**Operators**

Istilah operator dipinjam dari matematika dengan pengertian yang sedikit berbeda. Pada Dart operator menginstruksikan komputer untuk melakukan operasi.

Sebenarnya kita telah menerapkan beberapa contoh operator pada materi sebelumnya. Sebagai contoh operator *assignment* (=) yang kita gunakan untuk inisialisasi nilai variabel.

1. var firstNumber = 4;

Pada kode di atas kita menginstruksikan komputer untuk memasukkan nilai 4 ke dalam variabel firstNumber.

**Operator aritmatika**

Contoh operator lain yang telah Anda lihat adalah operator aritmatika yang digunakan untuk operasi seperti penjumlahan atau perkalian. Lihatlah contoh kode berikut:

1. var firstNumber = 4;
2. var secondNumber = 13;
3. var sum = firstNumber + secondNumber;
4. print(sum);
6. /\*
7. Output :
8. 17
9. \*/

Dart mendukung operator aritmatika umum sebagai berikut:

| **Operator** | **Deskripsi** |
| --- | --- |
| + | Penjumlahan |
| - | Pengurangan |
| \* | Perkalian |
| / | Pembagian |
| ~/ | Pembagian, mengembalikan nilai int |
| % | Modulo atau sisa hasil bagi |

1. print(5 + 2);   // int add = 7
2. print(5 - 2);   // int subtract = 3
3. print(5 \* 2);   // int multiply = 10
4. print(5 / 2);   // double divide = 2.5
5. print(5 ~/ 2);  // int intDivide = 2
6. print(5 % 2);   // int modulo = 1

Operator aritmatika pada pemrograman memiliki aturan yang sama dengan matematika, di mana perkalian dan pembagian akan didahulukan sebelum penjumlahan atau pengurangan.

1. print(2 + 4 \* 2); // output: 10

Jika Anda ingin melakukan operasi penjumlahan terlebih dahulu, gunakan tanda kurung atau *parentheses*.

1. print((1 + 3) \* (4 - 2)); // output: 8

Selain itu Dart mendukung *increment* dan *decrement*. Contohnya adalah seperti berikut:

1. var a = 0, b = 5;
2. a++;
3. b--;
4. print(a); // output = 1
5. print(b); // output = 4

Expression a++ di atas dapat diartikan dengan a = a + 1. Komputer akan mengambil nilai dari a kemudian menambahkan 1 lalu memasukkannya kembali ke variabel a. Bentuk *increment* lainnya adalah seperti berikut:

1. var c = 0;
2. c += 5; // c = c + 5 atau c = 0 + 5
3. print(c); // output 5

Operator ini juga bisa digunakan pada operator aritmatika lain seperti perkalian dan pembagian.

1. var d = 2;
2. d \*= 3;   // d = d \* 3 atau d = 2 \* 3
3. print(d); // output = 6

**Operator perbandingan**

Dart juga mendukung operasi perbandingan untuk membandingkan nilai-nilai yang dijadikan sebagai *operands*. Berikut ini adalah contoh operator perbandingan pada Dart:

| **Operator** | **Deskripsi** |
| --- | --- |
| == | Sama dengan |
| != | Tidak sama dengan |
| > | Lebih dari |
| < | Kurang dari |
| >= | Lebih dari sama dengan |
| <= | Kurang dari sama dengan |

Operator perbandingan ini akan mengembalikan nilai dalam bentuk *boolean*.

1. if (2 <= 3) {
2. print('Ya, 2 kurang dari sama dengan 3');
3. } else {
4. print(‘Anda salah’);
5. }
7. /\*
8. Output:
9. Ya, 2 kurang dari sama dengan 3
10. \*/

**Operator logika**

Kita juga dapat menggabungkan ekspresi *boolean* atau membaliknya dengan operator logika. Operator ini meliputi:

| **Operator** | **Deskripsi** |
| --- | --- |
| || | OR |
| && | AND |
| ! | NOT |

Kita telah membahas operator *NOT* atau *bang* pada materi *boolean*. Sementara itu operator *OR* atau *AND* digunakan untuk menguji logika dari beberapa nilai *boolean*. Operator AND akan menghasilkan nilai *true* jika semua *operand*-nya bernilai *true*, sedangkan OR jika salah satu saja dari *operand* bernilai true maka operator akan mengembalikan nilai *true*. Contohnya seperti kode berikut:

1. if (2 < 3 && 2 + 4 == 5) {
2. print('Untuk mencetak ini semua kondisi harus benar');
3. } else {
4. print('2 kurang dari 3, tapi 2 + 4 tidak sama dengan 5, maka ini akan tampil');
5. }
7. if (false || true || false) {
8. print('Ada satu nilai true');
9. } else {
10. print('Jika semuanya false, maka ini akan tampil');
11. }
13. /\*
14. Output:
15. 2 kurang dari 3, tapi 2 + 4 tidak sama dengan 5, maka ini akan tampil
16. Ada satu nilai true
17. \*/