

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

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



Disusun Oleh :

NAMA : Afief Amar Purnomo

NIM : 103112430067

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Abstract Data Type (ADT) adalah konsep fundamental dalam pemrograman, termasuk di C++, yang mendefinisikan tipe data beserta operasi-operasi dasar yang boleh dilakukan terhadap tipe tersebut. Pada ADT, fokusnya bukan pada bagaimana data disimpan atau diolah secara internal, melainkan pada interaksi yang tersedia bagi pengguna melalui antarmuka (interface). Di C++, konsep ADT biasa diwujudkan melalui kelas (class) dan struct, di mana data anggota (member) dan fungsi anggota (member function) dikelompokkan bersama.

Implementasi detail dari data disembunyikan menggunakan mekanisme enkapsulasi dengan access specifier seperti private dan protected. Pengguna kelas hanya memahami fungsi-fungsi publik yang disediakan untuk mengakses dan memanipulasi data, tanpa perlu tahu bagaimana data tersebut diorganisir atau diimplementasikan secara internal. Ini membuat kode lebih modular, mudah dipelihara, dan meminimalkan risiko kesalahan akibat akses langsung ke data. Dengan ADT di C++, programmer dapat membangun tipe data baru dengan perilaku yang jelas dan konsisten, sekaligus menjaga keamanan data dan abstraksi yang baik antara penggunaan dan implementasi.

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

Guided 1

(file mahasiswa.h)

```
C mahasiswa.h > ...
1  // Nama : Afief Amar Purnomo
2  // NIM  : 103112430067
3
4  #ifndef MAHASISWA_H_INCLUDED
5  #define MAHASISWA_H_INCLUDED
6  struct mahasiswa
7  {
8      char nim[10];
9      int nilai1, nilai2;
10 };
    Tabnine | Edit | Test | Explain | Document
11 void inputMhs(mahasiswa &m);
    Tabnine | Edit | Test | Explain | Document
12 float rata2(mahasiswa m);
13 #endif
```

(file mahasiswa.cpp)

```
➤ mahasiswa.cpp > ...
1  // Nama : Afief Amar Purnomo
2  // NIM  : 103112430067
3
4  #include "mahasiswa.h"
5  #include <iostream>
6  using namespace std;
7
8  Tabnine | Edit | Test | Explain | Document
void inputMhs(mahasiswa &m)
9  {
10     cout << "input nama = ";
11     cin >> (m).nim;
12     cout << "input nilai = ";
13     cin >> (m).nilai1;
14     cout << "input nilai2 = ";
15     cin >> (m).nilai2;
16 }
17 Tabnine | Edit | Test | Explain | Document
float rata2(mahasiswa m)
18 {
19     return float(m.nilai1 + m.nilai2) / 2;
20 }
```

(file main.cpp)

```
➤ main.cpp > ...
1  // Nama : Afief Amar Purnomo
2  // NIM  : 103112430067
3
4  #include <iostream>
5  #include "mahasiswa.h"
6  using namespace std;
7  Tabnine | Edit | Test | Explain | Document
int main()
8  {
9     mahasiswa mhs;
10     inputMhs(mhs);
11     cout << "rata - rata = " << rata2(mhs);
12     return 0;
13 }
```

Screenshots Output

```
● PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Guide>g++ main.cpp mahasiswa.cpp
● PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Guide> ./a.exe
input nama = Amar
input nilai = 100
input nilai2 = 90
rata - rata = 95
○ PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Guide> █
```

Deskripsi:

Program di atas merupakan contoh penerapan Abstract Data Type (ADT) dalam bahasa C++. ADT adalah tipe data yang didefinisikan oleh struktur data dan operasi yang dapat dilakukan terhadapnya, tanpa memperlihatkan detail implementasinya kepada pengguna. Dalam hal ini, tipe data abstrak yang digunakan adalah mahasiswa, yang didefinisikan di dalam file header **mahasiswa.h**. File **mahasiswa.h** berfungsi sebagai definisi tipe data abstrak, di mana dideklarasikan sebuah struct mahasiswa yang memiliki tiga anggota, yaitu nim (bertipe karakter array), nilai1, dan nilai2 (keduanya bertipe integer). Selain itu, dalam file ini juga terdapat deklarasi dua fungsi, yaitu inputMhs() untuk mengisi data mahasiswa, dan rata2() untuk menghitung rata-rata nilai mahasiswa. Dengan demikian, file ini hanya menjelaskan apa yang bisa dilakukan (interface) tanpa memperlihatkan bagaimana caranya dilakukan (implementasi). Implementasi fungsi-fungsi tersebut terdapat di dalam file **mahasiswa.cpp**. Fungsi inputMhs() digunakan untuk memasukkan data mahasiswa melalui input pengguna (NIM, nilai1, dan nilai2). Fungsi rata2() digunakan untuk menghitung rata-rata dua nilai mahasiswa dengan rumus $(\text{nilai1} + \text{nilai2}) / 2$. Kedua fungsi ini memanipulasi data dari struct mahasiswa tanpa perlu diketahui detail strukturnya oleh file utama **main.cpp**. File main.cpp berperan sebagai program utama yang memanfaatkan tipe data abstrak tersebut. Di sini, dibuat satu variabel mahasiswa mhs, kemudian dilakukan pemanggilan fungsi inputMhs(mhs) untuk mengisi data, dan hasil perhitungan rata-rata ditampilkan menggunakan rata2(mhs). Dengan pemisahan antara definisi tipe data (**mahasiswa.h**), implementasi fungsi (**mahasiswa.cpp**), dan penggunaan tipe data (**main.cpp**), program ini menunjukkan konsep utama ADT, yaitu enkapsulasi (encapsulation) dan abstraksi data (data abstraction). Pengguna cukup mengetahui bagaimana menggunakan fungsi ADT tanpa perlu memahami detail internal strukturnya.

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

Unguided 1 (soal 1)

```
soal1.cpp > main()
1 // Nama : Afief Amar Purnomo
2 // NIM : 103112430067
3
4 #include <iostream>
5 #include <string>
6 using namespace std;
7
8 float hitungNilaiAkhir(float uts, float uas, float tugas) {
9     return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
10 }
11
12 int main() {
13     const int MAX = 10;
14     string nama[MAX], nim[MAX];
15     float uts[MAX], uas[MAX], tugas[MAX], nilaiAkhir[MAX];
16     int jumlah;
17
18     cout << "Masukkan jumlah mahasiswa (maks 10): ";
19     cin >> jumlah;
20
21     if (jumlah > 10) {
22         cout << "Jumlah melebihi batas maksimum!" << endl;
23         return 0;
24     }
25
26     cout << endl;
27
28     for (int i = 0; i < jumlah; i++) {
29         cout << "Data mahasiswa ke-" << i + 1 << endl;
30         cout << "Nama : ";
```

```

31         cin >> ws;
32         getline(cin, nama[i]);
33         cout << "NIM      : ";
34         cin >> nim[i];
35         cout << "Nilai UTS   : ";
36         cin >> uts[i];
37         cout << "Nilai UAS   : ";
38         cin >> uas[i];
39         cout << "Nilai Tugas : ";
40         cin >> tugas[i];
41
42         nilaiAkhir[i] = hitungNilaiAkhir(uts[i], uas[i], tugas[i]);
43         cout << endl;
44     }
45
46     cout << "\n=== Data Mahasiswa ===\n";
47     for (int i = 0; i < jumlah; i++) {
48         cout << "\nMahasiswa ke-" << i + 1 << endl;
49         cout << "Nama      : " << nama[i] << endl;
50         cout << "NIM       : " << nim[i] << endl;
51         cout << "Nilai UTS : " << uts[i] << endl;
52         cout << "Nilai UAS : " << uas[i] << endl;
53         cout << "Nilai Tugas : " << tugas[i] << endl;
54         cout << "Nilai Akhir : " << nilaiAkhir[i] << endl;
55     }
56
57     return 0;
58 }
59

```

Screenshots Output

```

PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> g++ soal1.cpp -o soal1
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> ./soal1
Masukkan jumlah mahasiswa (maks 10): 2

Data mahasiswa ke-1
• Nama      : Amar
• NIM       : 103112430067
Nilai UTS   : 100
Nilai UAS   : 90
Nilai Tugas : 89

Data mahasiswa ke-2
Nama      : Afief
NIM       : 103218727678
Nilai UTS : 90
Nilai UAS : 96
Nilai Tugas : 88

```

```
=== Data Mahasiswa ===
```

```
Mahasiswa ke-1
```

```
Nama      : Amar  
NIM       : 103112430067  
Nilai UTS : 100  
Nilai UAS : 90  
Nilai Tugas : 89  
Nilai Akhir : 92.7
```

```
Mahasiswa ke-2
```

```
Nama      : Afief  
NIM       : 103218727678  
Nilai UTS : 90  
Nilai UAS : 96  
Nilai Tugas : 88  
Nilai Akhir : 91.8
```

```
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide>
```

Deskripsi:

Program di atas adalah program dalam bahasa C++ yang digunakan untuk menginput, menghitung, dan menampilkan nilai akhir mahasiswa berdasarkan nilai UTS, UAS, dan tugas. Program ini menggunakan array untuk menyimpan data hingga maksimal 10 mahasiswa, meliputi nama, NIM, UTS, UAS, dan tugas masing-masing. Perhitungan nilai akhir dilakukan melalui fungsi `hitungNilaiAkhir()` dengan rumus 30% UTS, 40% UAS, dan 30% tugas. Program meminta pengguna untuk memasukkan jumlah mahasiswa, kemudian melalui perulangan `for`, data setiap mahasiswa diinput dan nilai akhirnya dihitung serta disimpan. Setelah seluruh data dimasukkan, program menampilkan hasil lengkap berisi nama, NIM, nilai UTS, UAS, tugas, dan nilai akhir masing-masing mahasiswa. Secara keseluruhan, program ini menunjukkan penerapan array, fungsi, dan perulangan dalam mengolah serta menampilkan data mahasiswa secara terstruktur.

Unguided 2 (soal 2)

(file pelajaran.h)

```
C pelajaran.h > ...
1  // Nama : Afief Amar Purnomo
2  // NIM  : 103112430067
3
4  #ifndef PELAJARAN_H_INCLUDED
5  #define PELAJARAN_H_INCLUDED
6
7  #include <iostream>
8  #include <string>
9  using namespace std;
10
11  struct pelajaran {
12      string namaMapel;
13      string kodeMapel;
14  };
15
16  pelajaran create_pelajaran(string namapel, string kodepel);
17  void tampil_pelajaran(pelajaran pel);
18
19  #endif
20
```

(file pelajaran.cpp)


```

pelajaran.cpp > ...
1  // Nama : Afief Amar Purnomo
2  // NIM : 103112430067
3
4  #include <iostream>
5  #include "pelajaran.h"
6  using namespace std;
7
8  Tabnine | Edit | Test | Explain | Document
9  pelajaran create_pelajaran(string namapel, string kodepel) {
10     pelajaran p;
11     p.namaMapel = namapel;
12     p.kodeMapel = kodepel;
13     return p;
14 }
15
16 Tabnine | Edit | Test | Explain | Document
17 void tampil_pelajaran(pelajaran pel) {
18     cout << "Nama Pelajaran : " << pel.namaMapel << endl;
19     cout << "Kode Pelajaran : " << pel.kodeMapel << endl;
20 }

```

(file main.cpp)

```

main.cpp > ...
1  // Nama : Afief Amar Purnomo
2  // NIM : 103112430067
3
4  #include <iostream>
5  #include "pelajaran.h"
6  using namespace std;
7
8  Tabnine | Edit | Test | Explain | Document
9  int main() {
10     string namapel = "Struktur Data";
11     string kodepel = "STD";
12
13     pelajaran pel = create_pelajaran(namapel, kodepel);
14
15     tampil_pelajaran(pel);
16
17     return 0;
18 }

```

Screenshots Output

```
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> g++ main.cpp pelajaran.cpp
● PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> ./a.exe
Nama Pelajaran : Struktur Data
Kode Pelajaran : STD
○ PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> █
```

Deskripsi:

Program di atas merupakan contoh penerapan Abstract Data Type (ADT) dalam bahasa C++ dengan tipe data abstrak bernama pelajaran. Pada file **pelajaran.h** didefinisikan struktur pelajaran yang memiliki dua atribut, yaitu namaMapel dan kodeMapel, serta deklarasi dua fungsi, yaitu `create_pelajaran()` untuk membuat data pelajaran dan `tampil_pelajaran()` untuk menampilkannya. Implementasi kedua fungsi tersebut terdapat pada file **pelajaran.cpp**, di mana `create_pelajaran()` mengisi nilai atribut pelajaran dan mengembalikannya, sedangkan `tampil_pelajaran()` mencetak data pelajaran ke layar. File **main.cpp** berfungsi sebagai program utama yang memanfaatkan tipe data abstrak tersebut dengan membuat objek pelajaran, mengisinya melalui `create_pelajaran()`, dan menampilkannya menggunakan `tampil_pelajaran()`. Pemisahan antara definisi, implementasi, dan penggunaan menunjukkan penerapan konsep enkapsulasi dan abstraksi data dalam ADT.

Unguided 3 (soal 3)

```
soal3.cpp > tampilArray(int [3][3])
1 // Nama : Afief Amar Purnomo
2 // NIM : 103112430067
3
4 #include <iostream>
5 using namespace std;
6
7 void tampilArray(int arr[3][3]) {
8     for (int i = 0; i < 3; i++) {
9         for (int j = 0; j < 3; j++) {
10             cout << arr[i][j] << "\t";
11         }
12         cout << endl;
13     }
14 }
15
16 void tukarPosisi(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 void tukarPointer(int *p1, int *p2) {
23     int temp = *p1;
24     *p1 = *p2;
25     *p2 = temp;
26 }
27
28 int main() {
```

```

29     int A[3][3] = {
30         {1, 2, 3},
31         {4, 5, 6},
32         {7, 8, 9}
33     };
34
35     int B[3][3] = {
36         {10, 11, 12},
37         {13, 14, 15},
38         {16, 17, 18}
39     };
40
41     int *p1, *p2;
42
43     cout << "Array A:" << endl;
44     tampilArray(A);
45     cout << "\nArray B:" << endl;
46     tampilArray(B);
47
48     int baris = 1, kolom = 2;
49     tukarPosisi(A, B, baris, kolom);
50
51     cout << "\nSetelah menukar elemen pada posisi (" << baris << ", " << kolom << "):" << endl;
52     cout << "Array A:" << endl;
53     tampilArray(A);
54     cout << "\nArray B:" << endl;
55     tampilArray(B);
56
57     int x = 100, y = 200;
58     p1 = &x;
59     p2 = &y;
60     cout << "\nSebelum ditukar: x = " << x << ", y = " << y << endl;
61     tukarPointer(p1, p2);
62     cout << "Setelah ditukar : x = " << x << ", y = " << y << endl;
63
64     return 0;
65 }
66

```

Screenshots Output

```

PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> g++ soal3.cpp -o soal3
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide> ./soal3
Array A:
1      2      3
4      5      6
7      8      9

Array B:
10     11     12
13     14     15
16     17     18

Setelah menukar elemen pada posisi (1,2):
Array A:
1      2      3
4      5      15
7      8      9

Array B:
10     11     12
13     14     6
16     17     18

Sebelum ditukar: x = 100, y = 200
Setelah ditukar : x = 200, y = 100
PS D:\Kuliah\Semester 3\MATKUL\Praktikum Struktur Data\Praktikum\Modul_3\Unguide>

```

Deskripsi:

Program di atas merupakan program C++ yang menunjukkan penerapan fungsi, array dua dimensi, dan pointer untuk memanipulasi data dalam memori. Program ini mendefinisikan dua buah array 3x3 bernama A dan B yang masing-masing berisi angka berbeda, lalu menampilkan isinya menggunakan fungsi `tampilArray()`. Fungsi `tukarPosisi()` digunakan untuk menukar elemen array A dan B pada posisi tertentu dengan cara menukar nilainya berdasarkan indeks baris dan kolom yang diberikan. Selain itu, program juga memperlihatkan penggunaan pointer melalui fungsi `tukarPointer()`, yang menukar nilai dua variabel (x dan y) menggunakan alamat memorinya. Setelah pertukaran dilakukan, program menampilkan hasil perubahan pada kedua array dan variabel untuk menunjukkan perbedaan sebelum dan sesudah proses penukaran. Dengan demikian, program ini menggambarkan konsep dasar manipulasi array dua dimensi, penggunaan fungsi untuk modularisasi, serta penerapan pointer dalam pertukaran nilai secara langsung di memori.

D. Kesimpulan

Berdasarkan hasil percobaan pada bagian guided dan unguided, dapat disimpulkan bahwa bahasa pemrograman C++ sangat mendukung penerapan konsep Abstract Data Type (ADT) melalui penggunaan struct, fungsi, array, dan pointer. Pada percobaan guided, mahasiswa belajar bagaimana mendefinisikan tipe data abstrak seperti mahasiswa dan pelajaran, memisahkan antara definisi (header) dan implementasi (source), serta menggunakan fungsi untuk memanipulasi data tanpa mengetahui detail internalnya.

Pada bagian unguided, konsep tersebut diterapkan dalam kasus yang lebih kompleks, seperti pengelolaan data mahasiswa menggunakan array dan perhitungan nilai akhir menggunakan fungsi modular. Percobaan juga menekankan pentingnya pemahaman pointer dalam pertukaran nilai dan pengaksesan memori secara langsung. Secara keseluruhan, praktikum ini membuktikan bahwa penerapan ADT dalam C++ meningkatkan modularitas, efisiensi, dan kejelasan struktur program.

E. Referensi

Bernardinus Harnadi, Antonius Eldy Putra, Wilibrordus Endra, dkk. (2025). Dasar Algoritma dan Pemrograman dengan C++. Buku ini mencakup dasar pemrograman C++ termasuk array dan pointer. Diakses dari <https://www.unika.ac.id/wp-content/uploads/2025/02/ebook-Berdi-Dasar-Logika-Pemrograman-C.pdf>.

CodePolitan. (2024, Januari 31). Belajar Pointer C++: Dasar-Dasar, Fungsi, dan Contoh Kode. Diakses dari <https://codepolitan.com/blog/belajar-pointer-c-dasardasar-fungsi-dan-contoh-kode>

Datta, S. (2024, March 18). What is abstract data type? Baeldung on Computer Science. <https://www.baeldung.com/cs/adt>