**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

1. **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.

1. **TUGAS PENDAHULUAN**
2. 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.

1. 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.

1. 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.

1. 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;  } |
|  |

1. **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;  } |
|  |

* 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;  } |
|  |

* 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;  } |
|  |

* 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;  } |
|  |

* 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;  } |
|  |

**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

1. **TUGAS AKHIR**
2. 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;  } |
|  |

1. **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.