

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL III
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Taufik Hafit Zakaria

NIM : 103112430093

Dosen

WAHYU ANDI SAPUTRA

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

A. Dasar Teori

Abstract Data Type (ADT) adalah sebuah model konseptual untuk tipe data yang mendefinisikan sekumpulan nilai dan operasi yang berlaku padanya, tanpa memaparkan bagaimana implementasi internalnya. ADT berfokus pada apa yang dapat dilakukan oleh sebuah tipe data (antarmukanya), bukan bagaimana cara melakukannya (implementasinya).

Konsep Inti ADT:

1. **Abstraksi:** Menyembunyikan detail implementasi yang rumit dan hanya menampilkan fungsionalitas yang esensial kepada pengguna. Pengguna tidak perlu tahu bagaimana data disimpan atau bagaimana sebuah operasi bekerja.
2. **Enkapsulasi:** Menggabungkan data (struktur) dengan operasi-operasi yang memanipulasinya ke dalam satu unit tunggal. Ini melindungi data dari akses dan modifikasi yang tidak sah.

Implementasi dalam Pemrograman:

Dalam praktiknya, seperti pada bahasa C++, ADT diimplementasikan dengan memisahkan definisi dan realisasi ke dalam file yang berbeda untuk mendukung modularitas:

1. **File Header (.h):** Berisi deklarasi tipe data (misalnya, menggunakan struct) dan prototipe fungsi (primitif). File ini bertindak sebagai antarmuka (interface) publik dari ADT.
2. **File Implementasi (.cpp):** Berisi kode aktual atau definisi dari fungsi-fungsi yang telah dideklarasikan di file header. Bagian ini adalah implementasi (body) yang tersembunyi dari pengguna.

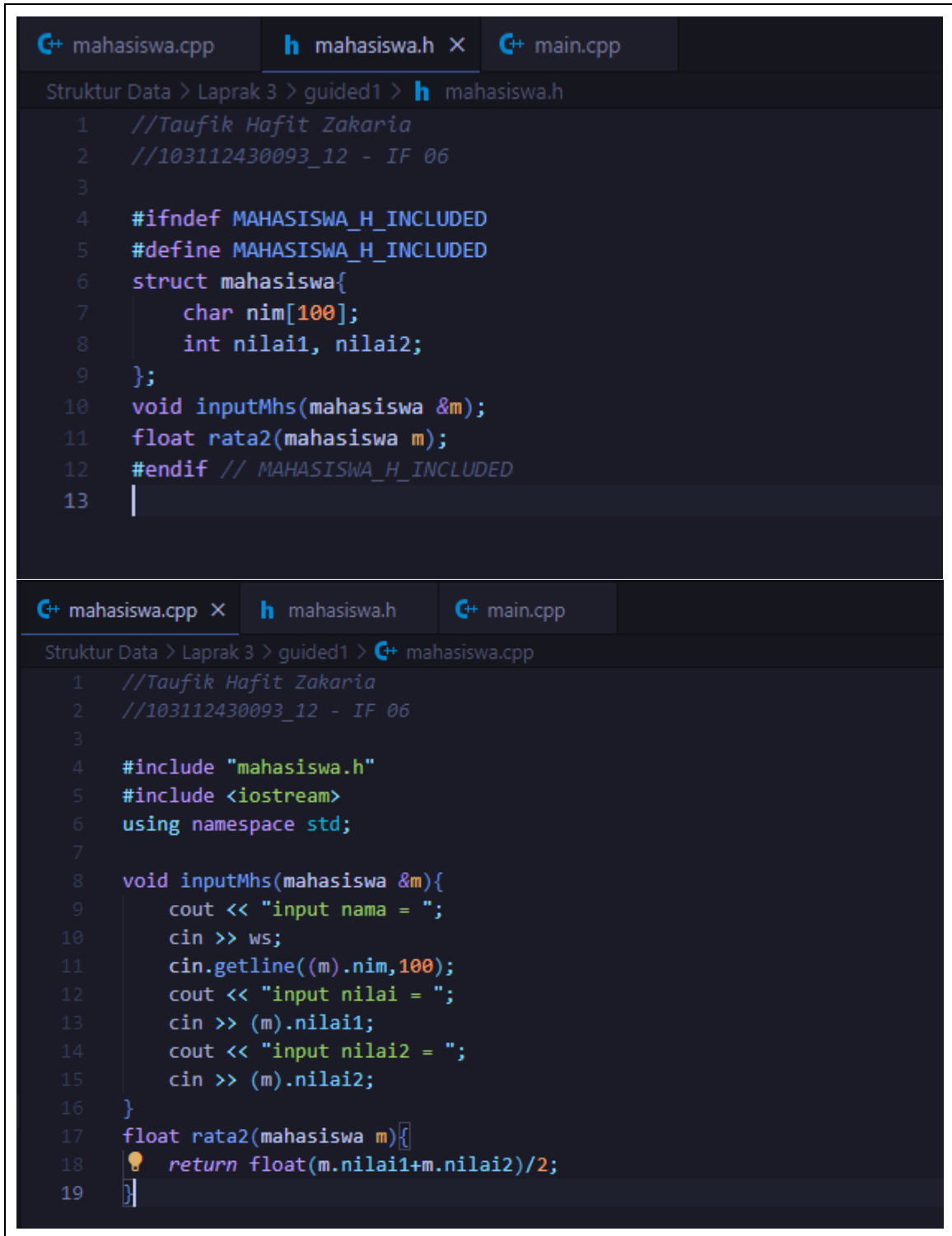
Operasi Dasar (Primitif) pada ADT:

Operasi dasar pada ADT umumnya dikelompokkan menjadi:

1. **Konstruktor/Kreator:** Membuat sebuah objek baru dari tipe ADT.
2. **Selektor:** Mengakses atau mengambil nilai dari komponen data dalam ADT.
3. **Mutator (Pengubah):** Mengubah nilai komponen dalam ADT.
4. **Destruktor:** Menghancurkan objek dan membebaskan memori yang digunakan.

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

Guided 1

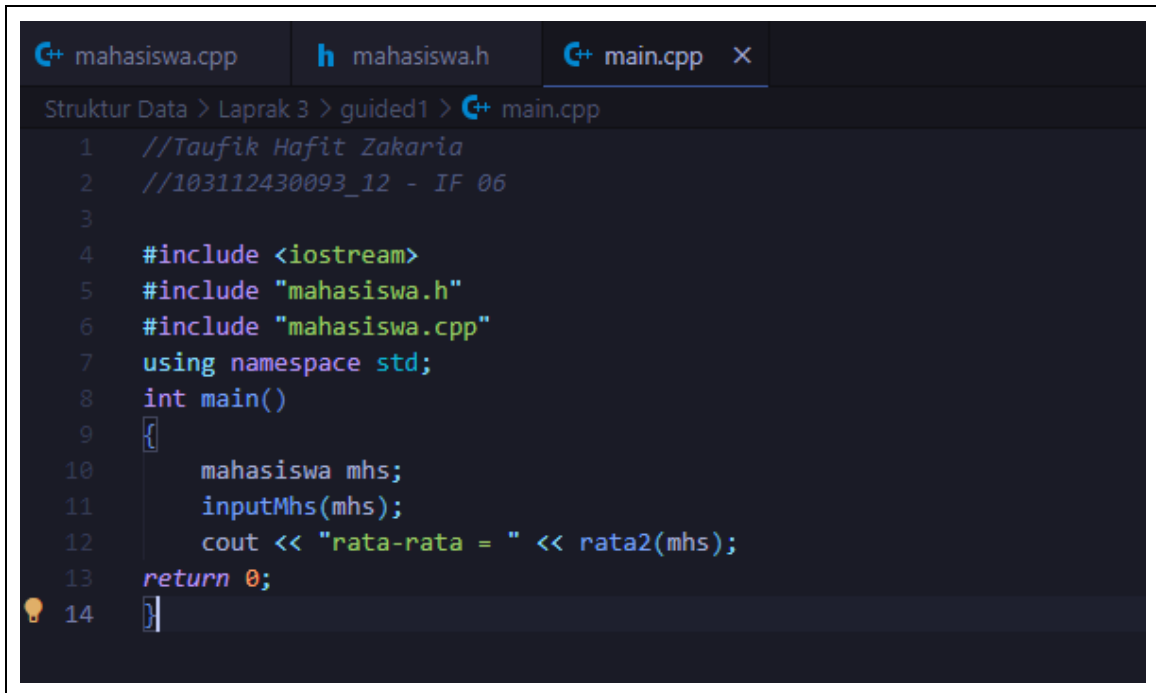


```

G++ mahasiswa.cpp | h mahasiswa.h X | G++ main.cpp
Struktur Data > Laprak 3 > guided1 > h mahasiswa.h
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #ifndef MAHASISWA_H_INCLUDED
5 #define MAHASISWA_H_INCLUDED
6 struct mahasiswa{
7     char nim[100];
8     int nilai1, nilai2;
9 };
10 void inputMhs(mahasiswa &m);
11 float rata2(mahasiswa m);
12 #endif // MAHASISWA_H_INCLUDED
13 |

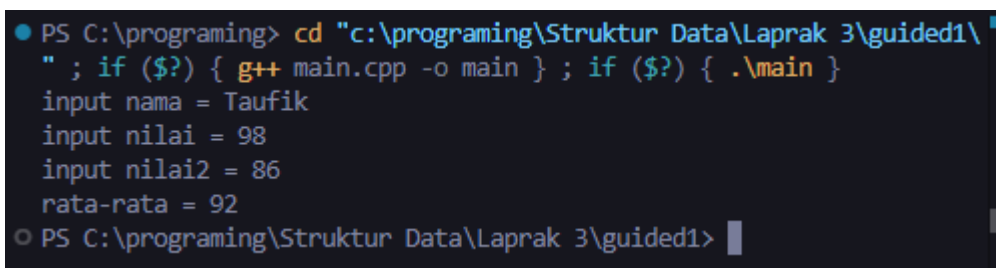
G++ mahasiswa.cpp X | h mahasiswa.h | G++ main.cpp
Struktur Data > Laprak 3 > guided1 > G++ mahasiswa.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include "mahasiswa.h"
5 #include <iostream>
6 using namespace std;
7
8 void inputMhs(mahasiswa &m){
9     cout << "input nama = ";
10    cin >> ws;
11    cin.getline((m).nim,100);
12    cout << "input nilai = ";
13    cin >> (m).nilai1;
14    cout << "input nilai2 = ";
15    cin >> (m).nilai2;
16 }
17 float rata2(mahasiswa m){
18     return float(m.nilai1+m.nilai2)/2;
19 }

```



```
C++ mahasiswa.cpp | h mahasiswa.h | C++ main.cpp x
Struktur Data > Laprak 3 > guided1 > C++ main.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 #include "mahasiswa.h"
6 #include "mahasiswa.cpp"
7 using namespace std;
8 int main()
9 {
10     mahasiswa mhs;
11     inputMhs(mhs);
12     cout << "rata-rata = " << rata2(mhs);
13     return 0;
14 }
```

Screenshots Output



```
PS C:\programming> cd "c:\programming\Struktur Data\Laprak 3\guided1\"
" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
input nama = Taufik
input nilai = 98
input nilai2 = 86
rata-rata = 92
PS C:\programming\Struktur Data\Laprak 3\guided1>
```

Deskripsi:

Program ini adalah implementasi sederhana dari konsep Abstract Data Type (ADT) untuk mengelola data mahasiswa. Tujuannya adalah untuk menerima masukan nama dan dua nilai dari pengguna, lalu menghitung dan menampilkan nilai rata-ratanya.

Struktur program dibagi menjadi tiga file utama, yang masing-masing memiliki peran spesifik:

1. mahasiswa.h (Antarmuka / Interface)

File ini mendefinisikan struktur data mahasiswa (terdiri dari nim, nilai1, dan nilai2).

File ini juga mendeklarasikan fungsi-fungsi yang dapat beroperasi pada data tersebut, yaitu inputMhs() dan rata2(). Ini adalah "wajah" dari ADT yang dilihat oleh pengguna.

2. mahasiswa.cpp (Implementasi / Body)

File ini berisi kode sumber atau implementasi detail dari fungsi inputMhs dan rata2.

Logika tentang bagaimana cara meminta input dan bagaimana cara menghitung rata-rata disembunyikan di dalam file ini.

3. main.cpp (Program Utama / Driver)

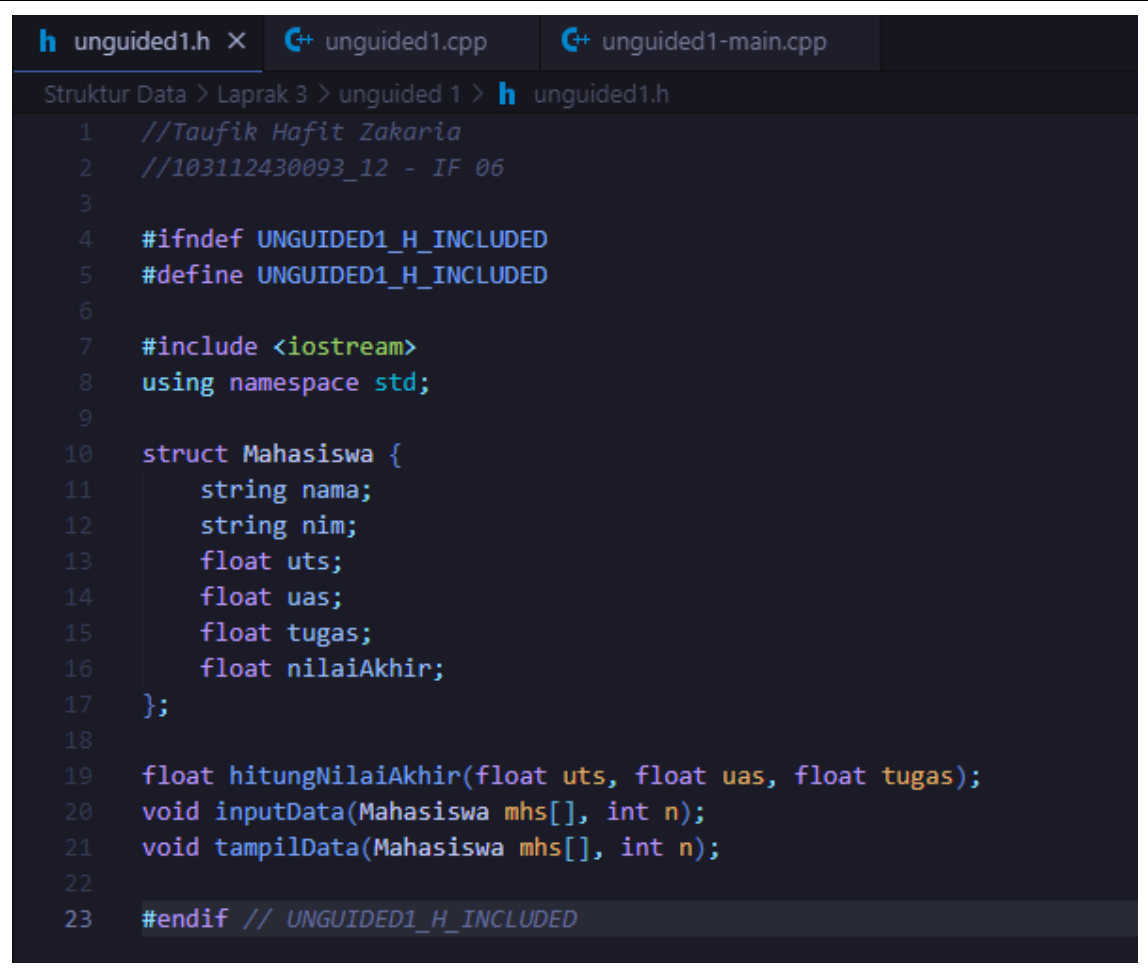
File ini bertindak sebagai pengguna (klien) dari ADT mahasiswa.

Fungsi main() di sini membuat sebuah variabel mahasiswa, memanggil fungsi inputMhs() untuk mengisi datanya, dan memanggil rata2() untuk menampilkan hasilnya.

Secara singkat, program ini menunjukkan pemisahan yang jelas antara definisi (apa yang bisa dilakukan) di file .h dan implementasi (bagaimana cara melakukannya) di file .cpp, yang merupakan prinsip inti dari ADT.

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

Unguided 1



The screenshot shows a C++ IDE with three tabs: 'unguided1.h', 'unguided1.cpp', and 'unguided1-main.cpp'. The 'unguided1.h' tab is active, displaying the following code:

```
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #ifndef UNGUIDED1_H_INCLUDED
5 #define UNGUIDED1_H_INCLUDED
6
7 #include <iostream>
8 using namespace std;
9
10 struct Mahasiswa {
11     string nama;
12     string nim;
13     float uts;
14     float uas;
15     float tugas;
16     float nilaiAkhir;
17 };
18
19 float hitungNilaiAkhir(float uts, float uas, float tugas);
20 void inputData(Mahasiswa mhs[], int n);
21 void tampilData(Mahasiswa mhs[], int n);
22
23 #endif // UNGUIDED1_H_INCLUDED
```

```

h unguided1.h  C++ unguided1.cpp X  C++ unguided1-main.cpp
Struktur Data > Laprak 3 > unguided 1 > C++ unguided1.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3 Ctrl+I for Command, Ctrl+L for Cascade
4 #include "unguided1.h"
5
6 float hitungNilaiAkhir(float uts, float uas, float tugas) {
7     return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);
8 }
9
10 void inputData(Mahasiswa mhs[], int n) {
11     cout << "\n=== Input Data Mahasiswa ===\n";
12     for (int i = 0; i < n; i++) {
13         cout << "\nMahasiswa ke-" << i + 1 << endl;
14         cout << "Nama : ";
15         cin >> mhs[i].nama;
16         cout << "NIM : ";
17         cin >> mhs[i].nim;
18         cout << "UTS : ";
19         cin >> mhs[i].uts;
20         cout << "UAS : ";
21         cin >> mhs[i].uas;
22         cout << "Tugas : ";
23         cin >> mhs[i].tugas;
24
25         mhs[i].nilaiAkhir = hitungNilaiAkhir(mhs[i].uts, mhs[i].uas, mhs[i].tugas);
26     }
27 }
28
29 void tampilData(Mahasiswa mhs[], int n) {
30     cout << "\n=== Data Mahasiswa ===\n";
31     for (int i = 0; i < n; i++) {
32         cout << "\nMahasiswa ke-" << i + 1 << endl;
33         cout << "Nama : " << mhs[i].nama << endl;
34         cout << "NIM : " << mhs[i].nim << endl;
35         cout << "UTS : " << mhs[i].uts << endl;
36         cout << "UAS : " << mhs[i].uas << endl;
37         cout << "Tugas : " << mhs[i].tugas << endl;
38         cout << "Nilai Akhir : " << mhs[i].nilaiAkhir << endl;
39     }
40 }

```

h unguided1.h C++ unguided1.cpp C++ unguided1-main.cpp X

Struktur Data > Laprak 3 > unguided 1 > C++ unguided1-main.cpp

```
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 #include "unguided1.h"
6 #include "unguided1.cpp"
7 using namespace std;
8
9 int main() {
10     Mahasiswa mhs[10];
11     int n;
12
13     cout << "Input jumlah mahasiswa (maks 10): ";
14     cin >> n;
15
16     if (n < 1 || n > 10) {
17         cout << "Jumlah mahasiswa tidak valid!" << endl;
18         return 0;
19     }
20
21     inputData(mhs, n);
22     tampilData(mhs, n);
23
24     return 0;
25 }
```


Screenshots Output

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

Mahasiswa ke-1
Nama      : Taufik
NIM       : 103112430093
UTS       : 87
UAS       : 96
Tugas    : 86

Mahasiswa ke-2
Nama      : Akmal
NIM       : 103112430093
UTS       : 84
UAS       : 82
Tugas    : 90

Mahasiswa ke-3
Nama      : Vincent
NIM       : 103112430095
UTS       : 79
UAS       : 89
Tugas    : 88

=== Data Mahasiswa ===

Mahasiswa ke-1
Nama      : Taufik
NIM       : 103112430093
UTS       : 87
UAS       : 96
Tugas    : 86
Nilai Akhir : 90.3

Mahasiswa ke-2
Nama      : Akmal
NIM       : 103112430093
UTS       : 84
UAS       : 82
Tugas    : 90
Nilai Akhir : 85

Mahasiswa ke-3
Nama      : Vincent
NIM       : 103112430095
UTS       : 79
UAS       : 89
Tugas    : 88
Nilai Akhir : 85.7
PS C:\programing\Struktur Data\Laprak 3\unguided 1> █
```

Deskripsi:

Program ini adalah sistem sederhana untuk mengelola data akademik dari beberapa mahasiswa (maksimal 10). Fungsinya adalah untuk menerima input data mahasiswa (nama, NIM, nilai UTS, UAS, dan tugas), menghitung nilai akhir secara otomatis, dan kemudian menampilkannya kembali ke layar.

Program ini menerapkan konsep Abstract Data Type (ADT) dengan memisahkan kode ke dalam tiga file:

1. `unguided1.h` (Antarmuka / Interface):

Mendefinisikan struct Mahasiswa sebagai tipe data.

Mendeklarasikan fungsi-fungsi utama: `inputData`, `tampilData`, dan `hitungNilaiAkhir`. File ini adalah "wajah publik" dari ADT.

2. `unguided1.cpp` (Implementasi / Body):

Berisi implementasi atau kode sumber dari fungsi-fungsi yang dideklarasikan di file header.

Logika untuk perulangan input data dan rumus perhitungan nilai akhir berada di sini, tersembunyi dari program utama.

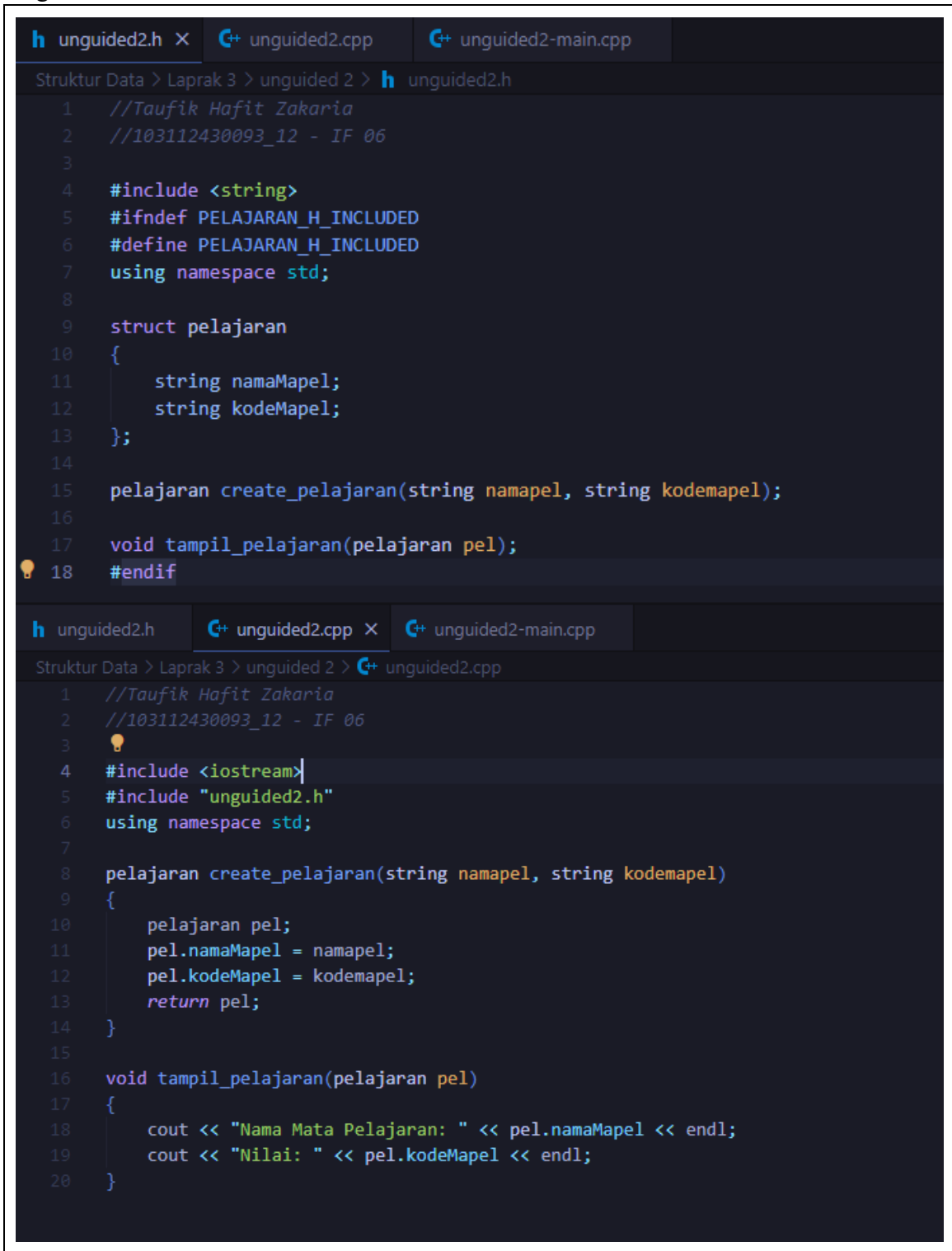
3. `main.cpp` (Program Utama / Driver):

Bertindak sebagai pengguna (klien) dari ADT Mahasiswa.

Mengatur alur utama program: meminta jumlah mahasiswa, memvalidasi input, serta memanggil fungsi `inputData` dan `tampilData` untuk menjalankan fungsionalitas inti.

Secara keseluruhan, program ini adalah contoh yang baik dari aplikasi ADT untuk mengelola sekumpulan data (array of structs), dengan struktur kode yang bersih dan modular.

Unguided 2



```
h unguided2.h X C++ unguided2.cpp C++ unguided2-main.cpp
Struktur Data > Laprak 3 > unguided 2 > h unguided2.h
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <string>
5 #ifndef PELAJARAN_H_INCLUDED
6 #define PELAJARAN_H_INCLUDED
7 using namespace std;
8
9 struct pelajaran
10 {
11     string namaMapel;
12     string kodeMapel;
13 };
14
15 pelajaran create_pelajaran(string namapel, string kodemapel);
16
17 void tampil_pelajaran(pelajaran pel);
18 #endif

h unguided2.h C++ unguided2.cpp X C++ unguided2-main.cpp
Struktur Data > Laprak 3 > unguided 2 > C++ unguided2.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 #include "unguided2.h"
6 using namespace std;
7
8 pelajaran create_pelajaran(string namapel, string kodemapel)
9 {
10     pelajaran pel;
11     pel.namaMapel = namapel;
12     pel.kodeMapel = kodemapel;
13     return pel;
14 }
15
16 void tampil_pelajaran(pelajaran pel)
17 {
18     cout << "Nama Mata Pelajaran: " << pel.namaMapel << endl;
19     cout << "Nilai: " << pel.kodeMapel << endl;
20 }
```

```
h unguided2.h  C++ unguided2.cpp  C++ unguided2-main.cpp x
Struktur Data > Laprak 3 > unguided 2 > C++ unguided2-main.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 #include "unguided2.h"
6 #include "unguided2.cpp"
7 using namespace std;
8
9 int main()
10 {
11     string namapel = "Struktur Data";
12     string kodemapel = "STD";
13     pelajaran pel = create_pelajaran(namapel, kodemapel);
14     tampil_pelajaran(pel);
15
16     return 0;
17 }
```

Screenshots Output

```
PS C:\programing\Struktur Data\Laprak 3\unguided 3> cd "c:\programing\Struktur Data\Laprak 3\unguided 2\" ; if ($?) { g++ unguided2-main.cpp -o unguided2-main } ; if ($?) { .\unguided2-main }
Nama Mata Pelajaran: Struktur Data
Nilai: STD
PS C:\programing\Struktur Data\Laprak 3\unguided 2> 
```

Deskripsi:

Program ini adalah contoh implementasi Abstract Data Type (ADT) sederhana untuk merepresentasikan sebuah pelajaran (mata pelajaran). Tujuannya adalah untuk menunjukkan cara mendefinisikan, membuat, dan menampilkan sebuah tipe data kustom.

Kodenya terstruktur secara modular menjadi tiga bagian utama:

1. File Header (.h) - Antarmuka (Interface):

Mendefinisikan struct pelajaran (namaMapel, kodeMapel).

Mendeklarasikan fungsi intinya: create_pelajaran (kreator) dan tampil_pelajaran.

2. File Implementasi (.cpp) - Badan Program (Body):

Berisi kode sumber yang menjelaskan bagaimana cara membuat dan menampilkan objek pelajaran.

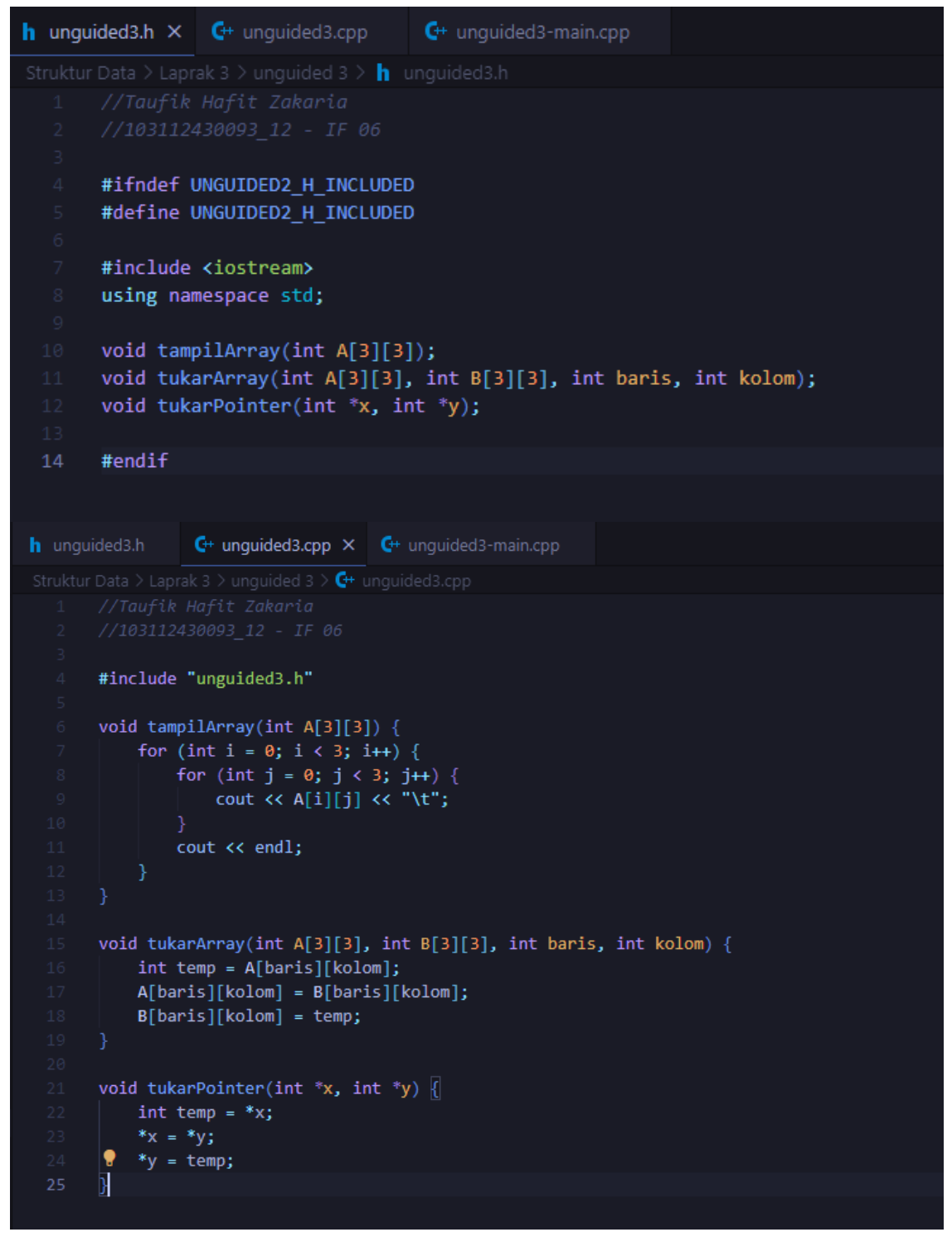
3. File Utama (main.cpp) - Program Pengguna (Driver):

Bertindak sebagai program driver yang menunjukkan cara menggunakan ADT:

membuat objek pelajaran dengan memanggil `create_pelajaran` lalu menampilkannya dengan `tampil_pelajaran`.

Secara singkat, program ini secara efektif mendemonstrasikan konsep dasar ADT, yaitu pemisahan antara definisi (antarmuka) dan implementasi (realisasi).

Unguided 3



```
h unguided3.h X C++ unguided3.cpp C++ unguided3-main.cpp
Struktur Data > Laprak 3 > unguided 3 > h unguided3.h
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #ifndef UNGUIDED2_H_INCLUDED
5 #define UNGUIDED2_H_INCLUDED
6
7 #include <iostream>
8 using namespace std;
9
10 void tampilArray(int A[3][3]);
11 void tukarArray(int A[3][3], int B[3][3], int baris, int kolom);
12 void tukarPointer(int *x, int *y);
13
14 #endif

h unguided3.h C++ unguided3.cpp X C++ unguided3-main.cpp
Struktur Data > Laprak 3 > unguided 3 > C++ unguided3.cpp
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include "unguided3.h"
5
6 void tampilArray(int A[3][3]) {
7     for (int i = 0; i < 3; i++) {
8         for (int j = 0; j < 3; j++) {
9             cout << A[i][j] << "\t";
10         }
11         cout << endl;
12     }
13 }
14
15 void tukarArray(int A[3][3], int B[3][3], int baris, int kolom) {
16     int temp = A[baris][kolom];
17     A[baris][kolom] = B[baris][kolom];
18     B[baris][kolom] = temp;
19 }
20
21 void tukarPointer(int *x, int *y) {
22     int temp = *x;
23     *x = *y;
24     *y = temp;
25 }
```

h unguided3.h

C++ unguided3.cpp

C++ unguided3-main.cpp X

Struktur Data > Laprak 3 > unguided 3 > C++ unguided3-main.cpp

```
1 //Taufik Hafit Zakaria
2 //103112430093_12 - IF 06
3
4 #include <iostream>
5 #include "unguided3.h"
6 #include "unguided3.cpp"
7 using namespace std;
8
9 int main() {
10     int A[3][3] = {{1,2,3},{4,5,6},{7,8,9}};
11     int B[3][3] = {{9,8,7},{6,5,4},{3,2,1}};
12     int *p1, *p2;
13     int a = 10, b = 20;
14
15     cout << "Array A sebelum ditukar:\n";
16     tampilArray(A);
17     cout << "\nArray B sebelum ditukar:\n";
18     tampilArray(B);
19
20     tukarArray(A, B, 1, 1);
21
22     cout << "\nArray A sesudah ditukar:\n";
23     tampilArray(A);
24     cout << "\nArray B sesudah ditukar:\n";
25     tampilArray(B);
26
27     p1 = &a;
28     p2 = &b;
29     cout << "\nSebelum tukar pointer: a=" << a << " b=" << b << endl;
30     tukarPointer(p1, p2);
31     cout << "Sesudah tukar pointer: a=" << a << " b=" << b << endl;
32
33     return 0;
34 }
```

Screenshots Output

```
PS C:\programming\Struktur Data\Laprak 3\unguided 3> cd "c:\programming\Struktur Data\Laprak 3\unguided 3\" ; if ($?) { g++ unguided3-main.cpp -o unguided3-main } ; if ($?) { .\unguided3-main }
Array A sebelum ditukar:
1      2      3
4      5      6
7      8      9

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

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

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

Sebelum tukar pointer: a=10 b=20
Sesudah tukar pointer: a=20 b=10
PS C:\programming\Struktur Data\Laprak 3\unguided 3> 
```

Deskripsi:

Program ini adalah sebuah demonstrasi yang menyoroti dua konsep fundamental dalam C++: manipulasi array 2D dan penggunaan pointer. Tujuannya adalah untuk menunjukkan bagaimana fungsi dapat memodifikasi data secara langsung, baik pada elemen array maupun pada variabel melalui alamat memorinya.

Struktur program dibagi menjadi tiga bagian logis:

1. File Header (.h) - Antarmuka (Interface):

Mendeklarasikan tiga fungsi: `tampilArray` untuk menampilkan matriks, `tukarArray` untuk menukar satu elemen antar dua array, dan `tukarPointer` untuk menukar nilai variabel menggunakan pointer.

2. File Implementasi (.cpp) - Badan Program (Body):

Berisi kode sumber (logika) dari ketiga fungsi tersebut. Implementasinya menggunakan loop untuk menampilkan data dan variabel sementara (`temp`) untuk melakukan operasi penukaran.

3. File Utama (main.cpp) - Program Pengguna (Driver):

Menginisialisasi dua array 2D dan dua variabel integer.

Memanggil fungsi-fungsi yang ada untuk melakukan operasi penukaran dan menampilkan hasilnya, dengan jelas menunjukkan kondisi "sebelum" dan "sesudah" untuk membuktikan bahwa data telah berhasil diubah.

D. Kesimpulan

Berdasarkan praktikum Modul 3 mengenai Abstract Data Type (ADT) yang telah dilaksanakan, dapat disimpulkan bahwa konsep ADT merupakan fondasi penting dalam membangun program yang terstruktur, modular, dan mudah dikelola. Melalui implementasi pada studi kasus data mahasiswa dan pelajaran, praktikan berhasil memahami cara memisahkan secara jelas antara antarmuka (interface) yang didefinisikan dalam file header (.h) dengan implementasi (body) yang direalisasikan dalam file (.cpp).

Pemisahan ini tidak hanya menerapkan prinsip enkapsulasi, di mana data dan operasi dibungkus menjadi satu kesatuan, tetapi juga penyembunyian informasi (information hiding), yang membuat program utama (driver) tidak perlu mengetahui detail teknis dari sebuah operasi. Selain itu, praktikum ini juga memberikan pemahaman praktis tentang penggunaan array of structs untuk mengelola sekumpulan data serta manipulasi data secara langsung melalui pointer, yang memperkuat pemahaman tentang manajemen memori dan pass-by-reference. Secara keseluruhan, penerapan ADT terbukti menghasilkan kode yang lebih bersih, mudah dibaca, dan dapat digunakan kembali (reusable).

E. Referensi

GeeksforGeeks. (2025). *Abstract Data Types*. Retrieved October 14, 2025, from <https://www.geeksforgeeks.org/dsa/abstract-data-types/>

CCBP. (2025, February 19). *Abstract Data Type in Data Structure*. Retrieved October 14, 2025, from <https://www.ccbp.in/blog/articles/abstract-data-type-in-data-structure>

Deepchecks. (2024, August 29). *What is Abstract Data Type? Benefits, Examples*. Retrieved October 14, 2025, from <https://www.deepchecks.com/glossary/abstract-data-type/>