**pointer pada object**

Pointer yang berarti menunjuk, adalah variable yang memungkinkan kita untuk membuat variabel yang dapat menunjuk ke sebuah alamat memory dan dapat kita ubah nilainya melalui variabel pointer, memesan memori (alokasi memori) atau membebaskan memori. Object memiliki persamaan seperti deklarasi variabel biasa. Saat kita membuat object disaat itu juga kita sedang memesan memori kepada komputer sebesar total memori pada class yang digunakan, dalam deklarasi object kita juga diperbolehkan untuk menggunakan pointer pada deklarasi object.

Bukan hanya akan berfungsi untuk menunjuk ke object lain yang memiliki class yang sama, kita juga bisa mengalokasi atau membebaskan memori dengan menggunakan keyword New dan Delete. Untuk penjelasan alokasi memori anda bisa kembali ke bab pointer. pointer memiliki dua macam yaitu reference dan dereference, dibawah ini adalah bentuk penulisan dan penjelasan dari object pointer reference dan dereference.

**Pointer Reference**

Bentuk umum penulisan

nama\_class &nama\_object = alias\_object;

Contoh penulisan

siswa &refBudi = budi;

Di atas adalah bentuk umum penulisan dan contoh penulisan dari penggunaan pointer reference pada object. pointer reference berfungsi untuk membuat alias atau identifier lain untuk variable yang di tunjuk. Sama artinya seperti satu alamat memori memiliki 2 identifier.

Contoh penulisan di atas berarti refBudi sama dengan budi. Karena kedua variabel adalah satu memori, mereka hanya dua identifier atau alias yang berbeda. Untuk mengakses juga masih sama seperti kita mengakses variable/object biasa. Untuk mengakses object dengan pointer reference kita membutuhkan “Member Access Operator (.)”, dengan menaruh tanda (.) di anatara nama object dan nama member. Tidak memiliki perbedaan seperti kita melakukanya pada object biasa.

**Pointer Dereference**

Bentuk umum penulisan

nama\_class \*nama\_object = &object\_lain;

Contoh penulisan

siswa \*pBudi = &budi;

pada bentuk umum penulisan dan contoh penulisan dicontohkan dari penggunaan object dengan pointer dereference. pointer dereference berfungsi untuk menunjuk ke alamat atau alamat variabel/object lain, kita dimungkinkan untuk mengakses melalui variable/object pointer. Pada contoh penulisan memiliki arti object pointer “pBudi” menunjuk ke pointer object “budi”, Kita dapat mengakses langsung memori yang dimiliki oleh object “budi” melalui “pBudi”.

Bukan hanya berfungsi untuk menunjuk ke memori yang dimiliki variable lain, tapi kita juga dapat memesan / alokasi memori pada memori heap sebesar total memori class yang digunakan dengan menggunakan keyword “new”. Seperti deklarasi object tapi tidak memiliki identitas, hanya ditunjuk oleh pointer. dan jangan lupa setelah kita selesai dengan memori itu, gunakan keyword “delete” untuk membebaskan memori.

Cara mengakses melalui pointer

Jika menggunakan pointer, untuk mengakses member pada object kita membutuhkan tanda “Arrow Operator ( -> )” di antara nama object dan nama member.

Bentuk umum penulisan

Nama\_object -> nama\_member;

Contoh penulisan

pbudi -> noinduk = 123;

Contoh program

#include <iostream>

using namespace std;

class siswa{

public:

int noInduk;

void showInduk(){

cout<<"No Induk = "<<noInduk<<endl;

}

};

int main (){

siswa budi{1}; // Object Budi

budi.showInduk(); //Member Access Operator

siswa &refBudi = budi; //Pointer Reference refBudi

refBudi.noInduk = 2; //Member Access Operator

budi.showInduk();

siswa \*pBudi = &budi; //pointer Dereference pBudi

pBudi->noInduk = 3;//Arrow Operator

budi.showInduk();

return 0;

}

Contoh program 2

#include <iostream>

using namespace std;

class siswa{

public:

int noInduk;

void showInduk(){

cout<<"No Induk = "<<noInduk<<endl;

}

};

int main (){

siswa \*budi = new siswa{1};

budi->noInduk = 2;

budi->showInduk();

delete budi;

return 0;

}

Pada contoh program ke 1 dicontoh cara penggunaan dari object pointer yang menunjuk ke object lainnya dengan class yang sama, dan contoh program kedua dicontohkan dari penggunaan alokasi pointer pada object.