

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AWAL PRAKTIKUM

Pertemuan ke-04

Pointer



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. RANGKUMAN MATERI

Pointer adalah salah satu konsep yang penting dalam bahasa pemrograman C++. Pointer memungkinkan kita untuk mengakses dan memodifikasi nilai atau variabel yang disimpan pada alamat memori tertentu. Dalam C++, setiap variabel memiliki alamat memori unik yang dapat diakses melalui operator address-of (&). Variabel pointer dapat digunakan untuk menyimpan alamat memori dari variabel lain, sehingga kita dapat mengakses atau memodifikasi nilai variabel tersebut melalui variabel pointer.

Selain itu, pointer juga memungkinkan pengalokasian memori secara dinamis dengan menggunakan operator new dan delete. Dalam penggunaannya, pointer harus diinisialisasi dengan nilai awal, yaitu alamat memori dari variabel atau objek yang ditunjuk. Kemudian, kita dapat menggunakan operator dereference (*) untuk mengakses nilai yang disimpan pada alamat memori yang ditunjuk oleh variabel pointer. Pemahaman tentang pointer sangat penting bagi para programmer untuk memaksimalkan penggunaan memori pada program dan mengoptimalkan kinerja program.

B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Pointer!

Jawab :

Pointer adalah sebuah variabel yang menyimpan alamat memori dari suatu nilai atau variabel lainnya. Pointer memungkinkan kita untuk mengakses dan memodifikasi nilai atau variabel yang disimpan pada alamat memori tertentu.

2. Apa yang dimaksud dengan Variabel Pointer!

Jawab :

Variabel pointer adalah sebuah variabel yang nilainya adalah alamat memori dari variabel lainnya. Variabel pointer digunakan untuk menyimpan alamat memori dari variabel lain, sehingga kita dapat mengakses atau memodifikasi nilai variabel tersebut melalui variabel pointer.

3. Sebutkan dan jelaskan jenis-jenis Operator Pointer!

Jawab :

Jenis-jenis operator pointer adalah:

- Operator dereference (*) : digunakan untuk mengakses nilai yang tersimpan pada alamat memori yang ditunjuk oleh pointer.
- Operator address-of (&) : digunakan untuk mendapatkan alamat memori dari suatu variabel.
- Operator assignment (=) : digunakan untuk menetapkan alamat memori ke dalam variabel pointer.
- Operator increment/decrement (++/--): digunakan untuk menggeser alamat yang ditunjuk oleh pointer, sehingga pointer mengarah ke alamat memori yang berbeda.

4. Buatlah contoh program sederhana menggunakan pointer!

Jawab :

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 10;
    int *ptr;

    ptr = &x;

    cout << "Nilai x = " << x << endl;
    cout << "Alamat memori x = " << &x << endl;
    cout << "Nilai yang ditunjuk oleh ptr = " << *ptr << endl;
    cout << "Alamat memori yang ditunjuk oleh ptr = " << ptr << endl;
    cout << "Alamat memori dari ptr = " << &ptr << endl;

    *ptr = 20;
    cout << "Nilai x sekarang = " << x << endl;

    return 0;
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai x = 10
Alamat memori x = 0x7ffd9fcaa31c
Nilai yang ditunjuk oleh ptr = 10
Alamat memori yang ditunjuk oleh ptr = 0x7ffd9fcaa31c
Alamat memori dari ptr = 0x7ffd9fcaa320
Nilai x sekarang = 20
```

C. TUGAS PRAKTIKUM

- Lat4_1

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 8, y;
    int *xPtr = &x;

    cout << "Nilai x \t: " << x << endl;
    cout << "Alamat x \t: " << &x << endl;
    cout << "Nilai xPtr \t: " << xPtr << endl;

    cout << "Nilai yang disimpan pada alamat " << xPtr << " adalah " << *xPtr <<
endl;

    return 0;
}
```

Output

```
/tmp/IxJMt2l2dz.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai x    : 8
Alamat x   : 0x7ffd085bcc0c
Nilai xPtr : 0x7ffd085bcc0c
Nilai yang disimpan pada alamat 0x7ffd085bcc0c adalah 8
```

- Lat4_2

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";
```

```

int x = 12, y;
int *xPtr = &x;

cout << "Nilai x \t: " << x << endl;
cout << "Alamat x \t: " << &x << endl;
cout << "Nilai xPtr \t: " << xPtr << endl;

cout << "Nilai yang disimpan pada alamat " << xPtr << " adalah " << *xPtr <<
endl;

return 0;
}

```

Output

```

/tmp/IxJMt2l2dz.o
Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai x      : 12
Alamat x     : 0x7ffc5ddd84fc
Nilai xPtr   : 0x7ffc5ddd84fc
Nilai yang disimpan pada alamat 0x7ffc5ddd84fc adalah 12
|

```

- Lat4_3

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[] = { 45, 23, 50, 8, 12, 10, 15 };
    int *NilaiPtr = Nilai;

    int i;
    NilaiPtr = Nilai;

    cout << "Nilai Ptr \t\t: " << NilaiPtr << endl;
    cout << "Alamat Nilai \t: " << &Nilai << endl;
    cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr
    << endl;

    cout << "Element array index pertama: " << NilaiPtr[0] << "\n\n";

    for (i = 1; i < 7; i++)
    {

```

```

    cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr
    << endl;
    NilaiPtr++;
}

return 0;
}

```

Output

```

/tmp/IxJMt2l2dz.o
Nama    : Nova Ardiansyah
NIM     : 211011401309
=====

Nilai Ptr      : 0x7ffa9f01630
Alamat Nilai   : 0x7ffa9f01630
Nilai yang disimpan pada alamat 0x7ffa9f01630 adalah 45
Element array index pertama: 45

Nilai yang disimpan pada alamat 0x7ffa9f01630 adalah 45
Nilai yang disimpan pada alamat 0x7ffa9f01634 adalah 23
Nilai yang disimpan pada alamat 0x7ffa9f01638 adalah 50
Nilai yang disimpan pada alamat 0x7ffa9f0163c adalah 8
Nilai yang disimpan pada alamat 0x7ffa9f01640 adalah 12
Nilai yang disimpan pada alamat 0x7ffa9f01644 adalah 10

```

- Lat4_4

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int x = 8;
    int *xPtr1;
    int **xPtr2;

    xPtr1 = &x;
    xPtr2 = &xPtr1;

    cout << "Nilai x \t\t: " << x << endl;
    cout << "Alamat x \t\t: " << &x << endl;
    cout << "Nilai xPtr1 \t: " << xPtr1 << endl;
    cout << "Alamat xPtr1 \t: " << &xPtr1 << endl;
    cout << "Nilai xPtr2 \t: " << xPtr2 << endl;
    cout << "Alamat xPtr2 \t: " << &xPtr2 << endl;
}

```

```
return 0;
}
```

Output

```
/tmp/IxJMt2l2dz.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Nilai x      : 8
Alamat x     : 0x7ffefba01b94
Nilai xPtr1  : 0x7ffefba01b94
Alamat xPtr1 : 0x7ffefba01b98
Nilai xPtr2  : 0x7ffefba01b98
Alamat xPtr2 : 0x7ffefba01ba0
|
```

- Lat4_5

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[] = { 11, 23, 17, 8, 21, 10, 25 };
    int *NilaiPtr = Nilai;

    NilaiPtr = Nilai;

    cout << "Nilai Ptr \t\t: " << NilaiPtr << endl;
    cout << "Alamat Nilai \t: " << &Nilai << endl;
    cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr
    << endl;

    cout << "Element array index pertama: " << NilaiPtr[0] << "\n\n";

    while (NilaiPtr < Nilai + 7)
    {
        cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr
        << endl;
        NilaiPtr++;
    }
}
```



```
return 0;  
}
```

Output

```
/tmp/IxJMt2l2dz.o
```

```
Nama      : Nova Ardiansyah
```

```
NIM       : 211011401309
```

```
=====
```

```
Nilai Ptr      : 0x7ffd7bcf8210
```

```
Alamat Nilai   : 0x7ffd7bcf8210
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8210 adalah 11
```

```
Element array index pertama: 11
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8210 adalah 11
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8214 adalah 23
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8218 adalah 17
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf821c adalah 8
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8220 adalah 21
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8224 adalah 10
```

```
Nilai yang disimpan pada alamat 0x7ffd7bcf8228 adalah 25
```

LAPORAN PRAKTIKUM STRUKTUR DATA

LAPORAN AKHIR PRAKTIKUM

Pertemuan ke-04

Pointer



Disusun Oleh:

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566
Tangerang Selatan - Banten

A. TUGAS AKHIR

1. Buatlah program dengan menggunakan kombinasi antara Pointer dan Array Dua Dimensi!

Jawab :

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[2][3] = { { 11, 23, 17 }, { 8, 21, 10 } };
    cout << "Array : " << "{ { 11, 23, 17 }, { 8, 21, 10 } }\n\n";

    for (int *NilaiPtr = Nilai[0]; NilaiPtr < Nilai[0] + 6; NilaiPtr++)
    {
        cout << "Nilai yang disimpan pada alamat " << NilaiPtr << " adalah " << *NilaiPtr << endl;
    }

    return 0;
}
```

Output

```
/tmp/IxJMt2l2dz.o
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Array : { { 11, 23, 17 }, { 8, 21, 10 } }

Nilai yang disimpan pada alamat 0x7fffa30b12a0 adalah 11
Nilai yang disimpan pada alamat 0x7fffa30b12a4 adalah 23
Nilai yang disimpan pada alamat 0x7fffa30b12a8 adalah 17
Nilai yang disimpan pada alamat 0x7fffa30b12ac adalah 8
Nilai yang disimpan pada alamat 0x7fffa30b12b0 adalah 21
Nilai yang disimpan pada alamat 0x7fffa30b12b4 adalah 10
```

2. KESIMPULAN

Dalam bahasa pemrograman C++, pointer memungkinkan kita untuk mengakses dan memodifikasi nilai atau variabel yang disimpan pada alamat memori tertentu. Dengan menggunakan pointer, kita dapat mengakses dan memodifikasi variabel melalui alamat memori yang ditunjuk oleh variabel pointer. Oleh karena itu, pemahaman tentang pointer sangat penting bagi para programmer untuk memaksimalkan penggunaan memori pada program dan mengoptimalkan kinerja program.