Static Member Variabel adalah member variabel yang menggunakan static keyword, membuat member variabel tersebut bersifat dimiliki oleh semua object. berbeda seperti member variable yang biasa kita dirikan, member variable biasa akan dimiliki oleh masing-masing object yang menggunakan class tersebut, member variabel akan berdiri sendiri di masing-masing object dan pastinya bisa memiliki nilai yang berbeda di masing-masing object.

Tapi jika member variabel tersebut menggunakan keyword static, static member variabel akan dimiliki oleh semua object, nilai dari static member variable pada masing-masing object yang menggunakan class bersangkutan akan memiliki nilai yang sama seperti object lainya, karena static member variable hanya ada satu dan dimiliki oleh semua object. static member variabel juga akan ada dan dapat di akses tanpa adanya instantiation. Static member variable biasanya digunakan untuk menghitung berapa banyak object yang telah dibuat dari class tersebut, atau membuat variable yang mempunyai nilai yang sama di semua object , satu perubahan nilai akan berlaku ke semua object dan tanpa melakukan perubahan satu demi satu melalui object.

Bentuk umum penulisan

static tipe\_data identifier;

contoh penulisan

class siswa {

public:

static int s\_id ;

};

Di atas adalah contoh penulisan dari deklarasi static member variable di dalam public akses, deklarasi static member variable di atas belum sepenuhnya bisa digunakan, kita masih membutuhkan definisi static member variable yang kita dirikan dalam class, deklarasi static member variable di atas hanya menyatakan kepada compiler bahwa member variable “s\_id” tersebut adalah static member variable.

Contoh penulisan

int siswa :: s\_id = 1 //Definisi dan Inisialisasi Static Member Variable

di atas adalah contoh penulisan dari definsi dan inisialisasi static member variable, dilakukan dengan cukup menulis tipe data yang dipakai tanpa menggunakan static keyword di depan tipe data, selanjutnya menggunakan scope operator di antara nama class dan member variable. definisi dan inisialisasi static member variable biasanya diletakan di luar class seperti halnya deklarasi global variable. defnisi contoh di atas, kita juga dapat untuk tidak menyertakan inisialisasi, sifat static sebagai member variable masih sama seperti sifat static yang sebelumnya penulis jelaskan di beberapa artikel sebelumnya. Static akan otomatis di inisialisasi menjadi “0” ketika tidak kita inisialisasi.

Contoh penulisan

int siswa::s\_id; //Definisi tanpa inisialisasi

kita juga dapat menginisialisasi static member variable di saat deklarasi static member variable di dalam class dan juga kita tidak membutuhkan untuk mendefinisi static member variable di luar class. Hal ini hanya akan terjadi jika static member variable merupakan konstanta.

Contoh penulisan

class siswa{

public:

static const int s\_id = 0;

};

**Cara mengakses Static Member Variable**

Static member variable hanya ada satu dan itu berlaku ke pada semua object yang menggunakan class tersebut, dan juga static member variable akan ada walaupun belum ada object yang dibuat dari class tersebut. static member variable dapat di akses menggunakan dua cara yaitu melalui object yang menggunakan class tersebut dan dapat melalui class langsung.

Untuk mengakses Static Member Variable melalui object adalah dengan menyertakan Member Access Operator ( . )di antara nama object dan nama member variable.

Contoh penulisan Akses melalui Object

nama\_object.s\_id

kita juga dapat mengakses langsung melalui class meskipun belum terjadinya instantiation. Untuk melakukanya kita membutuhkan Scope Resolution Operator ( :: ) di antara nama class dan nama member variable.

Contoh penulisan akses melalui class

nama\_class::s\_id

saat deklarasi static member variable di dalam class kita juga dapat mengikut sertakan static member variable di dalam akses specifier private atau protected. Jika kita melakukan itu kita juga akan tetap membutuhkan definisi dari static di luar class dan tidak ada yang berbeda dalam melakukan definisi static member variable. hal yang berbeda hanya kita tidak bisa mengakses static member variable dari luar class (tidak bisa mengakses static member variable melalui object maupun class).

Contoh program

#include <iostream>

#include <string>

using namespace std;

class siswa{

public:

static int s\_id;

int id;

string nama;

void setID();

void printAll();

siswa(string pnama):nama(pnama){setID();}

};

int siswa::s\_id=0;

void siswa::setID(){

id = ++s\_id;

}

void siswa::printAll(){

cout<<"ID = "<<id<<endl;

cout<<"Nama = "<<nama<<endl;

cout<<endl;

}

int main(){

siswa dadi("Dadi family");

siswa budi("Budi Dudi Dodi");

siswa andi("Andi Dadi Loro");

siswa william("Joko Will");

dadi.printAll();

budi.printAll();

andi.printAll();

william.printAll();

return 0;

}