

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 3  
ABSTRACT DATA TYPE (ADT)**



**Disusun Oleh :**

NAMA

NIM :

**Dosen**

WAHYU ANDI SAPUTRA

**PROGRAM STUDI STRUKTUR DATA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## A. Dasar Teori

C++ adalah pengembangan dari bahasa c yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), berorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

## B. Guided

### 1. Praktikum

#### a. Source Code :

mahasiswa.h

```
1  #ifndef MAHASISWA_H_INCLUDED
2  #define MAHASISWA_H_INCLUDED
3  struct mahasiswa
4  {
5      char nim[10];
6      int nilai1, nilai2;
7  };
9  void inoutMhs(mahasiswa &m);
10 float rata2(mahasiswa m);
11 #endif
```

mahasiswa.cpp

```
1  #include "mahasiswa.h"
2  #include <iostream>
3  using namespace std;
4
5
6  void inputMhs(mahasiswa &m)
7  {
8      cout << "input nama = ";
9      cin >> (m).nim;
10     cout << "input nilai 1 = ";
11     cin >> (m).nilai1;
12     cout << "input nilai 2 = ";
13     cin >> (m).nilai2;
14 }
15
16 float rata2(mahasiswa m)
17 {
18     return float(m.nilai1 + m.nilai2) / 2;
19 }
```

m3.cpp

```

1  #include <iostream>
2  #include "mahasiswa.h"
3  #include "mahasiswa.cpp"
4  using namespace std;
5
6  int main()
7  {
8      mahasiswa mhs;
9      inputMhs(mhs);
10     cout << "rata-rata = " << rata2(mhs);
11     return 0;
12 }

```

b. Screenshot Output :

```

input nama = Lia
input nilai 1 = 75
input nilai 2 = 80
rata-rata = 77.5

```

c. Deskripsi Program :

Program ini berfungsi untuk menghitung rata-rata dari dua nilai mahasiswa dengan langkah kerja sebagai berikut:

1. Program mendefinisikan tipe data mahasiswa yang memiliki tiga komponen, yaitu nim bertipe char untuk menyimpan nomor induk mahasiswa, serta nilai1 dan nilai2 bertipe integer untuk menyimpan dua nilai ujian.
2. Fungsi inputMhs digunakan untuk menerima input dari pengguna. Program meminta pengguna memasukkan nim, nilai1, dan nilai2, lalu menyimpannya ke dalam variabel bertipe mahasiswa.
3. Fungsi rata2 digunakan untuk menghitung rata-rata dari dua nilai mahasiswa. Nilai rata-rata diperoleh dengan rumus  $(\text{nilai1} + \text{nilai2}) / 2$  dan hasilnya dikembalikan dalam bentuk float.
4. Dalam fungsi main, program membuat variabel mhs bertipe mahasiswa. Program kemudian memanggil fungsi inputMhs untuk mengisi data mahasiswa dan setelah itu memanggil fungsi rata2 untuk menghitung nilai rata-rata.
5. Hasil perhitungan rata-rata ditampilkan ke layar menggunakan perintah cout.
6. Setelah menampilkan hasil, program berakhir dan mengembalikan nilai 0.

## 2. Guided 1

a. Source Code :

**mahasiswa.h**

```

1  #ifndef MAHASISWA_H_INCLUDED
2  #define MAHASISWA_H_INCLUDED
3
4  #include <iostream>
5  #include <string>
6  using namespace std;
7
8  struct Mahasiswa {
9      string nama;
10     string nim;
11     float uts;
12     float uas;
13     float tugas;
14     float nilaiAkhir;
15 };
16
17 void inputMahasiswa(Mahasiswa &m);
18 float hitungNilaiAkhir(Mahasiswa m);
19 void tampilMahasiswa(Mahasiswa m);
20
21 #endif

```

#### mahasiswa.cpp

```

1  #include "mahasiswa.h"
2  #include <iostream>
3  using namespace std;
4
5  void inputMahasiswa(Mahasiswa &m) {
6      cout << "Nama   : ";
7      getline(cin, m.nama);
8      cout << "NIM    : ";
9      getline(cin, m.nim);
10     cout << "Nilai UTS   : ";
11     cin >> m.uts;
12     cout << "Nilai UAS   : ";
13     cin >> m.uas;
14     cout << "Nilai Tugas : ";
15     cin >> m.tugas;
16     cin.ignore();
17 }
18
19 float hitungNilaiAkhir(Mahasiswa m) {
20     return (0.3 * m.uts) + (0.4 * m.uas) + (0.3 * m.tugas);
21 }
22
23 void tampilMahasiswa(Mahasiswa m) {
24     cout << "-----" << endl;
25     cout << "Nama       : " << m.nama << endl;
26     cout << "NIM        : " << m.nim << endl;
27     cout << "UTS        : " << m.uts << endl;
28     cout << "UAS        : " << m.uas << endl;
29     cout << "Tugas      : " << m.tugas << endl;
30     cout << "Nilai Akhir : " << m.nilaiAkhir << endl;
31 }

```

#### main.cpp

```

1  #include <iostream>
2  #include "mahasiswa.h"
3  using namespace std;
4
5  int main() {
6      Mahasiswa data[10];
7      int n;
8
9      cout << "Masukkan jumlah mahasiswa (maks 10): ";
10     cin >> n;
11     cin.ignore();
12
13     for (int i = 0; i < n; i++) {
14         cout << "\nData Mahasiswa ke-" << i + 1 << endl;
15         inputMahasiswa(data[i]);
16         data[i].nilaiAkhir = hitungNilaiAkhir(data[i]);
17     }
18
19     cout << "\n=== DAFTAR NILAI MAHASISWA ===\n";
20     for (int i = 0; i < n; i++) {
21         tampilMahasiswa(data[i]);
22     }
23
24     return 0;
25 }

```

b. Screenshot Output :

```

Masukkan jumlah mahasiswa (maks 10): 3

Data Mahasiswa ke-1
Nama    : Lala
NIM     : 12345
Nilai UTS : 75
Nilai UAS : 40
Nilai Tugas : 98

Data Mahasiswa ke-2
Nama    : Lali
NIM     : 54321
Nilai UTS : 57
Nilai UAS : 80
Nilai Tugas : 60

Data Mahasiswa ke-3
Nama    : Lalo
NIM     : 34521
Nilai UTS : 87
Nilai UAS : 10
Nilai Tugas : 67

```

```

=== DAFTAR NILAI MAHASISWA ===
-----
Nama      : Lala
NIM       : 12345
UTS       : 75
UAS       : 40
Tugas    : 98
Nilai Akhir : 67.9
-----
Nama      : Lali
NIM       : 54321
UTS       : 57
UAS       : 80
Tugas    : 60
Nilai Akhir : 67.1
-----
Nama      : Lalo
NIM       : 34521
UTS       : 87
UAS       : 10
Tugas    : 67
Nilai Akhir : 50.2

```

c. Deskripsi Program :

Program ini berfungsi untuk menyimpan dan menampilkan data mahasiswa beserta perhitungan nilai akhirnya dengan langkah kerja sebagai berikut:

1. Program mendefinisikan tipe data Mahasiswa yang berisi nama, nim, nilai uts, uas, tugas, dan nilaiAkhir. Tipe ini digunakan untuk menyimpan data setiap mahasiswa.
2. Fungsi inputMahasiswa digunakan untuk mengisi data mahasiswa. Program meminta pengguna memasukkan nama, nim, nilai uts, uas, dan tugas. Data tersebut disimpan dalam variabel bertipe Mahasiswa.
3. Fungsi hitungNilaiAkhir digunakan untuk menghitung nilai akhir berdasarkan rumus  $0.3 \times \text{uts} + 0.4 \times \text{uas} + 0.3 \times \text{tugas}$ . Nilai hasil perhitungan dikembalikan ke pemanggil fungsi.
4. Fungsi tampilMahasiswa digunakan untuk menampilkan semua data mahasiswa, yaitu nama, nim, nilai uts, uas, tugas, dan nilaiAkhir.
5. Dalam fungsi main, program membuat array data berukuran maksimal 10 elemen untuk menyimpan beberapa mahasiswa. Pengguna diminta memasukkan jumlah mahasiswa yang akan diinput.
6. Program kemudian memanggil fungsi inputMahasiswa untuk setiap mahasiswa dan menghitung nilai akhir dengan memanggil fungsi hitungNilaiAkhir. Hasilnya disimpan ke atribut nilaiAkhir masing-masing mahasiswa.
7. Setelah semua data diinput, program menampilkan daftar nilai mahasiswa dengan memanggil fungsi tampilMahasiswa untuk setiap elemen array.

8. Program selesai dijalankan dan mengembalikan nilai 0.

### 3. Guided 2

a. Source Code :

#### pelajaran.h

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

#### pelajaran.cpp

```
1  #include "pelajaran.h"
2  #include <iostream>
3  using namespace std;
4
5  pelajaran create_pelajaran(string namaPel, string kodePel) {
6      pelajaran p;
7      p.namaMapel = namaPel;
8      p.kodeMapel = kodePel;
9      return p;
10 }
11
12 void tampil_pelajaran(pelajaran pel) {
13     cout << "Nama Mata Kuliah : " << pel.namaMapel << endl;
14     cout << "Kode Mata Kuliah : " << pel.kodeMapel << endl;
15 }
```

#### main.cpp

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

b. Screenshot Output :

```
PS C:\Users\ASUS\Downloads\SEMESTER3STRUKDAT\MODUL3SOAL2> g++ main.cpp pelajaran.cpp -o program
PS C:\Users\ASUS\Downloads\SEMESTER3STRUKDAT\MODUL3SOAL2> ./program
>>
Nama Mata Kuliah : Struktur Data
Kode Mata Kuliah : STD
PS C:\Users\ASUS\Downloads\SEMESTER3STRUKDAT\MODUL3SOAL2> 
```

c. Deskripsi Program :

Program ini berfungsi untuk membuat dan menampilkan data pelajaran dengan langkah kerja sebagai berikut:

1. Program mendefinisikan tipe data pelajaran yang memiliki dua atribut, yaitu namaMapel untuk menyimpan nama mata kuliah dan kodeMapel untuk menyimpan kode mata kuliah.
2. Fungsi `create_pelajaran` digunakan sebagai konstruktor untuk membuat data pelajaran baru. Fungsi ini menerima dua parameter berupa namaPel dan kodePel, lalu mengembalikan variabel bertipe pelajaran yang berisi nilai dari kedua parameter tersebut.
3. Fungsi `tampil_pelajaran` digunakan untuk menampilkan informasi pelajaran dengan mencetak nama mata kuliah dan kode mata kuliah ke layar.
4. Pada fungsi main, program membuat dua variabel string yaitu `namapel` dan `kodepel` dengan nilai awal "Struktur Data" dan "STD".
5. Program kemudian memanggil fungsi `create_pelajaran` dengan mengirimkan `namapel` dan `kodepel`, lalu menyimpan hasilnya ke variabel `pel` bertipe pelajaran.
6. Selanjutnya, program memanggil fungsi `tampil_pelajaran` untuk menampilkan nama dan kode mata kuliah yang telah dibuat.
7. Setelah proses penampilan data selesai, program berakhir dan mengembalikan nilai 0.

#### 4. Guided 3

##### a. Source Code :

```

1  #include <iostream>
2  using namespace std;
3
4  void tampilArray(int A[3][3]) {
5      for (int i = 0; i < 3; i++) {
6          for (int j = 0; j < 3; j++) {
7              cout << A[i][j] << " ";
8          }
9          cout << endl;
10     }
11 }
12
13 void tukarArray(int A[3][3], int B[3][3], int baris, int kolom) {
14     int temp = A[baris][kolom];
15     A[baris][kolom] = B[baris][kolom];
16     B[baris][kolom] = temp;
17 }
18
19 void tukarPointer(int *p1, int *p2) {
20     int temp = *p1;
21     *p1 = *p2;
22     *p2 = temp;
23 }
24
25 int main() {
26     int A[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
27     int B[3][3] = {{9,8,7},{6,5,4},{3,2,1}};
28     int *ptr1, *ptr2;
29     int x = 10, y = 20;
30
31     cout << "Array A awal:\n";
32     tampilArray(A);
33     cout << "Array B awal:\n";
34     tampilArray(B);
35
36     tukarArray(A, B, 1, 2);

```



```

37
38     cout << "\nSetelah elemen [1][2] ditukar:\n";
39     cout << "Array A:\n";
40     tampilArray(A);
41     cout << "Array B:\n";
42     tampilArray(B);
43
44     ptr1 = &x;
45     ptr2 = &y;
46
47     cout << "\nSebelum tukar pointer:\n";
48     cout << "x = " << x << ", y = " << y << endl;
49
50     tukarPointer(ptr1, ptr2);
51
52     cout << "Setelah tukar pointer:\n";
53     cout << "x = " << x << ", y = " << y << endl;
54
55     return 0;
56

```

b. Screenshot Output :

```

Array A awal:
1 2 3
4 5 6
7 8 9
Array B awal:
9 8 7
6 5 4
3 2 1

Setelah elemen [1][2] ditukar:
Array A:
1 2 3
4 5 4
7 8 9
Array B:
9 8 7
6 5 6
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
7 8 9
Array B:
9 8 7
6 5 6
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
6 5 6
3 2 1

Sebelum tukar pointer:
x = 10, y = 20
Sebelum tukar pointer:
x = 10, y = 20
x = 10, y = 20
Setelah tukar pointer:
Setelah tukar pointer:
x = 20, y = 10

```

c. Deskripsi Program :

Program ini berfungsi untuk menampilkan isi dua array 2 dimensi dan melakukan pertukaran nilai baik pada elemen array maupun variabel yang ditunjuk oleh pointer dengan langkah kerja sebagai berikut:

1. Program mendeklarasikan dua buah array 2 dimensi A dan B berukuran 3x3 dengan nilai awal berbeda, serta dua variabel integer x dan y yang akan digunakan untuk contoh pointer.

2. Fungsi `tampilArray` digunakan untuk menampilkan isi dari array 2 dimensi. Fungsi ini menggunakan dua perulangan `for` untuk mencetak setiap elemen array dalam bentuk matriks 3x3.
3. Fungsi `tukarArray` digunakan untuk menukar isi elemen antara dua array A dan B pada posisi tertentu. Fungsi menerima parameter baris dan kolom untuk menentukan posisi elemen yang akan ditukar. Nilai pada posisi tersebut ditukar menggunakan variabel sementara `temp`.
4. Fungsi `tukarPointer` digunakan untuk menukar nilai dua variabel yang ditunjuk oleh pointer. Pertukaran dilakukan dengan cara menyimpan nilai yang ditunjuk pointer pertama ke variabel sementara, lalu mengganti nilainya dengan nilai yang ditunjuk pointer kedua.
5. Dalam fungsi `main`, program menampilkan isi awal dari array A dan B dengan memanggil fungsi `tampilArray`.
6. Program memanggil fungsi `tukarArray` untuk menukar elemen array A dan B pada posisi `[1][2]`, kemudian menampilkan kembali isi kedua array setelah pertukaran.
7. Program menginisialisasi dua pointer `ptr1` dan `ptr2` untuk menunjuk ke variabel `x` dan `y`, lalu menampilkan nilai awal keduanya.
8. Program memanggil fungsi `tukarPointer` untuk menukar nilai yang ditunjuk oleh `ptr1` dan `ptr2`, kemudian menampilkan hasil pertukaran nilai `x` dan `y`.
9. Setelah seluruh proses selesai, program berakhir dan mengembalikan nilai 0.

### C. Kesimpulan

Dalam praktikum Modul 3 ini, dilakukan pembelajaran mengenai konsep Abstract Data Type (ADT) dan penerapannya dalam bahasa pemrograman C++. Melalui berbagai latihan, peserta memahami cara membuat tipe data baru menggunakan `struct` untuk menyimpan informasi seperti data mahasiswa dan mata pelajaran. Program juga dipisahkan menjadi beberapa file, yaitu header, implementasi, dan main, agar struktur kode menjadi lebih teratur dan mudah dibaca.

Dalam Modul 3 juga mempraktikkan pembuatan fungsi untuk menghitung nilai rata-rata, menentukan nilai akhir mahasiswa dengan rumus tertentu, serta menampilkan hasilnya ke layar. Selain itu, peserta belajar menggunakan array untuk menyimpan beberapa data mahasiswa sekaligus dan memahami penggunaan pointer untuk menukar nilai antara dua variabel.

Melalui latihan-latihan tersebut, dari modul ini dapat dipahami pentingnya konsep modularitas dan abstraksi dalam pemrograman. Penerapan ADT, fungsi, array, dan pointer membantu membangun pemahaman tentang bagaimana program dapat disusun secara efisien dan terstruktur, sehingga menjadi dasar yang kuat untuk pengembangan program yang lebih kompleks pada tahap berikutnya.

#### D. Referensi

- Modul Praktikum Struktur Data – Modul 3: *Abstract Data Type (ADT)*, Program Studi Teknik Informatika.
- Deitel, H. M., & Deitel, P. J. (2017). *C++ How to Program* (10th Edition). Pearson Education.
- Malik, D. S. (2018). *Data Structures Using C++*. Cengage Learning.