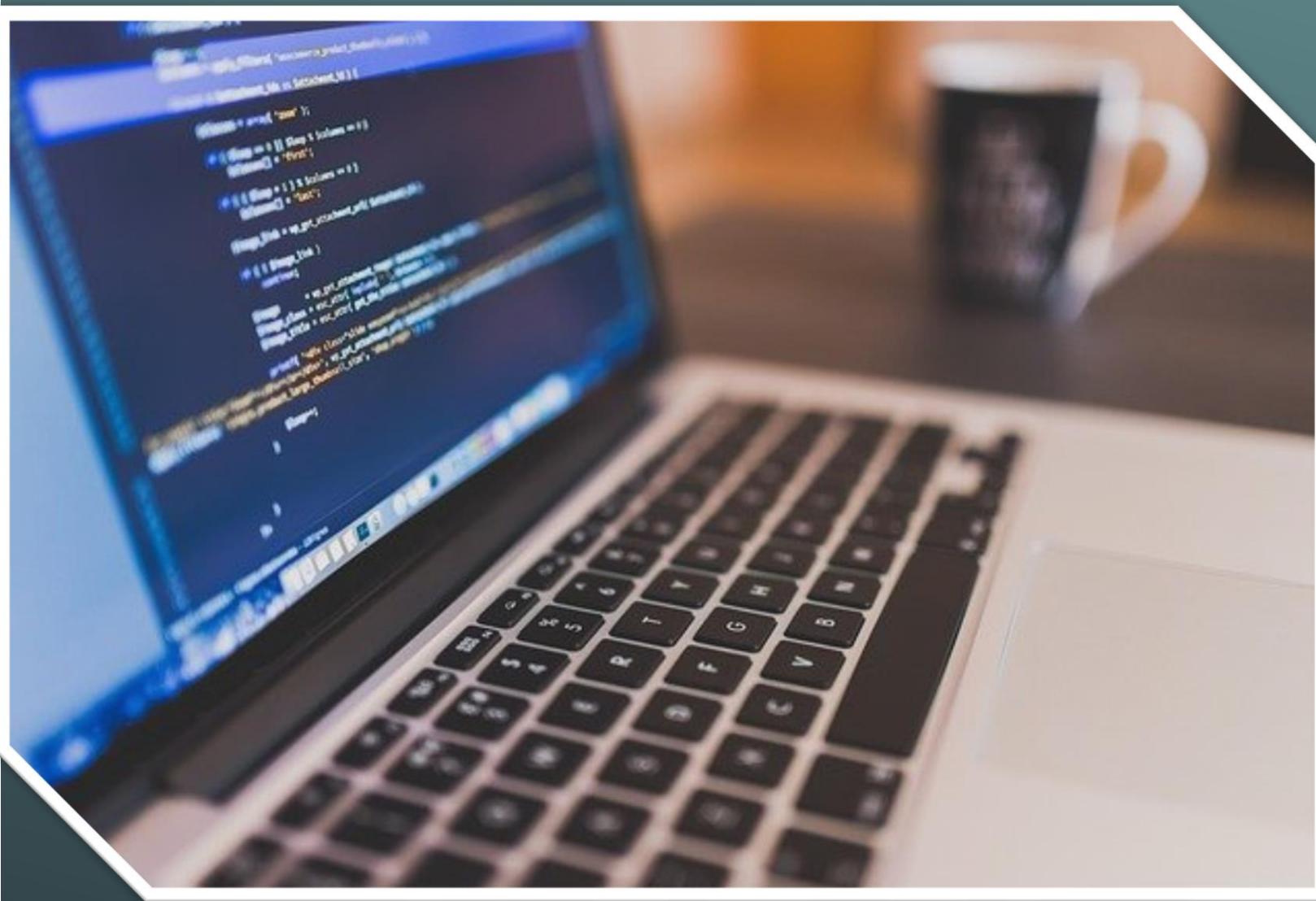


MATERI AJAR PEMROGRAMAN DASAR

OPERATOR ARITMATIKA

Oleh: Eko Zulkaryanto, S.Komp



**KELAS X
SMK TRI SUKSES
NATAR LAMPUNG SELATAN
2021**

MATERI AJAR PEMROGRAMAN DASAR

OPERATOR ARITMATIKA

Oleh:

EKO ZULKARYANTO, S.Komp

SMK TRI SUKSES NATAR LAMPUNG SELATAN

TAHUN 2021

KATA PENGANTAR

Kami ucapkan *Alhamdulillahi Rabbil Alamin*, puji syukur kepada Allah *Subhanahu Wa Ta'ala* yang telah memberkati kita sehingga materi ajar ini bisa kami selesaikan dengan baik dan tepat waktu. Materi ajar ini didesain untuk digunakan oleh peserta didik Kelas X Sekolah Menengah Kejuruan (SMK).

Materi ajar ini berisikan materi Perangkat Lunak Pengolah Angka. Harapan kami peserta didik dapat sungguh menguasai materi ajar ini sehingga nantinya dapat membantu peserta didik SMK menguasai materi Pemrograman Dasar yang dipelajari dengan baik dan benar.

Semoga materi ajar ini dapat membantu kemajuan peserta didik di Indonesia dan akhirnya Pelajaran Pemrograman Dasar menjadi salah satu pelajaran favorit di Indonesia.

Akhirnya kami mengucapkan banyak terima kasih kepada semua pihak yang telah membantu terselesaikannya penyusunan materi ajar ini. Kepada para dosen dan teman – teman sesama Pejuang PPG Teknik Komputer dan Informatika Angkatan 4 Universitas Negeri Malang (UM) Malang kami haturkan terima kasih. Semoga berkat dan karunia Allah menyertai Anda semua.

Lampung Selatan, September 2021

Penyusun,

Eko Zulkaryanto

DAFTAR ISI

KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
A. Kompetensi Dasar	1
B. Indikator Pencapaian Kompetensi	1
C. Tujuan Pembelajaran	1
D. Pokok-Pokok Materi	1
E. Peta Konsep	2
BAB II URAIAN MATERI	3
A. Operator Aritmatika	3
B. Rangkuman	9
C. Tes Formatif	9
DAFTAR PUSTAKA	10

BAB I PENDAHULUAN

A. Kompetensi Dasar

- 3.5 Menerapkan operasi aritmatika dan logika
- 4.5 Membuat kode program dengan operator aritmatika dan logika

B. Indikator Pencapaian Kompetensi

- 3.5.1 Menerapkan operator aritmatika (C3)
- 3.5.2 Menganalisis penerapan operator aritmatika (C4)
- 4.5.1 Menerapkan operator aritmatika pada kode program (P2)
- 4.5.2 Menunjukkan alur kode program operator aritmatika (P3)

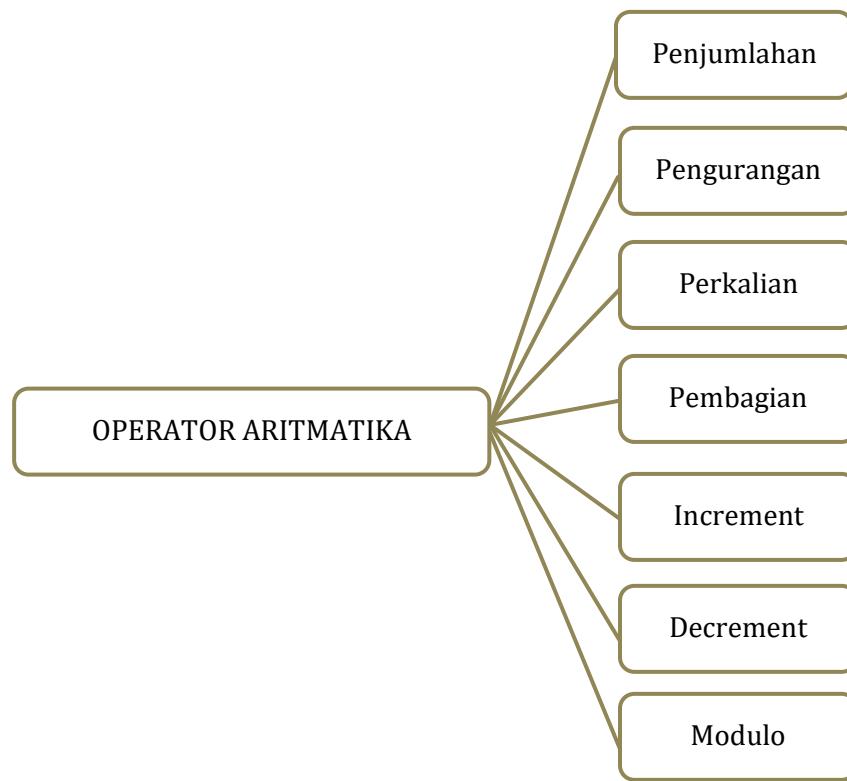
C. Tujuan Pembelajaran

- 1) Melalui tayangan slide presentasi peserta didik dapat menerapkan operator aritmatika dengan baik dan benar (C3)
- 2) Melalui tayangan slide presentasi peserta didik dapat menganalisis penerapan operator aritmatika dengan baik dan benar (C4)
- 3) Melalui kegiatan praktik penyelesaian studi kasus peserta didik dapat menerapkan operator aritmatika pada kode program dengan baik dan benar (P2)
- 4) Melalui kegiatan praktik penyelesaian studi kasus peserta didik dapat menunjukkan alur kode program operator aritmatika dengan baik dan benar (P3)

D. Pokok-Pokok Materi

- 1) Operator Aritmatika
- 2) Penjumlahan
- 3) Pengurangan
- 4) Perkalian
- 5) Pembagian
- 6) Increment
- 7) Decrement
- 8) Modulo

E. Peta Konsep



BAB II URAIAN MATERI

A. Operator Aritmatika

Masih ingatkah dengan operator di Bahasa Pemrograman yang telah kita pelajari pada Kompetensi Dasar sebelumnya? Operator adalah simbol atau tanda yang jika diletakkan pada dua buah operand (data) dapat menghasilkan sebuah hasil, operator berupa simbol yang digunakan untuk menyusun suatu ekspresi dengan melibatkan satu atau beberapa operand [1]. Contohnya pada matematika dimana tanda tambah ('+') jika diletakkan di antara dua buah angka akan menghasilkan angka lain hasil pertambahan dari dua angka tersebut.

Jenis-jenis operator pada Bahasa Pemrograman adalah sebagai berikut:

- 1) Operator Aritmatika
- 2) Operator Pemberi Nilai
- 3) Operator Pembanding
- 4) Operator Logika

Pada materi ajar ini akan kita ulas lebih lanjut tentang operator aritmatika.

Operator Aritmatika dalam pemrograman adalah operator yang dibutuhkan untuk melakukan operasi matematika yang meliputi penjumlahan, pengurangan, perkalian, pembagian, *increment*, *decrement*, dan modulo. Berikut operator-operator aritmatika:

Simbol Operator	Nama Operator	Keterangan
+	Penjumlahan	Menjumlahkan bilangan di sebelah kiri dan kanan
-	Pengurangan	Mengurangkan bilangan sebelah kiri dengan bilangan di sebelah kanan
*	Perkalian	Mengalikan bilangan sebelah kiri dengan bilangan di sebelah kanan
/	Pembagian	Membagi bilangan sebelah kiri dengan bilangan di sebelah kanan

++	Increment	Menjumlahkan bilangan dengan 1
--	Decrement	Mengurangkan bilangan dengan 1
%	Modulo	Sisa dari pembagian

Namun, yang perlu kita perhatikan adalah bahwa dalam penerapan operator aritmatika akan melibatkan operator pemberi nilai. Begitu juga dengan operator logika, akan melibatkan operator pembanding. Untuk itu, pada materi ajar ini kita akan sekaligus mempelajari keempat operator tersebut.

B. Penjumlahan

Penggunaan operator penjumlahan (+), dapat dilakukan terhadap dua bilangan bulat (integer), dua bilangan pecahan (float), ataupun campuran dari kedua bilangan tersebut.

Contoh:

```
10 + 20
```

Pada operasi di atas, penjumlahan digunakan untuk menjumlahkan bilangan bulat 10 dengan bilangan bulat 20. Dalam penerapannya di Bahasa Pemrograman C, digunakan variabel untuk dapat menampung hasil penjumlahan di atas menjadi sebagai berikut:

```
int hasil = 10 + 20;
```

Pada kode di atas, variabel `hasil` yang bertipe data integer digunakan untuk menampung hasil dari operasi penjumlahan bilangan bulat 10 dengan bilangan bulat 20. Hasilnya kemudian disimpan ke variabel `hasil` dengan bantuan operator pemberi nilai (=).

Terkadang operator pemberi nilai ini sering terjadi miskonsepsi pemahaman kita dengan operator sama dengan yang ada di Mata Pelajaran Matematika.

Di Bahasa Pemrograman C, operator pemberi nilai yang disimbolkan dengan “=” asalnya adalah tanda panah ke kiri (\leftarrow) dalam penulisannya dalam bentuk algoritma pseudocode. Sehingga, simbol “ \leftarrow ” diganti dengan simbol “=” untuk memudahkan dalam pengetikan kode program.

Berikut contohnya:

```
int hasil ← 10 + 20;
```

Dapat diartikan, hasil penjumlahan $10 + 20$ kemudian disimpan ke variabel hasil.

C. Pengurangan

Operator pengurangan (-) digunakan hampir sama seperti pada operator penjumlahan (-). Operator pengurangan digunakan untuk mengurangkan bilangan pertama dengan bilangan ke dua. Kedua bilangan tersebut dapat bertipe data bilangan bulat (integer) atau bilangan pecahan (float).

Contoh dalam bahasa C:

```
int hasil = 20 - 10;
```

Pada kode di atas, bilangan bulat 20 dikurangi dengan bilangan bulat 10, kemudian hasilnya disimpan ke variabel bilangan bulat `hasil`.

D. Perkalian

Operator perkalian (*) menggunakan karakter “*” (bintang). Tidak menggunakan karakter “x” atau “X”, karena karakter itu juga digunakan untuk variabel seperti karakter huruf lainnya.

Contoh dalam bahasa C:

```
int hasil = 20 * 10;
```

Pada kode di atas, bilangan bulat 20 dikalikan dengan bilangan bulat 10, kemudian hasilnya disimpan ke variabel bilangan bulat `hasil`.

E. Pembagian

Operator pembagian menggunakan karakter “/”, tidak dapat menggunakan karakter “:”. Cara penggunaannya masih sama seperti penggunaan operator sebelumnya.

Contoh dalam bahasa C:

```
float hasil = 20 / 10;
```

Pada kode di atas, bilangan bulat 20 dibagi dengan bilangan bulat 10, kemudian hasil dari pembagian tersebut disimpan ke variabel bilangan pecahan (float) `hasil`.

Kenapa variabel hasil tidak menggunakan bilangan bulat saja?

Apa yang terjadi jika variabel hasil dideklarasikan sebagai bilangan bulat?

Tentu kita tahu bahwa hasil pembagian tidak selalu bulat. Untuk menghasilkan hasil perhitungan yang akurat tentu pemilih jenis tipe data juga tidak boleh salah.

F. Increment

Operator increment menggunakan karakter “++”, operator ini digunakan untuk menjumlahkan dengan 1 (bilangan bulat 1). Operator ini ditempatkan disebelah kanan variabel atau sebelah kiri variabel. Jika ditempatkan di sebelah kanan variabel, penjumlahan dilakukan setelah baris instruksi tersebut dieksekusi oleh program. Sedangkan jika ditempatkan sebelum variabel, maka penjumlahan langsung terjadi di baris instruksi tersebut.

Contoh dalam bahasa C:

```
int a = 100;  
a++;
```

atau

```
int a = 100;  
++a;
```

Jika baris berikutnya adalah perintah untuk pencetak variabel a, maka dari kedua pilihan kode program di atas keluarannya adalah sama-sama 101.

Untuk membuktikan bahwa `a++` berbeda dengan `++a`, cobalah potongan kode di bawah ini:

```
int a = 100;  
printf("%d", a++);
```

bandingkan keluarannya dengan:

```
int a = 100;  
printf("%d", ++a);
```

G. Decrement

Operator decrement menggunakan karakter “`--`”, digunakan untuk mengurangkan suatu variabel bilangan bulat dengan 1. Penempatan oprand decrement sama seperti pada increment, yaitu setelah variabel atau setelah variabel, dengan ketentuan yang sama seperti pada operator increment.

Contoh dalam Bahasa C:

```
int a = 100;  
a--;
```

atau

```
int a = 100;  
--a;
```

Untuk membuktikan perbedaan penggunaan `a--` dan `--a`, silahkan bandingkan hasil keluaran dari potongan kode program di bawah ini:

```
int a = 100;  
printf("%d", a--);
```

bandingkan dengan:

```
int a = 100;  
printf("%d", --a);
```

H. Modulo

Operator modulo digunakan untuk menyatakan sisa bagi. Sisa hasil dari pembagian dalam bilangan bulat. Dalam bahasa C, karakter operand yang digunakan adalah "%" atau "mod". Untuk lebih memahami hasil perhitungan dari operator modulo, perhatikan contoh-contoh berikut:

$$10 \% 2 = 0$$

$$10 \% 6 = 4$$

$$10 \% 3 = 1$$

$$5 \% 3 = 2$$

$$10 \% 4 = 2$$

$$3 \% 2 = 1$$

$$10 \% 5 = 0$$

$$2 \% 3 = 2$$

Bilangan 10 mod 2 hasilnya adalah 0, karena bilangan 10 habis dibagi bilangan 2, sehingga hasil modulonya adalah 0. Bilangan 10 dibagi 3 akan sisa 1, sehingga 10 mod 3 adalah 1. Sedangkan untuk 2 mod 3, bilangan 2 dibagi 3 tidak bisa, sehingga sisa dari hasil pembagiannya adalah 2, jadi 2 mod 3 adalah 2.

Contoh dalam bahasa C:

```
int A = 10;  
int hasil = A % 2;  
printf("%d", hasil);
```

atau ringkasnya:

```
int A = 10;  
printf("%d", A % 2);
```

Apa keluaran dari program di atas?

Modulo 2 biasanya seringkali diterapkan untuk algoritma menentukan suatu bilangan apakah bilangan genap atau bilangan ganjil.

Berapa hasil modulo 2 jika bilangan tersebut genap?

Dan berapa hasil modulo 2 jika bilangan tersebut ganjil?

Untuk algoritma menentukan bilangan genap atau bilangan ganjil, *insyaAllah* akan kita pelajari lebih lanjut di Kompetensi Dasar berikutnya setelah Operator Logika.

I. Rangkuman

Operator Aritmatika dalam pemrograman adalah operator yang dibutuhkan untuk melakukan operasi matematika yang meliputi penjumlahan, pengurangan, perkalian, pembagian, *increment*, *decrement*, dan modulo. Dalam penerapan operator aritmatika akan melibatkan operator pemberi nilai.

Operator penjumlahan, pengurangan, perkalian dan pembagian sudah sangat sering ditemui dan digunakan dipelajaran matematika. Operator *increment* menambahkan dengan nilai 1 pada variabel yang menggunakan operator ini. Operator *decrement* mengurangkan dengan nilai 1 pada variabel yang menggunakan operator ini.

J. Tes Formatif

- 1) Perhatikan operasi aritmatika di bawah ini!

$$10 \dots 2 = 20$$

Untuk menghasilkan nilai 20, operator yang cocok untuk mengisi titik-titik di atas adalah ...

- a. %
- b. *
- c. /
- d. ++
- e. -

- 2) Perhatikan operasi aritmatika di bawah ini!

(1) $8 \% 5$

(2) $8 - 5$

(3) $10 \% 7$

(4) $10 - 7$

Hasil operasi secara berturut-turut di atas adalah ...

- a. 0 0 0 0
- b. 5 3 7 3
- c. 3 3 3 3
- d. 5 3 3 3
- e. 3 3 7 3

- 3) Perhatikan potongan kode program C di bawah ini!

```
int x = 90;  
x = x + 2;  
++x;
```

Dari operasi di atas, maka nilai x menjadi ...

- a. 90
- b. 91
- c. 92
- d. 93
- e. 94

- 4) Perhatikan kode program C di bawah ini!

```
int a = 100;  
printf("%d", ...);
```

Operator yang tepat untuk mengisi titik-titik pada kode di atas agar menghasilkan keluaran 101 adalah ...

- a. a--
- b. **a
- c. a++
- d. ++a
- e. -a

- 5) Manakah operasi aritmatika di bawah ini yang tidak 0 (nol)?

- a. $10 / 2 - 5$
- b. $3 * 2 - 6$
- c. $6 - 2 * 3$
- d. $10 - 2 * 5$
- e. $10 - 6 / 2 - 1$

DAFTAR PUSTAKA

- [1] <https://www.dicloud.id/tutorial/articles/mengenal-operator-dalam-pemrograman-c> [Diakses pada 27 September 2021]
- [2] <https://www.duniailkom.com/pengertian-dan-jenis-jenis-operator-logika-dalam-php/> [Diakses pada 27 September 2021]