DAFTAR ISI

[VARIABEL 2](#_Toc116980499)

[1. Apa itu variabel 2](#_Toc116980500)

[2. Cara Membuat Variabel di Javascript 3](#_Toc116980501)

[3. Menampilkan isi Variabel 4](#_Toc116980502)

[4. Mengisi Ulang Variabel 6](#_Toc116980503)

[5. Menghapus Variabel 6](#_Toc116980504)

[6. Mengenal Tipe Data 7](#_Toc116980505)

[7. Aturan Penulisan Nama Variabel di Javascript 9](#_Toc116980506)

[8. Apa itu Operator 10](#_Toc116980507)

[a. Opeartor Aritmatika pada Javascript 10](#_Toc116980508)

[b. Opeartor Penugasan pada Javascript 13](#_Toc116980509)

[c. Opeartor Perbandingan pada Javascript 16](#_Toc116980510)

[d. Opeartor Logika pada Javascript 19](#_Toc116980511)

[e. Opeartor Bitwise pada Javascript 20](#_Toc116980512)

[6. Opeartor Ternary pada Javascript 23](#_Toc116980513)

[Apa Selanjutnya? 27](#_Toc116980514)

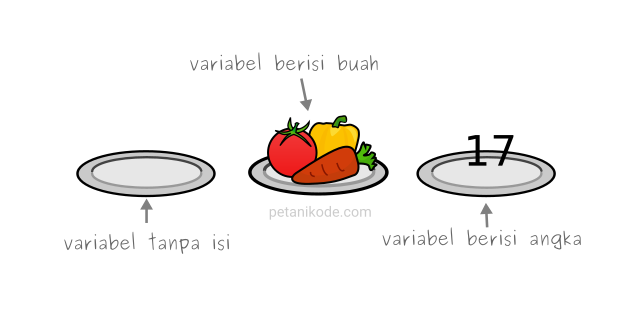
**Variable dan tipe data**

VARIABEL

1. Apa itu variabel

Variabel merupakan sebuah nama yang mewakili sebuah nilai. Variabel bisa diisi dengan berbagai macam nilai seperti **string** (teks), **number** (angka), objek, array, dan sebagainya.

Kita bisa ibaratkan, variabel itu seperti wadah untuk menyimpan sesuatu.



Pada kesempatan ini kita akan belajar tentang variabel dan tipe data dalam Javascript.

Dimulai dari cara membuat, cara mencetak variabel ke output, mengisi ulang, dan menghapusnya dari memori.

1. Cara Membuat Variabel di Javascript

Cara membuat variabel yang umum digunakan di javascript adalah menggunakan kata kunci **var** lalu diikuti dengan nama variabel dan nilainya.

Contoh:

var title = "Belajar Pemrograman Javascript";

Pada contoh di atas, kita membuat variabel bernama **title** dengan nilai berupa teks *(string)*: **"Belajar Pemrograman Javascript"**.

Contoh lagi:

var siteName = "SMK Negeri 1 Kinali";

var url = "[www.smkn1kinali.sch.id](http://www.smkn1kinali.sch.id)";

var visitorCount = 5921;

Perhatikan!

Pada contoh di atas, kita menggunakan huruf bersar atau kapital untuk nama variabel yang terdiri dari dua suku kata.

Kenapa tidak menggunakan *underscore*?

Pada Javascript kita dianjurkan menggunakan camelCase dalam penamaan. Boleh-boleh saja menggunakan *snake case* atau *underscore* seperti ini:

var site\_name = "SMK Negeri 1 Kinali";

var visitor\_count = 1243;

Hal tersebut tidak akan menjadi masalah, program masih tetap valid. Namun, mayoritas programmer Javascript menggunakan camelCase.

Pilihan ada ditangan anda, mau ikut yang mayoritas atau minoritas.

Perlu diketahui juga, selain kata kunci **var** kita juga bisa membuat variabel dengan kata kunci **let** atau tanpa awalan apapun.

Contoh:

// membuat variabel dengan kata kunci let

let price = 15000;

// membuat variabel tanpa awalan apapun

stock = 12

Apa bedanya yang menggunakan **var**, **let** dan yang tanpa awalan?

Perbedaannya terletak pada jangkauan dan penggunaannya. Nanti kita akan bahas lebih dalam lagi.

Satu lagi pertanyaan:

“Apa yang akan terjadi jika nilai variabel tidak diisi?”

Jawabannya, variabel akan bernilai **undefined** (belum ditentukan).

Contoh:

var x;

Maka variabel **x** akan bernilai **undefined**.

1. Menampilkan isi Variabel

Untuk menampilkan isi variabel, kita bisa memanfaatkan [fungsi-fungsi untuk menampilkan output](https://www.petanikode.com/javascript-output/) seperti:

Fungsi **console.log()** menampilkan output ke console javascript;

Fungsi **document.write()** menampilkan output ke dokumen HTML;

dan Fungsi **alert()** menampilkan output ke jendela dialog.

Mari kita coba…

Buatlah file bernama **lat2\_js.html**, lalu isi dengan kode berikut:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Belajar Variabel dalam Javascript</title>

<script>

// membuat variabel

var name = "SMK Negeri 1 Kinali";

var visitorCount = 50322;

var isActive = true;

var url = "<https://www.smkn1kinali.sch.id>";

// menampilkan variabel ke jendela dialog (alert)

alert("Selamat datang di " + name);

// menampilkan variabel ke dalam HTML

document.write("Nama Situs: " + name + "<br>");

document.write("Jumlah Pengunjung: " + visitorCount + "<br>");

document.write("Status Aktif: " + isActive + "<br>");

document.write("Alamat URL: " + url + "<br>");

</script>

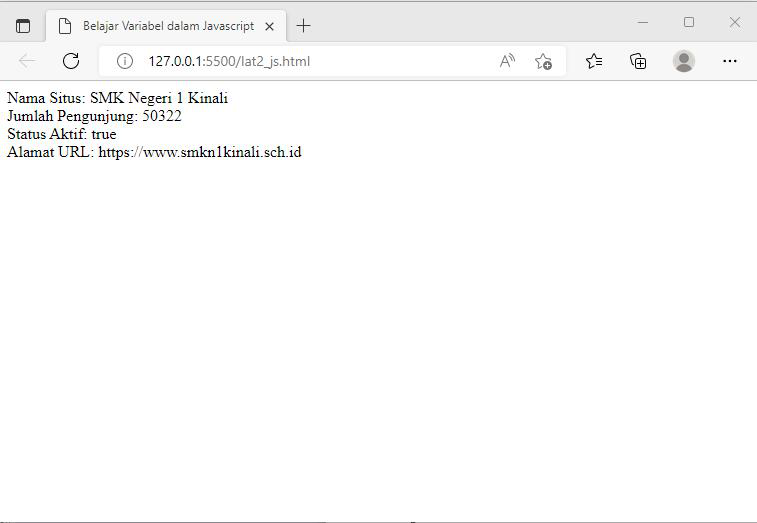
</head>

<body>

</body>

</html>

Setelah itu, coba buka dengan browser dan perhatikanlah hasilnya.



Arti simbol plus (+) pada contoh di atas adalah menggabungkan, bukan menjumlahkan.

1. Mengisi Ulang Variabel

Variabel bersifat *mutable*, artinya nilai yang tersimpan di dalamnya dapat kita isi ulang (berubah).

Contoh:

// mula-mula kita buat variabel dengan isi seperti ini

var age = 18;

// lalu kita isi ulang

age = 21;

Kenapa saat mengisi ulang nilai variabel tidak menggunakan kata kunci **var**?

Karena kata kunci **var** dibutuhkan saat membuat variabel saja. Sedangkan untuk mengisi ulang, kita cukup tulis seperti di atas.

Apabila kita menggunakan kata kunci **var**, berarti jadinya kita membuat variabel baru donk, bukan mengisi ulang.

1. Menghapus Variabel

Penghapusan variabel dalam Javscript memang jarang dilakukan. Namun, untuk program yang membutuhkan ketelitian dalam alokasi memori, penghapusan variabel perlu dilakukan agar penggunaan memori lebih optimal.

Penghapusan variabel dapat dilakukan dengan katakunci **delete**.

Contoh:

bookTitle = "Belajar Pemrograman Javascript";

delete bookTitle;

Maka variabel **bookTitle** akan menghilang dari memori.

Penghapusan variabel hanya bisa dilakukan pada variabel yang dibuat tanpa awalan. Sedangkan variabel yang dibuat dengan kata kunci **var** dan **let** akan dihapus otomatis.

1. Mengenal Tipe Data

Tipe data adalah jenis-jenis data yang bisa kita simpan di dalam variabel.

Ada beberapa tipe data dalam pemrograman Javascript:

1. String (teks)
2. Integer atau Number (bilangan bulat)
3. Float (bilangan Pecahan)
4. Boolean
5. Object

Javascript adalah bahasa yang bersifat dynamic typing, artinya kita tidak harus menuliskan tipe data pada saat pembuatan variabel seperti pada bahasa [C](https://www.petanikode.com/topik/c/), [C++](https://www.petanikode.com/topik/c++/), [Java](https://www.petanikode.com/topik/java/), dsb. yang bersifat static typing.

Contoh:

var name = "Dian";

var age = 22;

var single = true;

Javascript akan otomatis mengenali tipe data yang kita berikan pada variabel.

Pada contoh di atas variabel:

**name** bertipe data String;

**age** bertipe data integer;

dan **single** bertipe data boolean.

Hal ini bisa juga kita cek dengan kata kunci **typeof**.

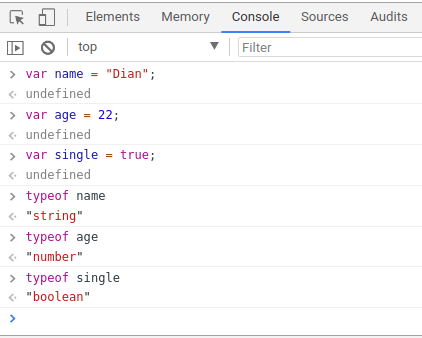
Contoh:

typeof name;

typeof age;

typeof single;

Hasilnya pada console:



Bagaimana cara kita mengenali tipe data selain menggunakan **typeof**?

Kita bisa mengenali melalui cara penulisannya.

Contoh:

// untuk tipe data teks atau string penulisannya selalu diapit dengan tanda petik

var author = "Petani Kode";

// sedangkan untuk tipe data number dan boolean tidak diapit dengan tanda petik

var available = true; // <- tipe data boolean hanya bernilai true dan false

var total = 14;

1. Aturan Penulisan Nama Variabel di Javascript

Ada beberapa aturan penulisan variabel dalam Javascript:

1. Penamaan variabel tidak boleh menggunakan angka di depannya.

contoh:

// salah

var 123nama = "SMK Negeri 1 Kinali";

// benar

var nama123 = "Petani Kode";

1. Penamaan variabel boleh menggunakan awal underscore.

contoh:

var \_nama = "Petani Kode";

1. Penamaan variabel dianjurkan menggunakan camelCase apabila tediri dari dua suku kata.

Contoh:

var fullName = "Petani Kode";

1. Penamaan variabel dianjurkan menggunakan bahasa inggris

Contoh:

var postTitle = "Tutorial Javascript untuk Pemula";

1. Apa itu Operator

Operator adalah simbol yang digunakan untuk melakukan operasi pada suatu nilai dan variabel.

Operator dalam pemrograman terbagi dalam 6 jenis:

1. Operator aritmatika;
2. Operator Penugasan (Assignment);
3. Opeartor relasi atau perbandingan;
4. Operator Logika;
5. Operator Bitwise;
6. Operator Ternary;

Operator wajib ada di setiap bahasa pemrograman. Ke 6 jenis operator di atas harus kamu pahami.

Mari kita bahas satu persatu…

1. Opeartor Aritmatika pada Javascript

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika seperti penjumlahan, pengurangan, pembagian, perkalian, dsb.

Operator aritmatika terdiri dari:

|  |  |
| --- | --- |
| Nama Operator | Simbol |
| Penjumlahan | + |
| Pengurangan | - |
| Perkalian | \* |
| Pemangkatan | \*\* |
| Pembagian | / |
| Sisa Bagi | % |

Untuk melakukan operasi perkalian, kita menggunakan simbol asterik **\***.

Jangan gunakan **x**, karena simbol **x** bukan termasuk dalam operator di dalam pemrograman.

Lalu untuk pemangkatan kita menggunakan asterik ganda **\*\***.

Untuk pembagian, kita gunakan simbol garis miring **/**.

Mari kita coba…

Contoh:

var a = 5;

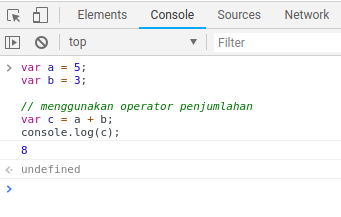
var b = 3;

// menggunakan operator penjumlahan

var c = a + b;

console.log(c);

Hasilnya:



Coba juga untuk operator yang lainnya:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Operator Aritmatika</title>

</head>

<body>

<script>

var a = 15;

var b = 4;

var c = 0;

// pengurangan

c = a - b;

document.write(`${a} - ${b} = ${c}<br/>`);

// Perkalian

c = a \* b;

document.write(`${a} \* ${b} = ${c}<br/>`);

// pemangkatan

c = a \*\* b;

document.write(`${a} \*\* ${b} = ${c}<br/>`);

// Pembagian

c = a / b;

document.write(`${a} / ${b} = ${c}<br/>`);

// Modulo

c = a % b;

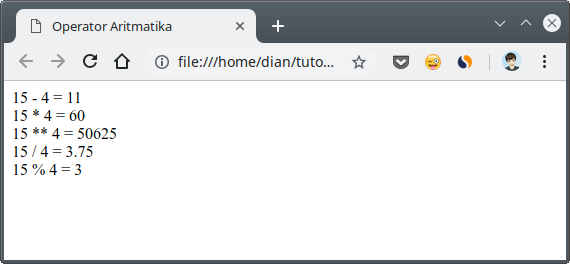
document.write(`${a} % ${b} = ${c}<br/>`);

</script>

</body>

</html>

Hasilnya:



Coba pertikan operator modulo (**%**) dan operator penjumlahan (**+**).

Operator modulo adalah operator untuk menghitung sisa bagi.

Misal **3** dibagi **2**, maka sisanya adalah **1**.

3 % 2 = 1

**Opertor Penggabungan Teks**

Mohon perhatikan!

Jangan sampai salah.

Pada Javascript, apabila kita akan melakukan operasi terhadap tipe data string atau teks menggunakan penjumlahan (+), maka yang akan terjadi adalah penggabungan; Bukan penjumlahan.

Contoh:

var a = "10" + "2";

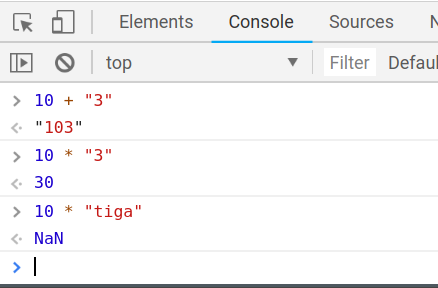
Maka hasilnya akan:

102

Kenapa tidak 12?

Karena kedua angka tersebut merupakan string—perhatikan, dia diapit dengan tanda petik.

Untuk operasi yang lainnya, silahkan dicoba-coba melalui console.



1. Opeartor Penugasan pada Javascript

Operator penugasan adalah operator yang digunakan untuk memberikan tugas kepada variabel. Biasanya digunakan untuk mengisi variabel.

Contoh:

var a = 19;

Variabel **a** kita berikan tugas untuk menyimpan nilai **19**.

Operator penugasan terdiri dari:

|  |  |
| --- | --- |
| Nama Operator | Sombol |
| Pengisian Nilai | = |
| Pengisian dan Penambahan | += |
| Pengisian dan Pengurangan | -= |
| Pengisian dan Perkalian | \*= |
| Pengisian dan Pemangkatan | \*\*= |
| Pengisian dan Pembagian | /= |
| Pengisian dan Sisa bagi | %= |

Operator penugasan sama seperti operator aritmatika. Ia juga digunakan untuk melakukan operasi aritmatika.

Contoh:

var jumlahView = 12;

// menggunakan operator penugasan penjumlahan

// untuk menambah nilai

jumlahView += 1;

Hasilnya:

Variabel **jumlahView** akan bertambah satu.

Maksud dari **jumlahView += 1** adalah seperti ini:

jumlahView = jumlahView + 1;

Bisa dibaca:

Isi variabel jumlahView dengan penjumlahan dari

nilai jumlahView sebelumnya dengan 1.

Khusus untuk operator penugasan yang dijumlahkan dan dikurangi dengan satu, bisa disingkat dengan ++ dan -- untuk pengurangan.

Contoh:

var a = 2;

a++;

Maka nilai dari variabel **a** akan menjadi 3.

Lalu pertanyaanya:

Apa bedanya dengan operator penugasan dengan operator aritmatika?

Operator aritmatika hanya melakukan operasi aritmatika saja, sedangkan operator penugasan… ia melakukan operasi aritmatika dan juga pengisian.

Berikut ini contoh operator penugasan:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Operator Penugasan</title>

</head>

<body>

<script>

document.write("Mula-mula nilai score...<br>");

// pengisian nilai

var score = 100;

document.write("score = "+ score + "<br/>");

// pengisian dan menjumlahan dengan 5

score += 5;

document.write("score = "+ score + "<br/>");

// pengisian dan pengurangan dengan 2

score -= 2;

document.write("score = "+ score + "<br/>");

// pengisian dan perkalian dengan 2

score \*= 2;

document.write("score = "+ score + "<br/>");

// pengisian dan pembagian dengan 4

score /= 4;

document.write("score = "+ score + "<br/>");

// pengisian dan pemangkatan dengan 2

score \*\*= 2;

document.write("score = "+ score + "<br/>");

// pengisian dan modulo dengan 3;

score %= 3;

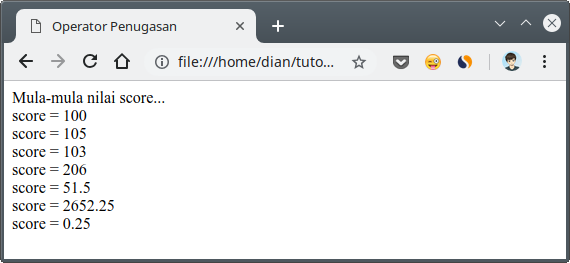
document.write("score = "+ score + "<br/>");

</script>

</body>

</html>

Hasilnya:



1. Opeartor Perbandingan pada Javascript

Operator relasi atau perbandingan adalah operator yang digunakan untuk membandingkan dua nilai.

Operator perbandingan akan menghasilkan sebuah nilai boolean **true** dan **false**.

Operator perbandingan terdiri dari:

|  |  |
| --- | --- |
| Nama Operator | Simbol |
| Lebih Besar | > |
| Lebih Kecil | < |
| Sama Dengan | == atau === |
| Tidak Sama dengan | != atau !== |
| Lebih Besar Sama dengan | >= |
| Lebih Kecil Sama dengan | <= |

Contoh:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Operator Perbandingan</title>

</head>

<body>

<script>

var aku = 20;

var kamu = 19;

// sama dengan

var hasil = aku == kamu;

document.write(`${aku} == ${kamu} = ${hasil}<br/>`);

// lebih besar

var hasil = aku > kamu;

document.write(`${aku} > ${kamu} = ${hasil}<br/>`);

// lebih besar samadengan

var hasil = aku >= kamu;

document.write(`${aku} >= ${kamu} = ${hasil}<br/>`);

// lebih kecil

var hasil = aku < kamu;

document.write(`${aku} < ${kamu} = ${hasil}<br/>`);

// lebih kecil samadengan

var hasil = aku <= kamu;

document.write(`${aku} <= ${kamu} = ${hasil}<br/>`);

// tidak samadengan

var hasil = aku != kamu;

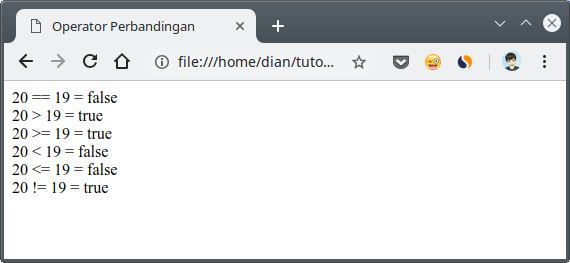
document.write(`${aku} != ${kamu} = ${hasil}<br/>`);

</script>

</body>

</html>

Hasilnya:



Pertanyaanya:

Apa perbedaan **==** (dua simbol samadengan) dengan **===** (tiga simbol samadengan)?

Perbandingan dengan menggunakan simbol **==** hanya akan membandingkan nilai saja. Sedangkan yang menggunakan **===** akan membandingkan dengan tipe data juga.

Contoh:

// ini akan bernilai true

var a = "4" == 4; //-> true

// sedangkan ini akan bernilai false

var b = "4" === 4; //-> false

Mengapa nilai **b** bernilai **false**?

Karena **"4"** (string) dan **4** (integer). Tipe datanya berbeda.

1. Opeartor Logika pada Javascript

Operator logika digunakan untuk melakukan operasi terhadap dua nilai **boolean**.

Operator ini terdiri dari:

|  |  |
| --- | --- |
| Nama Operator | Simbol |
| Logika AND | && |
| Logika OR | || |
| Negasi/kebalikan | ! |

Contoh:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Operator Logika</title>

</head>

<body>

<script>

var aku = 20;

var kamu = 19;

var benar = aku > kamu;

var salah = aku < kamu;

// operator && (and)

var hasil = benar && salah;

document.write(`${benar} && ${salah} = ${hasil}<br/>`);

// operator || (or)

var hasil = benar || salah;

document.write(`${benar} || ${salah} = ${hasil}<br/>`);

// operator ! (not)

var hasil = !benar

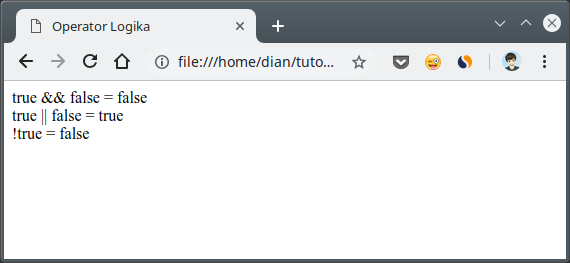
document.write(`!${benar} = ${hasil}<br/>`);

</script>

</body>

</html>

Hasilnya:



1. Opeartor Bitwise pada Javascript

Operator bitwise merupkan operator yang digunakan untuk operasi berdasarkan bit (biner).

Operator ini terdiri dari:

|  |  |
| --- | --- |
| Nama | Simbol di Java |
| AND | & |
| OR | | |
| XOR | ^ |
| Negasi/kebalikan | ~ |
| Left Shift | « |
| Right Shift | » |
| Left Shift (unsigned) | «< |
| Right Shift (unsigned) | »> |

Operator ini berlaku untuk tipe data **int**, **long**, **short**, **char**, dan **byte**.

Operator ini akan menghitung dari bit-ke-bit.

Misalnya, kita punya variabel **a = 60** dan **b = 13**.

Bila dibuat dalam bentuk biner, akan menjadi seperti ini:

a = 00111100

b = 00001101

(perhatikan bilangan binernya, angka 0 dan 1)

Kemudian, dilakukan operasi bitwise

Operasi AND

a = 00111100

b = 00001101

a & b = 00001100

Operasi OR

a = 00111100

b = 00001101

a | b = 00111101

Operasi XOR

a = 00111100

b = 00001101

a ^ b = 00110001

Opearsi NOT (Negasi/kebalikan)

a = 00111100

~a  = 11000011

Konsepnya memang hampir sama dengan opeartor Logika. Bedanya, Bitwise digunakan untuk biner.

Untuk lebih jelasnya…

Mari kita coba lihat contohnya:

<!DOCTYPE html>

<html lang="en">

<head>

<title>Operator Bitwise</title>

</head>

<body>

<script>

var x = 4;

var y = 3;

// operator bitwise and

var hasil = x & y;

document.write(`${x} & ${y} = ${hasil}<br/>`);

// operator bitwise or

var hasil = x | y;

document.write(`${x} | ${y} = ${hasil}<br/>`);

// operator bitwise xor

var hasil = x ^ y;

document.write(`${x} ^ ${y} = ${hasil}<br/>`);

// operator negasi

var hasil = ~x;

document.write(`~${x} = ${hasil}<br/>`);

// operator bitwise right shift >>

var hasil = x >> y;

document.write(`${x} >> ${y} = ${hasil}<br/>`);

// operator bitwise right shift <<

var hasil = x << y;

document.write(`${x} << ${y} = ${hasil}<br/>`);

// operator bitwise right shift (unsigned) >>>

var hasil = x >>> y;

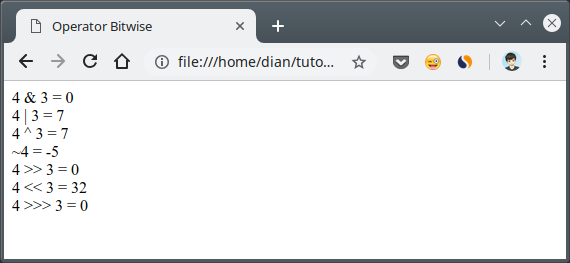
document.write(`${x} >>> ${y} = ${hasil}<br/>`);

</script>

</body>

</html>

Hasilnya:



1. Opeartor Ternary pada Javascript

Terakhir ada operator Ternary…

Operator ternary merupakan operator yang teridiri dari tiga bagian.

Operator-operator sebelumnya hanya dua bagian saja, yaitu: bagian kiri dan kanan. Ini disebut operator binary.

Sementara operator trinary ada bagian kiri, tengah, dan kanan.

bagian kiri <operator> bagian tengah <operator> bagian kanan

Opertor ternary pada Javascript, biasanya digunakan untuk membuat sebuah percabangan if/else.

Simbol opertor ternary terdiri dari tanda tanya dan titik dua (?:).

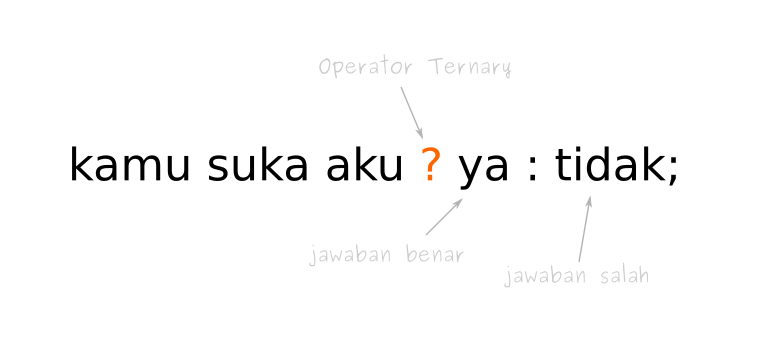
Bentuknya seperti ini:

<kodisi> ? "benar" : "salah"

Perhatikan! <kondisi> dapat kita isi dengan ekspresi yang menghasilkan nilai true dan false.

Apabila kondisi bernilai true, maka "benar" yang akan dipilih dan sebaliknya—apabila false—maka "salah" yang akan dipilih.

Opertor ini unik, seperti membuat pertanyaan.



Pada contoh di atas, “Kamu suka aku” adalah pertanyaan atau kondisi yang akan diperiksa.

Kalau jawabannya benar, maka **iya**. Sebaliknya akan **tidak**.

Lebih jelasnya, mari kita coba contohnya.

<!DOCTYPE html>

<html lang="en">

<head>

<title>Operator Ternary</title>

</head>

<body>

<script>

var pertanyaan = confirm("Apakah kamu berumur diatas 18 tahun?")

var hasil = pertanyaan ? "Selamat datang" : "Kamu tidak boleh di sini";

document.write(hasil);

</script>

</body>

</html>

Hasilnya:

