Nama : Alvin Febrianto

NIM : 21091397031

Prodi/Kelas : D4 Manajemen Informatika / A

Mata Kuliah : Pemrograman Dasar

Tipe Data pada Bahasa Pemrograman C++

Tipe data adalah sebuah pengelompokan data untuk memberitahu compiler atau interpreter bagaimana programmer ingin mengolah data tersebut. Sederhananya, tipe data adalah cara kita memberitahu komputer untuk mengelompokkan data berdasarkan apa yang dipahami oleh komputer.

Secara garis besar, terdapat 2 kelompok tipe data dalam bahasa C++, yakni tipe data sederhana (Primitive data types), dan tipe data kompleks (Non-primitive data types).

1. Primitive data types

|  |  |  |
| --- | --- | --- |
| Tipe Data | Ukuran / Rentang | Deskripsi |
| void | 0 byte | Tipe data khusus yang menyatakan tidak ada data. |
| boolean | 1 byte | Tipe data yang berisi nilai true atau false. |
| char | 1 byte /  -128 sampai 127 atau 0 sampai 255 | Tipe data yang berisi 1 digit karakter seperti “A”,”b”. |
| wchar\_t | 2 atau 4 bytes /  1 wide character | Tipe data yang digunakan untuk menyimpan karakter yang tidak ada di abjad biasa seperti huruf jepang, korea, cina. |
| int | 4 bytes /  -2,147,483,648 sampai 2,147,483,647 | Tipe data untuk angka bulat seperti 8, 10, atau 16. |
| float | 4 bytes /  3.4E +/- 38 (7 digit) | Tipe data yang berisi angka desimal dan hanya cukup untuk menyimpan sampai 7 digit. |
| double | 8 bytes /  1.7E +/- 308  (15 digit) | Tipe data yang berisi angka desimal dan hanya cukup untuk menyimpan sampai 15 digit. |

1. Non-primitive data types

* String: Tipe data untuk kumpulan karakter, seperti “java”
* Array: Tipe data untuk kumpulan tipe data lain yang sejenis
* Structure: Tipe data yang terdiri dari kumpulan tipe data dasar, bisa lebih dari 1 jenis
* Pointer: Tipe data untuk mengakses alamat memory secara langsung
* Enum: Tipe data bentukan yang dibuat sendiri oleh kita

Preprocessor

Preprocessor adalah bagian dari code yang bukan bagian langsung dari instruksi program. Bagian akan dieksekusi sebelum program dijalankan oleh compiler. Semua perintah preprocessor dimulai dengan (#) yang kemudian diikuti oleh header file atau macro. Ada beberapa preprocessor dalam C++, yaitu:

1. #include: Menyertakan isi file header ("\*.h") dalam kode program
2. #define: Menggantikan identifier dalam program dengan makro atau suatu nilai
3. #undef: Menghapus identifier yang sudah dibuat dengan #define
4. #Pragma: Perintah khusus compiler yang bisa berbeda untuk setiap compiler
5. Conditional Preprocessor ( #if, #elseif, #else): Menentukan bagian program yang akan di-compile berdasarkan suatu persyaratan seperti OS atau compilernya.

Compiler

Compiler adalah aplikasi yang berfungsi untuk membaca dan menerjemahkan bahasa pemrograman komputer (kode program) untuk menghasilkan sebuah program. Ada beberapa compiler yang bisa digunakan untuk mengcompile program C++. Ada MinGW, Cygwin, Clang, GCC (Gnu Compiler Collection), Embarcadero, Visual C++ 2017, Oracle Solaris Studio C++ Compiler, dan IBM XL C/C++ Compiler.

Assembler

Assembler merupakan sebuah program yang berfungsi untuk mengkonversi kode program ke dalam mesin. Sedangkan bahasa Assembly merupakan bahasa yang merespon perintah dengan bahasa mesin untuk memahami interaksi antara perangkat keras dan perangkat lunak di komputer.

Text Editor

Text Editor adalah suatu software aplikasi atau suatu program komputer yang memungkinkan Anda sebagai penggunanya untuk membuat, mengubah atau mengedit file teks yang ada berupa plain text. Ada Notepad, Notepad++, Geany, Atom, VS Code, dan Sublime Text.

Header

File Header pada C++ merupakan file yang berisi deklarasi variabel, fungsi, tipe data dan bertugas untuk menjalankan fungsi yang terdapat dalam library file header tersebut. Berikut ini beberapa header file yang sering digunakan dalam bahasa C++:

1. **<iostream.h>**

* **Cin**  
  Merupakan fungsi masukan (digunakan untuk menyimpan data dalam suatu variabel). Bentuk umum: cin>>var x;
* **Cout**  
  Merupakan fungsi keluaran (digunakan untuk menampilkan data ataupun tulisan). Bentuk umum: cout<<”tulisan”; atau cout<<var x;
* **Endl**  
  Digunakan untuk pindah baris/ enter. Bentuk umum: cout<<”tulisan”<<endl;
* **Ends**  
  Merupakan suatu fungsi manipulator yang digunakan untuk menambah karakter null (nilai ASCII NOL) ke deretan suatu karakter. Fungsi ini akan berguna untuk mengirim sejumlah karakter ke file di disk atau modem dan mangakhirinya dengan karakter NULL.

1. **<conio.h>**

* **Getch**  
  Berfungsi untuk menahan tampilan. Bentuk umum: getch();
* **Clrscr**  
  Berfungsi untuk membersihkan layar. Bentuk umum: clrscr;
* **Getche**berfungsi untuk membaca sebuah karakter dengan sifat karakter yang dimasukkan tidak perlu diakhiri dengan menekan tombol ENTER, dan karakter yang dimasukan ditampilkan di layar.
* **Putch**  
  Akan menampilkan karakter ASCII dari nilai x ke layer monitor tanpa memindahkan letak kursor ke baris berikutnya.
* **Clreol**  
  Fungsi ini digunakan untuk membersihkan layar mulai dari posisi kursor hingga kolom terakhir, posisi kursor tidak berubah.
* **Gotoxy**  
  Fungsi gotoxy digunakan untuk memindahkan kursor ke kolom x, baris y.
* **Wherex**  
  Fungsi wherex digunakan untuk mengembalikan posisi kolom kursor.
* **Wherey**  
  Fungsi wherey digunakan untuk mengembalikan posisi baris kursor.
* **Window**  
  Fungsi window digunakan untuk mendefinisikan sebuah window berdasarkan koordinat kiri atas dan kanan bawah.

1. **<stdio.h>**

* **Printf**  
  Merupakan fungsi keluaran. Bentuk umum: printf (“tulisan”);
* **Scanf**  
  Merupakan fungsi masukan. Bentuk umum: scanf (“%c”, &karakter);
* **Gets**  
  Fungsi inputan yang bisa membaca spasi. Bentuk umum: gets(var x);

1. **<string.h>**

* **Strcpy**  
  Digunakan untuk menyalin nilai string.
* **Strlen**  
  Digunakan untuk mengetahui panjang string.
* **Strupr**  
  Digunakan untuk membuat string menjadi kapital.
* **Strcmp**  
  Digunakan untuk membandingkan dua buah string. Hasil dari fungsi ini bertipe integer dengan nilai: Negative, jika string pertama kurang dari string kedua. Nol, jika string pertama sama dengan string kedua Positif, jika string pertama lebih besar dari string kedua. Bentuk umum: strcmp(string1, string2);
* **Strlwr**  
  Digunakan untuk mengubah huruf menjadi kecil semua. Bentuk umum: strlwr(string);
* **Strcat**  
  Digunakan untuk menggabungkan string. Bentuk umum: strcat(string1,string2);

1. **<math.h>**

* **Sqrt**  
  Fungsi akar
* **Pow**  
  Fungsi pangkat  
  Sin (), Cos(), Tan()  
  Masing-masing digunakan untuk menghitung nilai sinus, cosinus dan tangens dari suatu sudut. Bentuk umum: Sin(sudut); Cos(sudut); Tan(sudut);
* **Max**  
  Digunakan untuk menghitung hasil pembagian dan sisa pembagian.  
  Bentuk umum: max(bilangan1, bilangan2);
* **Min**  
  Digunakan untuk menentukan bilangan terkecil dari dua buah bilangan.   
  Bentuk umum: min(bilangan1, bilangan2).