

# OPERASI DASAR MASUKAN DAN KELUARAN



## Overview

---

Dalam sebuah bahasa pemrograman terdapat suatu masukan (input) dan keluaran (output). Masukan dan keluaran ini digunakan sebagai interaksi pengguna terhadap program yang akan dieksekusinya. Masukan adalah suatu inputan yang diberikan oleh pengguna suatu program dimana masukan tersebut akan diteruskan kedalam proses dengan aturan-aturan yang disepakatin oleh bahasa pemrograman dan akan ditampilkan hasil berupa keluaran (output) dari proses yang telah dilakukan.



## Tujuan

---

- ☑ Dapat membedakan mana operasi masukan dan keluaran
- ☑ Dapat membuat program dengan operasi masukan dan keluaran
- ☑ Memahami penggunaan variabel atau konstanta dalam operasi masukan dan keluaran

## 6.1 Cout

Dengan cout pemrogram dapat meletakkan suatu informasi ke *standart output* (normalnya berupa layar). Sebagai contoh, pernyataan berikut akan menampilkan tulisan “Pilihan Anda salah”

```
cout << "Pilihan Anda salah !\a\n";
```

Untuk lebih jelasnya dapat memerhatikan program dibawah ini :

Contoh 12	Hasil
<pre>/*-----* /* Contoh 12 : Memperkenalkan cout * /*-----* #include &lt;iostream.h&gt; void main() {     cout &lt;&lt; "Pilihan Anda Salah !";     return 0; }</pre>	Pilihan Anda Salah!

## 6.2 Manipulator

Manipulator umumnya digunakan untuk mengatur tampilan data. Misalnya untuk mengatur agar suatu nilai ditampilkan dengan lebar 10 karakter dan diatur rata kanan terhadap lebar tersebut.

### 6.2.1 Manipulator endl

Manipulator **endl** digunakan untuk menyisipkan karakter *newline*. Dengan kata lain manipulator ini identik dengan ‘\n’. Contoh program berikut menunjukkan penggunaan endl.

Contoh 13	Hasil
<pre>/*-----* /* Contoh 13 : Menampilkan 3 jumlah barang dan * /* menggunakan manipulator endl * /*-----* #include &lt;iostream.h&gt; #include &lt;conio.h&gt; void main() {     int jumbar1 = 150;     int jumbar2 = 23;     int jumbar3 = 1401;     clrscr(); // Hapus layar     cout &lt;&lt; "Barang 1 = " &lt;&lt; jumbar1 &lt;&lt; endl;     cout &lt;&lt; "Barang 2 = " &lt;&lt; jumbar2 &lt;&lt; endl;     cout &lt;&lt; "Barang 3 = " &lt;&lt; jumbar3 &lt;&lt; endl;     return 0; }</pre>	Barang 1=150 Barang 2=23 Barang 3=1401

## 6.2.2 Manipulator setw()

Manipulator **setw()** bermanfaat untuk mengatur lebar dari suatu tampilan data. Untuk lebih jelasnya dapat dilihat dalam program berikut :

Contoh 14	Hasil
<pre>/*-----* /* Contoh 14 : Menampilkan 3 jumlah barang dgn * /* menggunakan manipulator setw() * /*-----* #include &lt;iostream.h&gt; #include &lt;conio.h&gt; void main() { int jumbar1 = 150; int jumbar2 = 23; int jumbar3 = 1401; clrscr(); // Hapus layar cout &lt;&lt; "Barang 1 = " &lt;&lt; setw(4) &lt;&lt; jumbar1 &lt;&lt; endl; cout &lt;&lt; "Barang 2 = " &lt;&lt; setw(4) &lt;&lt; jumbar2 &lt;&lt; endl; cout &lt;&lt; "Barang 3 = " &lt;&lt; setw(4) &lt;&lt; jumbar3 &lt;&lt; endl; return 0; }</pre>	Barang 1= 150 Barang 2= 23 Barang 3= 1401

Dengan menggunakan **setw()**, terlihat hasil eksekusi adalah rata kanan. Ini dapat di bedakan dengan hasil eksekusi pada contoh 13 yang rata kiri.

## 6.3 cin

Obyek **cin** bermanfaat untuk untuk membaca data dari *standart input* (normalnya adalah keyboard).

### 6.3.1 cin dengan sebuah variabel

Bentuk pernyataan **cin** untuk membaca data dari keyboard dan meletakkan ke sebuah variabel variabel bernama *var* :

**cin >> var**

Contoh program yang menunjukkan pemakaian **cin** untuk membaca data bertipe **int** dan **float**.

Contoh 15	Hasil
<pre> /*-----* /* Contoh 15 : Membaca data bertipe int dan * /* float dari keyboard * /*-----* #include &lt;iostream.h&gt; void main () { int bil_x ; // Definisi bilangan bulat float bil_y ; // Definisi bilangan pecahan cout &lt;&lt; "Masukkan sebuah bilangan bulat = " ; cin &gt;&gt; bil_x ; cout &lt;&lt; "Masukkan sebuah bilangan pecahan = " ; cin &gt;&gt; bil_y ;  cout &lt;&lt; " Bilangan Bulat = " &lt;&lt; bil_x &lt;&lt; endl; cout &lt;&lt; " Bilangan Pecahan = " &lt;&lt; bil_y &lt;&lt; endl;  return 0; } </pre>	<p>Masukkan sebuah bilangan bulat = 123  Masukkan sebuah bilangan pecahan = 23.1  Bilangan Bulat = 123  Bilangan Pecahan 23.1</p>

Tampak bahwa bentuk perintah untuk membaca data bertipe **int** ataupun **float** sama saja. Hal ini berlaku untuk semua jenis data dasar (**char**, **int**, **long**, **float** ataupun **double**). Salah satu manfaat dari adanya fasilitas pemasukkan dari keyboard adalah memungkinkan untuk membuat program yang membaca data yang berubah-ubah.

### 6.3.2 cin dengan lebih dari satu variabel

Bentuk cin dapat juga sebagai berikut :

```
cin >> var1 >> var2
```

Pada contoh ini kedua bilangan dapat bilangan dapat dimasukkan dari keyboard dengan pemisah berupa spasi, tab atau enter.

Contoh 16	Hasil
<pre> /*-----* /* Contoh 16 : Untuk membaca data dengan lebih * /* dari satu variabel * /*-----*  #include &lt;iostream.h&gt; void main () { int bil_x ; // Definisi bilangan bulat float bil_y ; // Definisi bilangan pecahan cout &lt;&lt; "Masukkan sebuah bilangan bulat dan " &lt;&lt; endl; cout &lt;&lt; "Masukkan sebuah bilangan pecahan " &lt;&lt; endl ; cin &gt;&gt; bil_x &gt;&gt; bil_y;  cout &lt;&lt; " Bilangan Bulat = " &lt;&lt; bil_x &lt;&lt; endl; cout &lt;&lt; " Bilangan Pecahan = " &lt;&lt; bil_y &lt;&lt; endl; return 0; } </pre>	<p>Masukkan sebuah bilangan bulat dan Masukkan sebuah bilangan pecahan 20 23.2 ↵ Bilangan Bulat = 20 Bilangan Pecahan 23.2</p>

### 6.2.1. cin untuk membaca karakter

cin juga dapat dipakai untuk membaca sebuah karakter.  
Sebagai contoh :

Contoh 17	Hasil
<pre> /*-----* /* Contoh 17 : Membaca karakter dengan cin * /*-----*  #include &lt;iostream.h&gt; #include &lt;iomanip.h&gt; #include &lt;conio.h&gt; void main () { char karakter; clrscr(); cout &lt;&lt; "Masukkan sebuah karakter "; cin &gt;&gt; karakter; cout &lt;&lt; "Anda mengetik " &lt;&lt; karakter &lt;&lt; endl; cout &lt;&lt; "Masukkan sebuah karakter "; cin &gt;&gt; karakter; cout &lt;&lt; "Anda mengetik " &lt;&lt; karakter &lt;&lt; endl; return 0; } </pre>	<p>Masukkan sebuah karakter a ↵ Anda mengetik a Masukkan sebuah karakter b ↵ Anda mengetik b</p>

Tampak pada contoh program diatas, untuk setiap karakter yang dimasukkan harus diakhiri dengan enter (↵). Penekanan enter pada pemasukkan sebuah karakter terkadang tidak dikehendaki oleh pemakai. Jika demikian, anda dapat menggunakan fungsi **getch()** ataupun **getche()**.

### 6.3.3 Fungsi **getc()** dan **getche()**

Fungsi **getc()** dan **getche()** berguna untuk membaca sebuah karakter tanpa perlu menekan enter. Selain itu, fungsi ini juga dapat dipakai untuk membaca tombol seperti spasi, tab ataupun enter.

Bentuk pemakaiannya :

```
karakter = getc()
karakter = getche()
```

Perbedaan kedua fungsi ini adalah :

- a. **getc()** : tidak menampilkan karakter dari tombol yang ditekan
- b. **getche()** : menampilkan karakter dari tombol yang ditekan

Contoh 18	Hasil
<pre> /*-----* /* Contoh 18 : Membaca karakter dengan getch() * /*-----* #include &lt;iostream.h&gt; #include &lt;iomanip.h&gt; #include &lt;conio.h&gt; void main() { char karakter; clrscr(); cout &lt;&lt; "Masukkan sebuah karakter "; karakter = getch(); cout &lt;&lt; "Anda mengetik " &lt;&lt; karakter &lt;&lt; endl; cout &lt;&lt; "Masukkan sebuah karakter "; karakter = getch(); cout &lt;&lt; "Anda mengetik " &lt;&lt; karakter &lt;&lt; endl; return 0; } </pre>	<p>Masukkan sebuah karakter a «</p> <p>Tidak ditampilkan</p> <p>Anda mengetik a</p> <p>Masukkan sebuah karakter b «</p> <p>Tidak ditampilkan</p> <p>Anda mengetik b</p>

Sebuah **getc()** bisa pula dimanfaatkan untuk menunggu sembarang tombol ditekan. Pada keadaan seperti ini tidak perlu diletakkan ke variabel.

Contoh 19	Hasil
<pre> /*-----* /* Contoh 19 : getch() untuk memebaca sembarang * /* tombol * /*-----* #include &lt;iostream.h &gt; #include &lt;conio.h&gt; void main() { cout &lt;&lt; "Tekanlah sembarang tombol" &lt;&lt; endl; cout &lt;&lt; "Untuk mengakhiri program ini " ; getch(); } </pre>	<p>Tekanlah sembarang tombol Untuk mengakhiri program ini</p>



## Kuis Benar Salah

1. Deklarasi tipe data mempengaruhi besarnya tempat dalam memori yang disediakan untuk menyimpan data tersebut.
2. Tipe data real memiliki cakupan nilai antara  $2.9 \times 10^{-39}$  hingga  $1.1 \times 10^{4932}$ .
3. Penentuan nama variabel dan cara mengaksesnya dilakukan oleh program secara otomatis.
4. Output dari operasi aritmatik akan memiliki tipe data yang sama dengan kedua operannya.
5. Operator relasional dapat membandingkan dua nilai boolean.

(Untuk nomor 6-10) Perhatikan potongan algoritma berikut ini:

```

...
Kamus data
  a, b, d, e : integer
  c, f : real
  g : char
  h : boolean

```

#### Konstanta

```
d = 100;  
a ← 5  
b ← 3  
c ← 7.0  
a ← a + c  
e ← d / a  
ReadLn (g)  
h ← (g = 'k') OR (d MOD b > 3)  
f ← g + c  
Output (a, b, c, d, e, f, g, h)
```

6. Nilai akhir variabel a adalah 12.
7. Operasi  $e \leftarrow d / a$  tidak dapat dilakukan sehingga nilai e tidak dapat diketahui.
8. Jika pengguna memberikan input 'k' untuk variabel g, maka nilai akhir h adalah TRUE.
9. Variabel h tidak dapat diketahui nilainya.
10. Output program tersebut akan mengeluarkan empat nilai variabel dan empat error.
11. Tipe data word memiliki cakupan nilai yang lebih luas daripada tipe data integer.
12. Salah satu dari empat tipe data real dapat menyimpan nilai negatif.
13. String adalah kumpulan karakter dengan panjang tertentu.
14. Nilai dari sebuah konstanta tidak dapat diubah dalam badan program.
15. Pengisian nilai ke dalam sebuah variabel dapat dilakukan dengan dua cara, yaitu secara langsung atau dengan inputan.

(Untuk nomor 16-20) Perhatikan ekspresi berikut ini:

Hasil =  $(50 - 37) + 50 * 2 \% 35 > 21 / 3 \&\& ! (200 - 7 != 25 * 7 + 18)$

16. Operasi yang pertama kali dieksekusi adalah  $(50 - 37)$ .
17. Operator yang terakhir dieksekusi adalah '>'.
18. Hasil bertipe integer.
19. Hasil = 116.
20. Hasil = True.





## Pilihan Ganda

1. Berikut ini adalah pernyataan-pernyataan yang benar tentang tipe data, kecuali ... .
  - A. Tipe data adalah himpunan nilai yang dapat dimiliki oleh sebuah data.
  - B. Tipe data menentukan operasi apa yang dapat dilakukan pada data tersebut.
  - C. Pemilihan tipe data untuk sebuah variabel atau konstanta harus mempertimbangkan kapasitas memori yang tersedia.
  - D. Tipe data dari setiap variabel dan konstanta harus dideklarasikan di awal program.
  - E. Tipe data dasar dapat langsung digunakan dalam sebuah program.
2. Berikut ini adalah karakter yang sebaiknya tidak digunakan dalam penamaan variabel/ konstanta, kecuali ... .
  - A. '#'
  - B. '/'
  - C. ';'
  - D. ' '
  - E. '<'
3. Perhatikan ekspresi berikut ini:  
 $5 + 2 * 3 \geq 100 \text{ MOD } 45 \text{ OR } 75 - 30 / 2 < 10$   
Apakah output dari ekspresi tersebut?
  - A. 25
  - B. True
  - C. 11
  - D. False
  - E. 60
4. Perhatikan kedua kalimat tentang urutan eksekusi operator berikut ini:  
Kalimat A: Operasi dalam tanda kurung dieksekusi pertama kali.  
Kalimat B: Operasi relasional dieksekusi terlebih dahulu dibandingkan operasi aritmatik.  
Manakah pernyataan yang tepat tentang kedua kalimat di atas?

- A. Kalimat A dan kalimat B benar.
- B. Kalimat A dan kalimat B salah.
- C. Kalimat A benar dan kalimat B salah.
- D. Kalimat A salah dan kalimat B benar.
- E. Salah satu dari kalimat A atau kalimat B tidak dapat ditentukan benar/ salah.

5. Perhatikan potongan program berikut ini:

```
#include <stdio.h>
```

```
main () {  
    int bilangan1, bilangan2;  
    int bilangan3, bilangan4;  
  
    bilangan1 := 10;  
    bilangan1 := bilangan1 * 2;  
    bilangan2 := bilangan1 + 10;  
    bilangan3 := bilangan2 * bilangan1;  
  
    printf("%i",bilangan3);  
}
```

Manakah yang merupakan output dari program di atas?

- |        |        |
|--------|--------|
| A. 200 | D. 600 |
| B. 300 | E. 800 |
| C. 400 |        |



## Latihan

1. Jelaskan perbedaan variabel dan konstanta!
2. Perhatikan ekspresi berikut ini:  
 $4 + 7 * 10 - 5 \bmod 13$   
Apakah output dari ekspresi di atas?
3. Jelaskan tentang konsep runtunan dalam eksekusi sebuah program!
4. Perhatikan ekspresi berikut ini:  
 $x * 3 + (7 \bmod 5) * y$   
dengan  $x \leftarrow 5$  dan  
 $y \leftarrow 3$   
Apakah output dari ekspresi di atas?
5. Perhatikan potongan program berikut ini:

```
# include <stdio.h>

main () {
    int iPertama, iKedua;
    iPertama = 50;
    iPertama %= 9;
    iKedua = iPertama - 2;
    printf("%i",iKedua);
}
```

Apakah output dari program di atas?