

Nama : Mohamad Alit Zikrillah Maulana

BAHASA PEMROGRAMAN C++

Konsep Dasar Bahasa Pemrograman C++

Kalau Anda ingin belajar C++, penting untuk memahami konsep dasar bahasa pemrograman C++ berikut ini:

Variabel

Variabel adalah penanda identitas yang digunakan untuk menampung suatu nilai. Artinya, variabel akan menunjukkan suatu lokasi yang ada di memori komputer atau RAM. Jadi, saat Anda membuat satu variabel, akan ada satu slot memori untuk menampung nilai tersebut.

Bahasa C++ termasuk bahasa pemrograman dengan konsep **strongly typed**. Artinya, saat melakukan deklarasi, Anda harus memberi tipe data di dalam setiap variabel.

Berikut ini adalah tipe-tipe data yang ada di dalam variabel C++:

- **bool**: tipe data berupa nilai boolean, yaitu True or False
- **char**: tipe data huruf dari A sampai Z
- **int**: tipe data berupa angka
- **float** dan **double**: tipe data berupa angka pecahan, contohnya 1,33
- **string**: tipe data berupa kumpulan karakter seperti "belajar bahasa c++"

Lalu bagaimana cara menuliskan variabel dalam bahasa C++? Ada dua langkah untuk melakukannya, yaitu **deklarasi** dan **inisialisasi**.

Deklarasi dilakukan sebelum program dimulai. Namun, Anda juga bisa melakukan deklarasi di tengah program. Akan tetapi, deklarasi harus sudah selesai sebelum Anda menggunakan program. Berikut ini adalah contoh deklarasi:

```
int i;    // contoh deklarasi
char c; int i, j, k; // banyak
deklarasi
```

Sedangkan **inisialisasi** artinya mengisi nilai di dalam variabel yang sudah dideklarasikan. Contohnya seperti ini:

```
int i; // deklarasi i
= 10; // inisialisasi
```

Jika dilakukan secara bersamaan, hasilnya seperti ini:

```
int i=10;    // contoh deklarasi dan inisialisasi
int i=10, j=11;
```

Jika Anda melakukan deklarasi tanpa inisialisasi, atau melakukan deklarasi ulang pada variabel yang ada dalam satu cakupan, maka akan muncul error. Contohnya seperti ini:

```
int i,j;
i=10; j=20; int j=i+j; //
compile time error
```

Seluruh variabel yang Anda tulis memiliki cakupan fungsinya masing-masing. Artinya, variabel tidak akan berfungsi jika Anda menuliskannya di luar cakupan. Cakupan suatu variabel biasanya berupa kurung kurawal.

Ada dua cakupan dalam bahasa C++, yaitu **Variabel Global** dan **Variabel Lokal**.

Variabel global adalah variabel yang ketika Anda deklarasikan, bisa digunakan lagi dalam class dan function apapun selama program masih berjalan. Variabel ini dideklarasikan di luar function main(). Contohnya seperti ini:

```
#include <iostream>
using namespace std;
int x;          // Deklarasi variabel global
int main()
{
    x=10;        // inisialisasi
    pertama     cout <<"first value of x =
    "<< x;      x=20;          // inisialisasi
    kedua
    cout <<"diinisialisasi lagi dengan value = "<< x; }
```

Sedangkan **variabel lokal** adalah variabel yang hanya bisa berfungsi jika dideklarasikan di dalam function main(). Artinya, jika dideklarasikan di luar main () maka Anda akan menemui error.

```
#include <iostream> using
namespace std;
int main()
{
    int
    i=10;
    if(i<20)    // batas awal cakupan
    {
        int n=100; // variabel lokal dideklarasikan dan diinisialisasi
    }           // batas akhir cakupan
    cout << n;   // Compile time error
}
```

Control Structure

Control structure adalah konsep dasar untuk menentukan keputusan dalam setiap baris pembacaan kode atau code flow.

Bagaimana cara kerjanya?

Saat program membaca baris kode satu persatu menggunakan compiler, compiler akan menemukan titik pengambilan keputusan. Titik keputusan itu berupa penyeleksian dan pengulangan kode.

Nah, control structure akan membantu compiler menetapkan keputusan apakah penyeleksian atau perulangan itu akan dipecahkan atau dilewati.

Berikut ini adalah **control structure penyeleksian** bahasa pemrograman C++:

- **Statement if**

Statement if digunakan untuk mencari kebenaran dari conditional expression yang berupa nilai boolean dengan nilai True atau False. Begini contohnya:

```
#include< iostream.h>
int main( )
{   int
x,y;
x=37;
y=25;
    if (x > y )
    {
        cout << "x lebih besar daripada y";
    }
}
```

- **Statement else if**

Ketika compiler menemukan penyeleksian dengan statement if, CPU akan memeriksa kebenaran dari conditional expression. Jika hasilnya True, maka perintah kode akan dijalankan. Namun, jika hasilnya False, maka compiler akan memeriksa kode dengan statemen else if.

```
void main( )
{   int
x,y;
x=15;
y=19;
    if (x > y )
    {
        cout << "x lebih besar daripada y";
    }
    else
    {
        cout << "y lebih besar daripada x";
    }
}
```

Pemeriksaan akan dilakukan secara berulang di setiap baris kode hingga menemukan nilai True. Jika tidak ada nilai True, maka compiler akan melakukan perintah statemen else. Namun, apabila statemen else juga tidak memiliki nilai True, maka pernyataan if akan diabaikan seluruhnya.

Selain memiliki control structure untuk penyeleksian, bahasa C++ juga memiliki **control structure untuk pengulangan atau loops**, yaitu:

- **While**

While adalah statemen yang digunakan untuk mengulangi sejumlah perintah selama conditional expression bernilai True. Di bawah ini adalah contohnya:

```
int x=0 ; while
(x<=10)
{
    statements;  x++
or x-- or x=x+2; }
```

- **Do While**

Do while adalah statemen untuk mengeksekusi sebuah perintah, baru kemudian memeriksa conditional expression berupa bilangan boolean. Statemen **do** akan mengeksekusi perintah, kemudian **while** akan memeriksa conditional expression.

Contoh statemen do while:

```
do{
    pernyataan;
    ...
}while(conditional expression);
```

Jadi pada saat compiler bertemu dengan statemen do while, compiler akan langsung mengeksekusi perintah yang ada di dalam badan do while. Untuk pengulangan perintah, compiler akan memeriksa conditional expression terlebih dahulu.

Jika conditional expression bernilai 1 atau true, maka compiler akan kembali mengeksekusi perintah. Pengulangan akan terus berlanjut hingga conditional expression bernilai 0 atau false.

- **For**

For adalah statemen untuk mengeksekusi banyak perintah secara berulang-ulang. Artinya, statemen ini berguna untuk membuat program yang dinamis. Berikut adalah contoh penggunaannya:

```
for(initialization; condition; increment/decrement)
{
    statement-block;
}
```

Data Structure

Data structure adalah fitur yang berfungsi untuk membuat deklarasi berisi kelompok variabel dengan tipe data berbeda. Data structure memiliki statemen struct. Dengan fitur ini, Anda bisa membuat tipe struktur yang mendirikan suatu objek.

Berikut ini adalah data structure bahasa pemrograman C++:

- Struct: identitas dari struktur yang akan dibuat
- Variabel: sepasang kurung kurawal yang mengelompokkan semua variabel
- Object: deklarasi yang menggunakan struktur menjadi tipe data

Di bawah ini adalah data structure bahasa pemrograman C++:

```
struct nama_struct{
/*macam-macam deklarasi variabel
*...
*/
}nama_object;
```

Anda bisa menggunakannya seperti ini:

```
struct Mahasiswa {
    char NIM[11];
    char Nama[27];    char
    Alamat[41];    float Ipk;
};
Syntax
```

Sintaks adalah aturan penulisan kode untuk membuat sebuah program atau aplikasi. Sintaks meliputi sebuah layout, ekspresi, dan simbol. Anda perlu memastikan sintaks yang digunakan tepat untuk mencegah terjadinya error pada program yang dibuat.

Berikut ini adalah contoh sintaks dasar dari bahasa pemrograman C++:

```
#include <iostream.h> → include using
namespace std; → namespace
int main() → fungsi main
{
    cout << "belajar bahasa pemrograman C++"; }
```

Jika dilihat dari contoh di atas, bahasa pemrograman C++ memiliki tiga sintaks dasar, yaitu:

1. Include

Include adalah sintaks untuk mendefinisikan library yang akan digunakan dalam suatu program. Artinya, include bisa Anda sebut sebagai sebuah program yang akan digunakan dalam program.

Ada beberapa library yang bisa digunakan dalam bahasa C++. Pertama, iostream yang berisi fungsi-fungsi input dan output. Kedua, library yang dimasukkan dalam ekstensi seperti .h, .cpp, .hpp, .c untuk mengimpor header file. Header file adalah file berisi fungsi yang sudah pernah dideklarasikan.

Berikut ini adalah contoh penulisannya:

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

1. Using namespace std

Using namespace std adalah sebuah instruksi terhadap compiler untuk menggunakan semua fungsi yang dideklarasikan. Misalnya, class, object, dan variabel. Berikut ini adalah contoh penulisannya: using namespace std;

1. Fungsi main ()

Fungsi main () adalah fungsi yang akan dieksekusi pertama kali saat Anda membuka program. Fungsi yang lain baru bisa dieksekusi setelah fungsi main. Artinya, fungsi ini wajib Anda gunakan. Berikut ini adalah contoh penulisannya:

```
int main() → fungsi main
{
    cout << "belajar bahasa pemrograman C++";
Tools
```

Dalam penulisan kode C++, Anda memerlukan editor sebagai tool untuk menuliskan kode pemrograman dan compiler untuk menerjemahkan perintah ke dalam komputer. Namun dengan Integrated Development Environment (IDE) Anda bisa bekerja lebih efektif karena terdapat editor dan compiler dalam satu aplikasi.