Call by Pointer atau bisa juga disebut sebagai call by Address, adalah metode dari salah satu cara untuk memberikan argument atau parameter actual kepada function parameter atau parameter formal dengan menggunakan Pointer Dereference (\*).

Pointer dereference atau ada juga yang memanggil Pointer atau Address. Pointer dereference adalah variabel yang akan berisikan alamat dari variabel yang di tunjuk. Mendeklarasikan Pointer dereference sama halnya mendrikan sebuah variabel biasa hanya saja kita cukup menambahkan tanda (\*) di depan identifier, Pendeklarasian Pointer dereference membutuhkan memori untuk mendeklarasikanya. Fungsi Parameter adalah memberikan Fungsi parameter adalah memberikan nilai atau argument dari variabel pada function sebelumnya diserahkan kepada dan bersamaan dengan pemanggilan function tersebut, biasanya jika kita hanya menyerahkan arguments, nilai tersebut diolah di dalam function tersebut. dan jika terjadi perubahan pada parameter tersebut maka tidak akan mengubah nilai atau arguments asli pada variabel sebelumnya.

Dengan menggunakan kombinasi fungsi diatas. Call by Pointer berfungsi untuk memberikan alamat argument pada function sebelumnya kepada parameter di function yang dipanggil, dengan itu kita bisa mengubah isi dari variabel asli sebagai argument tersebut melalui pointer dereference yang berada pada function lain, meskipun mereka berada pada luar scope. Berbeda dengan pointer reference,Kelebihanya adalah kita bebes untuk mengganti alamat yang akan di tunjuk jika suatu waktu kita butuh untuk menunjuk ke alamat lainya.

Bentuk Umum Penulisan

//Parameter

tipeDat myFunction (tipeData \*identifier){

…

}

//Pemanggilan

myFunction(&arguments);

Contoh Penulisan

//Parameter

void myFunction (){

}

//Pemanggilan

myFunction(&argument1,&argument2);

Contoh Program

#include <iostream>

using namespace std;

void myFunction (int \*x);

int main(){

int var=1;

cout<<"var = "<<var<<endl;

myFunction(&var);

cout<<"var = "<<var<<endl;

return 0;

}

void myFunction (int \*x){

\*x+=2;

}

Contoh program diatas akan menampilkan perubahan yang terjadi pada variabel yang ditunjuk oleh pointer. saat pemanggilan function “myFunction(&var)” kita sekaligus memberikan alamat argument variabel “var” dengan menggunakan tanda ‘&’. Maka variabel ‘x’ akan memuat alamat dari variabel “var”, dan kita diperbolehkan untuk mengoperasi atau mengubah nilai dari variabel “var” melalui pointer ‘x’ meskipun mereka mempunyai scope yang berbeda.