

## **I. COVER**

### **LAPORAN PRAKTIKUM**

#### **Modul 3**

#### **Implementasi Struktur Data Mahasiswa dengan C++**



**Disusun Oleh:**

**Haza Zaidan Zidna Fann**

**(2311104056)**

**Dosen :**

**Wahyu Andi Saputra**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY**

**PURWOKERTO**

**2024**

## **II. TUJUAN**

Menjelaskan bagaimana mendefinisikan sebuah struktur (struct) untuk merepresentasikan data yang terdiri dari beberapa anggota (dalam kasus ini, NIM, nilai1, dan nilai2)

Melihat bagaimana parameter mahasiswa &m dalam fungsi inputMhs memungkinkan perubahan pada objek asli (mhs) di dalam fungsi.

Menggunakan cin dan cout untuk berinteraksi dengan pengguna dan menampilkan hasil perhitungan.

### **III. Landasan Teori**

Struktur Data:

Struct: Struktur mahasiswa menyimpan beberapa data terkait mahasiswa, yaitu NIM (array karakter) dan dua nilai (nilai1 dan nilai2).

Fungsi:

inputMhs: Fungsi ini digunakan untuk menginput data mahasiswa dengan parameter referensi, sehingga perubahan pada parameter mempengaruhi variabel asli.

rata2: Fungsi ini menghitung rata-rata dari dua nilai mahasiswa dan mengembalikan hasil dalam tipe float.

Konsep Perhitungan:

Rata-Rata: Perhitungan rata-rata dilakukan dengan menjumlahkan dua nilai lalu membaginya dengan dua.

Input/Output:

Input: Pengguna memasukkan NIM dan dua nilai mahasiswa.

Output: Hasil rata-rata dari dua nilai mahasiswa ditampilkan ke layar.

Referensi:

Pass by Reference: Fungsi inputMhs menggunakan referensi untuk langsung mengubah nilai variabel yang diberikan tanpa membuat salinan.

### **IV. GUIDED**



```
1  #include<iostream>
2
3
4  using namespace std;
5
6  struct mahasiswa{
7      char nim[10];
8      int nilai1,nilai2;
9  };
10
11 void inputMhs(mahasiswa &m);
12 float rata2(mahasiswa m);
13
14 int main(){
15     mahasiswa mhs;
16     inputMhs(mhs);
17     cout << "rata -rata = "<<rata2(mhs);
18     return 0;
19
20
21 }
22 void inputMhs(mahasiswa &m){
23     cout << "input nim= ";
24     cin >> (m).nim;
25     cout << "input nilai= ";
26     cin >> (m).nilai1;
27     cout << "input nilai= ";
28     cin >> (m).nilai2;
29 }
30
31 float rata2(mahasiswa m){
32     return(m.nilai1 + m.nilai2) /2;
33 }
```

### 1. Struktur mahasiswa:

struct mahasiswa: Mendefinisikan sebuah struktur yang menyimpan informasi mahasiswa. Dalam hal ini, struktur tersebut berisi NIM mahasiswa dalam bentuk array karakter (maksimal 10 karakter) dan dua nilai ujian/tugas (nilai1 dan nilai2) dalam bentuk bilangan bulat (integer).

### 2. Fungsi inputMhs:

Fungsi ini menerima parameter berupa referensi ke variabel mahasiswa (ditandai dengan tanda &), sehingga perubahan pada parameter m akan langsung mempengaruhi variabel yang dikirim dari luar fungsi.

Fungsi ini meminta input dari pengguna untuk memasukkan **NIM, nilai pertama, dan nilai kedua**.

cin >> m.nim: Membaca input NIM dari pengguna dan menyimpannya di variabel m.nim.

cin >> m.nilai1 dan cin >> m.nilai2: Membaca dua nilai ujian atau tugas dari pengguna dan menyimpannya di variabel m.nilai1 dan m.nilai2.

### 3. Fungsi rata2:

Fungsi ini menerima parameter berupa objek mahasiswa, kemudian menghitung rata-rata dari dua nilai (nilai1 dan nilai2) dan mengembalikannya sebagai nilai float.

(m.nilai1 + m.nilai2) / 2: Menjumlahkan nilai1 dan nilai2, kemudian membaginya dengan dua untuk mendapatkan rata-rata.

### 4. Fungsi main :

Variabel mhs dideklarasikan sebagai objek dari struktur mahasiswa.

Fungsi inputMhs(mhs) dipanggil untuk mengisi data mhs (NIM, nilai1, nilai2) yang dimasukkan oleh pengguna.

Setelah data diisi, fungsi rata2(mhs) dipanggil untuk menghitung rata-rata dua nilai yang dimasukkan, dan hasilnya dicetak menggunakan cout.

Deklarasi Struktur: Struktur mahasiswa digunakan untuk mengelompokkan data mahasiswa yang terdiri dari NIM, nilai1, dan nilai2.

Input Data: Fungsi inputMhs meminta pengguna untuk memasukkan NIM dan dua nilai mahasiswa, yang kemudian disimpan dalam variabel mhs.

Penghitungan Rata-Rata: Fungsi rata2 menghitung rata-rata dua nilai yang telah dimasukkan oleh pengguna.

Output Rata-Rata: Hasil rata-rata dua nilai dicetak ke layar menggunakan cout.

## V. UNGUIDED

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  const int MAX_MAHASISWA = 10;
7
8  struct Mahasiswa {
9      string nama;
10     string nim;
11     int uts, uas, tugas;
12     double nilaiAkhir;
13 };
14
15 // Fungsi untuk menghitung nilai akhir
16 double hitungNilaiAkhir(int uts, int uas, int tugas) {
17     return 0.3 * uts + 0.4 * uas + 0.3 * tugas;
18 }
19
20 // Fungsi untuk input data mahasiswa
21 void inputDataMahasiswa(Mahasiswa mhs[]) {
22     for (int i = 0; i < MAX_MAHASISWA; i++) {
23         cout << "Mahasiswa ke-" << i+1 << endl;
24         cout << "Nama: ";
25         cin >> mhs[i].nama;
26         cout << "NIM: ";
27         cin >> mhs[i].nim;
28         cout << "Nilai UTS: ";
29         cin >> mhs[i].uts;
30         cout << "Nilai UAS: ";
31         cin >> mhs[i].uas;
32         cout << "Nilai Tugas: ";
33         cin >> mhs[i].tugas;
34         mhs[i].nilaiAkhir = hitungNilaiAkhir(mhs[i].uts, mhs[i].uas, mhs[i].tugas);
35     }
36 }
37
38 // Fungsi untuk menampilkan data mahasiswa
39 void tampilkanDataMahasiswa(Mahasiswa mhs[]) {
40     cout << "\nData Mahasiswa:\n";
41     cout << "-----\n";
42     cout << "No\tNama\tNIM\tUTS\tUAS\tTugas\tNilai Akhir\n";
43     cout << "-----\n";
44     for (int i = 0; i < MAX_MAHASISWA; i++) {
45         cout << i+1 << "\t" << mhs[i].nama << "\t" << mhs[i].nim << "\t"
46             << mhs[i].uts << "\t" << mhs[i].uas << "\t" << mhs[i].tugas << "\t"
47             << mhs[i].nilaiAkhir << endl;
48     }
49 }
50
51 int main() {
52     Mahasiswa mahasiswa[MAX_MAHASISWA];
53
54     inputDataMahasiswa(mahasiswa);
55     tampilkanDataMahasiswa(mahasiswa);
56
57     return 0;
58 }

```

```

Mahasiswa ke-1
Nama: Haza
NIM: 12345
Nilai UTS: 100
Nilai UAS: 100
Nilai Tugas: 100
Mahasiswa ke-2
Nama: Zidan
NIM: 74856
Nilai UTS: 80
Nilai UAS: 80
Nilai Tugas: 80
Mahasiswa ke-3
Nama: Fanzzy
NIM: 44359
Nilai UTS: 60
Nilai UAS: 60
Nilai Tugas: 60
Mahasiswa ke-4
Nama: Zidna
NIM: 55676
Nilai UTS: 90
Nilai UAS: 90
Nilai Tugas: 90
Mahasiswa ke-5
Nama: Asep
NIM: 78765
Nilai UTS: 50
Nilai UAS: 50
Nilai Tugas: 50
Mahasiswa ke-6
Nama: Ujang
NIM: 33245
Nilai UTS: 95
Nilai UAS: 95

```

```

NIM: 56447
Nilai UTS: 55
Nilai UAS: 55
Nilai Tugas: 55
Mahasiswa ke-9
Nama: Nabil
NIM: 11234
Nilai UTS: 65
Nilai UAS: 65
Nilai Tugas: 65
Mahasiswa ke-10
Nama: Rizki
NIM: 99876
Nilai UTS: 40
Nilai UAS: 40
Nilai Tugas: 40

```

Data Mahasiswa:

No	Nama	NIM	UTS	UAS	Tugas	Nilai Akhir
1	Haza	12345	100	100	100	100
2	Zidan	74856	80	80	80	80
3	Fanzzy	44359	60	60	60	60
4	Zidna	55676	90	90	90	90
5	Asep	78765	50	50	50	50
6	Ujang	33245	95	95	95	95
7	Kipli	98743	75	75	75	75
8	Kurniadi	56447	55	55	55	55
9	Nabil	11234	65	65	65	65
10	Rizki	99876	40	40	40	40

```

Process returned 0 (0x0)   execution time : 469.172 s
Press any key to continue.

```

## 1. Struktur Mahasiswa:

Struktur ini digunakan untuk menyimpan informasi tentang mahasiswa. Setiap mahasiswa memiliki beberapa atribut:

Nama: Nama mahasiswa.

NIM: Nomor Induk Mahasiswa.

UTS: Nilai Ujian Tengah Semester.

UAS: Nilai Ujian Akhir Semester.

Tugas: Nilai Tugas.

Nilai Akhir: Dihitung dari kombinasi nilai UTS, UAS, dan Tugas.

## 2. Konstanta MAX\_MAHASISWA:

Digunakan untuk menentukan jumlah maksimum mahasiswa yang dapat diinput dalam program. Dalam hal ini, batasnya adalah 10 mahasiswa.

## 3. Fungsi hitungNilaiAkhir:

Fungsi ini bertugas untuk menghitung nilai akhir mahasiswa. Nilai akhir dihitung dengan rumus:

30% dari nilai UTS.

40% dari nilai UAS.

30% dari nilai Tugas.

## 4. Fungsi inputDataMahasiswa:

Fungsi ini menerima input data dari pengguna. Pengguna akan diminta untuk memasukkan:

Nama mahasiswa.

NIM mahasiswa.

Nilai UTS, UAS, dan Tugas mahasiswa.

Setelah semua data dimasukkan, fungsi ini juga menghitung nilai akhir untuk setiap mahasiswa dengan memanggil fungsi hitungNilaiAkhir dan menyimpannya ke dalam atribut nilaiAkhir.

## 5. Fungsi tampilkanDataMahasiswa:

Fungsi ini digunakan untuk menampilkan data mahasiswa yang telah dimasukkan. Untuk setiap mahasiswa, ditampilkan:

Nomor urut mahasiswa.

Nama mahasiswa.

NIM mahasiswa.

Nilai UTS, UAS, dan Tugas.

Nilai akhir yang telah dihitung.

## 6. Fungsi main:

Dalam fungsi utama, pertama-tama program membuat array yang menyimpan data maksimum 10 mahasiswa. Fungsi inputDataMahasiswa kemudian dipanggil untuk memasukkan data dari pengguna, diikuti dengan fungsi tampilkanDataMahasiswa untuk menampilkan hasil input serta nilai akhir mahasiswa.

## Alur Program:

1. Program memulai dengan mendefinisikan array untuk menyimpan data mahasiswa.
2. Pengguna diminta memasukkan data untuk setiap mahasiswa, yaitu nama, NIM, nilai UTS, UAS, dan Tugas.
3. Setelah data dimasukkan, program menghitung nilai akhir berdasarkan rumus yang diberikan.
4. Program kemudian menampilkan data lengkap dari setiap mahasiswa, termasuk nilai UTS, UAS, Tugas, dan Nilai Akhir.





```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct pelajaran {
7      string namaMapel;
8      string nilaiMapel;
9  };
10
11 // Fungsi untuk membuat pelajaran baru
12 pelajaran create_pelajaran(string nama, string nilai) {
13     pelajaran p;
14     p.namaMapel = nama;
15     p.nilaiMapel = nilai;
16     return p;
17 }
18
19 // Fungsi untuk menampilkan informasi pelajaran
20 void tampil_pelajaran(pelajaran p) {
21     cout << "Nama Mata Pelajaran: " << p.namaMapel << endl;
22     cout << "Nilai Mata Pelajaran: " << p.nilaiMapel << endl;
23 }
24
25 int main() {
26     // Membuat objek pelajaran
27     pelajaran p1 = create_pelajaran("Struktur Data", "100");
28
29     // Menampilkan informasi pelajaran
30     tampil_pelajaran(p1);
31
32     return 0;
33 }
34
35
```

```
Nama Mata Pelajaran: Struktur Data
Nilai Mata Pelajaran: 100
```

### **1. Struktur pelajaran:**

Struktur ini digunakan untuk menyimpan informasi tentang mata pelajaran. Setiap mata pelajaran memiliki dua atribut:

**namaMapel:** Nama mata pelajaran.

**nilaiMapel:** Nilai mata pelajaran.

### **2. Fungsi create\_pelajaran:**

Fungsi ini bertanggung jawab untuk membuat objek pelajaran baru. Pengguna memberikan nama mata pelajaran dan nilai sebagai input, dan fungsi ini mengembalikan objek pelajaran yang baru dibuat dengan data tersebut. Fungsi ini menyimpan nama dan nilai mata pelajaran dalam struktur pelajaran.

### **3. Fungsi tampil\_pelajaran:**

Fungsi ini bertugas untuk menampilkan informasi tentang mata pelajaran yang telah dibuat. Informasi yang ditampilkan meliputi:

Nama mata pelajaran.


Nilai mata pelajaran. Data ini ditampilkan menggunakan cout untuk mencetak hasilnya ke layar.

### **4. Fungsi main:**

Dalam fungsi utama, pertama kali program membuat objek pelajaran baru dengan memanggil fungsi create\_pelajaran, yang memberikan nama mata pelajaran dan nilai sebagai input. Setelah objek pelajaran dibuat, program kemudian memanggil fungsi tampil\_pelajaran untuk menampilkan nama dan nilai dari mata pelajaran tersebut.

### **Alur Program:**

1. Program mulai dengan mendefinisikan struktur pelajaran, yang menyimpan data berupa nama dan nilai mata pelajaran.
2. Kemudian, fungsi create\_pelajaran digunakan untuk membuat objek pelajaran berdasarkan input nama dan nilai mata pelajaran.
3. Setelah objek dibuat, fungsi tampil\_pelajaran digunakan untuk menampilkan informasi mata pelajaran.
4. Program mencetak nama dan nilai mata pelajaran yang diberikan.



```

1  #include <iostream>
2
3  using namespace std;
4
5  // Fungsi untuk menampilkan isi array 2D
6  void tampilkanArray(int arr[3][3]) {
7      for (int i = 0; i < 3; i++) {
8          for (int j = 0; j < 3; j++) {
9              cout << arr[i][j] << " ";
10         }
11         cout << endl;
12     }
13 }
14
15 // Fungsi untuk menukar isi dua array 2D pada posisi tertentu
16 void tukarElemen(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
17     int temp = arr1[baris][kolom];
18     arr1[baris][kolom] = arr2[baris][kolom];
19     arr2[baris][kolom] = temp;
20 }
21
22 // Fungsi untuk menukar isi dua variabel yang ditunjuk oleh pointer
23 void tukarPointer(int *ptr1, int *ptr2) {
24     int temp = *ptr1;
25     *ptr1 = *ptr2;
26     *ptr2 = temp;
27 }
28
29 int main() {
30     int arr1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
31     int arr2[3][3] = {{10, 11, 12}, {13, 14, 15}, {16, 17, 18}};
32
33     cout << "Array 1 sebelum ditukar:\n";
34     tampilkanArray(arr1);
35
36     cout << "\nArray 2 sebelum ditukar:\n";
37     tampilkanArray(arr2);
38
39     // Tukar elemen pada posisi baris 1, kolom 1
40     tukarElemen(arr1, arr2, 1, 1);
41
42     cout << "\nArray 1 setelah ditukar:\n";
43     tampilkanArray(arr1);
44
45     cout << "\nArray 2 setelah ditukar:\n";
46     tampilkanArray(arr2);
47
48     // Menggunakan pointer
49     int *ptr1 = &arr1[0][0];
50     int *ptr2 = &arr2[0][0];
51
52     tukarPointer(ptr1, ptr2);
53
54     cout << "\nSetelah menukar menggunakan pointer:\n";
55     cout << "Elemen pertama dari arr1: " << *ptr1 << endl;
56     cout << "Elemen pertama dari arr2: " << *ptr2 << endl;
57
58     return 0;
59 }

```

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

Array 2 sebelum ditukar:
10 11 12
13 14 15
16 17 18

Array 1 setelah ditukar:
1 2 3
4 14 6
7 8 9

Array 2 setelah ditukar:
10 11 12
13 5 15
16 17 18

Setelah menukar menggunakan pointer:
Elemen pertama dari arr1: 10
Elemen pertama dari arr2: 1
```

### 1. Fungsi tampilkanArray:

Fungsi ini bertugas untuk menampilkan isi dari array 2D yang berukuran 3x3. Fungsi ini menggunakan dua loop bersarang (nested loop), satu untuk baris dan satu lagi untuk kolom. Setiap elemen pada array diakses berdasarkan indeks baris dan kolom, kemudian dicetak ke layar menggunakan cout. Setelah setiap baris selesai dicetak, program menambahkan endl untuk memulai baris baru.

### 2. Fungsi tukarElemen:

Fungsi ini digunakan untuk menukar elemen tertentu antara dua array 2D pada posisi yang ditentukan oleh parameter baris dan kolom. Prosesnya adalah:

Elemen dari array pertama (arr1) pada posisi tertentu disimpan sementara di variabel temp.

Elemen dari array kedua (arr2) pada posisi yang sama kemudian disalin ke array pertama (arr1).

Nilai yang tersimpan di temp kemudian dipindahkan ke array kedua (arr2).

Ini menghasilkan penukaran elemen pada posisi yang sama di kedua array.

### 3. Fungsi tukarPointer:

Fungsi ini melakukan pertukaran antara dua elemen yang ditunjuk oleh pointer. Pointer yang diterima sebagai argumen menunjuk ke elemen pertama dari array. Pertukaran dilakukan dengan cara:

Nilai yang ditunjuk oleh ptr1 disimpan sementara di variabel temp.

Nilai yang ditunjuk oleh ptr2 kemudian dipindahkan ke ptr1.

Nilai yang tersimpan di temp kemudian dipindahkan ke ptr2.

Dengan cara ini, nilai yang ditunjuk oleh dua pointer saling bertukar.

#### **4. Fungsi main:**

Fungsi utama berfungsi sebagai titik awal program. Prosesnya adalah:

Dua array 2D, arr1 dan arr2, dideklarasikan dan diinisialisasi dengan nilai-nilai berbeda.

Fungsi tampilkanArray dipanggil untuk menampilkan isi dari kedua array sebelum penukaran dilakukan.

Fungsi tukarElemen dipanggil untuk menukar elemen pada posisi tertentu, dalam hal ini posisi baris 1 kolom 1 (elemen tengah array).

Setelah penukaran, isi dari kedua array ditampilkan kembali untuk menunjukkan perubahan.

Kemudian, pointer ptr1 dan ptr2 digunakan untuk menunjuk ke elemen pertama dari masing-masing array. Fungsi tukarPointer dipanggil untuk menukar nilai yang ditunjuk oleh kedua pointer.

Hasil penukaran dengan pointer ditampilkan dengan mencetak elemen pertama dari kedua array setelah penukaran.

#### **Alur Program:**

1. Program dimulai dengan inisialisasi dua array 2D, arr1 dan arr2, masing-masing berukuran 3x3.
2. Program menampilkan isi dari kedua array sebelum dilakukan penukaran.
3. Elemen array pada baris 1 kolom 1 (elemen tengah) dari kedua array ditukar menggunakan fungsi tukarElemen.
4. Setelah itu, isi dari kedua array ditampilkan kembali untuk melihat hasil penukaran.
5. Program juga melakukan penukaran menggunakan pointer dengan menunjuk ke elemen pertama dari masing-masing array, lalu menampilkan hasil penukarannya.

## **VI. Kesimpulan**

Penggunaan struktur data, array, dan pointer dalam C++ memungkinkan pengelolaan dan manipulasi data secara efisien. Melalui implementasi struktur mahasiswa, data seperti NIM dan nilai dapat disimpan dengan terorganisir, dan pass by reference memungkinkan perubahan langsung tanpa salinan, meningkatkan efisiensi. Selain itu, manipulasi array 2D menggunakan indeks dan pointer memperlihatkan fleksibilitas

dalam mengubah data dua dimensi, serta memfasilitasi akses langsung ke lokasi memori.