

**LAPORAN PRAKTIKUM**  
**PERTEMUAN 3**  
**ABSTRACT DATA TYPE**



**Nama :**

Martryatus Sofia      2311104003

**Dosen :**

Yudha Islami Sulistya, S.Kom., M.Cs

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO 2024**

## A. Soal Tugas Pendahuluan

1. Jelaskan apa yang dimaksud dengan pointer

Jawab :

Pointer adalah sebuah variabel dalam bahasa pemrograman c++ yang menyimpan alamat memori dari variabel lain, dengan kata lain, pointer menunjuk ke lokasi memori dimana sebuah nilai ata data disimpan.

2. Bagaimana cara menampilkan alamat memori dari suatu variabel dalam program C++? Berikan contoh!

Jawab :

Untuk menampilkan alamat memori dari suatu variabel dalam program C++, bisa menggunakan operator &(Ampersand). Operator ini digunakan untuk mendapatkan alamat memori dari sebuah variabel.

Contoh :



```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 42;
6      float y = 3.14;
7
8
9      cout << "Nilai dari x: " << x << endl;
10     cout << "Nilai dari y: " << y << endl;
11
12     cout << "Alamat memori dari x: " << &x << endl;
13     cout << "Alamat memori dari y: " << &y << endl;
14
15     return 0;
16 }
17
```

&x mengambil alamat memori dari variabel x

&y mengambil alamat memori dari variabel y

3. Bagaimana cara menggunakan pointer dalam program c++? Berikan contoh data menampilkan nilai yang tersimpan pada suatu alamat melalui pointer!

Jawab :

Pointer dalam C++ adalah variabel yang menyimpan alamat memori dari variabel lain. Untuk menggunakan pointer, kita perlu mendeklarasikan variabel pointer, menyimpan alamat memori dari variabel lain, dan kemudian dapat mengakses atau mengubah nilai di alamat memori tersebut melalui pointer.

1. Deklarasikan pointer dengan tipe data diikuti dengan tanda bintang \*. Misalnya `int*`
2. Gunakan operator `&` untuk menyimpan alamat memori dari variabel ke dalam pointer.
3. Untuk mengakses nilai yang ditunjuk oleh pointer, gunakan operator dereferensi `*`.

Contoh :



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int x = 25;
6     int* ptr;
7
8     ptr = &x;
9
10    cout << "Nilai dari x: " << x << endl;
11    cout << "Alamat memori dari x (disimpan dalam ptr): " << ptr << endl;
12
13    cout << "Nilai yang ditunjuk oleh ptr: " << *ptr << endl;
14
15    return 0;
16 }
17
```

**\*ptr:** Operator dereferensi digunakan untuk mengakses nilai yang disimpan di alamat yang ditunjuk oleh ptr. Dengan kata lain, \*ptr memberikan nilai dari x, karena pointer ptr menunjuk ke x.

4. Jelaskan apa yang dimaksud dengan Abstract Data Type(ADT)!

Jawab :

ADT adalah TYPE dan sekumpulan PRIMITIF (operasi dasar) terhadap TYPE tersebut. Selain itu, dalam sebuah ADT yang lengkap, disertakan pula definisi invarian dari TYPE dan aksioma yang berlaku. ADT merupakan definisi STATIK.

5. Berikan contoh ilustrasi sederhana di dalam dunia nyata, tetapi diluar konteks pemrograman!

Jawab :

Contoh mesin ATM sebagai ADT :

Rekening bank : tipe data yang bisa menyimpan informasi saldo, nomor rekening dan riwayat transaksi.

Operasi :

Tarik Tunai : Mengurangi saldo rekening

Setor Tunai : Menambah saldo rekening

Cek Saldo : Melihat saldo

Transfer Uang : Mengirim uang ke rekening lain

6. Tuliskan ADT dari bangun ruang kerucut dalam bahasa C++!

Jawab :

Atribut (Data) :

Jari-jari : menyimpan nilai jari-jari alas kerucut

Tinggi : menyimpan nilai tinggi kerucut

Operasi (Fungsi/Prosedur)

hitungVolume();  $\frac{1}{3} \times \pi \times r^2 \times t$

hitungLuasPermukaan();  $\pi \times r \times (r + s)$

tampilkanInfo(); menampilkan nilai jari-jari, tinggi volume, dan luas permukaan kerucut.

Constructor :

Kerucut(double r, double t)

## B. Soal Latihan Modul

1. Buat program yang dapat menyimpan data mahasiswa (max. 10) ke dalam sebuah *array* dengan field nama, nim, uts, uas, tugas, dan nilai akhir. Nilai akhir diperoleh dari FUNGSI dengan rumus  $0.3 \times \text{uts} + 0.4 \times \text{uas} + 0.3 \times \text{tugas}$ .

```

1  #include <iostream>
2  #include <conio.h>
3  #include <stdlib.h>
4  using namespace std;
5
6  struct Mahasiswa {
7      string nama;
8      char nim [10];
9      float uts, uas, tugas, nilaiakhir;
10 };
11
12 float hitungNilaiAkhir(float uts, float uas, float tugas){
13     return(0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
14 }
15
16 int main(){
17     Mahasiswa mahasiswa[10];
18     int jumlahMahasiswa;
19
20     cout << "Masukkan jumlah mahasiswa (maks 10): ";
21     cin >> jumlahMahasiswa;
22
23     if (jumlahMahasiswa > 10) {
24         cout << "Jumlah mahasiswa tidak boleh lebih dari 10!" << endl;
25         return 1;
26     }
27     for (int i = 0; i < jumlahMahasiswa; i++) {
28         cout << "\nData mahasiswa ke-" << i+1 << endl;
29         cout << "Nama: ";
30         cin >> mahasiswa[i].nama;
31         cout << "NIM: ";
32         cin >> mahasiswa[i].nim;
33         cout << "Nilai UTS: ";
34         cin >> mahasiswa[i].uts;
35         cout << "Nilai UAS: ";
36         cin >> mahasiswa[i].uas;
37         cout << "Nilai Tugas: ";
38         cin >> mahasiswa[i].tugas;
39
40         mahasiswa[i].nilaiakhir = hitungNilaiAkhir(mahasiswa[i].uts,
mahasiswa[i].uas, mahasiswa[i].tugas);
41     }
42     cout << "\n\nData Mahasiswa dan Nilai Akhir:\n";
43     for (int i = 0; i < jumlahMahasiswa; i++) {
44         cout << "\nMahasiswa ke-" << i+1 << endl;
45         cout << "Nama      : " << mahasiswa[i].nama << endl;
46         cout << "NIM       : " << mahasiswa[i].nim << endl;
47         cout << "Nilai UTS : " << mahasiswa[i].uts << endl;
48         cout << "Nilai UAS : " << mahasiswa[i].uas << endl;
49         cout << "Nilai Tugas: " << mahasiswa[i].tugas << endl;
50         cout << "Nilai Akhir: " << mahasiswa[i].nilaiakhir << endl;
51     }
52     return 0;
53 }

```

```

Masukkan jumlah mahasiswa (maks 10): 1

Data mahasiswa ke-1
Nama: sopiya
NIM: 2311104003
Nilai UTS: 90
Nilai UAS: 94
Nilai Tugas: 89

Data Mahasiswa dan Nilai Akhir:

Mahasiswa ke-1
Nama      : sopiya
NIM       : 2311104003
Nilai UTS : 90
Nilai UAS : 94
Nilai Tugas: 89
Nilai Akhir: 91.3

```

2.

2. Buatlah ADT pelajaran sebagai berikut di dalam file "pelajaran.h":

```

tipe pelajaran <
    namaMapel : string
    kodeMapel : string
>
fungsi create_pelajaran( namaMapel : string, kodeMapel : string ) →
    pelajaran
prosedur tampil_pelajaran( pel : pelajaran )

```

Buatlah implementasi ADT pelajaran pada file "pelajaran.cpp"

Cobalah hasil implementasi ADT pada file "main.cpp"

```

using namespace std;
int main(){
    string namaMapel = "Struktur Data";
    string kodeMapel = "STD";
    pelajaran pel = create_pelajaran(namaMapel, kodeMapel);
    tampil_pelajaran(pel);

    return 0;
}

```

Jawab :

Pelajaran.h

```

1  #ifndef PELAJARAN_H
2  #define PELAJARAN_H
3
4  #include <string>
5  using namespace std;
6
7  struct Pelajaran {
8      string namaMapel;
9      string kodeMapel;
10 };
11
12 Pelajaran create_pelajaran(string namaMapel, string kodeMapel);
13
14 void tampil_pelajaran(Pelajaran pel);
15
16 #endif
17

```

Pelajaran.cpp

```

1 #include "pelajaran.h"
2 #include <iostream>
3
4 // Fungsi untuk membuat pelajaran baru
5 Pelajaran create_pelajaran(string namaMapel, string kodeMapel)
6 {
7     Pelajaran pel;
8     pel.namaMapel = namaMapel;
9     pel.kodeMapel = kodeMapel;
10    return pel;
11 }
12
13 // Prosedur untuk menampilkan pelajaran
14 void tampil_pelajaran(Pelajaran pel) {
15     cout << "Nama Mata Pelajaran: " << pel.namaMapel << endl;
16     cout << "Kode Mata Pelajaran: " << pel.kodeMapel << endl;
17 }

```

Main.cpp

```

1 #include <iostream>
2 #include "pelajaran.h"
3 using namespace std;
4
5 int main() {
6     // Inisialisasi pelajaran
7     string namapel = "Struktur Data";
8     string kodepel = "STD";
9
10    // Membuat pelajaran
11    Pelajaran pel = create_pelajaran(namelapel, kodepel);
12
13    // Menampilkan pelajaran
14    tampil_pelajaran(pel);
15
16    return 0;
17 }

```

```

Nama Mata Pelajaran: Struktur Data
Kode Mata Pelajaran: STD

```

3. Buatlah program dengan ketentuan :

- 2 buah *array 2D integer* berukuran 3x3 dan 2 buah *pointer integer*
- fungsi/prosedur yang menampilkan isi sebuah *array integer 2D*
- fungsi/prosedur yang akan menukarkan isi dari 2 *array integer 2D* pada posisi tertentu
- fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2

buah *pointer*

Jawab :

```
1  #include <iostream>
2  using namespace std;
3
4  void tampilkanArray(int arr[3][3]) {
5      for (int i = 0; i < 3; i++) {
6          for (int j = 0; j < 3; j++) {
7              cout << arr[i][j] << " ";
8          }
9          cout << endl;
10     }
11     cout << endl;
12 }
13
14 void tukarArray(int arr1[3][3], int arr2[3][3], int row, int col) {
15     if (row >= 0 && row < 3 && col >= 0 && col < 3) {
16         int temp = arr1[row][col];
17         arr1[row][col] = arr2[row][col];
18         arr2[row][col] = temp;
19     } else {
20         cout << "Indeks di luar batas array!" << endl;
21     }
22 }
23
24 void tukarPointer(int* ptr1, int* ptr2) {
25     int temp = *ptr1;
26     *ptr1 = *ptr2;
27     *ptr2 = temp;
28 }
29
30 int main() {
31     int array1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
32     int array2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
33
34     int a = 10, b = 20;
35     int* ptr1 = &a;
36     int* ptr2 = &b;
37
38     cout << "Array 1 sebelum pertukaran:" << endl;
39     tampilkanArray(array1);
40
41     cout << "Array 2 sebelum pertukaran:" << endl;
42     tampilkanArray(array2);
43
44     cout << "Menukar elemen di posisi (1, 1):" << endl;
45     tukarArray(array1, array2, 1, 1);
46
47     cout << "Array 1 setelah pertukaran:" << endl;
48     tampilkanArray(array1);
49
50     cout << "Array 2 setelah pertukaran:" << endl;
51     tampilkanArray(array2);
52
53     cout << "\nNilai sebelum pertukaran pointer:" << endl;
54     cout << "a = " << *ptr1 << ", b = " << *ptr2 << endl;
55
56     tukarPointer(ptr1, ptr2);
57
58     cout << "Nilai setelah pertukaran pointer:" << endl;
59     cout << "a = " << *ptr1 << ", b = " << *ptr2 << endl;
60
61     return 0;
62 }
63
```

Semoga Selalu diberi kemudahan^^