**Modul 2: Pointer**

## Tujuan Pembelajaran

Setelah mengikuti praktikum ini, mahasiswa diharapkan mampu:

1. Memahami konsep dasar pointer serta cara deklarasinya.
2. Menggunakan operator \* (dereference) dan & (reference).
3. Melakukan operasi aritmatika pada pointer.
4. Menghubungkan pointer dengan array.
5. **Materi**

## 1. Pengertian Pointer

* **Pointer** adalah variabel yang menyimpan **alamat memori** dari variabel lain.
* Pointer tidak langsung menyimpan nilai data, melainkan alamat dari data tersebut.
* Untuk mengakses nilai yang ditunjuk digunakan operator dereference (\*).

## 2. Deklarasi Pointer

Bentuk umum:

TipeData \*nama\_pointer;

Contoh:

int \*a; // pointer ke int

char \*p; // pointer ke char

## 3. Operator Pointer

1. **& (Address-of operator)** → mengambil alamat suatu variabel.
2. **\* (Dereference operator)** → mengakses nilai pada alamat yang ditunjuk pointer.

Contoh:

int b = 4;

int \*a = &b; // a menunjuk alamat b

cout <<a;

cout << \*a; // tampilkan isi variabel b

## 4. Aritmatika Pointer dan Array

* Pointer dapat dioperasikan dengan penambahan (+) atau pengurangan (-).
* Nilai bertambah sesuai ukuran tipe data (misal int = 4 byte, double = 8 byte).
* Berguna untuk **traversing array**.
* Nama array (x) sama dengan alamat elemen pertama (&x[0]).
* Pointer dapat digunakan untuk mengakses seluruh elemen array.

Contoh:

int x[]={8, 2, 9};

int \*p = x;

cout << \*p; // 8 x[0]

p++;

cout << \*p; // 2 x[1]

Contoh:

int x[]={9, 2, 1, 7, 4};

int \*p = x;

for (int i = 0; i < 5; i++)

cout << (p + i) << " ";

## 5. Reference

* Alternatif pointer di C++.
* Lebih sederhana karena langsung menjadi alias dari variabel.
* Tidak bisa berpindah alamat setelah diinisialisasi.

Contoh:

int x = 8;

int& r = x; // r adalah alias dari x

cout << r;

## 6. Ukuran Data

* Ukuran variabel dapat dicek dengan sizeof().

Contoh:

cout << sizeof(int); // 4

cout << sizeof(double); // 8

## C. Praktikum

### Percobaan 1. Deklarasi Pointer dan Operator

Program berikut menunjukkan penggunaan operator & dan \*:

#include <iostream>

using namespace std;

int main() {

int b = 4;

int \*a = &b;

cout << "Alamat b : " << a << endl;

cout << "Nilai b melalui pointer : " << \*a << endl;

\*a = 9;

cout << "Nilai b setelah diubah pointer: " << b << endl;

return 0;

}

### Percobaan 2. Aritmatika Pointer

Program berikut menunjukkan perpindahan pointer dalam array:

#include <iostream>

using namespace std;

int main() {

int x[]={8, 2, 9, 4, 6, 3, 1};

int \*p = x;

cout << "Elemen pertama: " << \*p << endl;

p++;

cout << "Elemen kedua: " << \*p << endl;

cout << "Elemen keempat: " << \*(p + 2) << endl;

return 0;

}

### Percobaan 3. Pointer dan Fungsi (Swap)

Program berikut menunjukkan perbedaan passing by pointer:

#include <iostream>

using namespace std;

void swap(int\* x, int\* y) {

int tmp = \*x;

\*x = \*y;

\*y = tmp;

}

int main() {

int a = 3;

int b = 4;

cout << "Sebelum swap: a = " << a << ", b = " << b << endl;

swap(&a, &b);

cout << "Sesudah swap: a = " << a << ", b = " << b << endl;

return 0;

}

### 4. Alokasi Dinamis

Program berikut menunjukkan alokasi dan dealokasi memori:

#include <iostream>

using namespace std;

int main() {

int \*a = new int;

\*a = 10;

cout << "Alamat a: " << a << endl;

cout << "Nilai a: " << \*a << endl;

delete a;

cout << "Memori a sudah dibebaskan." << endl;

return 0;

}

**D. Tugas**

Buatlah sebuah program C++ yang:

* Mendeklarasikan variabel bertipe char, int, float, double, dan string.
* Buat pointer untuk masing-masing variabel.
* Tampilkan:
  + **Alamat variabel asli (&variabel)**
  + **Alamat variabel setelah pointer + 1 (ptr+1)**
  + **Selisih alamat dalam byte**
  + **Ukuran tipe data dengan sizeof**

**Contoh output yang diharapkan**

|  |
| --- |
| === char ===  Alamat c : 0x7ffee22c5a3b  Alamat c + 1 : 0x7ffee22c5a3c  Selisih alamat : 1 byte  Ukuran via sizeof : 1 byte  === int ===  Alamat x : 0x7ffee22c5a20  Alamat x + 1 : 0x7ffee22c5a24  Selisih alamat : 4 byte  Ukuran via sizeof : 4 byte  === float ===  ... dst ...  === double ===  ... dst ...  === string ===  Alamat s : 0x7ffee22c59f0  Alamat s + 1 : 0x7ffee22c5a10  Selisih alamat : 32 byte  Ukuran via sizeof : 32 byte |

Pertanyaan:

1. Bandingkan hasil antara pointer arithmetic dan sizeof. Apakah sama?
2. Mengapa hasil sizeof(int) selalu 4 byte? Sizeof(char) selalu 1 byte?
3. Apakah hasil alamat variabel selalu sama setiap kali program dijalankan? Jelaskan alasannya!