[](https://1.bp.blogspot.com/-9tFE8B0u19c/VrQ10uG9VgI/AAAAAAAADCs/365qJ2guwtw/s1600/pointer+pada+konstanta.jpg)

Pointer dengan Konstanta adalah salah satu pemanfaatan dari Pointer dan Konstanta yang memungkinkan kita untuk membatasi fungsi atau fasilitas dari pointer tersebut, Pointer adalah adalah variabel yang mempuyai atau menunjuk ke alamat dari variabel yang di tunjuk, yang memungkinkan kita untuk mengubah dan membaca nilai dari variabel yang di tunjuk tanpa melalui identifier yang di tunjuk. Dan Konstana adalah salah satu nilai yang sifatnya akan selalu tetap dan tidak bisa kita ubah. Kesimpulanya adalah Pointer Berkonstanta memperbolehan kita untuk membaca variabel yang di tunjuk atau akan ditunjuk tapi tidak untuk mengubah nilai variabel yang di tunjuk oleh pointer berkonstanta melalui pointer tersebut, atau juga tidak diperbolehkan untuk mengubah alamat yang akan di tunjuk.

Bentuk Umum Penulisan :

const [TipeData] \*[identifier]=[initialization/variable];

Contoh :

const int \*nilai=10;

Contoh Program :

int main( )

{

int var1=10;

const int \*ptr =&var1;

\*ptr = 22;

return 0;

}

Pada contoh program di atas adalah contoh dari penggunaan konstanta pada pointer.pada baris pertama kita dirikan variabel integer “nilai” bernilai ”10”, baris kedua kita dirikan variabel pointer “ptr” dengan konstanta dan kita arahkan ke variabel “nilai”, dan baris ke tiga merupakan contoh pembuktian dari pengertian di atas, kita tau bahwa pernyataan pada baris ke tiga adalah salah, dan tidak akan dilksanakan karena variabel yang kita coba untuk kita ubah merupakan variabel konstanta. Dari program di atas akan menghasilkan pesan error seperti “Cannot modify a const object.” yang terjadi karena pengubahan nilai variabel berkonstanta. Pesan yang akan di tampilkan mungkin akan berbeda-beda tergantung oleh compiler apa yang anda gunakan. Pada compiler Code Block menghasilkan pesan “error: assignment of read-only location ‘\* ptr’. ” tetapi kesimpulaya masih sama yaitu kita tidak bisa mengubah nilai yang di tunjuk oleh pointer berkonstanta melalui variabel pointer tersebut, yang bisa kita lakukan hanyalah membaca nilai dari variabel yang ditunjuk.

Pada contoh penulisan pointer berkonstanta sebenarnya pointer dengan konstanta memiliki banyak variasi bentuk penulisan, begitu juga dengan fungsinya. Berikut adalah macam-macam bentuk penulisan.

Tips : Untuk mempermudah pembacaan dan pemahaman, Silahkan lakukan pembacaan Statement dimulai dari belakang.

Macam-Macam Fungsi Pointer Konstanta

Constant Reference

Constanta Reference adalah perpaduan konstanta dengan pointer reference, fungsi reference adalah untuk membuat nama lain/identifier lain tanpa memesan memori baru. Jika kita buat konstanta kita bisa membuat 2 identifier dengan satu variabel, seperti membuat 2 jalan untuk menuju satu tempat dengan beberapa halangan yang berbeda, variabel dengan konstanta hanya di izinkan untuk pembacaan tidak untuk di edit dan variabel lain bebeas untuk mengedit.

Bentu Penulisan :

const [tipeData] &[identifier] = [initialization/variable]; //atau

[tipeData] const &[identifier] = [initialization/variable];

Contoh Penulisan :

const int &ptr\_const\_nilai = nilai; //atau

int const &ptr\_const\_nilai = nilai;

//Cara Membacanya adalah Pointer reference ke konstanta int.

Contoh Program :

int main( )

{

int var1 = 10;

const int &ptr1 = var1;

int const &ptr2 = var1;

var1 = 50; //OK

ptr1 = 23; //Error (Read-only)

ptr2 = 98; //Error (Read-only)

return 0;

}

Constanta Pointer

Merupakan perpaduan konstanta dengan pointer dereference, pointer difungsikan untuk membuat variable yang akan menunjuk ke variabel lain dan konstanta berfungsi untuk membuat variabel mempunyai nilai tetap. Dan ingat penulisan konstanta pada pointer bermacam-macam begitu pula fungsinya untuk kali ini berfungsi untuk membuat variabel yang akan menunjuk ke variabel lain dan kita tidak bisa mengubah atau memperbaharui alamat memori yang akan di tunjuk.

Bentuk Penulisan :

[tipeData] \* const [identifier];

Contoh Penulisan :

int \*const ptr; //pointer konstanta ke int

Contoh Program :

int main( )

{

int var1 = 20;

int var2 = 50;

int \*const ptr = &var1;

\*ptr=230; //OK

ptr=&var2;//Error

return 0;

}

Pointer to Constant

Merupakan perpaduan dari pointer dereference dengan konstanta. Fungsinya berlawanana seperti fungsi penulisan di atas “Constanta Pointer”. Berfungsi untuk membuat variabel pointer dereference yang menunjuk ke variabel lainya, dari pernyataan tersebut kita diperbolehkan untuk mengubah alamat variabel yang akan di tunjuk tetapi kita tidak bisa mengubah nilai variabel yang di tunjuk.

Bentuk Penulisan :

[tipeData] const \*[identifier]; //atau

const [tipeData] \*[identifier];

Contoh Penulisan :

int const \*ptr1; //atau

const int \*ptr2;

//Cara Membacanya adalah Pointer dereference ke konstanta int.

Contoh Program :

int main( )

{

int var1 = 20;

int var2 = 50;

int const \*ptr1 = &var1;

const int \*ptr2 = &var1;

ptr1=&var2;//OK

\*ptr1=230; //Error

return 0;

}

Constant pointer to a Constant

Merupakan kombinasi dari pointer dan konstanta. Memiliki kedua fungsi dari “Pointer to Constant” dan “Constanta Pointer”. berfungsi untuk mendirikan variabel ponter dereference yang menunjuk ke variabel lain, dari pernyataan tersebut kita hanya diperbolehan untuk membaca alamat memori yang ditunjuk dan nilai variabel yang di tunjuk. Mereka akan menjadi nilai tetap yang tidak bisa diubah.

Bentuk Penulisan :

[tipeData] const \*const [identifier]; //atau

const [tipeData] \*const [identifier];

Contoh Penulisan :

int const \*const ptr1; //atau

const int \*const ptr2;

//Cara Membacanya adalah Konstanta Pointer dereference ke konstanta int.

Contoh Program :

int main( )

{

int var1 = 20;

int var2 = 50;

int const \* const ptr1 = &var1;

const int \* const ptr2 = &var1;

ptr1=&var2;//Error

\*ptr1=230; //Error

return 0;

}

Bentuk tata cara penulisan konstanta pointer akan menghasilkan fungsi berbeda, tetapi tidak untuk beberapa penulisan ini.

const int &ptr\_const\_nilai = nilai; == int const &ptr\_const\_nilai = nilai;

int const \*ptr1; == const int \*ptr2;

int const \* const ptr1; == const int \* const ptr2;

sebenarnya cukup mudah untuk membedakanya. Cukup hapalkan ini saja, jika “const” berada pada di sebelah kiri “const int \*ptr” konstanta itu milik variabel yang di tunjuk pointer tetapi jika “const” berada pada sebelah kanan “int \*const ptr” konstanta itu milik pointer tersebut. Di bawah ada beberapa pernyataan yang juga bekerja pada Pointer Berkonstanta.

int \*\* //Pointer ke Pointer dan ke int

int \*\* const //konstanta pointer ke pointer ke int

int \* const \* //Pointer ke Konstanta pointer ke int

int const \*\* //Pointer ke pointer ke konstanta int

int \* const \* const //konstanta pointer ke konstanta pointer ke int

int const \* const \* const //Konstanta pointer ke Konstanta pointer ke konstanta int

//dan seterusnya

Cukup sekian dari belajarcpp.com semoga artikel ini bisa bermanfaat untuk kita semua. Mohon maaf atas kekuranganya dan Terima Kasih atas dukungan dan kunjungan anda ke BelajarCPP. Have a nice day.

Last update : 4 Februari 2016