Bukan hanya variable dan member variable saja yang dapat menggunakan keyword static tapi juga Member Function, Static Member Function adalah Member function yang menggunakan keyword static yang akan bersifat dimiliki oleh semua object termasuk class tersebut. berbeda dengan member function yang basanya, yang memiliki sifat setiap object memiliki member function mereka masing-masing jika kita menambahkan keyword static di member function akan membuat member static tersebut hanya ada satu dan bisa di gunakan di semua object yang menggunakan class tersebut. dan juga static member function aka nada meskipun belum ada instance dari class.

Pada artikel sebelumnya dijelaskan mengenai Static Member Variable, tepat pada akhir artikel sebelum contoh program diberitahukan bahwa Static member Variable dapat didirikan di dalam akses specifier private atau protected hal itu dlakukan untuk meningkatkan keamanan program. Dengan arti akan menyebabkan static member function tersebut tidak dapat di akses dari luar class.

Di artikel sebelumnya dicontohkan secara tidak langsung mengenai setter dan getter,yang memungkinkan kita untuk mengakses member variable atau function yang bersifat private melalui setter atau getter. Kita bisa saja menggunakan setter dan getter pada private atau protected static member variable, tapi hal itu hanya memungkinkan kitauntuk mengakses static member variable melalui instance, kita harus melakukan instantiation dan mengaksesnya melalui instance class. Hal itu tidak sama seperti cara-cara mengakses static member variable yang sesungguhnya, karena static member variable juga dapat diakses melalui nama class tanpa melalui atau adanya object dari class tersebut.

Dari masalah di atas kita dapat mengatasinya menggunakann Static Member Function, sama seperti setter atau getter dengan menambahkan keyword static di depan deklarasi mereka setter atau getter. dengan static member function kita dimungkinkan untuk mengakses static member variable yang bersifat private atau protected secara tidak langsung dari luar class melalui object maupun class.

Bentuk umum penulisan

static tipe\_data nama\_function(parameters…){

//Definsi..

}

Contoh penulisan

class siswa{

private :

static int s\_id;

public :

static int getS\_id(){return s\_id;}

};

Int siswa::s\_id =0;

**Cara Mengakses Static Member Function**

Untuk mengakses static member function memilki dua cara sama seperti kita melakunya pada static member variable, yang pertama kita bisa melakukanya melalui object, dengan menambahkan Member Akses Operator ( . ) di antara nama object dan nama member.

Contoh penulisan Akses Melalui Object

nama\_object.getS\_id();

di setiap object akan memiliki static member function yang sama karena static member function hanya ada satu. cara kedua adalah kita dapat mengakses melalui nama dari class langsung dengan menambahkan tanda Scope Resolution Operator ( :: ) di antara nama class dan member function, hal ini dapat dilakukan ada atau tidak ada instantiation pada class.

Contoh penulisan Akses Melalui Class

nama\_class::getS\_id();

untuk static member function tidak seperti static member variable yang mengharuskan kita mendefinisi member di luar class. Static member function dapat langsung digunakan setelah deklarasi static member Function.

Contoh program

#include <iostream>

#include <string>

using namespace std;

class siswa{

private:

static int s\_id;

public:

int id;

string nama;

void setID();

void printAll();

static void setS\_id(int pS\_id){s\_id=pS\_id; /\*Definisi Function\*/}

static int getS\_id(){return s\_id; /\*Definisi Function\*/}

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::setS\_id(9); //mengakses s\_id melalui static member function “setS\_id”

siswa andi("Andi Dadi Loro");

siswa william("Joko Will");

dadi.printAll();

budi.printAll();

andi.printAll();

william.printAll();

cout<<"akses dari luar object = "<<siswa::getS\_id()<<endl; //mengakses s\_id melalui static member function “getS\_id”

return 0;

}

Untuk definisi static member function kita harus memperhatikan apa yang akan kita panggil. Static member function hanya dapat memanggil static member function lainya atau static member variable. tapi kita juga bisa memanggil member biasa, untuk memanggil non static member variable dan non static member function kita membutuhkan object untuk dapat melakukanya. Karena kita tahu bahwa non static member hanya ada pada object dan dibutuhan object untuk mengakses member tersebut. Hal ini terjadi sama seperti kasus dalam “friend” keyword yang membutuhkan object untuk mengakses non-static member.