. Sementara itu, statement merujuk pada aksi. Sehingga, pada bagian kode tertentu yang membutuhkan nilai tidak bisa kita isi dengan sebuah statement. Contohnya seperti kode berikut:

1. let fullName = let lastName; // Error karena let lastName adalah sebuah statement untuk deklarasi variabel. Statement tidak bisa berada di posisi expression.
2. let fullName = (lastName = "Skywalker"); // (lastName = "Skywalker") merupakan expression, sehingga kode ini tidak error.
3. let fullName = "Luke" + "Skywalker"; // "Luke" + "Skywalker" juga merupakan expression, sehingga kode ini tidak error.

Melalui contoh kode di atas, kita bisa bayangkan variabel sebagai sebuah kotak atau wadah yang menyimpan nilai. Proses inisialisasi atau assignment berarti kita memasukkan nilai ke dalam kotak tersebut.

[](https://www.dicoding.com/academies/256/tutorials/13872?from=13867)

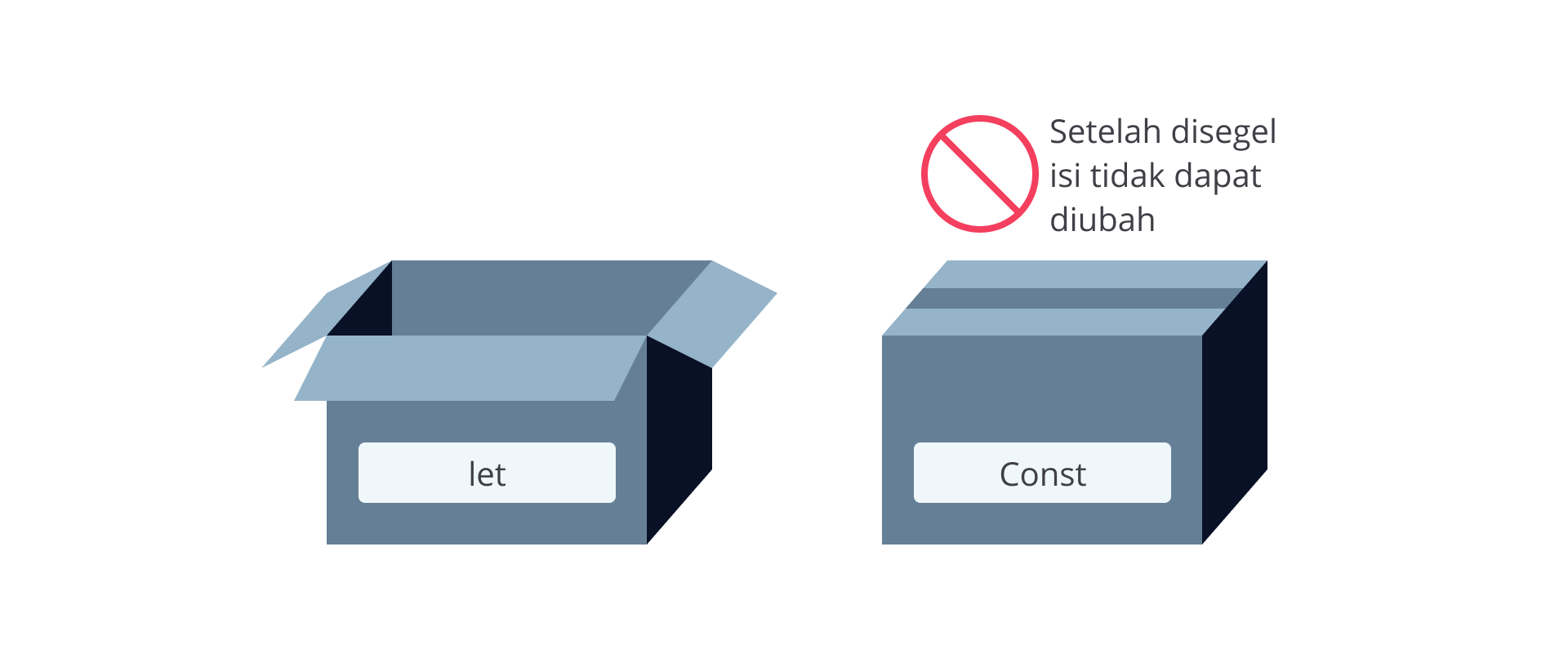
Variabel lastName di atas akan tersimpan di dalam memori komputer.

Setiap variabel memiliki nama yang dapat kita panggil dan gunakan. Kita dapat menamai variabel dengan nama apa pun, tetapi pastikan penamaannya masih masuk akal dengan konteksnya supaya 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 boleh mengandung spasi (whitespace). Jika penamaan variabel lebih dari dua kata, tuliskan secara camelCase. Contoh firstName, lastName, catName, dll.
* Tidak boleh mengandung karakter spesial (! . , / \ + \* = dll.)

Lalu, bagaimana dengan const? Const merupakan kependekan dari constant. Artinya, kita akan mendeklarasikan sebuah variabel dengan const ketika ingin variabel bernilai konstan dan tidak bisa diubah setelah diinisialisasi nilainya. Bayangkan variabel bernilai const sebagai sebuah kotak yang ditutup dan disegel setelah diisi, sehingga nilainya tidak bisa diubah lagi.

[](https://www.dicoding.com/academies/256/tutorials/13872?from=13867)

Jika menginisialisasi kembali nilai variabel yang menggunakan const, kita akan mendapati eror “TypeError: Assignment to constant variable.”

* [**main.js**](https://www.dicoding.com/academies/256/tutorials/13872?from=13867#run3-editor1)



1

const z = 100;

2

console.log(z);

3

z = 200;

4

console.log(z)

5

​

6

/\* TypeError: Assignment to constant variable. \*/

 Input  Reset

 Jalankan