

Aturan Praktikum Struktur Data

1. **Akun GitHub:** Setiap praktikan wajib memiliki akun GitHub yang aktif dan digunakan selama praktikum berlangsung.
2. **Invite Collaborator:** Setiap praktikan diwajibkan untuk menambahkan collaborator di setiap repository
 - a. Asisten Praktikum: AndiniNH
 - b. Asisten Praktikum: 4ldiputra
3. **Repository Praktikum:** Setiap praktikan diwajibkan untuk membuat satu repository di GitHub yang akan digunakan untuk seluruh tugas dan laporan praktikum. Repository ini harus diatur dengan rapi dan sesuai dengan instruksi yang akan diberikan di lampiran.
4. **Penamaan Folder:** Penamaan folder dalam repository akan dibahas secara rinci di lampiran. Praktikan wajib mengikuti aturan penamaan yang telah ditentukan.

Nomor	Pertemuan	Penamaan
1	Pengalaman Bahasa C++ Bagian Pertama	01_Pengenalan_CPP_Bagian_1
2	Pengenalan Bahasa C++ Bagian Kedua	02_Pengenalan_CPP_Bagian_2
3	Abstract Data Type	03_Abstract_Data_Type
4	Single Linked List Bagian Pertama	04_Single_Linked_List_Bagian_1
5	Single Linked List Bagian Kedua	05_Single_Linked_List_Bagian_2
6	Double Linked List Bagian Pertama	06_Double_Linked_List_Bagian_1
7	Stack	07_Stack
8	Queue	08_Queue
9	Assessment Bagian Pertama	09_Assessment_Bagian_1
10	Tree Bagian Pertama	10_Tree_Bagian_1
11	Tree Bagian Kedua	11_Tree_Bagian_2
12	Asistensi Tugas Besar	12_Asistensi_Tugas_Besar
13	Multi Linked List	13_Multi_Linked_List
14	Graph	14_Graph
15	Assessment Bagian Kedua	15_Assessment_Bagian_2
16	Tugas Besar	16_Tugas_Besar

5. Jam Praktikum:

- Jam masuk praktikum adalah **1 jam lebih lambat** dari jadwal yang tercantum. Sebagai contoh, jika jadwal praktikum adalah pukul 06.30 - 09.30, maka aturan praktikum akan diatur sebagai berikut:
 - **06.30 - 07.30:** Waktu ini digunakan untuk **Tugas Praktikum dan Laporan Praktikum** yang dilakukan di luar laboratorium.
 - **07.30 - 08.30:** Sesi ini mencakup **tutorial, diskusi, dan kasus problem-solving**. Kegiatan ini berlangsung di dalam laboratorium dengan alokasi waktu sebagai berikut:
 - **60 menit pertama:** Tugas terbimbing.
 - **60 menit kedua:** Tugas mandiri.

6. **Pengumpulan Tugasn Pendahuluan:** Tugas Pendahuluan (TP) wajib dikumpulkan melalui GitHub sesuai dengan format berikut:

nama_repo/nama_pertemuan/TP_Pertemuan_Ke.md

Sebagai contoh:

STD_Yudha_Islalmi_Sulistya_XXXXXXXX/01_Running_Modul/TP_01.md

7. **Pengecekan Tugas Pendahuluan:** Pengumpulan laporan praktikum akan diperiksa **1 hari sebelum praktikum selanjutnya** dimulai. Pastikan tugas telah diunggah tepat waktu untuk menghindari sanksi.

8. **Struktur Laporan Praktikum**

1. **Cover :**

LAPORAN PRAKTIKUM
Modul 3
“ABSTRACT DATA TYPE (ADT)”



Disusun Oleh:
KAFKA PUTRA RIYADI -2311104041
Kelas:
SE 07-02

Dosen :
Wahyu Andi Saputra, S.pd, M.Eng

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY
PURWOKERTO
2024

2. **Tujuan**

1. Memahami konsep Abstract Data Type (ADT) dan penggunaannya dalam pemrograman.

3. **Landasan Teori**

ADT (Abstract Data Type) adalah konsep dalam ilmu komputer yang merujuk pada tipe data yang didefinisikan secara abstrak, dengan fokus pada operasi yang dapat

dilakukan terhadap data tersebut, tanpa mengungkapkan implementasi konkret dari struktur data di belakangnya. Dengan kata lain, ADT menekankan apa yang bisa dilakukan dengan data, bukan bagaimana data disimpan atau diimplementasikan.

Komponen ADT:

1. Model Data: Definisi dari jenis data yang disimpan dan dioperasikan. Contohnya, list, stack, queue, atau tree.
2. Operasi: Sekumpulan operasi yang dapat dilakukan pada data tersebut. Operasi ini biasanya meliputi manipulasi data (seperti menambah, menghapus, atau mencari elemen) dan pengecekan keadaan (seperti memeriksa apakah struktur kosong).

ADT (Abstract Data Type) biasanya diimplementasikan menggunakan dua pendekatan utama:

1. Struktur Data

DT diimplementasikan menggunakan berbagai struktur data. Struktur data ini digunakan untuk menyimpan dan mengelola data dengan cara yang efisien. Struktur data yang dipilih untuk implementasi ADT tergantung pada kebutuhan operasi yang harus dilakukan pada data tersebut.

2. Algoritma

Selain struktur data, ADT juga diimplementasikan menggunakan algoritma. Algoritma ini digunakan untuk mendefinisikan bagaimana operasi pada ADT dilakukan, seperti penambahan, penghapusan, pencarian, atau manipulasi data lainnya. Implementasi ADT menggunakan algoritma mengacu pada cara di mana operasi dilakukan dengan efisien berdasarkan struktur data yang dipilih.

4. Guided

```
1  #include<iostream>
2
3  using namespace std;
4
5  struct mahasiswa{
6      char nim[10];
7      int nilai1, nilai2;
8  };
9
10 void inputMhs(mahasiswa &m);
11 float rata2(mahasiswa m);
12
13
14 int main()
15 {
16     mahasiswa mhs;
17     inputMhs(mhs);
18     cout<<"Rata-rata = "<<rata2(mhs);
19     return 0;
20 }
21
22 void inputMhs(mahasiswa &m){
23     cout << "input nim = ";
24     cin >> (m).nim;
25     cout << "input nilai1 = ";
26     cin >> (m).nilai1;
27     cout << "input nilai2 = ";
28     cin >> (m).nilai2;
29 }
30 float rata2(mahasiswa m){
31     return (m.nilai1+m.nilai2)/2;
32 }
33
34
```

Code program/codingan diatas digunakan untuk mencari rata rata, dengan menambahkan nilai kedua dengan nilai pertama lalu di bagi dengan 2 yang Dimana nilai tersebut di input oleh user dan system akan menentukan rata” nya

5. Unguided

1.

```
1  #include <iostream>
2  #include <iomanip>
3  #include <string>
4  using namespace std;
5
6  class Mahasiswa {
7  public:
8      string nama;
9      string nim;
10     double uts;
11     double uas;
12     double tugas;
13     double nilaiAkhir;
14
15     // Constructor
16     Mahasiswa(string n, string id, double u, double a, double t) {
17         nama = n;
18         nim = id;
19         uts = u;
20         uas = a;
21         tugas = t;
22         nilaiAkhir = calculateNilaiAkhir(); // Hitung nilai akhir saat objek dibuat
23     }
24
25     // Fungsi untuk menghitung nilai akhir
26     double calculateNilaiAkhir() {
27         return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
28     }
29
30     // Method untuk menampilkan data mahasiswa
31     void displayData() {
32         cout << "Nama: " << nama << endl;
33         cout << "NIM: " << nim << endl;
34         cout << "Nilai UTS: " << fixed << setprecision(2) << uts << endl;
35         cout << "Nilai UAS: " << fixed << setprecision(2) << uas << endl;
36         cout << "Nilai Tugas: " << fixed << setprecision(2) << tugas << endl;
37         cout << "Nilai Akhir: " << fixed << setprecision(2) << nilaiAkhir << endl;
38     }
39 }
40
41
42 int main() {
43     Mahasiswa* mahasiswaArray[10]; // Array untuk menyimpan data mahasiswa
44     int count = 0; // Menghitung jumlah mahasiswa yang disimpan
45
46     while (count < 10) {
47         string nama, nim;
48         double uts, uas, tugas;
49
50         cout << "Masukkan data mahasiswa ke-" << (count + 1) << endl;
51
52         cout << "Nama: ";
53         getline(cin, nama);
54
55         cout << "NIM: ";
56         getline(cin, nim);
57
58         cout << "Nilai UTS: ";
59         cin >> uts;
60
61         cout << "Nilai UAS: ";
62         cin >> uas;
63
64         cout << "Nilai Tugas: ";
65         cin >> tugas;
66         cin.ignore(); // membersihkan buffer cin
67
68         // Membuat objek mahasiswa dan menyimpannya dalam array
69         mahasiswaArray[count] = new Mahasiswa(nama, nim, uts, uas, tugas);
70         count++;
71
72         // Menanyakan apakah ingin menambah data mahasiswa lagi
73         if (count < 10) {
74             string response;
75             cout << "Apakah Anda ingin menambahkan data mahasiswa lagi? (ya/tidak): ";
76             getline(cin, response);
77             if (response != "ya" && response != "YA") {
78                 break;
79             }
80         }
81     }
82
83     // Menampilkan data mahasiswa
84     cout << "\nData Mahasiswa:" << endl;
85     for (int i = 0; i < count; i++) {
86         mahasiswaArray[i]->displayData();
87         delete mahasiswaArray[i]; // Menghapus objek setelah digunakan
88     }
89
90     return 0;
91 }
92
```

Outputannya:

```
Masukkan data mahasiswa ke-1
Nama: agus
NIM: 123456789
Nilai UTS: 90
Nilai UAS: 90
Nilai Tugas: 90
Apakah Anda ingin menambahkan data mahasiswa lagi? (ya/tidak): tidak

Data Mahasiswa:
Nama: agus
NIM: 123456789
Nilai UTS: 90.00
Nilai UAS: 90.00
Nilai Tugas: 90.00
Nilai Akhir: 90.00
PS D:\LAPRAK 3>
```

2. Pelajaran.cpp

```
#include "pelajaran.h"
#include <iostream>

using namespace std;

Pelajaran create_pelajaran(string namaMapel, string kodeMapel) {
    Pelajaran pel;
    pel.namaMapel = namaMapel;
    pel.kodeMapel = kodeMapel;
    return pel;
}

void tampil_pelajaran(const Pelajaran& pel) {
    cout << "nama pelajaran : " << pel.namaMapel << endl;
    cout << "nilai : " << pel.kodeMapel << endl;
}
```

Pelajaran.h

```
#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>
using namespace std;

struct Pelajaran {
    string namaMapel;
    string kodeMapel;
};

Pelajaran create_pelajaran(string namaMapel, string kodeMapel);

void tampil_pelajaran(const Pelajaran& pel);

#endif
```

Main.cpp

```
#include "pelajaran.h"
#include "pelajaran.cpp"

int main() {
    string namaPel = "Struktur Data";
    string kodePel = "STD";

    Pelajaran pel = create_pelajaran(namaPel, kodePel);

    tampil_pelajaran(pel);

    return 0;
}
```

Codingan tersebut dijadikan 1 dengan jenis bahasa pemrogramannya yaitu c++ (.cpp) dan c (.h) dengan cara memanggil file nya dengan #include “pelajaran.cpp” dan #include “pelajaran.h”

Dan ini hasil outputannya:

```
"C:\Users\slame\OneDrive\ID" x + v
nama pelajaran : Struktur Data
nilai : STD
Process returned 0 (0x0) execution time : 0.031 s
Press any key to continue.
```

3.

```
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 tukarArrayPosisi(int arr1[3][3], int arr2[3][3], int x, int y) {
15     int temp = arr1[x][y];
16     arr1[x][y] = arr2[x][y];
17     arr2[x][y] = temp;
18 }
19
20 void tukarPointer(int* a, int* b) {
21     int temp = *a;
22     *a = *b;
23     *b = temp;
24 }
25
26 int main() {
27     int array1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
28     int array2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
29
30     int a = 10, b = 20;
31     int *pA = &a;
32     int *pB = &b;
33
34     cout << "Array 1 sebelum ditukar:\n";
35     tampilkanArray(array1);
36     cout << "Array 2 sebelum ditukar:\n";
37     tampilkanArray(array2);
38
39     tukarArrayPosisi(array1, array2, 1, 1);
40
41     cout << "Array 1 setelah ditukar:\n";
42     tampilkanArray(array1);
43     cout << "Array 2 setelah ditukar:\n";
44     tampilkanArray(array2);
45
46     cout << "Nilai sebelum pointer ditukar: a = " << a << ", b = " << b << endl;
47
48     tukarPointer(pA, pB);
49
50     cout << "Nilai setelah pointer ditukar: a = " << a << ", b = " << b << endl;
51
52     return 0;
53 }
54
```

Outputannya :

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

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

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

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

Nilai sebelum pointer ditukar: a = 10, b = 20
Nilai setelah pointer ditukar: a = 20, b = 10
PS C:\Users\slame>
```

6. Kesimpulan

Setelah saya menyelesaikan laporan praktikum ini yang dimana saya harus mengerjakan soal yang diberikan pada modul saya menjadi tahu fungsi dari `#include`, karena pada unguided no.2 saya sedikit kebingungan kenapa codingan saya tidak bisa jalan padahal sudah hijau semua tidak ada yang error, ternyata kurang `#include "pelajaran.cpp"`