

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



**Nama :**

RIFKI TAUFIKURROHMAN (2311104033)

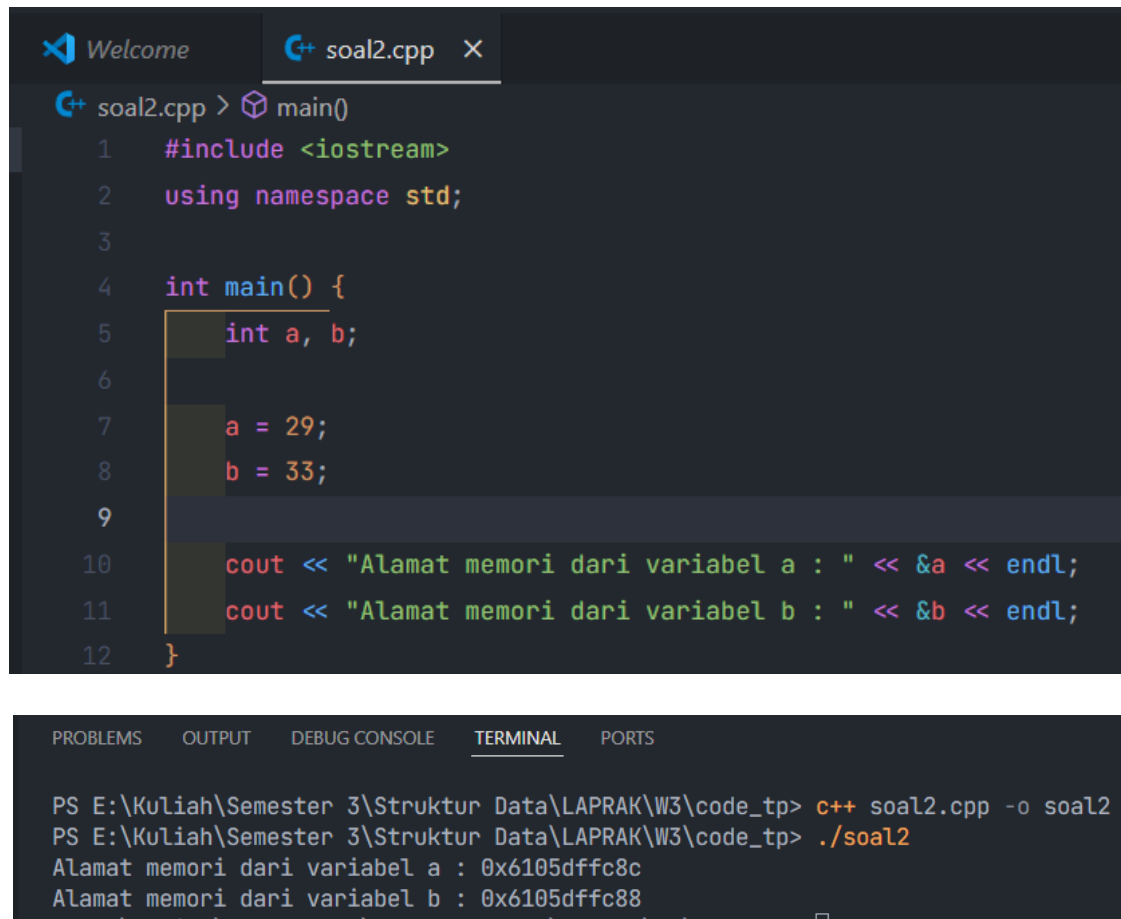
**Dosen :**

YUDHA ISLAMI SULISTYA, S.Kom., M.Kom.

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

## Soal TP

1. Pointer adalah sebuah variabel yang menyimpan alamat memori dari variabel lain, bukan nilai dari variabel itu sendiri.
2. Cara menampilkan Alamat memori dari variabel :



```

Welcome | C++ soal2.cpp x
C++ soal2.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int a, b;
6
7      a = 29;
8      b = 33;
9
10     cout << "Alamat memori dari variabel a : " << &a << endl;
11     cout << "Alamat memori dari variabel b : " << &b << endl;
12 }

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS E:\Kuliah\Semester 3\Struktur Data\LAPRAK\W3\code_tp> g++ soal2.cpp -o soal2
PS E:\Kuliah\Semester 3\Struktur Data\LAPRAK\W3\code_tp> ./soal2
Alamat memori dari variabel a : 0x6105dffc8c
Alamat memori dari variabel b : 0x6105dffc88
PS E:\Kuliah\Semester 3\Struktur Data\LAPRAK\W3\code_tp>

```

3. Cara menggunakan pointer dan menampilkan nilai yang tersimpan melalui pointer

```
soal3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int x = 22;
6      int y = 12;
7
8      int *ptr1 = nullptr;
9      ptr1 = &x;
10
11     int *ptr2 = &y;
12
13     cout << "Alamat memori x : " << &x << endl;
14     cout << "Alamat memori y : " << &y << endl;
15     cout << "=====" << endl;
16     cout << "nilai dari variabel ptr1 : " << ptr1 << endl; // Akan menyimpan alamat dari variabel X
17     cout << "nilai pointer dari variabel ptr1 : " << *ptr1 << endl; // Mengeluarkan nilai dari variabel X
18     cout << "=====" << endl;
19     cout << "nilai dari variabel ptr2 : " << ptr2 << endl; // Akan menyimpan alamat dari variabel Y
20     cout << "nilai pointer dari variabel ptr2 : " << *ptr2 << endl; // Mengeluarkan nilai dari variabel Y
21 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Kuliah\Semester 3\Struktur Data\LAPRAK\W3\code_tp> ./soal3
Alamat memori x : 0x5b5dff86c
Alamat memori y : 0x5b5dff868
=====
nilai dari variabel ptr1 : 0x5b5dff86c
nilai pointer dari variabel ptr1 : 22
=====
nilai dari variabel ptr2 : 0x5b5dff868
nilai pointer dari variabel ptr2 : 12
```

4. ADT adalah Type dan sekumpulan Primitif ( operasi dasar ) terhadap data type tersebut. Dalam sebuah ADT yang lengkap disertai juga definisi invarian dari TYPE dan aksioma yang berlaku.

5. Contoh ADT Klub Bola :

**Deskripsi :** Klub bola merupakan suatu klub atau organisasi olahraga yang bergerak di bidang sepakbola yang memiliki visi, misi dan tujuan yang sudah ditentukan oleh klub itu sendiri.

**Operasi :**

1. Mengikuti kompetisi atau liga.
2. Beli pemain.
3. Jual pemain

## 6. ADT bangun ruang kerucut

```
Welcome | C++ soal2.cpp | C++ soal3.cpp | C++ soal5.cpp X
C++ soal5.cpp > main()
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  struct Kerucut {
6      float jariJari;
7      float tinggi;
8
9      float hitungSisiMiring() {
10         return sqrt(jariJari * jariJari + tinggi * tinggi);
11     }
12
13     float luasAlas() {
14         return M_PI * jariJari * jariJari;
15     }
16
17     float luasSelimut() {
18         return M_PI * jariJari * hitungSisiMiring();
19     }
20
21     float luasPermukaan() {
22         return luasAlas() + luasSelimut();
23     }
24
25     float volume() {
26         return (1.0 / 3.0) * M_PI * jariJari * jariJari * tinggi;
27     }
28 };
```

```
PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS
PS E:\Kuliah\Semester 3\Struktur Data\LAPRAK\W3\code_tp> ./soal5
Masukkan jari-jari kerucut: 5
Masukkan tinggi kerucut: 10
Luas Alas: 78.5398
Luas Selimut: 175.62
Luas Permukaan: 254.16
Volume: 261.799
```

## Unguided

### 1. Soal 1

```
latihan1.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  struct Mahasiswa {
5      string nama;
6      string nim;
7      float uts;
8      float uas;
9      float tugas;
10     float nilaiAkhir;
11 };
12
13 float totalNilai( float uts, float uas, float tugas ) {
14     return ( 0.3 * uts ) + ( 0.4 * uas ) + ( 0.3 * tugas );
15 }
16
17 void inputDataMhs(Mahasiswa mhs[], bool& tambahData, int& jumlahData) {
18     char pilihan;
19
20     while( tambahData && jumlahData < 10 ) {
21         cout << "Masukkan nama : ";
22         cin >> mhs[jumlahData].nama;
23
24         cout << "Masukkan nim : ";
25         cin >> mhs[jumlahData].nim;
26
27         cout << "Masukkan nilai UTS : ";
28         cin >> mhs[jumlahData].uts;
29
30         cout << "Masukkan nilai UAS : ";
31         cin >> mhs[jumlahData].uas;
```

latihan1.cpp > ...

```
32
33     cout << "Masukkan tugas : ";
34     cin >> mhs[jumlahData].tugas;
35
36     mhs[jumlahData].nilaiAkhir = totalNilai(
37         mhs[jumlahData].uts, mhs[jumlahData].uas, mhs[jumlahData].tugas
38     );
39     jumlahData++;
40
41     if ( jumlahData < 10 ) {
42         cout << "Tambah data mahasiswa ? ( y / t ) : ";
43         cin >> pilihan;
44
45         if (pilihan != 'y' && pilihan != 'Y') {
46             tambahData = false;
47         }
48     } else {
49         cout << "Data mahasiswa sudah ada 10.\n";
50         tambahData = false;
51     }
52 }
53 }
```

latihan1.cpp > inputDataMhs(Mahasiswa [], bool &, int &)

```
54
55 void coutDataMhs( Mahasiswa mhs[], int jumlahData) {
56     for (int i = 0; i < jumlahData; i++) {
57         cout << "Nama : " << mhs[i].nama << endl;
58         cout << "Nim : " << mhs[i].nim << endl;
59         cout << "Nilai UTS : " << mhs[i].uts << endl;
60         cout << "Nilai UAS : " << mhs[i].uas << endl;
61         cout << "Nilai Tugas : " << mhs[i].tugas << endl;
62         cout << "Nilai akhir : " << mhs[i].nilaiAkhir << endl;
63         cout << " " << endl;
64         cout << "<=====>\n\n";
65     }
66 }
67
68 int main() {
69     int MAX_ARRAY = 10;
70     Mahasiswa mhs[MAX_ARRAY];
71     int jumlahData = 0;
72     bool tambahData = true;
73
74     inputDataMhs(mhs, tambahData, jumlahData);
75
76     cout << "\nData Mahasiswa:\n";
77     coutDataMhs(mhs, jumlahData);
78
79     return 0;
80 }
```

```

PS E:\Kuliah\Semester 3\Struktur Data\LA
Masukkan nama : rifki
Masukkan nim : 2333
Masukkan nilai UTS : 90
Masukkan nilai UAS : 95
Masukkan tugas : 99
Tambah data mahasiswa ? ( y / t ) : y
Masukkan nama : rafi
Masukkan nim : 2310
Masukkan nilai UTS : 90
Masukkan nilai UAS : 93
Masukkan tugas : 95
Tambah data mahasiswa ? ( y / t ) : t

Data Mahasiswa:
Nama : rifki
Nim : 2333
Nilai UTS : 90
Nilai UAS : 95
Nilai Tugas : 99
Nilai akhir : 94.7

<=====>

Nama : rafi
Nim : 2310
Nilai UTS : 90
Nilai UAS : 93
Nilai Tugas : 95
Nilai akhir : 92.7

<=====>

```

Penjelasan kode :

Program di atas adalah contoh sederhana untuk mengelola data mahasiswa di c++

1. Struct Mahasiswa, struct ini digunakan untuk mendefinisikan sebuah struktur data untuk menyimpan informasi terkait mahasiswa. di dalamnya terdapat nama, nim, nilai UTS dan yang lainnya.
2. Fungsi Total nilai, fungsi ini digunakan untuk menghitung nilai akhir dari mahasiswa dengan rumus 30% nilai UTS, 30% Tugas dan 40% UAS.
3. Fungsi inputDataMhs, fungsi ini untuk menginputkan data mahasiswa maksimal data yang diinputkan adalah 10.
4. Fungsi coutDataMhs, fungsi ini untuk menampilkan semua data mahasiswa yang sudah diinputkan.
5. Fungsi Main, fungsi ini merupakan program utama yang membuat array berisi maksimal 10 mahasiswa, memanggil fungsi inputDataMhs dan memanggil coutDataMhs.

## 2. Soal 2

```
latihan2.cpp > Pelajaran > kodeMapel
1  #include <iostream>
2
3  using namespace std;
4
5  struct Pelajaran {
6      string namaMapel;
7      string kodeMapel;
8  };
9
10 void createPelajaran(string& namaMapel, string& kodeMapel, Pelajaran& pel) {
11     pel.namaMapel = namaMapel;
12     pel.kodeMapel = kodeMapel;
13 }
14
15 void tampilPelajaran(Pelajaran& pel) {
16     cout << "Nama Pelajaran : " << pel.namaMapel << endl;
17     cout << "Kode Pelajaran : " << pel.kodeMapel << endl;
18 }
19
20 int main() {
21     string namaPel = "Struktur Data";
22     string kodePel = "STD";
23
24     Pelajaran pel;
25     createPelajaran(namaPel, kodePel, pel);
26
27     tampilPelajaran(pel);
28
29     return 0;
30 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS E:\Kuliah\Semester 3\Struktur Data
PS E:\Kuliah\Semester 3\Struktur Data
Nama Pelajaran : Struktur Data
Nama Pelajaran : STD
```

Penjelasan kode :

Program di atas adalah contoh menampilkan nama dan kode pelajaran di c++

1. Struct Pelajaran di gunakan untuk mendefinisikan apa saja yang dibutuhkan untuk mengisi Struct Pelajaran, di dalamnya terdapat nama dan kode Pelajaran.
2. Void createPelajaran digunakan untuk memasukkan data yang ada ke dalam struct Pelajaran.
3. Void tampilPelajaran digunakan untuk menampilkan nama dan kode Pelajaran ke terminal saat di jalankan
4. Fungsi Main digunakan untuk membuat variabel namaPel dan kodePel,



memanggil Struct Pelajaran, memanggil fungsi createPelajaran dan memanggil fungsi tampilPelajaran.

### 3. Soal 3

```
latihan3.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  void tampilArray2D(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 tukarArray2D(int arr1[3][3], int arr2[3][3], int i, int j) {
15     int temp = arr1[i][j];
16     arr1[i][j] = arr2[i][j];
17     arr2[i][j] = temp;
18 }
19
20 void tukarPointer(int* p1, int* p2) {
21     int temp = *p1;
22     *p1 = *p2;
23     *p2 = temp;
24 }
25
26 int main() {
27     int arr1[3][3] = {
28         {1, 2, 3},
29         {4, 5, 6},
30         {7, 8, 9}
31     };
```

latihan3.cpp > main()

```
33     int arr2[3][3] = {
34         {9, 8, 7},
35         {6, 5, 4},
36         {3, 2, 1}
37     };
38
39     cout << "Array 1 sebelum pertukaran: " << endl;
40     tampilArray2D(arr1);
41
42     cout << "Array 2 sebelum pertukaran: " << endl;
43     tampilArray2D(arr2);
44
45     tukarArray2D(arr1, arr2, 1, 1);
46
47     cout << "Array 1 setelah pertukaran elemen di posisi [1][1]: " << endl;
48     tampilArray2D(arr1);
49
50     cout << "Array 2 setelah pertukaran elemen di posisi [1][1]: " << endl;
51     tampilArray2D(arr2);
52
53     int a = 100;
54     int b = 200;
55     int* ptr1 = &a;
56     int* ptr2 = &b;
57
58     cout << "Sebelum pertukaran pointer: " << endl;
59     cout << "Nilai yang ditunjuk ptr1: " << *ptr1 << endl;
60     cout << "Nilai yang ditunjuk ptr2: " << *ptr2 << endl;
61
62     tukarPointer(ptr1, ptr2);
63
```

```
63
64     cout << "Setelah pertukaran pointer: " << endl;
65     cout << "Nilai yang ditunjuk ptr1: " << *ptr1 << endl;
66     cout << "Nilai yang ditunjuk ptr2: " << *ptr2 << endl;
67
68     return 0;
69 }
```

```

PS E:\Kuliah\Semester 3\Struktur Data\LAPRAK\W3\code
Array 1 sebelum pertukaran:
1 2 3
4 5 6
7 8 9

Array 2 sebelum pertukaran:
9 8 7
6 5 4
3 2 1

Array 1 setelah pertukaran elemen di posisi [1][1]:
1 2 3
4 5 6
7 8 9

Array 2 setelah pertukaran elemen di posisi [1][1]:
9 8 7
6 5 4
3 2 1

Sebelum pertukaran pointer:
Nilai yang ditunjuk ptr1: 100
Nilai yang ditunjuk ptr2: 200

Setelah pertukaran pointer:
Nilai yang ditunjuk ptr1: 200
Nilai yang ditunjuk ptr2: 100

```

Penjelasan kode :

Program di atas adalah contoh menampilkan array dengan pointer di c++

1. Array 2D, arr1 dan arr2 masing-masing berukuran 3x3. arr1 berisi angka 1 sampai 9, dan arr2 berisi angka 9 sampai 1.
2. Pointer: ptr1 menunjuk ke variabel a (bernilai 100) dan ptr2 menunjuk ke variabel b (bernilai 200).
3. Fungsi tampilArray2D: Menerima array 2D sebagai parameter dan menampilkan isi array ke konsol.
4. Fungsi tukarArray2D: Menerima dua array 2D dan menukar elemen pada posisi tertentu (i dan j). Pada contoh, posisi yang ditukar adalah [1][1].
5. Fungsi tukarPointer: Menukar nilai yang ditunjuk oleh dua pointer. Nilai yang ditunjuk oleh ptr1 dan ptr2 akan saling bertukar.