

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

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



Disusun Oleh :

NAMA : Loh Suwargi Nitis Hamengku Bintang
NIM : 103112400116

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Struct dalam bahasa C++ merupakan salah satu bentuk struktur data yang digunakan untuk membuat tipe data baru dengan cara menggabungkan sejumlah variabel yang memiliki tipe data berbeda. Melalui struct, seorang programmer dapat membentuk tipe data yang lebih kompleks dan sesuai dengan kebutuhan program yang dikembangkan, di luar tipe data bawaan yang telah disediakan oleh C++. Berbeda dengan *array* yang hanya mampu menyimpan elemen dengan tipe data sejenis, struct memungkinkan pengelompokan beberapa variabel dengan tipe data berbeda dalam satu kesatuan. Sebagai contoh, data mahasiswa yang terdiri dari nama, NIM, program studi, dan universitas dapat disatukan dalam satu struct, meskipun masing-masing data tersebut memiliki tipe data yang berbeda.

Selain struct, konsep penting lain dalam C++ adalah *pointer*. Pointer merupakan variabel yang menyimpan alamat memori dari variabel lain. Dalam memori komputer, satuan terkecil yang digunakan disebut bit, dan delapan bit membentuk satu byte. Memori utama (RAM) terdiri atas sekumpulan byte yang diberi nomor secara berurutan, di mana setiap nomor menunjukkan alamat tertentu. Ketika sebuah variabel dideklarasikan, sistem akan menentukan tipe serta nama variabel tersebut, lalu sistem operasi akan mengalokasikan sejumlah byte pada lokasi memori tertentu untuk menyimpan nilainya. Besarnya alokasi memori yang diberikan bergantung pada tipe data variabel tersebut; misalnya, variabel bertipe integer umumnya memerlukan empat byte ruang memori.

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided 1

```
#ifndef MAHASISWA_H_INCLUDED
#define MAHASISWA_H_INCLUDED

struct mahasiswa
{
    char nim[10];
    int nilai1, nilai2;
};

void inputMhs(mahasiswa &m);

float rata2(mahasiswa m);

#endif
```

```
#include "mahasiswa.h"

#include <iostream>
```

```
using namespace std;

void inputMhs(mahasiswa &m)
{
    cout << "input nama = ";
    cin >> (m).nim;
    cout << "input nilai1 = ";
    cin >> (m).nilai1;
    cout << "input nilai2 = ";
    cin >> (m).nilai2;
}

float rata2(mahasiswa m)
{
    return float (m.nilai1 + m.nilai2) / 2;
}
```

```
#include <iostream>
#include "mahasiswa.h"
using namespace std;
int main()
{
    mahasiswa mhs;
    inputMhs(mhs);
    cout << "rata - rata = " << rata2(mhs);
    return 0;
}
```

Screenshots Output

```

PS C:\semester3\struktur data\modul3\guided1> g++ main.cpp mahasiswa.cpp -o mahasiswa.exe
PS C:\semester3\struktur data\modul3\guided1> .\mahasiswa.exe
input nama = bintang
input nilai1 = 89
input nilai2 = 88
rata - rata = 88.5
PS C:\semester3\struktur data\modul3\guided1>

```

Deskripsi: File mahasiswa.h berisi deklarasi struktur dan fungsi yang digunakan dalam program. Struktur bernama mahasiswa memiliki tiga data anggota, yaitu nim untuk menyimpan nomor induk mahasiswa serta nilai1 dan nilai2 untuk menyimpan dua nilai ujian atau tugas. Di dalam file ini juga terdapat deklarasi dua fungsi, yaitu inputMhs() untuk menginput data mahasiswa dan rata2() untuk menghitung rata-rata dari dua nilai. File mahasiswa.cpp berisi implementasi dari fungsi-fungsi tersebut. Fungsi inputMhs() meminta pengguna untuk memasukkan NIM, nilai pertama, dan nilai kedua melalui input konsol, sedangkan fungsi rata2() menghitung rata-rata dari dua nilai dan mengembalikannya dalam bentuk bilangan pecahan (float). File main.cpp merupakan program utama yang memanggil kedua fungsi tersebut. Di dalamnya, program membuat variabel bertipe mahasiswa, kemudian memanggil fungsi inputMhs() untuk menerima input dari pengguna dan fungsi rata2() untuk menghitung serta menampilkan hasil rata-rata ke layar. Program ini menerapkan pemisahan kode dengan konsep modularisasi dalam bahasa C++, di mana struktur data dan fungsi dipisahkan ke dalam file header dan implementasi terpisah. Program ini membantu memahami cara kerja fungsi, struktur (struct), dan penggunaan file header (.h) serta file implementasi (.cpp) dalam proyek C++.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

```

#include <iostream>
#include <string>
using namespace std;

struct Mahasiswa {
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAkhir;
};

```

```
float hitungNilaiAkhir(float uts, float uas, float tugas) {  
    return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);  
}
```

```
int main() {  
    const int MAX = 10;  
    Mahasiswa mhs[MAX];  
    int jumlah;  
  
    cout << "Masukkan jumlah mahasiswa (maks 10): ";  
    cin >> jumlah;  
  
    for (int i = 0; i < jumlah; i++) {  
        cout << "\nData mahasiswa ke-" << i + 1 << endl;  
        cout << "Nama: ";  
        cin >> mhs[i].nama;  
        cout << "NIM: ";  
        cin >> mhs[i].nim;  
        cout << "Nilai UTS: ";  
        cin >> mhs[i].uts;  
        cout << "Nilai UAS: ";  
        cin >> mhs[i].uas;  
        cout << "Nilai Tugas: ";  
        cin >> mhs[i].tugas;  
  
        mhs[i].nilaiAkhir = hitungNilaiAkhir(mhs[i].uts, mhs[i].uas, mhs[i].tugas);  
    }  
  
    cout << "\n=== DATA MAHASISWA ===" << endl;  
    for (int i = 0; i < jumlah; i++) {  
        cout << "Nama: " << mhs[i].nama << endl;
```

```

        cout << "NIM: " << mhs[i].nim << endl;
        cout << "UTS: " << mhs[i].uts << ", UAS: " << mhs[i].uas
            << ", Tugas: " << mhs[i].tugas << endl;
        cout << "Nilai Akhir: " << mhs[i].nilaiAkhir << endl;
        cout << "-----" << endl;
    }

    return 0;
}

```

Screenshots Output

```

Data mahasiswa ke-1
Nama: bintang
NIM: 10311400116
Nilai UTS: 80
Nilai UAS: 87
Nilai Tugas: 84

Data mahasiswa ke-2
Nama: azzam
NIM: 103112400112
Nilai UTS: 88
Nilai UAS: 81
Nilai Tugas: 822

=== DATA MAHASISWA ===
Nama: bintang
NIM: 10311400116
UTS: 80, UAS: 87, Tugas: 84
Nilai Akhir: 84
-----
Nama: azzam
NIM: 103112400112
UTS: 88, UAS: 81, Tugas: 822
Nilai Akhir: 305.4
-----

```

Deskripsi: Program ini digunakan untuk menginput dan menampilkan data beberapa mahasiswa serta menghitung nilai akhir mereka. Setiap mahasiswa memiliki data berupa nama, NIM, nilai UTS, nilai UAS, dan nilai tugas. Nilai akhir dihitung dengan rumus:

$$\text{nilai akhir} = (0.3 \times \text{UTS}) + (0.4 \times \text{UAS}) + (0.3 \times \text{Tugas})$$

Pengguna diminta memasukkan jumlah mahasiswa (maksimal 10), lalu mengisi data setiap mahasiswa satu per satu. Setelah semua data dimasukkan, program akan

menghitung nilai akhir masing-masing mahasiswa dan menampilkannya di layar.

D. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 2

```
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <iostream>
#include <string>
using namespace std;

typedef struct {
    string namaMapel;
    string kodeMapel;
} pelajaran;

pelajaran create_pelajaran(string namaMapel, string kodeMapel);

void tampil_pelajaran(pelajaran pel);

#endif
```

```
#include "pelajaran.h"
#include <iostream>
using namespace std;

pelajaran create_pelajaran(string namaMapel, string kodeMapel) {
    pelajaran p;
    p.namaMapel = namaMapel;
    p.kodeMapel = kodeMapel;
```

```
    return p;
}

void tampil_pelajaran(pelajaran pel) {
    cout << "Nama Pelajaran : " << pel.namaMapel << endl;
    cout << "Nilai : " << pel.kodeMapel << endl;
}
```

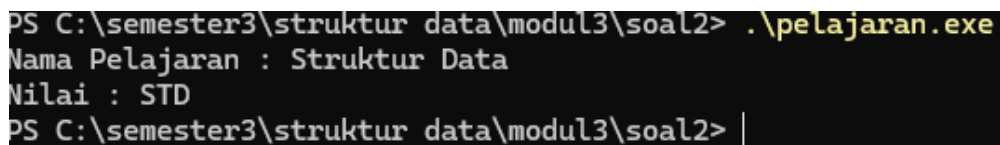
```
#include <iostream>
#include "pelajaran.h"
using namespace std;

int main() {
    string namaMapel = "Struktur Data";
    string kodeMapel = "STD";

    pelajaran pel = create_pelajaran(namaMapel, kodeMapel);
    tampil_pelajaran(pel);

    return 0;
}
```

Screenshots Output



```
PS C:\semester3\struktur data\modul3\soal2> .\pelajaran.exe
Nama Pelajaran : Struktur Data
Nilai : STD
PS C:\semester3\struktur data\modul3\soal2> |
```

Deskripsi: Struktur program ini terbagi ke dalam tiga file utama: pelajaran.h, pelajaran.cpp, dan main.cpp. File pelajaran.h berfungsi sebagai header file yang berisi

deklarasi struktur pelajaran serta dua fungsi, yaitu `create_pelajaran()` untuk membuat data baru dan `tampil_pelajaran()` untuk menampilkannya. Struktur pelajaran sendiri terdiri dari dua komponen: `namaMapel` dan `kodeMapel`. File `pelajaran.cpp` merealisasikan fungsi-fungsi tersebut. Fungsi `create_pelajaran()` akan mengisi nilai atribut sesuai input, sedangkan `tampil_pelajaran()` digunakan untuk menampilkan hasil data ke layar. Pada `main.cpp`, program utama membuat variabel `namaMapel` dan `kodeMapel` dengan nilai "Struktur Data" dan "STD", kemudian memanggil fungsi `create_pelajaran()` dan `tampil_pelajaran()` untuk menampilkan hasil akhirnya.

E. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 3

```
#include <iostream>

using namespace std;

void tampilArray(int arr[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
}

void tukarElemenArray(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
    int temp = arr1[baris][kolom];
    arr1[baris][kolom] = arr2[baris][kolom];
    arr2[baris][kolom] = temp;
}

void tukarPointer(int *p1, int *p2) {
    int temp = *p1;
    *p1 = *p2;
    *p2 = temp;
}
```

```
}
```

```
int main() {
```

```
    int A[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
```

```
    int B[3][3] = {{9,8,7},{6,5,4},{3,2,1}};
```

```
    cout << "Array A sebelum ditukar:\n";
```

```
    tampilArray(A);
```

```
    cout << "Array B sebelum ditukar:\n";
```

```
    tampilArray(B);
```

```
    tukarElemenArray(A, B, 1, 1);
```

```
    cout << "\nSetelah menukar elemen [1][1]:\n";
```

```
    cout << "Array A:\n";
```

```
    tampilArray(A);
```

```
    cout << "Array B:\n";
```

```
    tampilArray(B);
```

```
    int x = 10, y = 20;
```

```
    int *ptr1 = &x;
```

```
    int *ptr2 = &y;
```

```
    cout << "\nSebelum tukar pointer: x=" << x << ", y=" << y << endl;
```

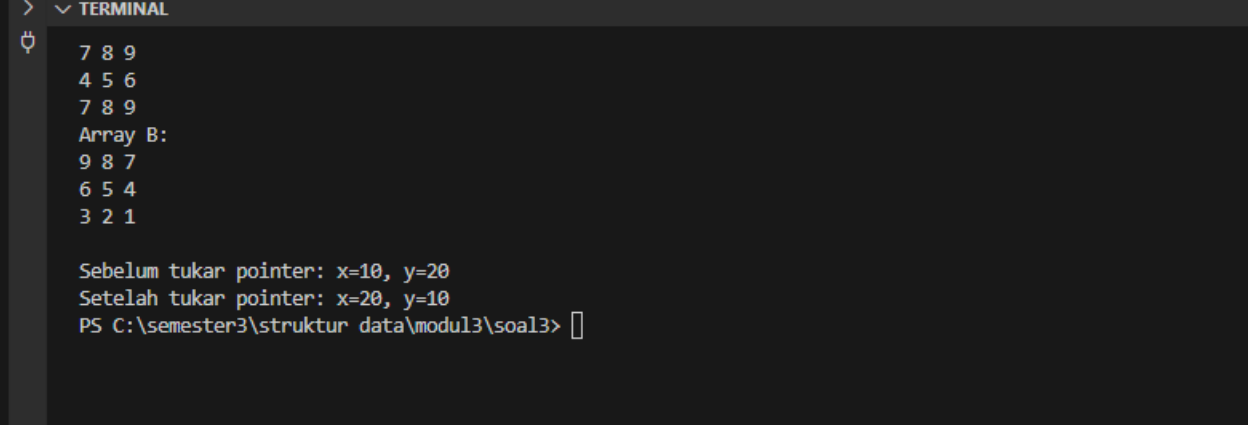
```
    tukarPointer(ptr1, ptr2);
```

```
    cout << "Setelah tukar pointer: x=" << x << ", y=" << y << endl;
```

```
    return 0;
```

```
}
```

Screenshots Output



```
> ▾ TERMINAL
7 8 9
4 5 6
7 8 9
Array B:
9 8 7
6 5 4
3 2 1

Sebelum tukar pointer: x=10, y=20
Setelah tukar pointer: x=20, y=10
PS C:\semester3\struktur data\modul3\soal3>
```

Deskripsi: Program mendefinisikan dua buah array 3x3 bernama A dan B dengan nilai yang berbeda. Fungsi `tampilArray()` digunakan untuk menampilkan isi array dalam bentuk matriks 3x3. Fungsi `tukarElemenArray()` digunakan untuk menukar satu elemen pada posisi tertentu antara dua array. Dalam program ini, elemen yang ditukar adalah elemen pada posisi `[1][1]` (baris ke-2, kolom ke-2) antara array A dan B. Setelah pertukaran dilakukan, program menampilkan kembali kedua array untuk memperlihatkan hasilnya.

Selain itu, program juga memiliki fungsi `tukarPointer()` yang menunjukkan cara menukar nilai dua variabel menggunakan pointer. Dua variabel, yaitu `x` dan `y`, masing-masing bernilai 10 dan 20, ditukar nilainya melalui pointer `ptr1` dan `ptr2`. Setelah proses pertukaran, nilai `x` dan `y` saling berubah.

F. Kesimpulan

Program ini dibuat untuk mengelola data pelajaran menggunakan struktur pelajaran dalam bahasa C++. File `pelajaran.h` berisi deklarasi struktur dan fungsi, `pelajaran.cpp` berisi isi fungsi, dan `main.cpp` berisi program utama untuk membuat dan menampilkan data pelajaran.

Fungsi `create_pelajaran()` digunakan untuk membuat data pelajaran, sedangkan `tampil_pelajaran()` menampilkan hasilnya. Contoh data yang digunakan adalah “Struktur Data” dengan kode “STD”.

Secara keseluruhan, program ini memperlihatkan cara kerja struct, fungsi, dan pemisahan file dalam C++. Dengan struktur kode seperti ini, program menjadi lebih rapi, mudah dibaca, dan mudah dikembangkan.

G. Referensi

Putra, M. T. D., Munawir, M., & Yuniarti, A. R. (2023). BELAJAR PEMROGRAMAN LANJUT DENGAN C++.

Siahaan, V., & Sianipar, R. H. (2020). *TUTORIAL Bahasa C++ Untuk Programmer*. BALIGE PUBLISHING.