Reference Variabel

Pengertian Reference

Reference adalah fitur yang berasal dari C++ memungkinkan kita untuk membuat variabel yang dapat memegang alamat memori dan bertingkah layaknya variabel tapi sebenarnya adalah alias dari suatu alamat memori, suatu obyek, atau suatu variabel.

Ketika kita mendirikan variabel, kita juga memesan sebuah memori. Variabel tersebut akan bertindak layaknya sebagai pengenal yang memegang alamat dari memori itu secara langsung. Dan reference adalah variabel yang saat mendirikan tidak memesan memori untuk dirinya sendiri tapi memegang alamat memori dari sebuah memori, variabel atau object yang sudah ada.

Setelah kita mendirikan variabel reference kita dimungkinkan untuk bebas mengakses nilai dari memori melalui variabel asli atau variabel reference, jadi variabel reference hanyalah sebuah variabel yang bertingkah sebagai alias dari suatu penyimpanan.

Untuk mendirikan variabel reference dibutuhkan tanda ( & ) “Operator Reference”, diletakan sebelum identitas dari variabel.

Bentuk penulisan

tipeData &identifier = var;

contoh penulisan

int &ref = myVar;

di dalam materi pointer kita juga diperkenalkan dengan operator bertanda ( & ) yang bernama “address-of”, meskipun menggunakan tanda yang sama tapi mereka akan berbeda tergantung dimana kita meletakannya.

Jika tanda ( & ) diletakan saat memanggil sebuah variabel maka tanda itu akan bekerja sebagai “address-of” tapi jika anda meletakanya saat mendirikan variabel maka akan bekerja sebagai “reference”.

Pada contoh penulisan di atas, pernyataan mendirikan sebuah variabel reference yang bernama “ref” akan bertingkah sebagai alias dari variabel “myVar”. Pada variabel “ref” kita dimungkinkan untuk mengakses memori yang dipegang oleh “myVar” melalui variabel reference bernama “ref”.

Variabel reference mewajibkan inisialisasi pada saat didirikan dan arah tujuan dari variabel reference tidak bisa diperbaharui. Tujuan (menunjuk) dari variabel pointer hanya bisa ditentukan saat mendirikan variabel reference dan tidak akan bisa diperbaharui.

Int &myRef; //Error

int &myRef = myVar; //Ok

myRef = &myOtherVar //Error

Macam-macam Reference

Reference pada Non-const

Variabel reference non-const hanya dapat memegang atau menunjuk ke variabel non-const, tidak bias menunjuk atau memegang variabel berkonstanta, ini adalah peraturan yang di buat oleh C++ untuk keamanan program anda. Jika itu dimungkinkan maka hal itu akan membuat nilai dari variabel konstanta tidak aman (dapat diubah).

int const a = 1;

int &b = a; //Error

Contoh Program ;

#include <iostream>

int main()

{

int a = 1;

int &b = a;

std::cout<<a<<"\n";

b = 2;

std::cout<<a;

return 0;

}

Pada contoh di atas kita diperlihatkan bahwa perubahan yang terjadi pada variabel reference “b” juga berpengaruh pada variable “a”, hal itu dikarenakan mereka memiliki satu memori yang sama. Itu alasan mengapa perubahan pada satu variabel akan berpengaruh pada variabel lainnya.

Reference pada Const

Untuk mendirikan sebuah variabel reference yang dapat menunjuk ke variabel Konstanta, dibutuhkan variabel reference yang bersifat konstanta.

cons int var = 1;

cons tint &ref = var; //Ok

Kita tidak dimungkinkan untuk mendirikan variabel reference non-const menunjuk ke variabel const, tapi kita dimungkinkan untuk mendirikan variabel reference konstanta menunjuk ke variabel biasa.

Contoh Program

#include <iostream>

int main()

{

int a = 1;

int const &b = a;

std::cout<<b<<"\n";

a = 2;

std::cout<<b;

return 0;

}

Perbedaan Pointer dan Reference

Berdasarkan penjelasan pointer pada artikel sebelumnya dan penjelasan mengenai reference di atas, sekelebat kita bisa menyimpulkan bahwa mereka memiliki kemiripan. Kedua hal tersebut memang sama-sama bisa menunjuk ke sebuah memori dari variabel atau obyek, kita bisa menyimpulkan mereka bersaudara tapi mereka berbeda jauh. Terdapat banyak perdaaan antara pointer dan reference.

Cara penggunaan reference sangat berbeda dengan pointer

Untuk mendirikan variabel pointer membutuhkan memori sedangkan variabel reference tidak membutuhkan memori.

Reference tidak bisa didirikan jika tidak disertai inisialisasi sedangkan pointer inisialisasi merupakan opsional.

Tujuan yang ditunjuk oleh pointer dapat kita ubah setelah deklarasi sedanhkan reference tidak dimungkinkan.

Reference bertindak layaknya alias dari variabel sedangkan pointer bertindak sebagai variabel individu.

Contoh Program

#include <iostream>

int main()

{

int myVar = 1;

int &myRef = myVar;

int \*myPtr;

std::cout<<myVar<<"\n";

myRef = 2;

std::cout<<myVar<<"\n";

myPtr = &myVar;

\*myPtr = 3;

std::cout<<myVar<<"\n";

return 0;

}

Fungsi dari variabel reference adalah variabel alias untuk memori dari variabel atau obyek lain yang sudah ada. Biasanya variabel reference digunakan dalam parameter (melakukan perubahan pada parameter sekaligus pada argumen), membuat jalan pintas (seperti mengakses member dari class yang memiliki jalan yang cukup panjang) dan lain-lain.

Tambahan :

Pada variabel reference bersifat konstanta kita dimungkinkan untuk menginisialisasi dengan nilai (bukan variabel), Nilai itu dapat berupa nilai tunggal atau merupakan operasi/ekspresi.

Rvalue merupakan nilai sumber dan lvalue merupakan memori tujuan. Sebenarnya pada saat proses penugasan (assignment) untuk mendirikan rvalue dibutuhkan memori yang bersifat sementara untuk memegang nilai tersebut, setelah pernyataan tersebut dijalankan maka memori itu akan segera dihancurkan.

Tapi jika kita mendirikan variabel reference dengan inisialisasi berupa nilai maka hidup dari memori rvalue (sumber) akan bertahan lebih panjang sampai variabel reference dihancurkan. Di atas dijelaskan bahwa reference itu tidak memiliki memori dan bertingkah sebagai alias/pengenal dari suatu memori yang dapat berupa variabel, obyek atau memori. Itulah alasanya kenapa memori rvalue dari variabel reference tidak dihancurkan.

#include <iostream>

int main()

{

int const &b = 2; //Hidup memori rvalue diperpanjang

std::cout<<b<<"\n";

return 0;

}//memori rvalue dihancurkan bersamaan dengan variabel reference