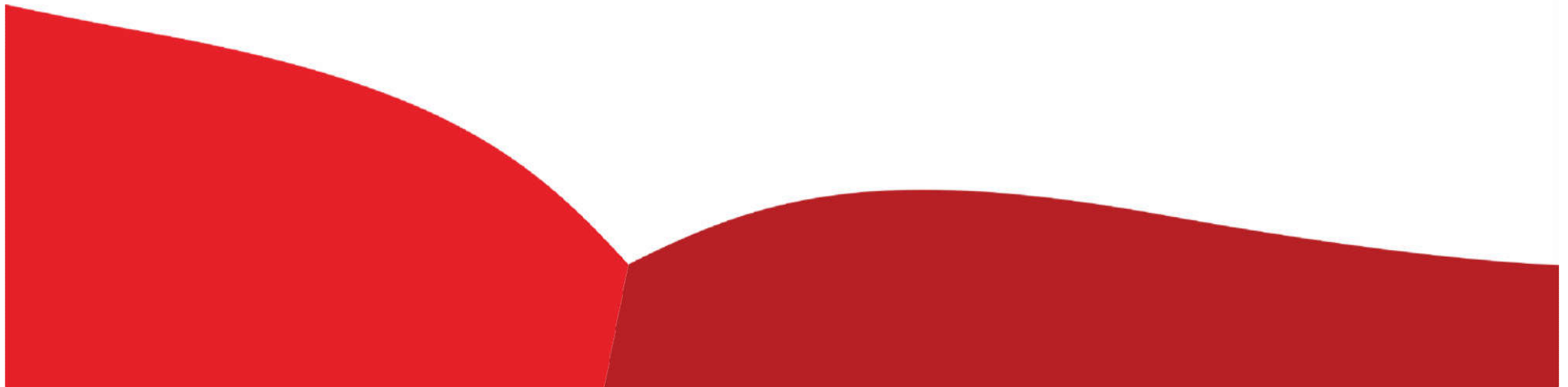


Pengenalan Bahasa C++

Oleh : Agus Priyanto, M.Kom



Tujuan Pembelajaran

- Mahasiswa dapat memahami konsep dasar pemrograman dengan bahasa C++
- Mahasiswa dapat membuat aplikasi sederhana menggunakan bahasa C++, secara khusus untuk bidang informatika



Bahasa Pemrograman

- Komputer bekerja seperti **switching** dan hanya mengenali 0 dan 1
- Manusia **tidak (paham) berbicara** dengan bahasa 0 dan 1



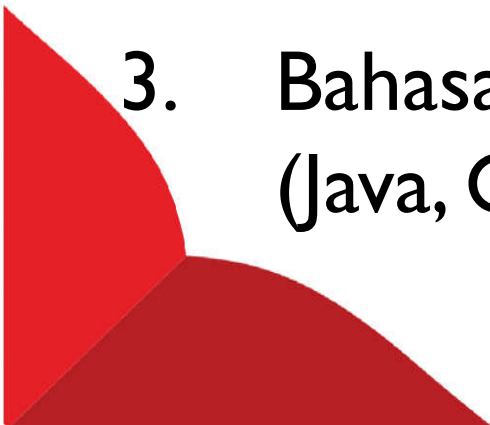
- Perlu bahasa pemrograman yang dapat menjadi **perantara percakapan** antara komputer dan manusia
- Bahasa pemrograman diubah ke dalam bahasa yang dipahami oleh komputer dengan menggunakan **interpreter** atau **kompiler**



Tingkat Bahasa Pemrograman



1. Bahasa Pemrograman Tingkat **Rendah**
(Assembler)
2. Bahasa Pemrograman Tingkat **Sedang**
(C, Pascal, Fortran)
3. Bahasa Pemrograman Tingkat **Tinggi**
(Java, C++, C#)



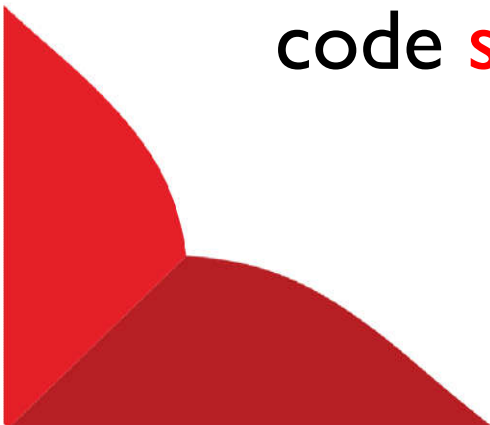
Compiler or Interpreter

1. Compiler

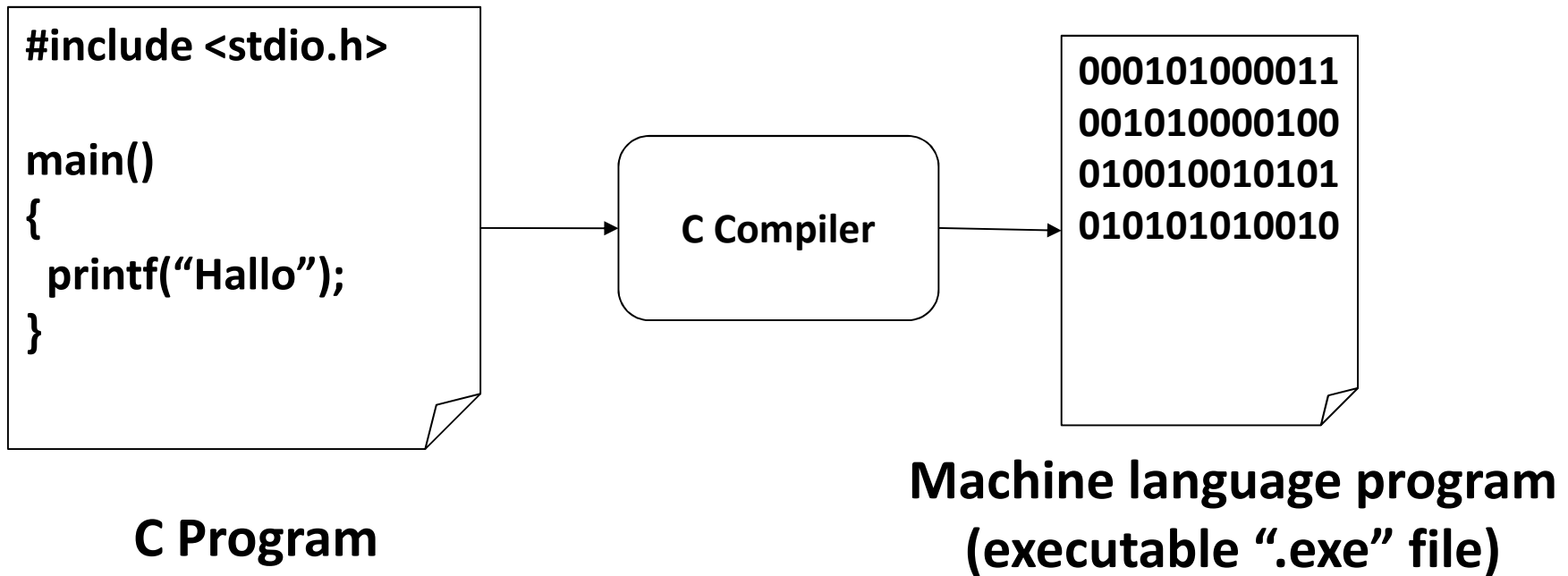
Mengkompilasi source code menjadi bentuk
file yang bisa dieksekusi

2. Interpreter

Mengkompilasi dan menjalankan source
code secara langsung



C Language (Compiler)



Sejarah Bahasa C++

- Tahun 1978, Brian W. Kernighan & Dennis M. Ritchie dari AT & T Laboratories mengembangkan bahasa **B** menjadi bahasa **C**
- Bahasa **B** yang diciptakan oleh Ken Thompson merupakan pengembangan dari bahasa **BCPL** (Basic Combined Programming Language) yang diciptakan oleh Martin Richard.



- Tahun 1989, dikeluarkannya standar bahasa C oleh American National Standards Institute (ANSI). Bahasa C yang diciptakan Kernighan & Ritchie kemudian dikenal dengan nama **ANSI C**

- 
- Tahun 1980, Bjarne Stroustrup dari AT & T Bell Laboratories mulai mengembangkan bahasa C

- Tahun 1983, lahirlah secara resmi bahasa baru hasil pengembangan C yang dikenal dengan nama C++.
- Bahasa ini bersifat kompatibel dengan bahasa pendahulunya yaitu C. Pada mulanya C++ disebut dengan “a better C”.
- Nama C++ sendiri diberikan oleh Rick Mascitti pada musim panas 1983. Adapun tanda ++ berasal dari nama operator penaikan pada bahasa C.



Kenapa C++ ?

- Mudah dan familiar dalam membuat program dengan konsep Object Oriented Programming
- Secara De-Facto merupakan standar bahasa pemrograman yang digunakan dalam dunia pendidikan
- Architecture neutral (platform independent)
- Rank in TIOBE Index

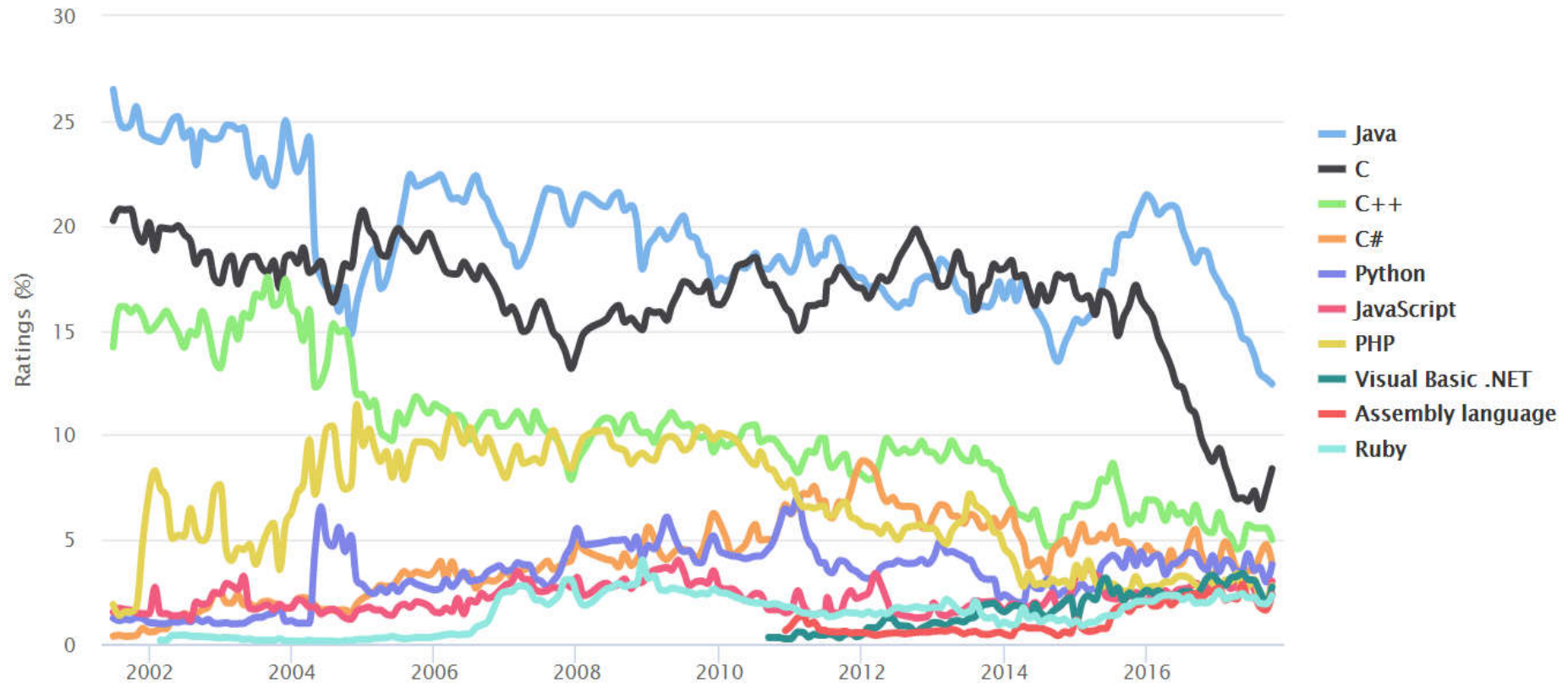


Oct 2017	Oct 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.431%	-6.37%
2	2		C	8.374%	-1.46%
3	3		C++	5.007%	-0.79%
4	4		C#	3.858%	-0.51%
5	5		Python	3.803%	+0.03%
6	6		JavaScript	3.010%	+0.26%
7	7		PHP	2.790%	+0.05%
8	8		Visual Basic .NET	2.735%	+0.08%
9	11	⬆	Assembly language	2.374%	+0.14%
10	13	⬆	Ruby	2.324%	+0.32%
11	15	⬆	Delphi/Object Pascal	2.180%	+0.31%
12	9	⬇	Perl	1.963%	-0.53%
13	19	⬆	MATLAB	1.880%	+0.26%
14	23	⬆	Scratch	1.819%	+0.69%
15	18	⬆	R	1.684%	-0.06%

Rating
Oktober
2017

TIOBE Programming Community Index

Source: www.tiobe.com



TIOBE Indeks Programming Community merupakan indikator popularitas bahasa pemrograman. Indeks diperbaharui sebulan sekali. Peringkat tersebut didasarkan pada jumlah tenaga ahli di seluruh dunia, pendidikan dan kursus serta vendor pihak ketiga.

Struktur Bahasa C++

Hellow.cpp

```
// my first program in C++  
  
#include <iostream.h>  
int main ()  
{  
    cout << "Hello World!";  
    return 0;  
}
```



Keterangan :

1. `// my first program in C++`

- Baris ini adalah komentar
- Semua baris yang diawali dengan dua garis miring (`//`) akan dianggap sebagai komentar dan tidak akan berpengaruh terhadap program

2. `#include <iostream.h>`

- Kalimat yang diawali dengan tanda (`#`) adalah preprocessor directive
- Bukan merupakan baris kode yang dieksekusi, tetapi indikasi untuk kompiler



3. **int main()**

- Fungsi **main** merupakan titik awal dimana seluruh program C++ akan mulai dieksekusi
- Diletakan diawal, ditengah atau diakhir program, isi dari fungsi main akan selalu dieksekusi pertama kali
- Pada dasarnya, seluruh program C++ memiliki fungsi **main**



4. **cout << "Hello World";**

- **cout** merupakan standard output stream dalam C++ (biasanya monitor)
- **cout** dideklarasikan dalam header file `iostream.h`, sehingga agar dapat digunakan maka file ini harus disertakan

5. **return 0;**

- Instruksi **return** menyebabkan fungsi `main()` berakhir dan mengembalikan kode yang mengikuti instruksi tersebut



Contoh : Wellcome to C++

```
#include <iostream.h>

int main()
{
    cout << "Welcome to C++!\n";
    return 0;
}
```



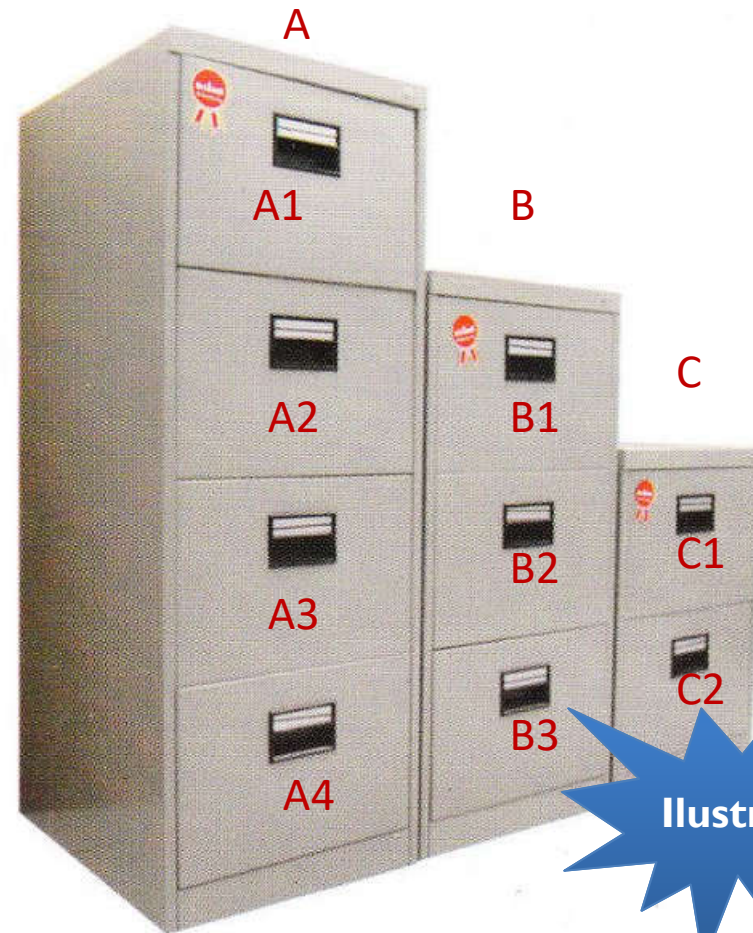
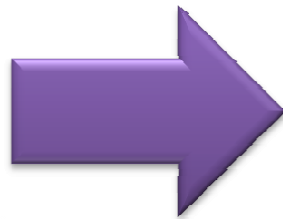
```
#include <iostream.h>

int main()
{
    cout << "Welcome ";
    cout << "to C++!\n";
    return 0;
}
```

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.





Syarat penamaan variabel

Untuk C++ dan Java, nama variabel bersifat **case sensitive**.

Harus diawali dengan huruf

Boleh diikuti dengan huruf/angka/symbol

Tidak boleh ada spasi

Bukan merupakan keyword dalam sebuah bahasa pemrograman

misalkan read, write (pascal), private, protected (java), cout, cin (c++)



Perhatikan!

$$X = 10 + Y$$

Diagram illustrating the equation $X = 10 + Y$ with annotations:

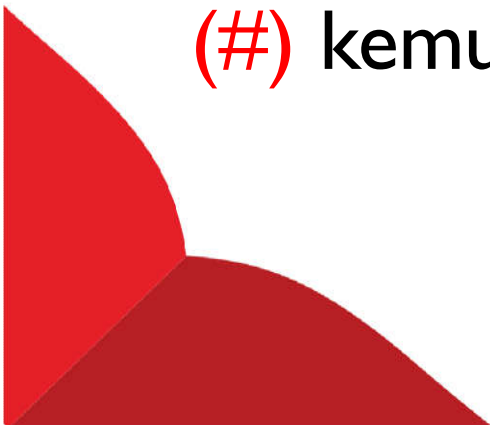
- variabel** (variable): Points to X and Y (circles in red).
- konstanta** (constant): Points to 10 (circle in green).

- Berapakah nilai X , jika $Y=3$?
- Berapakah nilai X , jika $Y=5$?
- Berapakah nilai Y , jika $X=15$?



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**.

```
#include <iostream.h>

#define iMaxs 100
#define rPhi 3.14159
#define sSapa "Hello"
#define newLine "\n"
```





```
#include <iostream.h>

int main()
{
    int jari;
    float luas;
    cout<<"Program Menghitung Luas Lingkaran : \n";
    cout<<"Masukan Jari-Jari Lingkaran";
    cin>>jari;
    luas = 3,14 * jari * jari;
    Cout<<"Luas  Lingkaran Jari-jari %i adalah %f", jari, luas;
    return 0;
}
```

Contoh



Karakter Escape

- Dalam bahasa C++, juga terdapat karakter khusus yang disebut karakter escape:

Karakter	Keterangan
\0	Karakter NULL (Tulisannya berupa \ dan angka nol)
\a	Karakter bel atau alert
\b	Karakter <i>backspace</i> (mundur satu spasi)
\f	<i>Formfeed</i> (ganti halaman)
\n	<i>Linefeed</i> (disebut juga <i>newline</i> atau pindah baris)
\r	<i>Carriage return</i> (menuju kolom pertama, baris yang sama)
\t	Tab horizontal
\v	Tab vertikal
\\	Karakter \
\'	Karakter petik tunggal
\"	Karakter petik ganda
\?	Karakter tanda tanya

Tipe Data

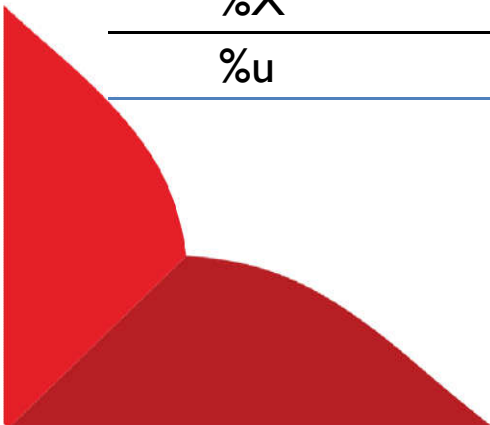
Tipe	Kapasitas Memori	Size (Range)	Keterangan
char	1 byte	-128 s/d 127	Character
int	2 byte	-32768 s/d 32768	Integer (Bilangan)
long	4 byte	-21474386 s/d 21474386	Bilangan Bulat
float	4 byte	3.4E-38 s/d 3.4+38	Real presisi tunggal
double	8 byte	1.7E-308 s/d 1.7E+308	Real presisi ganda



Tipe Data
Dasar

Tabel penentu format data dalam C++

Kode Penentu Format	Keterangan
%c	Menampilkan karakter
%d atau %i	Menampilkan bilangan integer bertanda dalam desimal
%f	Menampilkan bilangan float (pecahan)
%g	Menampilkan bilangan double (pecahan)
%s	Menampilkan karakter (string)
%o	Menampilkan bilangan integer tidak bertanda dalam oktal
%X	Menampilkan bilangan integer tidak bertanda dalam hexsadesimal
%u	Menampilkan bilangan tidak bertanda (unsigned)



Contoh Penggunaan Tipe Data

Tipe data : bilangan bulat

Satu variabel

```
int lebar;
```

Lebih dari satu variabel

```
int lebar, tinggi;
```

Konstanta

```
int lebar = 3;
```



```
lebar; int
```



Tipe data : bilangan pecahan

Satu variabel

```
float tinggi;
```

Lebih dari satu variabel

```
float lebar, panjang, tinggi;
```

Konstanta

```
float tinggi = 3.5;
```



tinggi; float



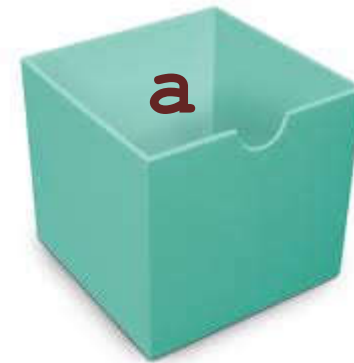
Tipe data : character

Satu variabel
`char kode;`

Lebih dari satu variabel
`char kode, huruf, angka;`

Konstanta

`char kode = 'a'; [petik satu]`
`char kode[4] = "aaaa"; [petik dua]`



`Kode; char`

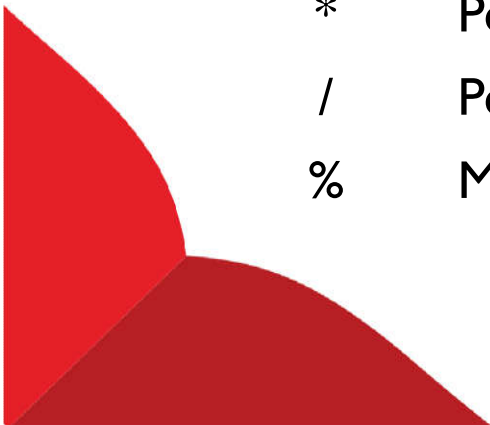


Operator

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

- **Operator aritmatik**

+	Penjumlahan	$x = y + z$
-	Pengurangan	$x = y - z$
*	Perkalian	$x = y * z$
/	Pembagian	$x = y / z$
%	Modulo (siswa bagi)	$x = y \% z$




```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    int iAngka1, iAngka2, iTambah,iKurang,iMod;
```

```
    float fBagi;
```

```
    cout<<"Masukan Angka Pertama : \n";
```

```
    cin>>iAngka1;
```

```
    cout<<"Masukan Angka kedua : \n";
```

```
    cin>>iAngka2;
```

```
    iTambah = iAngka1 + iAngka2;
```

```
    iKurang = iAngka1 - iAngka2;
```

```
    iMod = iAngka1 % iAngka2;
```

```
    fBagi = iAngka1 / iAngka2;
```

```
    cout<<"Hasil  %i  +  %i  adalah  %i", iAngka1, iAngka2, iTambah;
```

```
    cout<<"Hasil  %i  -  %i  adalah  %i", iAngka1, iAngka2, iKurang;
```

```
    cout<<"Hasil  %i  mod  %i  adalah  %i", iAngka1, iAngka2, iMod;
```

```
    cout<<"Hasil  %i  /  %i  adalah  %f", iAngka1, iAngka2, fBagi;
```

```
    return 0;
```

```
}
```



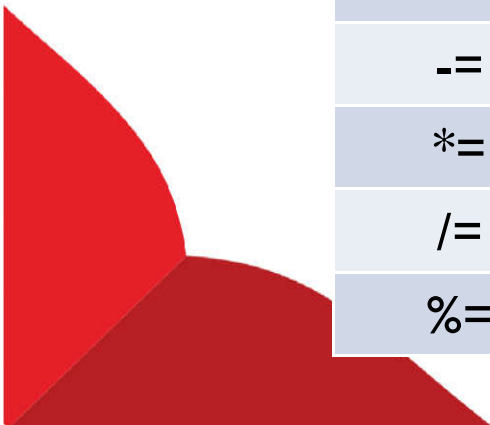
Contoh



- **Operator Assignment**

Operator ini digunakan memasukan nilai kedalam sebuah variabel, tanpa menghilangkan atau mengosongkan nilai variabel sebelumnya

Notasi	Deskripsi	Contoh
<code>+=</code>	Menambahkan	<code>x += 1</code>
<code>-=</code>	Mengurangkan	<code>x -= 1</code>
<code>*=</code>	Mengalikan	<code>x *= 2</code>
<code>/=</code>	Membagi	<code>x /= 2</code>
<code>%=</code>	Mem-mod	<code>x %= 2</code>



```
#include <iostream.h>

int main()
{
    int i = 5;
    cout<<"Nilai i sebelumnya : %i",i
    i += 3;
    cout<<"Nilai i setelahnya : %i",i
    return 0;
}
```

Contoh



- **Increase and decrease**

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



```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    int b = 3;
```

```
    int a;
```

```
    a = ++b;
```

```
    cout<<"Nilai A = %i  \n  Nilai  B  = %i",a,b;
```

```
    return 0;
```

```
}
```

```
#include <iostream.h>
```

```
int main()
```

```
{
```

```
    int b = 3;
```

```
    int a;
```

```
    a = b++;
```

```
    cout<<"Nilai A = %i  \n  Nilai  B  = %i",a,b;
```

```
    return 0;
```

```
}
```

Contoh



- **Operator relasional**

Operator ini membandingkan dua operan dan hasilnya berupa nilai boolean (BENAR atau SALAH)

Notasi	Deskripsi	Contoh
==	Sama dengan	$x == y$
!=	Tidak sama dengan	$x != y$
>	Lebih dari	$x > y$
<	Kurang dari	$x < y$
>=	Lebih dari atau sama dengan	$x >= y$
<=	Kurang dari atau sama dengan	$x <= y$



```
#include <iostream.h>

int main()
{
    int iAngka1, iAngka2;
    iAngka1 = 6;
    cout<<"Masukan Angka Tebakan : \n";
    cin>>iAngka2;
    if (iAngka1 == iAngka2) {
        cout<<"Tebakan Anda Benar";
    } else {
        cout<<"Tebakan Anda Salah";
    }
    return 0;
}
```

Contoh



- **Operator logika**

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

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



Output dari penggunaan operator AND dan OR adalah sebagai berikut:

True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False



```
#include <iostream.h>

int main()
{
    int iNilai;
    cout<<"Masukan Nilai : \n";
    cin<<iNilai;
    if (iNilai <= 100 && iNilai >= 0)
    {
        cout<<"Nilai Valid";
    } else {
        cout<<"Nilai Tidak Valid";
    }
    return 0;
}
```

Contoh



Jawablah !!!

- Tuliskan tipe data yang sesuai untuk data-data berikut ini (gunakan bahasa C++) !



Jumlah anak dalam keluarga



Gaji pegawai

NAMA

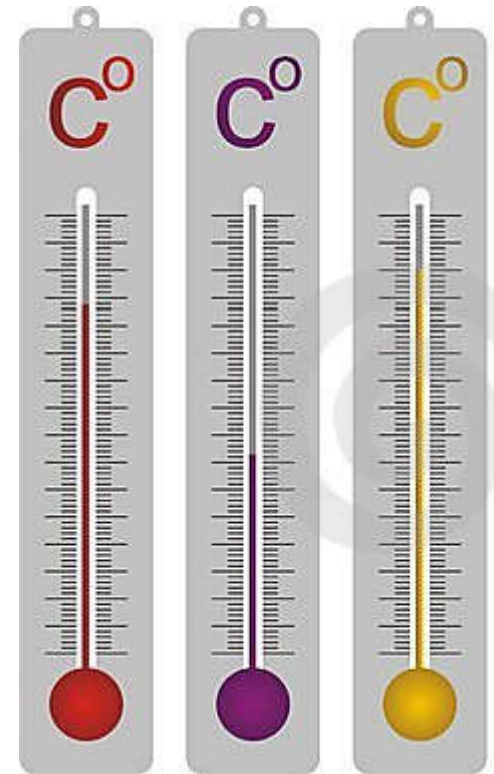
Nama mahasiswa



Nilai huruf



Berat badan



Suhu udara



Latihan di kelas



I. Menghitung luas segitiga

Input : alas, tinggi

Output : luas

Algoritma luas segitiga :

- Masukkan alas, tinggi
- $\text{Luas} \leftarrow \text{alas} * \text{tinggi} * 0.5$
- Tampilkan luas



- a. Tentukanlah tipe data yang tepat untuk alas, tinggi dan luas!
- b. Tuliskan dalam bahasa C++ penamaan variabel yang tepat untuk alas, tinggi dan luas!
- c. Ubahlah algoritma menghitung luas segitiga ke dalam bahasa C++!



2. Konversi suhu Celcius ke Reamur

$$C = \frac{5}{4} R$$

- Tentukanlah tipe data yang tepat untuk r !
- Tuliskan dalam bahasa C++ penamaan variabel yang tepat untuk r !
- Ubahlah algoritma konversi Celcius ke Reamur ke dalam bahasa C++!

KUMPULKAN !





