**LAPORAN PRAKTIKUM**

**STRUKTUR DATA DAN ALGORITME**

**MODUL III**

**POINTER**

Logo

Description automatically generated

**Disusun Oleh :**

Nama : Fatkhurrohman Purnomo

NIM : 21102125

**Dosen Pengampu**

Ipam Fuaddina Adam, S.T., M.Kom.

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**PURWOKERTO**

**2022**

1. **Dasar Teori**

**Tipe data pointer** adalah tipe data yang digunakan untuk menunjukkan alamat memori dari suatu variabel yang ditunjuknya. Variabel yang dapat dirujuk oleh suatu pointer adalah variabel yang memiliki **tipe data yang sama** dengan tipe data pointer. Setiap variabel tersimpan di dalam suatu alamat memori yang berbeda-beda. Alamat memori disebut dengan *address* atau *reference*.

Misalkan memiliki variabel x dan terletak di memori 0x000001. Jika ingin memasukkan nilai 100 kedalam variabel x, maka processor harus membawa nilai 100 tersebut kedalam variabel x yang terletak di alamat memori 0x000001. Hal yang perlu ketahui adalah, setiap variabel ternyata memiliki ukuran byte yang berbeda-beda dalam memori. Sebagai contoh suatu variabel bertipe int memiliki ukuran 4 byte dalam memori. Maka variabel tersebut akan menempati 4 kapling lokasi dalam memori, misalkan 0x000001, 0x000002, 0x000003, dan 0x000004. Jika terdapat dua buah variabel bertipe int yang bersebelahan, maka alamat variabel pertama terletak di 0x000001 dan variabel kedua terletak dialamat 0x000005.

Deference (&) merupakan suatu operator yang berfungsi untuk menanyakan alamat dari suatu variabel. Apabila memberikan simbol & pada awal variabel dan mencetak hasilnya pada jendela CLI, maka yang akan tercetak adalah alamat dari variabel tersebut bukan nilai yang ditampung oleh variabel tersebut.

Reference (\*) merupakan suatu operator yang berfungsi menyatakan suatu variabel adalah variabel pointer. Sama halnya dengan operator deference, peletakan simbol operator reference diletakan diawal variabel. Operator reference ini akan membuat suatu variabel pointer untuk menampung alamat.

*Ref:*

[Pointer - Belajar C++ (belajarcpp.com)](https://www.belajarcpp.com/tutorial/cpp/pointer/)

[Pointer Pada Pemrograman C++ – SinauArduino](https://www.sinauarduino.com/artikel/pointer-pada-pemrograman-cpp/)

Modul 3 Pointer

1. **Guided**
2. **Program Pointer 1**

|  |
| --- |
| // Nama : Fatkhurrohman Purnomo  // NIM : 21102125  #include <iostream>  using namespace std;  // pointer  int main(){      int v = 7; // menginisialisasi variabel v dengan nilai 7      int \*p; // menginisialisasi pointer p dengan nilai NULL      cout << "v = " << v << endl; // mencetak nilai variabel v      cout << "p = " << \*p << endl; // mencetak nilai pointer p      cout << "v = " << &v << endl; // mencetak alamat dari variabel v      cout << "p = " << &p << endl; // mencetak alamat dari pointer p      return 0;  } |

**Deskripsi:**

Program diatas dibuat untuk melihat dimana letak memori yang menyimpan.

Yang pertama melakukan inisialisasi variabel integer, dilanjut dengan pointer. Kemudian menampilkan hasilnya di layar, dan akan terlihat letak memorinya.

**Output:**

Text

Description automatically generated

1. **Re-inisialisasi variabel dengan pointer**

|  |
| --- |
| // Nama : Fatkhurrohman Purnomo  // NIM : 21102125  #include <iostream>  using namespace std;  // pointer  int main(){      int value1 = 5, value2 = 15; // menginisialisasi variabel value1 dengan nilai 5 dan value2 dengan nilai 15      int \*mypointer; // menginisialisasi pointer mypointer dengan nilai NULL      cout << "value1 = " << value1 << endl; // mencetak nilai variabel value1      cout << "value2 = " << value2 << endl; // mencetak nilai variabel value2      cout << endl;      // inisialisasai ulang dengan menggnakan pointer      mypointer = &value1; // menginisialisasi pointer mypointer dengan alamat dari variabel value1      \*mypointer = 10; // mengisi nilai dari pointer mypointer dengan 10      cout << "value1 = " << value1 << endl; // mencetak nilai variabel value1      cout << "value2 = " << value2 << endl; // mencetak nilai variabel value2      return 0;  } |

**Deskripsi:**

Program diatas dibuat untuk merubah nilai yang ada di dalam variabel menggunakan pointer.

Yang pertama melakukan inisialisasi variabel integer, dilanjut dengan pointer. Program akan menampilkan nilai awal dari setiap variabel. Lalu menyamakan variabel dengan pointer, dan mencetaknya ke layar dengan nilai baru yang dimasukan melalui pointer.

**Output:**

Text

Description automatically generated

1. **Array Dengan Pointer**

|  |
| --- |
| // Nama : Fatkhurrohman Purnomo  // NIM : 21102125  #include <iostream>  using namespace std;  int main(){      int data[] = {1,2,3,4,5}; // menginisialisasi array data dengan nilai 1,2,3,4,5      int \*pData = data; // menginisialisasi pointer pData dengan alamat dari array data      cout << pData[0] << endl; // mencetak nilai dari array data dengan index 0      cout << pData[1] << endl; // mencetak nilai dari array data dengan index 1      cout << pData[2] << endl; // mencetak nilai dari array data dengan index 2      cout << pData[3] << endl; // mencetak nilai dari array data dengan index 3      cout << pData[4] << endl; // mencetak nilai dari array data dengan index 4      return 0;  } |

**Deskripsi:**

Program diatas dibuat untuk mencetak / menampilkan nilai di dalam array dengan menggunakan pointer.

Yang pertama melakukan inisialisasi variabel array integer, dilanjut dengan pointer. Pointer disamakan dengan variabel, lalu hasilnya dicetak dengan memanggil pointer yang ada.

**Output:**

Text

Description automatically generated with medium confidence

1. **Tugas (Unguided)**

**Source code:**

|  |
| --- |
| // Nama : Fatkhurrohman Purnomo  // Nim : 21102125  /\*  \* Buatlah array yang menampung data bebas (minimal 10 data), kemudian buatlah variabel pointer yang  menunjuk ke array tersebut.    1. Tampilkan semua nilai data/elemen dari array melalui pointer yang menunjuknya.    2. Tampilkan semua alamat memori dari tiap elemen array melalui pointer yang menunjuknya.    3.  Lakukan perubahan (re-inisialisasi) beberapa elemen array (minimal 5 data yang diubah) melalui pointer yang menunjuknya.  \*/  #include <iostream>  using namespace std;  int main()  {      //menampung data 10 data      // buat variabel pointer yang menunjuk ke array      int data[] = {1,2,3,4,5,6,7,8,9,10};      int \*pData = data;      cout << "\tSemua data dari array\t" << endl;      cout << "=================================" << endl;      //menampilkan semua nilai data/elemen dari array melalui pointer yang menunjuknya      for (int i = 0; i < 10; i++) //Looping 10 kali      {          cout << "Data ke-" << i + 1 << " : \t" << \*(pData + i) << " \t\t" << endl; //menampilkan data      }      cout << endl << endl;      cout << "\tSemua alamat memori \t" << endl;      cout << "==================================" << endl;      //menampilkan semua alamat memori dari tiap elemen array melalui pointer yang menunjuknya      for (int i = 0; i < 10; i++) //Looping 10 kali      {          cout << " Alamat data ke-" << i + 1 << " : \t" << pData + i << endl; //menampilkan alamat memori      }      cout << endl << endl;      cout << "\tPerubahan data array \t" << endl;      cout << "=================================" << endl;      cout << "=> Data yang diubah\t\t" << endl;      //lakukan perubahan (re-inisialisasi) beberapa elemen array (minimal 5 data yang diubah) melalui pointer yang menunjuknya      for (int i = 0; i < 5; i++) //Looping 5 kali      {          \*(pData + i) = \*(pData + i) + 10; //menambahkan 10 pada data nilai array di ubah          cout << " Data ke-" << i + 1 << " : " << \*(pData + i) << " \t\t" << endl; //menampilkan data yang telah diubah      }      cout << "=> Data yang Tidak diubah\t" << endl;      //menampilkan semua nilai data/elemen array dari data ke 6 sampai terakhir      for (int i = 5; i < 10; i++) //Looping 5 kali      {          cout << " Data ke-" << i + 1 << " : " << \*(pData + i) << " \t\t" << endl; //menampilkan data yang tidak diubah dari data ke 6 sampai terakhir      }      return 0;  } |

**Deskripsi:**

Program yang pertama adalah melakukan inisialisasi array dengan besar array kosong, kemudian melakukan Deklarasi pointer yang disamakan dengan array tadi. Kemudian membuat perulangan for untuk menampilkan isi array dengan menggunakan pointer tadi. Dilanjut dengan menampilkan memori array yang disimpan dengan perulangan for. Kemudian membuat perulangan for lagi, lalu melakukan perkalian dari isi array yang tadi sudah disimpan di pointer, kemudian di cetak. Dan yang terakhir membuat pengulangan for untuk mencetak isi array sebelum di ubah.

1. **Tampilkan semua nilai/elemen dari array melalui pointer yang menunjuknya.**

**Output:**

A picture containing text

Description automatically generated

1. **Tampilkan semua alamat memori dari setiap elemen array melalui pointer yang menunjuknya**

**Output:**Text

Description automatically generated with medium confidence

1. **Melakukan perubahan (re-inisialisasi) beberapa elemen array (minimal 5 data yang diubah) melalui pointer yang menunjuknya**

**Output:**

Text

Description automatically generated

1. **Kesimpulan**
   * + 1. Bisa membuat pointer
       2. Dari modul pointer saya dapat lebih memahami penggunaan pointer
       3. Saya lebih mahir dalam menggunakan bahasa C++
       4. Saya bisa melakukan problem solving bagi program yang error
       5. Lebih paham dalam membuat program
       6. Melatih daya pikir, imajinasi, dan langkah-langkah dalam membuat program
       7. Pointer berguna untuk mengetahui letak dari memori atau bisa untuk menandai