Call by value adalah salah satu metode pemberian argument kepada parameter, metode ini menggunakan cara, memberikan argument dalam bentuk nilai atau variabel yang memiliki nilai di dalamnya, yang berasal dari function sebelumnya yang memanggil function dengan parameter dengan tujuan memberikan argument itu, setelah melakukan pemanggilan dengan memberikan argument kepada parameter maka argument itu akan disalin atau dicopy dan diberikan kepada parameter tersebut.

Seperti penjelasan sebelumnya mengenai parameter. Parameter adalah suatu tempat untuk menyimpan dan mengantarkan nilai atau data sebesar memori tipe data yang digunakan. Parameter akan menyimpan argument yang diberikan disaat pemanggilan function bersangkutan, arguments itu akan disimpan sebagai variabel local pada function tersebut dan biasanya akan digunakan untuk membantu operasi yang di dalamnya. Alokasi memori untuk Parameter akan dibuat ketika function itu dipanggil dan akan dihapus ketika compiler selesai dengan pembacaan function tersebut.

Call by value bisa kita golongkan sebagai cara yang aman untuk mengisi parameter. Karena, Fungsi dari call by value adalah membuat salinan atau copy dari arguments yang diberikan dan disimpan di dalam parameter tujuan. dan itu bisa menjaga data atau nilai asli, jika kita melakukan perubahan dengan nilai diberikan, tidak akan berdampak pada nilai asli, karena yang mendapatkan perubahan hanyalah salinan dari nilai yang asli.

Bentuk Umum Deklarasi Parameter :

typeData identifier (parameters)

Bentuk umum Pemanggilan :

Identifier(arguments)

Contoh Deklarasi Parameter :

void myFunction(int a)

Int myFunction(int a, int b)

double myFunction(double a, char b)

Contoh Pemanggilan Berdasarkan Nilai (Call by Value) :

myFunction(3)

myFunction(var+1,2)

myFunction(var,‘B’)

Contoh Program :

#include <iostream>

using namespace std;

void myFunction(int,double,char);

int main(){

int var=29029;

myFunction(var-29000,2,'A');

return 0;

}

void myFunction(int a,double b,char c){

cout<<"a"<<" = "<<a<<endl;

cout<<"b"<<" = "<<b<<endl;

cout<<"c"<<" = "<<c<<endl<<endl;

}