Algoritma dan Pemrograman Lanjut

Pertemuan Ke-11
Pointer 2

Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.

Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional "Veteran"
Yogyakarta

Algoritma dan Pemrograman Lanjut

Judul Materi : Pointer 2

Deskripsi Materi : Materi ini membahas tipe data pointer, operator-operator yang dapat dipakai dana penggunaannya bersama tipe data terstrutur yang lain

Tujuan Instruksional Khusus :

1. Mendefinisikan dan menggunakan tipe data pointer
2. Mendeskripsikan tipe data pointer
3. Memahami kegunaan pointer

Referensi :

- Buku Teks

Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 2, Edisi Ketiga, Penerbit Informatika Bandung.
Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman II*, Edisi Kedua, Yogyakarta

- Buku Acuan/Referensi

Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.
Jarne, Stroustrup B. (1997), *C++ Programming language,* AT &T.
Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.
Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.
Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

POINTER 2

PENDAHULUAN

Pointer adalah vriabel yang berisi alamat memori sebagai nilainya dan berbeda dengan variable biasa yang berisi nilai tertentu. Dengan kata lain, pointer berisi alamat dari variable yang mempunyai nilai tertentu.

ISI

A. Operator Pointer

Ada beberapa operator yang bisa digunakan dalam pointer, yaitu :

1. Operator alamat (yang dilambangkan dengan symbol &)
2. Operator unary yang mengembalikan alamat dari *unary* yang mengembalikan alamat dari *operandnya.*

Contoh Program 1:

```
#include<iostream.h>
main()
{
  int *ptr, num;
  ptr = &num;
  *ptr = 100;
  cout << num << " ";
  (*ptr)++;
  cout << num << " ";
  (*ptr)*2;
  cout << num << "\n";
}
```

B. Expresi Pointer

- **Pointer Aritmatika**

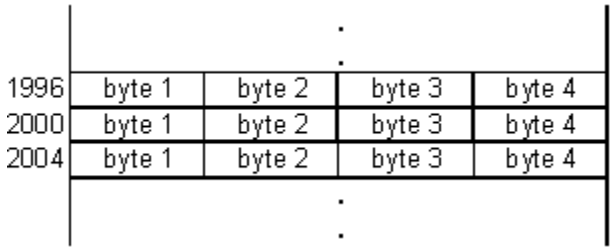
  Hanya 4 operator aritmatik dapat digunakan pada pointer + +, = = , + , dan -. Asumsi integer 32 bit.

Perhatikan contoh berikut.

int *p1 // assume; p1==2000

p1++;

p1--;



Pointer Aritmatika

Contoh program 2 pointer Aritmatika.

```
#include<iostream.h>
main()
{ int i[10], *i_ptr;
  double f[10], *f_ptr;
  int x;
  i_ptr = i;   //i_ptr points to first element of i
  f_ptr = f;   //f_ptr points to first element of f
  for(x=0;x<10;x++)
  cout << i_ptr+x << " " << f_ptr+x << "\n";
}
```

- **Pointer Perbandingan**

  Pointer dapat dibandingkan dengan menggunakan operator hubungan, seperti != , = =, < , dan >.

  Contoh Program 3 :

```
#include<iostream.h>
main()
{
  int num[10];
  int *start, *end;

  start = num;
  end = &num[9];
  while(start != end) {
    cout << "Masukkan bilangan sebanyak 9 data: ";
    cin >> *start; start++;
  }
}
```

C. Pointer Dan Array

Array dan pointer adalah dua struktur data yang saling berkaitan satu sama lain didalam C, dan dapat saling dipertukarkan penggunaannya. Hal ini karena suatu array dapat didefinisikan sebagai pointer ke elemen pertama dari array tersebut.

Pointer dapat di-array seperti tipe data yang lain dalam C++. Sebagai contoh, untuk menyatakan sebuah array pi dari pointer sebanyak 10 buah data yang bertipe 10 integer, dapat dituliskan sebagai berikut :

int *pi[10];

Untuk menentukan alamat dari variable integer disebut var ke elemen ketiga dari pointer array, dapat dituliskan sebagai berikut :

int var;

Pi[2] = &var

Contoh program 4 array pointer

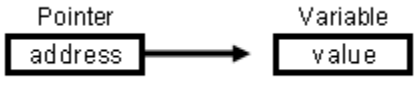
```
#include<iostream.h>
main()
{
  int numbers[5];
  int *p;

  p = numbers; *p = 10;
  p++; *p = 20;
  p = &numbers[2]; *p = 30;
  p = numbers + 3; *p = 40;
  p = numbers; *(p+4) = 50;
  for(int n=0;n<5;n++)
    cout << numbers[n] << ", ";
 }
```

D. Pointer dalam Pointer.

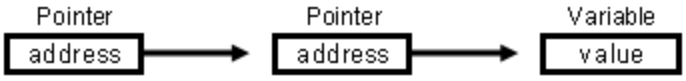
Diperbolehkan dalam C++ menggunakan pointer dalam pointer yang masih bisa berisi data yang sama atau berbeda.

Dalam kondisi pointer biasa atau pointer tunggal, diagramnya adalah sebagai berikut :



Pointer Tunggal

Untuk Pointer dalam Pointer, diagramnya adalah sebagai berikut :

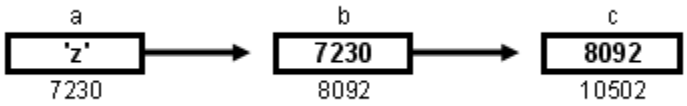


Pointer dalam Pointer

Contoh :

```
char a;
char *b;
char **c;
a = 'z';
b = &a;
c = &b;
```

dan misalnya berisi data-data yang acak pada memori 7230, 8092, dan 10502, maka diagramnya adalah sebagai berikut :



Dari diagram diatas dapat disimpulkan :

- c adalah sebuah variable dengan tipe (char **) yang berisi 8092.
- *c adalah sebuah variable dengan tipe (char *) yang berisi 7230.
- **c adalah sebuah variable dengan tipe (char) yang berisi 'z'.

Contoh program 5 pointer dalam pointer

```
#include<iostream.h>
main()
{
  int x, *p, **q;
  x = 10;
  p = &x;
  q = &p;
  cout << **q; //prints the value of x
}
```

Penjelasan Program :

Proses pendeklarasian pointer dalam pointer dapat dilihat pada pernyataan :

```
int x, *p, **q;
x = 10;
p = &x;
q = &p;
```

dan hasilnya adalah : 10

E. Masalah yang kadang terjadi pada pointer.

1. Pointer yang tidak di inisialisasikan.
2. Null pointer.
    - Sesudah pointer dinyatakan dan sebelum nilainya ditentukan, pointer akan berisi sebuah nilai yang berubah-ubah. Jika pointer berisi Null atau nilai (0), diasumsikan untuk tidak menunjuk apapun.
    - Beberapa tipe pointer dapat dimulai ke Null ketika dinyatakan sehingga menghindari salah penggunaan atau pointer yang tidak di inisialisasikan.
3. Kesalahan dalam pointer perbandingan.

   Membuat perbandingan pointer antara dua obyek yang berbeda, mungkin menghasilkan hasil yang tidak diharapkan.
4. Pengembalian nilai pointer.

   Kadang setelah mendeklarasikan nilai pointer, kita lupa mengembalikan ke nilai semula.

PENUTUP

Pointer adalah tipe data dalam pemrograman yang dapat dikenai operator aritmatika tertentu dan operator perbandingan. Pointer dapat dipadukan dengan tipe data terstruktur array.

SOAL-SOAL

1. Apa output dari contoh program 1 sampai dengan 4 di atas.
2. Apa output dari program berikut :

```
#include<iostream.h>
main()
{
  int m, n, *x, **y;
  m = 10;
  x = &m;
  y = &x;
  n = **y + 5;
  x = &n;
  cout << m << endl;
  cout << n << endl;
  cout << *x << endl;
  cout << **y << endl;
 }
```