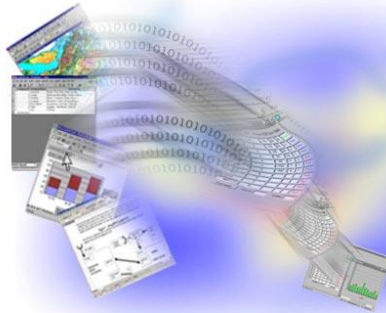


# I. Tipe Data, Operator dan Runtutan



## Overview

---

Tipe data, operator, dan runtutan merupakan suatu kesatuan konsep yang paling mendasar didalam pemrograman komputer, karena tipe-tipe data dasar dan operator dapat membentuk berbagai macam ekspresi yang akan digunakan dalam program.

Sedangkan runtutan merupakan konsep dasar yang dapat memberikan gambaran tentang cara kerja sebuah program dalam komputer.



## Tujuan

---

1. Mengenal dan membedakan tipe-tipe data dasar
2. Memahami dan dapat menggunakan tipe-tipe data dasar dalam program
3. Memahami dan dapat menggunakan operator dan penggunaannya dalam program
4. Memahami dan dapat menggunakan konsep runtutan dalam program

## 1.1 Tipe Data Dasar

Tipe data dasar adalah tipe data yang dapat langsung digunakan. Secara umum terdapat 2 tipe data dasar, yaitu **numerik** dan **kategorik**. Tipe data numerik terdiri atas angka/ kumpulan angka serta dapat mengalami operasi perhitungan, sedangkan tipe data kategorik dapat berupa angka maupun huruf namun tidak dapat mengalami operasi perhitungan.

Berikut merupakan contoh beberapa tipe data dasar :

- **Integer/ bilangan bulat**

Integer adalah tipe data dasar berupa bilangan yang tidak mengandung pecahan desimal.

Contoh integer:     2            5            -10        -135       2008

Dalam pemrograman yang menggunakan bahasa pemrograman C++, secara umum dikenal beberapa macam tipe data integer, yaitu:

*Tabel 1. Tipe data integer*

| Tipe         | Ukuran | Nilai                     |
|--------------|--------|---------------------------|
| <b>Short</b> | 8 bit  | -128 .. 127               |
| <b>Int</b>   | 16 bit | -32768 .. 32767           |
| <b>Long</b>  | 32 bit | -2147483648 .. 2147483647 |

- **Float/ bilangan pecahan**

Float adalah tipe data dasar berupa bilangan yang memiliki pecahan desimal. Dalam pemrograman, nilai dengan tipe data ini harus ditulis dengan sebuah titik sebagai pemisah bilangan utuh dan bilangan pecahannya.

Contoh real:            .5            0.17        -3.465    92.0        4.3000+E9

Dalam pemrograman, secara umum dikenal beberapa macam tipe data real, yaitu:

*Tabel 2. Tipe data real*

| Tipe          | Ukuran | Nilai   |
|---------------|--------|---|
| <b>float</b>  | 32 bit | $2.9 \times 10^{-39}$ .. $1.7 \times 10^{38}$   |
| <b>Double</b> | 48 bit | $5.0 \times 10^{-324}$ .. $1.7 \times 10^{308}$ |

Nilai pada tabel diatas berbeda dengan nilai yang ada pada tabel tipe data integer, pada tabel diatas nilai untuk tipe data merupakan tingkat ketelitian untuk masing-masing tipe data, bukan berdasarkan rentang nilai.

- **Char/ Karakter**

Char adalah tipe data dasar yang terdiri atas satu buah angka, huruf, tanda baca atau karakter khusus. Untuk menyimpan sebuah karakter, diperlukan 1 byte atau 8 bit tempat didalam memori. Dalam sebuah program, penulisan tipe data char diawali dan diakhiri dengan tanda kutip ganda.

Contoh char:        "5"        "A"        "?"        "+"        "\$"

Perhatikan bahwa 5 adalah integer sedangkan "5" adalah char.

- **String**

String adalah tipe data dasar yang berupa kumpulan karakter dengan panjang tertentu. Panjang string minimal adalah 0 artinya string tersebut bernilai *null*. Sama halnya dengan penulisan karakter, penulisan sebuah string juga harus diawali dan diakhiri dengan tanda petik ganda.

Contoh string:

- "Politeknik Telkom Bandung"
- "30202001"
- "z"
- ""

- **Boolean/ bilangan logika**

Sebuah data boolean memiliki tepat dua buah kemungkinan nilai, yang biasanya direpresentasikan sebagai Benar dan Salah, atau True dan False, atau dapat juga dilambangkan dengan 1 dan 0.

## 1.2 Variabel

Variabel atau peubah adalah obyek yang nilainya dapat berubah-ubah dalam sebuah program. Variabel harus dideklarasikan sebelum digunakan didalam program dengan tujuan untuk 'memesan' tempat dengan ukuran tertentu (sesuai tipe datanya) pada memori. Kita dapat memberikan nama yang mencirikan variabel tersebut untuk mempermudah pemanggilan di dalam program. Pada saat mendeklarasikan sebuah variabel, pemrogram harus menyebutkan nama variabel dan tipe data dari variabel tersebut.

Sebelum kita menuliskan beberapa program dalam bahasa C++, ada baiknya kita mengenal terlebih dahulu struktur dan format penulisan program dalam bahasa C++.

Buatlah contoh progrm dibawah ini :

```
1. // Contoh Program C++
```

```
2. #include <stdio.h>
3. #include <conio.h>
4. /* Program
5.    Utama */
6. main () {
7.    printf("Selamat Datang");
8.    getch();
9.    return 0;
10. }
```

Jika dijalankan, maka hasil dari program tersebut akan menampilkan kata “Selamat Datang”.

Pada contoh program diatas :

- Pada baris pertama dituliskan diawalannya tanda *doubleslash*. Maksudnya adalah sebagai komentar, artinya baris tersebut tidak akan dieksekusi oleh program. Selain dengan menggunakan “//”, kita juga dapat menggunakan “/\* ... \*/” dimana kita dapat menuliskan komentar diantara tanda tersebut. Dengan menggunakan yang kedua, kita dapat menuliskan komentar lebih dari 1 baris, seperti terlihat pada baris ke 3 dan 4.
- Pada baris kedua, code `#include <stdio.h>`, yang diawali dengan tanda crash (#). Ini berfungsi untuk memberitahukan kepada preprocessor untuk meng-includekan file `stdio.h`, karena file tersebut akan digunakan didalam program untuk melakukan perintah.
- Pada baris ke 6 – 10 merupakan isi dari program. Dimana program utama merupakan fungsi dengan nama `main()`. Program utama adalah tempat dimana kode yang pertama kali dieksekusi saat program dijalankan. Fungsi `main` tersebut harus ada untuk setiap kali kita membuat program. Pada baris ke-7 berfungsi untuk mencetak tulisan ke layar, baris ke-8 agar instruksi berhenti sehingga hasil dilayar dapat dilihat karena program tidak langsung berakhir, baris ke-9 merupakan nilai kembali dari fungsi `main`, sedangkan 10 adalah penutup fungsi `main`.

Hal yang perlu diingat adalah : Pemrograman bahasa C++ adalah *case sensitive*, artinya penulisan huruf besar berbeda dengan penulisan huruf kecil.

Untuk menuliskan variabel, kita dapat menuliskannya pada bagian isi program, dimana format menuliskannya adalah penulisan tipe data terlebih dahulu, selanjutnya diikuti oleh spasi kemudian nama variabel. Contoh program penulisan variabelnya adalah :

```

1.  #include <stdio.h>
2.  #include <conio.h>
3.
4.  main () { // Program Utama
5.      // Deklarasi Variabel
6.      string sNama;
7.      int iUmur;
8.      char jKelamin;
9.
10.     // Membaca Inputan
11.     printf("Masukan Nama : ");
12.     scanf("%i",&sNama);
13.     printf("Masukan Umur : ");
14.     scanf("%i",&iUmur);
15.     printf("Masukan Jenis Kelamin (L/P) : ");
16.     scanf("%i",&jKelamin);
17.
18.     // Menampilkan Inputan
19.     printf("Nama : %s", sNama);
20.     printf("Umur : %i", iUmur);
21.     printf("Jenis Kelamin : %c", jKelamin);
22.     getch();
23. }

```

Pengisian nilai ke dalam sebuah variabel dalam sebuah program dapat dilakukan dengan 2 cara, yaitu:

- **Secara langsung**

Contoh program memberikan nilai secara langsung :

```

1.  #include <stdio.h>
2.  main () { // Program Utama
3.      // Deklarasi Variabel
4.      int x;
5.      float TB;
6.      bool status;
7.
8.      // Inisialisasi Nilai secara Langsung
9.      x = 10;
10.     TB = 25.654;
11.     status = true;
12.
13.     // .....

```

```
14.  
15. }
```

- **Dengan inputan**

Contoh program memberikan nilai secara langsung :

```
1. #include <stdio.h>  
2. main () { // Program Utama  
3. // Deklarasi Variabel  
4.     int x;  
5.     string nama;  
6.  
7.     // Meminta Inputan Nilai  
8.     scanf("%i", &x);  
9.     scanf("%s", &nama);  
10.  
11.     // ...  
12.  
13. }
```

### 1.3 Konstanta

Berbeda dengan variabel, konstanta memiliki nilai yang tetap dan tidak dapat diubah sejak dideklarasikan hingga program berakhir.

Cara penulisan konstanta didalam program, di tulis dengan diawali dengan tanda crash (#) kemudian diikuti dengan `define`, selanjutnya nama konstantanya dan selanjutnya nilainya dan ditulis diluar program utama setelah pendeklarasian librari namespace. Contoh penulisannya adalah sebagai berikut :

```
1. ...  
2. #define iMaxs 100  
3. #define rPhi 3.14159  
4. #define sSapa 'Hello'  
5. #define newLine '\n'  
6. ...
```

Contoh program menghitung luas lingkaran berikut :

```
1. #include <stdio.h>  
2. #include <conio.h>  
3.  
4. #define Pi 3.14159  
5. #define NewLine '\n'  
6.
```

```

7.  main () {
8.      int jari;
9.      float luas;
10.
11.     printf("Program Menghitung Luas Lingkaran
    %c", NewLine);
12.     printf("Masukan Jari-Jari Lingkaran");
13.     scanf("%i", &jari);
14.     luas = Pi * jari * jari;
15.     printf("Luas Lingkaran dengan Jari-jari
    %i adalah %f", jari, luas);
16.     getch();
17. }

```

Pada baris ke-4 kita mendefinisikan “\n” dengan nama `NewLine`. Didalam pemrograman dengan bahasa C++ kita mengenal istilah karakter spesial. Karakter ini dikatakan spesial karena karakter ini tidak terdapat pada keyboard. Untuk menuliskan karakter spesial diawali dengan tanda *backslash* (\), kemudian diikuti karakter yang melambangkannya. Misalnya :

- \n : untuk Baris Baru (enter)
- \t : untuk tabulasi (tab)
- \r : akhir dari string
- \0 : *null terminated* dalam array
- \a : untuk menghasilkan bunyi beep
- Dll.

## 1.4 Operator

Operator adalah pengendali operasi yang akan dilakukan pada beberapa operan sehingga membentuk sebuah ekspresi.

- **Operator aritmatik**

| Lambang | Deskripsi        | Contoh       |
|---------|------------------|--------------|
| +       | Penjumlahan      | $x = y + z$  |
| -       | Pengurangan      | $x = y - z$  |
| *       | Perkalian        | $x = y * z$  |
| /       | Pembagian        | $x = y / z$  |
| %       | Modulo (sisanya) | $x = y \% z$ |

Catatan : Output dari operasi aritmatik akan memiliki tipe data yang sama dengan tipe data kedua operannya.

Contoh program dengan operasi aritmatik:

```
1  /* Contoh :
2     Program Aritmatik*/
3  #include <stdio.h>
4  #include <conio.h>
5
6  main () {
7      // Deklarasi Variabel
8      int iAngka1, iAngka2;
9      int iTambah, iKurang, iMod;
10     float fBagi;
11
12     printf("Masukan Angka Pertama : ");
13     scanf("%i", &iAngka1);
14     printf("Masukan Angka kedua : ");
15     scanf("%i", &iAngka2);
16
17     // Penjumlahan
18     iTambah = iAngka1 + iAngka2;
19     iKurang = iAngka1 - iAngka2;
20     iMod = iAngka1 % iAngka2;
21     fBagi = iAngka1 / iAngka2;
22
23     printf("Hasil %i + %i adalah %i",
24           iAngka1, iAngka2, iTambah);
25     printf("Hasil %i - %i adalah %i",
26           iAngka1, iAngka2, iKurang);
27     printf("Hasil %i mod %i adalah %i",
28           iAngka1, iAngka2, iMod);
29     printf("Hasil %i / %i adalah %f",
30           iAngka1, iAngka2, fBagi);
31     return 0;
32     getch();
33 }
```

- **Operator Assignment**

Dalam pemrograman bahasa C++, Operator ini digunakan memasukan nilai kedalam sebuah variabel, tanpa menghilangkan atau



mengosongkan nilai variabel sebelumnya. Contoh penggunaan operator ini adalah sebagai berikut :

*Tabel 4. Operator relasional*

| Lambang   | Deskripsi    | Contoh |
|-----------|--------------|--------|
| <b>+=</b> | Menambahkan  | x += 1 |
| <b>-=</b> | Mengurangkan | x -= 1 |
| <b>*=</b> | Mengalikan   | x *= 2 |
| <b>/=</b> | Membagi      | x /= 2 |
| <b>%=</b> | Mem-mod      | x %= 2 |

Contoh :

```

1  #include <stdio.h>
2  #include <conio.h>
3  main () {
4      int i = 5;
5      printf("Nilai i sebelumnya : %i",i)
6      i += 3;
7      printf("Nilai i setelahnya : %i",i)
8      getch();
9  }
```

- **Increase and decrease**

Penulisan ini dilambangkan dengan ++ (Increase) dan -- (decrease). Operator ini berfungsi untuk menaikkan atau menurunkan satu satuan nilai pada sebuah variabel.

Ada dua macam penulisan operator ini, yaitu simbol dapat ditulis sebelum nama variabel dan setelah variabel. Adapun perbedaan antara keduanya adalah :

|  |  |
|--|--|
| <pre> 1  #include &lt;stdio.h&gt; 2  #include &lt;conio.h&gt; 3  main () { 4      int b = 3; 5      int a; 6      a = ++b; 7      printf("Nilai A = %i \n Nilai B = %i",a,b); 8      getch(); 9      return 0;</pre> | <pre> 1  #include &lt;stdio.h&gt; 2  #include &lt;conio.h&gt; 3  main () { 4      int b = 3; 5      int a; 6      a = b++; 7      printf("Nilai A = %i \n Nilai B = %i",a,b); 8      getch(); 9      return 0;</pre> |
|--|--|

|      |      |
|------|------|
| 10 } | 10 } |
|------|------|

Perhatikan perbedaan kedua program tersebut dan perhatikan perbedaan hasilnya.

- **Operator relasional**

Operator ini membandingkan dua operan dan hasilnya berupa nilai boolean (BENAR atau SALAH). Operasi relasional dapat dilakukan pada dua nilai dengan tipe data yang sama: tipe data integer, riil, char, string, maupun boolean. Tipe data ini sering digunakan dalam pengkondisian. Berikut ini adalah operator relasional:

*Tabel 4. Operator relasional*

| Lambang      | Deskripsi                    | Contoh                 |
|--------------|------------------------------|------------------------|
| <b>==</b>    | Sama dengan                  | <code>x == y</code>    |
| <b>!=</b>    | Tidak sama dengan            | <code>x != y</code>    |
| <b>&gt;</b>  | Lebih dari                   | <code>x &gt; y</code>  |
| <b>&lt;</b>  | Kurang dari                  | <code>x &lt; y</code>  |
| <b>&gt;=</b> | Lebih dari atau sama dengan  | <code>x &gt;= y</code> |
| <b>&lt;=</b> | Kurang dari atau sama dengan | <code>x &lt;= y</code> |

Contoh penggunaan operator relasional dalam program :

```

1  #include <stdio.h>
2  #include <conio.h>
3  main () {
4      int iAngka1, iAngka2;
5      iAngka1 = 6 //pengisian variabel langsung
6      printf("Masukan Angka Tebakan : ");
7      scanf("%i", &iAngka2);
8      if (iAngka1 == iAngka2) {
9          printf("Tebakan Anda Benar");
10     } else {
11         printf("Tebakan Anda Salah");
12     }
13     getch();
14 }
```

- **Operator logika**

Operator logika adalah operator yang digunakan untuk mengkombinasikan hasil ekspresi yang mengandung operator relasional.

Tiga macam operator logika adalah:

Tabel 5. Operator logika

| Lambang | Deskripsi   | Contoh                  |
|---------|-------------|-------------------------|
| &&      | And / Dan   | $x > 7 \ \&\& \ x = y$  |
|         | Or / Atau   | $x \neq y \    \ x > 3$ |
| !       | Not / Tidak | $! (x > y)$             |

Pola penggunaan operator logika adalah:

*ekspresi1* **OPERATOR** *ekspresi2*

Output dari penggunaan operator AND dan OR adalah sebagai berikut:

Tabel 6. Output operator logika

| ekspresi1 | ekspresi2 | Kombinasi dengan |       |
|-----------|-----------|------------------|-------|
|           |           | AND              | OR    |
| True      | True      | True             | True  |
| True      | False     | False            | True  |
| False     | True      | False            | True  |
| False     | False     | False            | False |

Pola yang mudah untuk mengingat output kedua operator logika tersebut adalah: True AND True = True, False OR False = False.

Contoh penggunaan operator logika dalam program :

```
1  #include <stdio.h>
2  #include <conio.h>
3  main () {
4      int iNilai;
5      printf("Masukan Nilai : ");
6      scanf("%i", & iNilai);
7      if (iNilai <= 100 && iNilai >= 0) {
8          printf("Nilai Valid");
9      } else {
10         printf("Nilai Tidak Valid");
11     }
12     getch();
13 }
```



## Latihan

---

1. Buatlah program persamaan linier yang menerima inputan  $x$  dan menghasilkan nilai  $y$  dengan persamaan  $y = 3x + 4$  :
2. Buatlah program menghitung keliling lingkaran dimana program menerima inputan jari-jari lingkaran.
3. Buatlah program menghitung luas segitiga dimana program menerima inputan lebar alas dan tinggi.
4. Buatlah program yang memeriksa apakah sebuah garis persamaan linier bersinggungan dengan persamaan  $x = y$  dimana inputan program adalah persamaan parameter.
5. Buatlah program yang dapat menghitung rentang waktu untuk perjalanan kendaraan, dimana program menerima inputan kecepatan perjalanan dan jarak tempuh perjalanan.
6. perbaiki program berikut ini sehingga bisa dijalankan :

```
#include <stdio.h>
main () {
    int iPertama, iKedua;
    float fHasil;
    printf("Masukan Bilangan Pertama : ");
    scanf(iPertama);
    printf("Masukan Bilangan Pertama : ");
    scanf(iKedua);
    fHasil = iPertama / iKedua;
    printf("Hasilnya : ",fHasil);
}
```