# BAB I

# PENGENALAN BAHASA C++

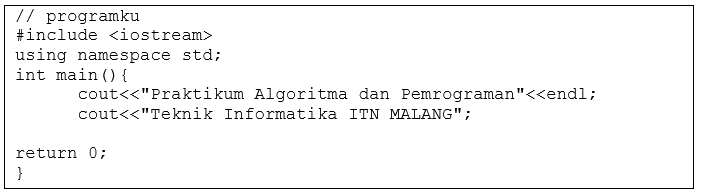
Bahasa Pemrograman C++ adalah bahasa Pemrograman Komputer Tingkat Tinggi (High Level Language), tapi C++ juga dimungkinkan untuk menulis Bahasa Pemrograman Tinggkat Rendah (Low Level Language) di dalam pengkodingan

C++ merupakan penyempurnaan dari bahasa pemrograman sebelumnya yaitu bahasa C, oleh Bjarne Stroustrup pada tahun 1980. Awal C++ mempunyai nama yaitu “C with Classes” dan berganti nama menjadi C++ pada tahun 1983. Bjarne Stroustrup membuat bahasa pemrograman C++ dengan tambahan fasilitas, yang sangat berguna pada tahun itu sampai sekarang, yaitu bahasa pemrograman yang mendukung OOP (Object Oriented Programming).

Setiap program C++ mempunyai bentuk umum seperti dibawah ini :

|  |
| --- |
| #include <iostream> //prepocessor directive  using namespace std;    int main() {  Pernyataan\_yang\_akan\_dilakukan;  …  return 0;  } |

Seperti pada umumnya, langkah pertama kita harus berkenalan dengan bahasa pemrograman c++. silahkan buka IDE Dev-C++ kalian dan ketikan kode berikut :



## Header dan Fungsi Main

Header adalah tempat kita memasukan library yang ada di c++ ke dalam program, library tersebut berisi deklarasi berbagai fungsi yang dibutuhkan oleh program baik itu sebagai standar Input/Output maupun sebagai syntax.

Apabila kita lihat pada program baris pertama terdapat #include <iostream>. Tanda ‘#’ adalah **Preprocessor directive**. Preprocessor directive sendiri adalah intruksi untuk compiler memerintahkan untuk memproses suatu arahan atau informasi sebelum memulai melakukan kompilasi pada program yang telah kita tulis, misal untuk memasukan file library, dll.

Sebagai contoh, jika dilihat dari program diatas kita akan memasukan (include) library iostream ke dalam program kita, library iostream sendiri adalah header yang di butuhkan untuk melakukan input dan output, dll.

Jika di perhatikan terdapat kode using namespace std, Using adalah sebuah kata kunci yang digunakan untuk memperkenalkan suatu Namespace kepada suatu ruang lingkup, dengan ini kita tidak diwajibkan lagi untuk menyertakan nama dari Namespace dan Scope Operator :: setiap kali mencoba untuk mengakses entitas dari Namespace tersebut.

Jika tidak menggunakan using namespace std maka cout dan cin tidak dapat dikenali kecuali didepannya terdapat syntax std::, contohnya :

|  |
| --- |
| // programku  #include <iostream>  int main(){  std::cout<<"Praktikum Algoritma dan Pemrograman "<<endl;  std::cout<<"Teknik Informatika ITN MALANG";  return 0;  } |

Selanjutnya fungsi main(), main() adalah blok program utama kita yang di tulis diantara { hingga }. Fungsi inilah yang akan dipanggil pertama kali pada saat eksekusi program. Apabila ada fungsi lain yang dibuat, maka fungsi tersebut akan dijankan ketika dipanggil di fungsi utama. Hal ini karena bahasa C/C++ merupakan bahasa procedural yang menerapkan konsep runtutan (program dieksekusi baris-perbaris dari atas ke bawah secara berurutan).

## Return 0;

Intruksi return menyebabkan fungsi main() berakhir dan mengembalikan kode yang mengikuti instruksi tersebut, dalam kasus ini 0. Ini merupakan cara yang paling sering digunakan untuk mengakhiri program.

## Komentar Program

Komentar adalah bagian dari kode program yang tidak ikut dibaca pada proses kompilasi. Dengan kata lain, komentar tidak akan mempengaruhi jalannya program. Dalam C++ terdapat 2 jenis komentar, yaitu :

1. Menggunakan tanda //

Tanda ini digunakan untuk menuliskan komentar yang banyaknya hanya satu baris.

|  |
| --- |
| // ini adalah komentar untuk satu baris |

1. Menggunakan tanda /\*\*/

Berbeda dengan sebelumnya, tanda ini dapat digunakan untuk menuliskan komentar yang banyaknya satu baris atau lebih. Komentar ini dimulai dari tanda **/\*** sampai ditemukan tanda **\*/**.

|  |
| --- |
| /\*ini adalah komentar panjang  yang banyaknya  lebih dari satu baris\*/ |

## Tanda Semicolon

Tanda semicolon “ ; ” digunakan untuk mengakhiri sebuah pernyataan. Setiap pernyataan harus diakhiri dengan sebuah tanda semicolon.

## Input dan Output

Statemen Input adalah Statemen / fungsi yang digunakan untuk menerima data dari input keyboard, contoh: cin (character in).

|  |
| --- |
| cin>>a; |

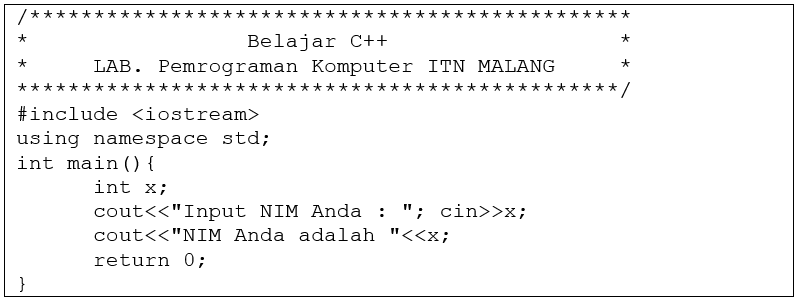
Selain cin ada juga getline, getline digunakan jika ingin mengambil input lebih dari 1 kata. getline memiliki struktur :

|  |
| --- |
| getline(cin,nama\_identifier); |

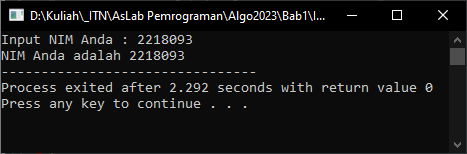
Statemen Output adalah Statemen yang digunakan untuk menampilkan data ke layar monitor., contoh: cout (character out).

|  |
| --- |
| cout<<”RPL”; |

Contoh :



Tampilan :



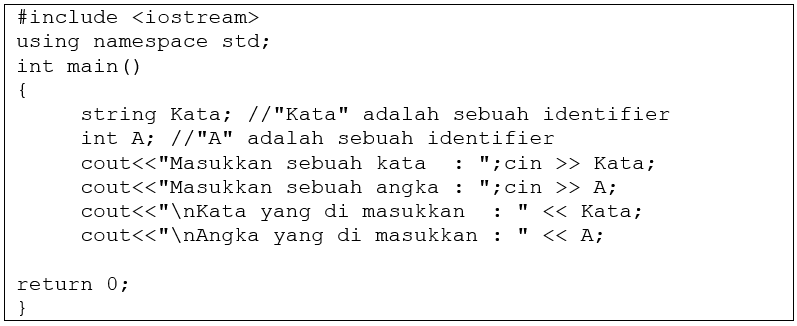
Analisa :

Pada saat di jalan kan, cin>> akan menerima data inputan program dan menyimpan data inputan ke variable x, lalu data yang ada di variable x di tampilkan dengan cout<<

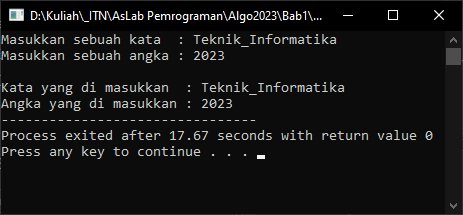
## Jenis Identifier

Identifier atau pengenal adalah suatu nama yang biasa dipakai dalam pemrograman untuk menyatakan variabel, konstanta, tipe data, dan fungsi. Identifier yang berperan sebagai variabel dan konstanta berfungsi untuk menampung sebuah nilai yang digunakan dalam program.

Contoh program yang menggunakan dua buah identifier :



Tampilan :



Analisa :

Pada saat program dijalankan, identifier tersebut akan digunakan untuk menyimpan nilai yang dimasukkan dari keyboard. Dalam C++, proses penyimpanan nilai seperti ini dinyatakan dengan perintah cin (si in). berbeda dengan perintah cout (si out) yang menggunakan operator <<. Pada perintah cin operator yang digunakan adalah operator >>.

### Konstanta

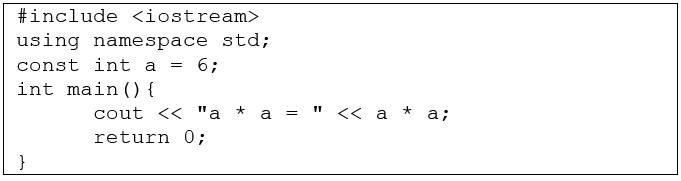
Konstanta adalah jenis identifier yang bersifat konstan atau tetap, artinya nilai dari konstanta di dalam program tidak berubah. Terdapat dua cara untuk membuat sebuah konstanta, yaitu dengan menggunakan preposesor directive #define dan menggunakan kata kunci const.

1. **Menggunakan kata kunci *const***

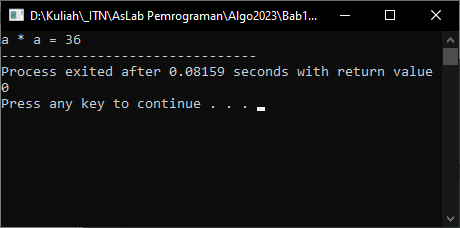
Mendefinisikan sebuah konstanta dapat dilakukan dengan menggukan kata kunci const. Berikut ini adalah bentuk umumnya.

|  |
| --- |
| const tipe\_data nama\_konstanta = nilai; |

Contoh :



Tampilan :



Analisa :

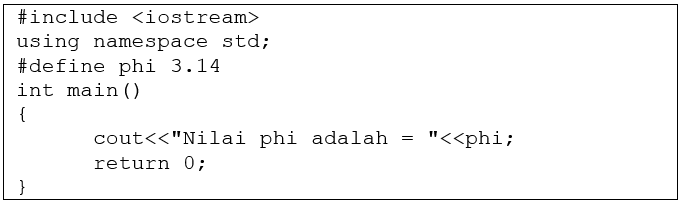
Konstanta MAX didefinisikan menggunakan kata kunci const, penggunaan kata kunci const harus diakhiri tanda titik koma.

1. **Menggunakan *Preprocessor Directive* #*define***

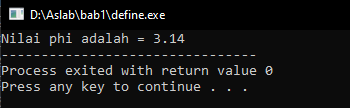
Keuntungan menggunakan #define apabila dibandingkan dengan const adalah kecepatan kompilasi, karena sebelum kompilasi dilaksanakan, kompiler pertama kali mencari symbol #define (oleh sebab itu mengapa # dikatakan preprocessor directive). #define adalah preprocessor directive, oleh sebab itu tidak diakhiri dengan tanda semicolon (;).

|  |
| --- |
| #define nama\_konstanta nilai\_tetapan |

Contoh :



Tampilan :



Analisa :

#define phi 3.14 akan di kompiler terlebih dahulu setelah preposesor directive. Oleh sebab itu, ketika <<phi di panggil kembali akan menampilkan = 3.14 pada output.

### Variabel

Variabel adalah sebuah identifier yang mempunyai nilai dinamis. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variabel bisa diubah-ubah sesuai kebutuhan.

1. **Deklarasi variabel**

Untuk menggunakan variabel pada C++, kita harus mendeklarasikan tipe data yang akan digunakan. Sintaks penulisan deklarasi variabel adalah dengan menuliskan tipe data yang akan digunakan diikuti dengan identifier yang benar. Pengertian deklarasi disini berarti mengenalkan sebuah pengenal ke program dan menentukan jenis data yang disimpam didalamnya.

Bentuk pendeklarasian variabel :

|  |
| --- |
| tipe\_data nama\_variabel; |

Contoh :

|  |
| --- |
| int A;  float a;  char kata; |

Jika akan menggunakan tipe data yang sama untuk beberapa identifier maka dapata dituliskan dengan menggunakan tanda koma.

|  |
| --- |
| int a,b,c;  float d,e,f; |

Nama dari suatu variabel dapat ditentukan sendiri oleh program dengan aturan sebagai berikut :

* Tidak boleh dimulai dengan karakter non huruf
* Terdiri dari gabungan huruf dan angka dengan karakter pertama harus berupa huruf. Bahasa C++ bersifat case-sensitive, yang artinya huruf besar dan kecil dianggap berbeda. Jadi antara **nim, NIM, dan Nim** dianggap berbeda.
* Tidak boleh mengandung spasi.
* Tidak boleh mengandung simbol-simbol khusus, kecuali garis bawah (underscore). Yang termasuk simbol khusus yang tidak diperbolehkan antara lain : **$, ?, %, #, !, &, \*, (, ), -, +, =, dsb**.
* Panjang bebas, tetapi hanya 32 karakter pertama yang terpakai.

Contoh penamaan variabel yang benar dan salah :

|  |  |
| --- | --- |
| Penamaan yang Benar | Penamaan yang Salah |
| namasiswa | nama siswa (salah karena menggunakan spasi) |
| XY12 | 12X (salah karena dimulai dengan angka) |
| harga\_total | harga.total (salah karena menggunakan karakter .) |
| JenisMotor | Jenis Motor (salah karena menggunakan spasi) |
| alamatrumah | for (salah karena menggunakan kata kunci bahasa pemrograman) |

1. **Inisialisasi variabel**

Inisialisasi dapat didefinisikan sebagai proses pengisian nilai awal (nilai default) ke dalam suatu variabel. Dalam C++, pengisian nilai dilakukan dengan menggunakan operator sama dengan (=).

Bentuk pernyataan yang digunakan untuk memberikan nilai ke variabel yang telah dideklarasikan atau didefinisikan :

|  |
| --- |
| nama\_variabel = nilai\_awal; |

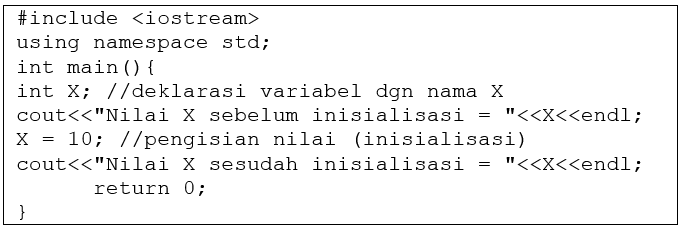
Adakalanya dalam penulisan program, variabel langsung diberi nilai setelah didefinisikan. Sebagai contoh :

|  |
| --- |
| int A;  A = 10; |

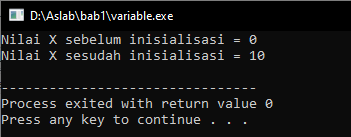
Dua pernyataan seperti diatas sebenarnya dapat disingkat melalui pendefinisian yang disertai penugasan nilai, sebagai berikut :

|  |
| --- |
| int A = 10; |

Contoh :



Tampilan :



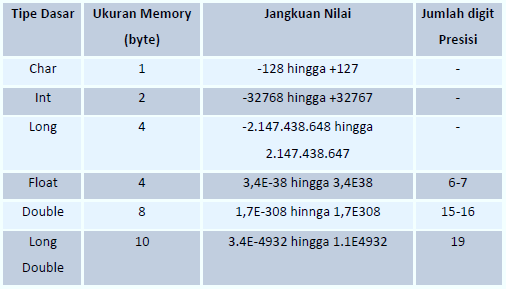
Analisa :

Apabila kita tidak melakukan inisialisasi dan kita sudah memanggil variabel tersebut, maka nilai yang akan ditampilkan adalah nilai 0, setelah itu di lakukan inisialisasi dengan nilai X = 10, maka nilai X yang akan di tampilkan adalah nilai 10

## Tipe Data

### Tipe Data Dasar

Adalah tipe data yang dapat langsung dipakai. Daftarnya dapat dilihat pada tabel dibawah ini :

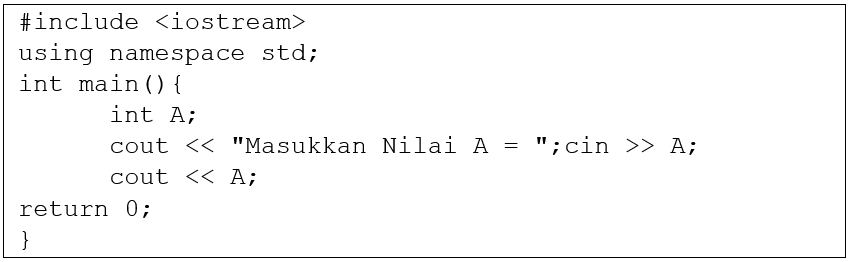


Perhatikan contoh dibawah ini, tipe data dapat dirubah (*type cast)* dengan cara dibawah ini :

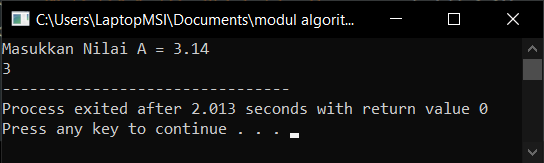
|  |
| --- |
| float x = 3.345;  int p = int(x); |

Maka nilai P adalah 3 (terjadi *truncating*).

Contoh :



Tampilan :



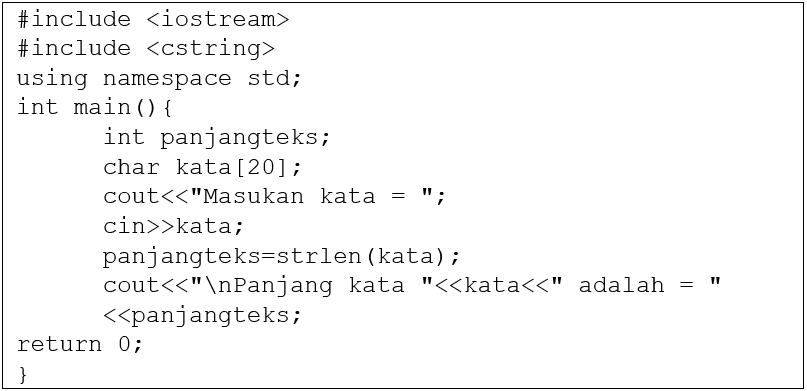
Analisa :

Tidak adanya tipe data pecahan menyebabkan bilangan 3,14 dibulatkan menjadi 3 karena int adalah tipe data bilangan bulat.

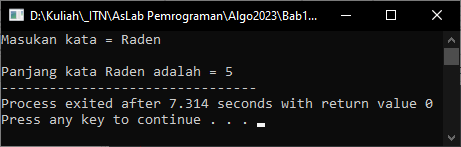
### Karakter dan String Literal

String adalah tipe data text yang memungkinkan kita menyimpan nilai dengan bentuk text, kumpulan dari karakter. Berbeda dengan char gabungan dari karakter. Perbedaannya dengan tipe data *char* adalah mengenai penyimpanannya. Char menyimpan hanya beberapa karakter saja, sedangkan String dapat menyimpan banyak karakter, kata, dan kalimat sekaligus.

Contoh :



Tampilan :



Analisa :

Dengan adanya operasi pengolahan string, fungsi strlen dapat menghitung panjangnya karakter yang diinputkan.

### Escape Sequence Characters

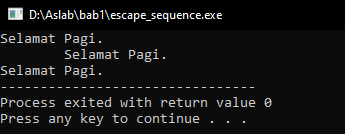
Escape sequence characters adalah urutan sebuah table, penulisan kode yang mewakili sebuah karakter yang tidak dapat diwakili secara langsung seperti newline yang diwakili dengan tanda \n dan petik dua dengan menggunakan tanda \".



Contoh :

|  |
| --- |
| #include <iostream>  using namespace std;  int main()  {  cout<<"Selamat Pagi.\n";  cout<<"\tSelamat Pagi."<<endl;  cout<<"Selamat Pagi.\a";  return 0;  } |

Tampilan :



Analisa :

Karakter \n adalah digunakan untuk membuat baris baru, lalu \t untuk membuat tab horizontal (jarak), dan \a adalah memberikan bunyi beep.

|  |  |
| --- | --- |
| **#taugaksih**  **Global & Lokal Variable**  Berdasarkan ruang linkupnya, variabel dibedakan menjadi dua yaitu variabel lokal dan variabel global. Penentuan variabel untuk dijadikan / dikatakan sebagai variabel global atau lokal tentu akan tergantung dari penempatan deklarasi variabel pada program yang sedang dibuat, Contoh :   |  | | --- | | #include <iostream>  using namespace std;  int a=3; //Global variable  int main(){  int b=2; //Lokal variable  cout<<a<<" adalah lokal variable"<<endl;  cout<<b<<" adalah global variable";  } |   Tampilan:    Perbedaan utama lokal dan global variable:   1. Variabel Global dideklarasian diluar fungsi main() atau diluar sebuah fungsi atau prosedur lain. 2. Variabel Lokal dideklarasian didalam sebuah fungsi atau didalam sebuah prosedur. 3. Variabel Global dapat digunakan atau dipanggil atau dikenali oleh semua fungsi / prosedur / dikenali diseluruh program. 4. Variabel Lokal hanya dikenali oleh suatu fungsi / prosedur saja (hanya dikenali pada fungsi / prosedur tempat variabel tersebut dideklarasikan) |

**Latihan !**

1. Buatlah program menampilkan 3 digit angka NIM terakhir anda seperti pada gambar berikut, kemudian berikan komentar keterangan pada setiap baris pada source code kalian.

Dengan ketentuan :

1. Terdapat *Escape Sequence Characters*
2. Menggunakan Perintah *cout*

