

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

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



Disusun Oleh :

NAMA : Faris Walid Awwal Aidi

NIM : 103112430133

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

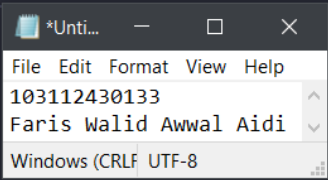
Abstract Data Type (ADT) adalah sebuah model matematis yang mendefinisikan suatu tipe data secara abstrak berdasarkan perilaku dan operasinya, tanpa terikat pada implementasi spesifik. Konsep ini berfokus pada apa yang dapat dilakukan oleh tipe data tersebut, bukan bagaimana cara melakukannya. Sebuah ADT terdiri dari dua komponen utama: definisi tipe data itu sendiri dan sekumpulan operasi dasar (primitif) yang dapat diterapkan padanya. Operasi-operasi ini dikelompokkan menjadi beberapa kategori, seperti konstruktor untuk menciptakan objek, selector untuk mengakses komponen nilai, mutator untuk mengubah nilai, serta operator relasional dan aritmatika untuk perbandingan dan perhitungan.

Dalam implementasinya, ADT mempromosikan modularitas dengan memisahkan spesifikasi dari realisasi. Spesifikasi, yang berisi definisi tipe dan deklarasi fungsi (prototipe), biasanya ditempatkan dalam file header (.h). Sementara itu, realisasi atau implementasi dari setiap fungsi dan prosedur ditulis dalam file source terpisah (.c atau .cpp). Pemisahan ini memungkinkan program utama (driver) untuk menggunakan ADT hanya dengan mengacu pada spesifikasinya, sehingga kode menjadi lebih terstruktur, mudah dikelola, dan implementasinya dapat diubah tanpa memengaruhi program yang menggunakannya.

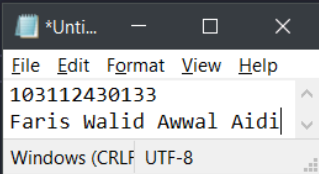
B. Guided (berisi screenshot source code & output program disertai penjelasannya)

1. Guided

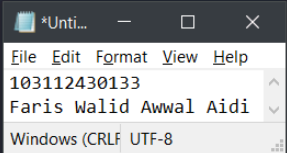
```
C mahasiswa.h > ...
1  #ifndef MAHASISWA_H_INCLUDED
2  #define MAHASISWA_H_INCLUDED
3  struct mahasiswa
4  {
5      char nim[10];
6      int nilai1, nilai2;
7  };
8  void inputMahasiswa(mahasiswa &m);
9  float rerata(mahasiswa m);
10 #endif
```



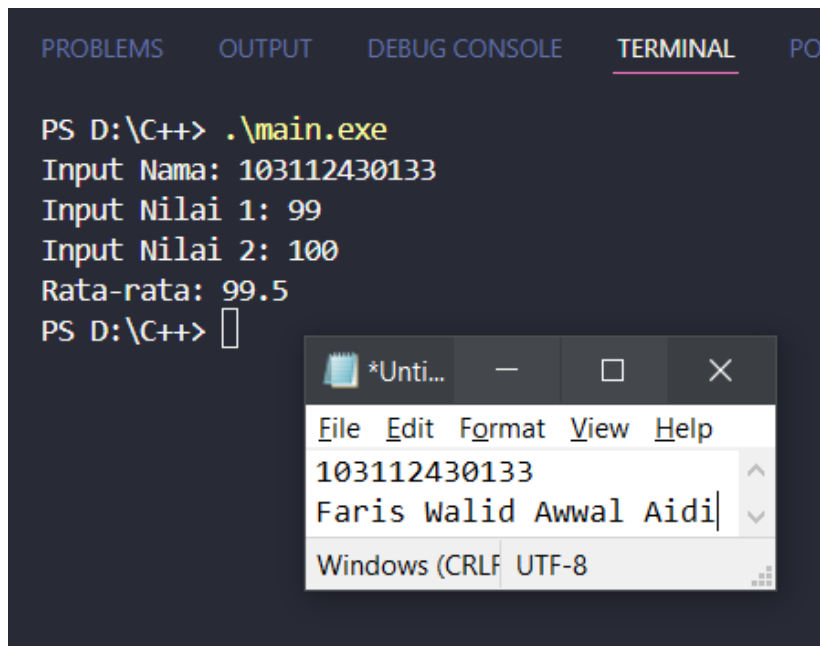
```
G mahasiswa.cpp > inputMahasiswa(mahasiswa &)
1  #include "mahasiswa.h"
2  #include <iostream>
3  using namespace std;
4
5  void inputMahasiswa(mahasiswa &m)
6  {
7      cout << "Input Nama: ";
8      cin >> (m.nim);
9      cout << "Input Nilai 1: ";
10     cin >> (m.nilai1);
11     cout << "Input Nilai 2: ";
12     cin >> (m.nilai2);
13 }
14
15 float rerata(mahasiswa m)
16 {
17     return float(m.nilai1 + m.nilai2) / 2;
18 }
```



```
G main.cpp > ...
1  #include <iostream>
2  #include "mahasiswa.h"
3  #include "mahasiswa.cpp"
4  using namespace std;
5
6  int main()
7  {
8      mahasiswa mhs;
9      inputMahasiswa(mhs);
10     cout << "Rata-rata: " << rerata(mhs) << endl;
11     return 0;
12 }
```



Screenshots Output



The screenshot shows a terminal window with the following output:

```
PS D:\C++> .\main.exe
Input Nama: 103112430133
Input Nilai 1: 99
Input Nilai 2: 100
Rata-rata: 99.5
PS D:\C++> 
```

Overlaid on the terminal is a standard Windows text editor window titled '*Unti...'. It contains the input data from the program:

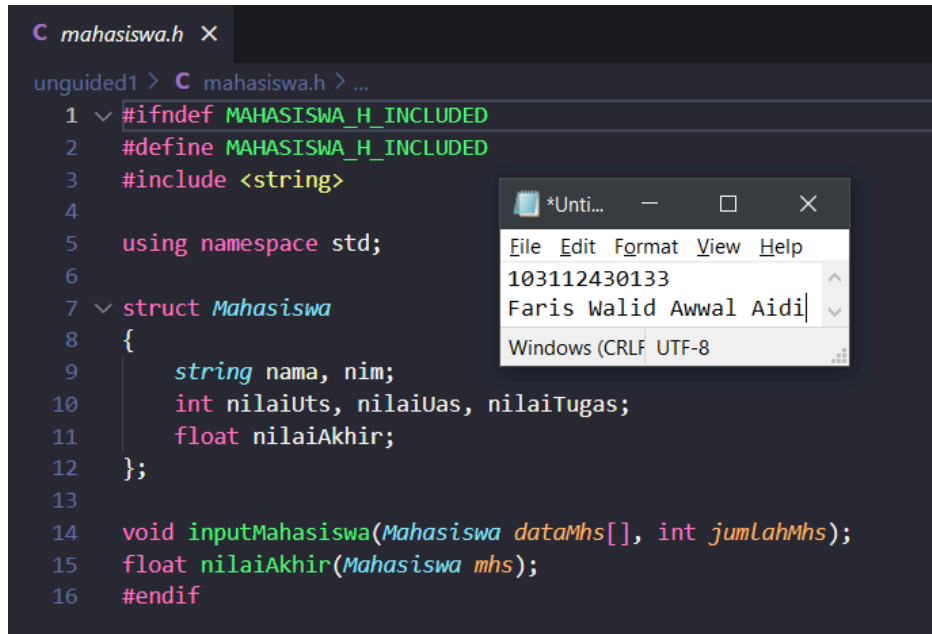
```
File Edit Format View Help
103112430133
Faris Walid Awwal Aidi
Windows (CRLF) UTF-8
```

Deskripsi:

Program ini mendemonstrasikan implementasi dasar Abstract Data Type (ADT) dengan memisahkan deklarasi struct dan fungsi ke dalam file header (mahasiswa.h). Program utama menggunakan fungsi dari file implementasi (mahasiswa.cpp) untuk meminta pengguna menginput NIM dan dua nilai, kemudian memanggil fungsi lain untuk menghitung dan menampilkan nilai rata-ratanya.

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

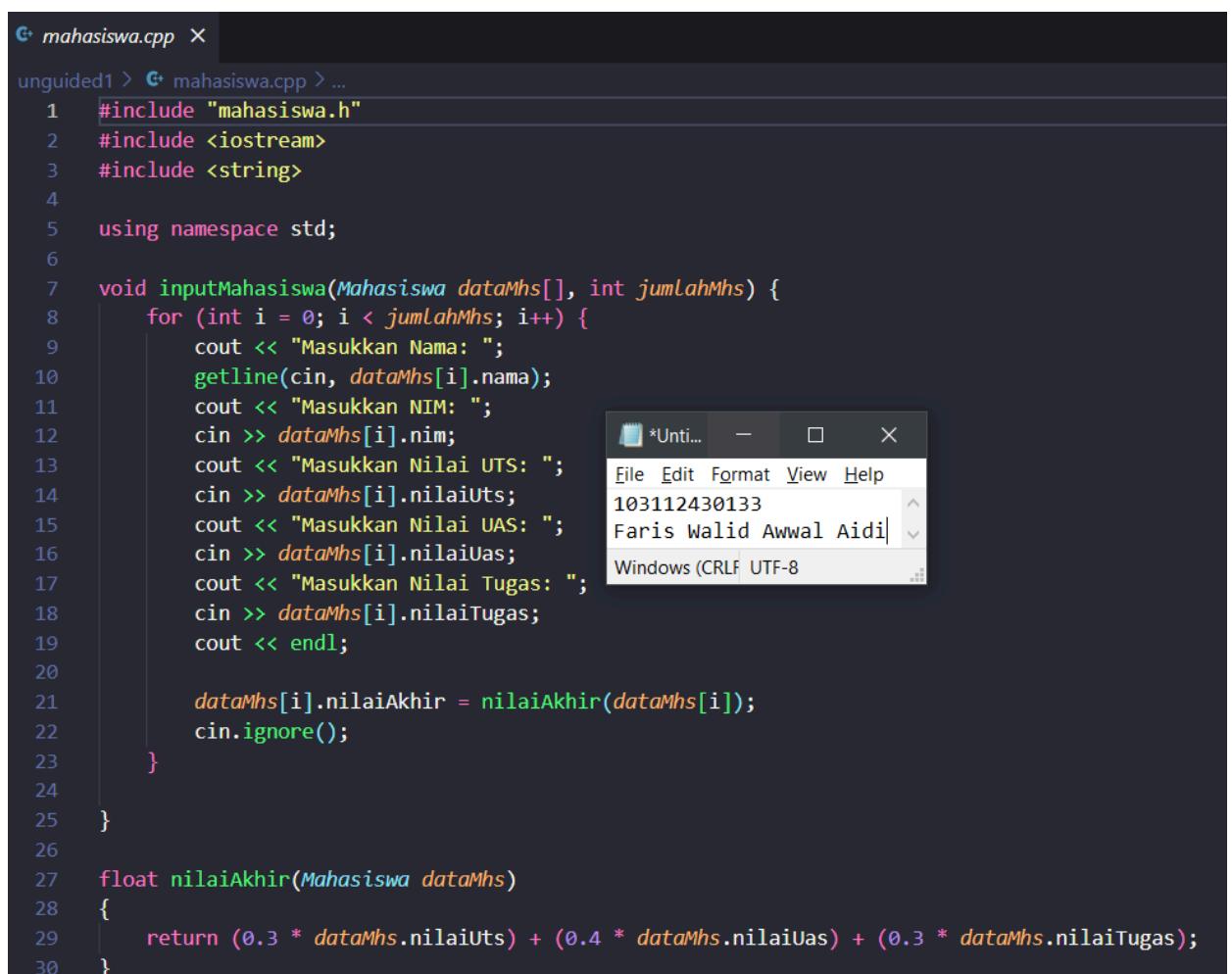
1. Unguided I



```
C mahasiswa.h X
unguided1 > C mahasiswa.h > ...
1  #ifndef MAHASISWA_H_INCLUDED
2  #define MAHASISWA_H_INCLUDED
3  #include <string>
4
5  using namespace std;
6
7  struct Mahasiswa
8  {
9      string nama, nim;
10     int nilaiUts, nilaiUas, nilaiTugas;
11     float nilaiAkhir;
12 };
13
14 void inputMahasiswa(Mahasiswa dataMhs[], int jumlahMhs);
15 float nilaiAkhir(Mahasiswa mhs);
16 #endif
```

Terminal window output:

```
*Unti...
File Edit Format View Help
103112430133
Faris Walid Awwal Aidi
Windows (CRLF) UTF-8
```



```
C mahasiswa.cpp X
unguided1 > C mahasiswa.cpp > ...
1  #include "mahasiswa.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  void inputMahasiswa(Mahasiswa dataMhs[], int jumlahMhs) {
8      for (int i = 0; i < jumlahMhs; i++) {
9          cout << "Masukkan Nama: ";
10         getline(cin, dataMhs[i].nama);
11         cout << "Masukkan NIM: ";
12         cin >> dataMhs[i].nim;
13         cout << "Masukkan Nilai UTS: ";
14         cin >> dataMhs[i].nilaiUts;
15         cout << "Masukkan Nilai UAS: ";
16         cin >> dataMhs[i].nilaiUas;
17         cout << "Masukkan Nilai Tugas: ";
18         cin >> dataMhs[i].nilaiTugas;
19         cout << endl;
20
21         dataMhs[i].nilaiAkhir = nilaiAkhir(dataMhs[i]);
22         cin.ignore();
23     }
24 }
25
26
27 float nilaiAkhir(Mahasiswa dataMhs)
28 {
29     return (0.3 * dataMhs.nilaiUts) + (0.4 * dataMhs.nilaiUas) + (0.3 * dataMhs.nilaiTugas);
30 }
```

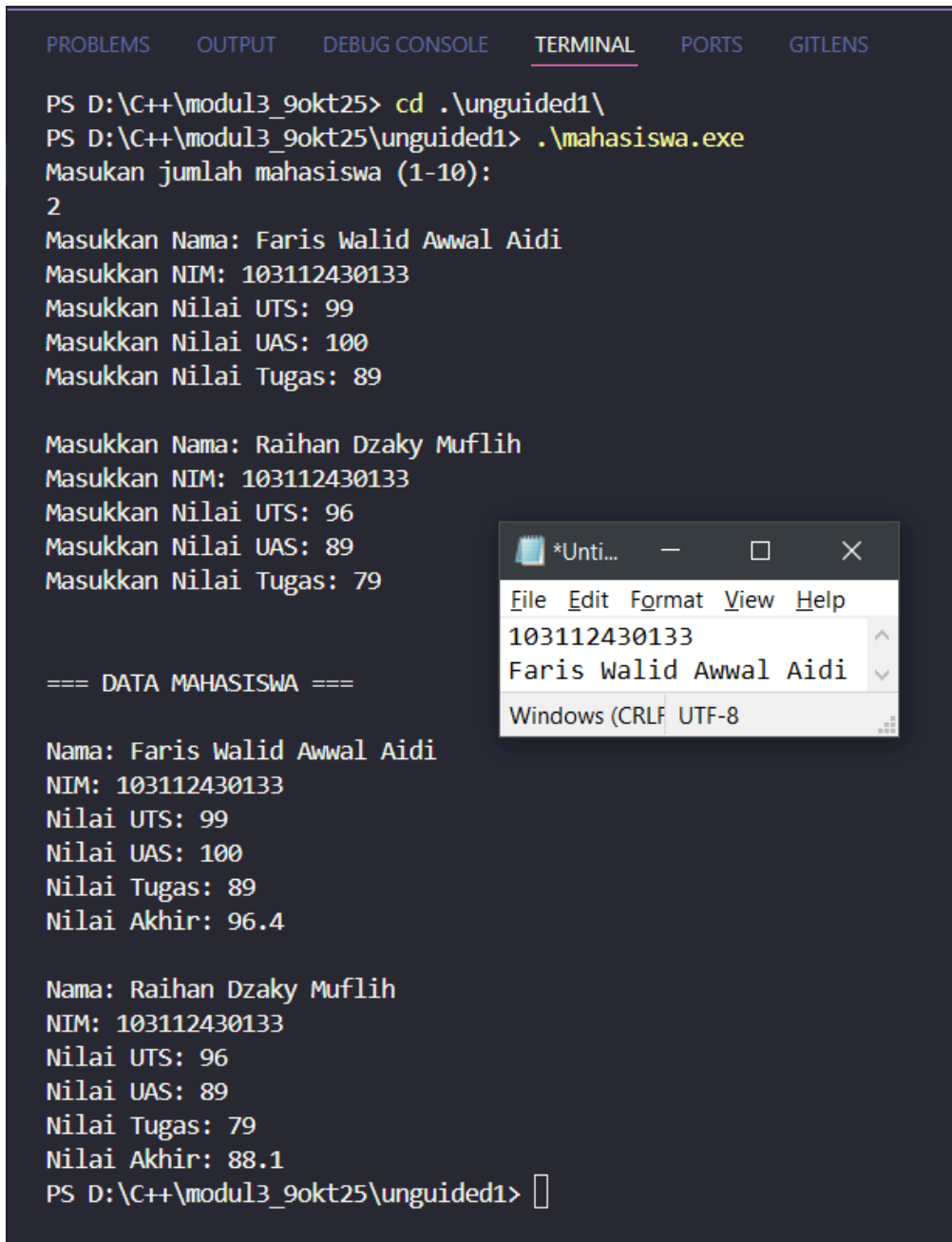
Terminal window output:

```
*Unti...
File Edit Format View Help
103112430133
Faris Walid Awwal Aidi
Windows (CRLF) UTF-8
```

```
main.cpp X
unguided1 > main.cpp > main()
1  #include "mahasiswa.h"
2  #include <iostream>
3
4  using namespace std;
5
6  int main()
7  {
8      Mahasiswa dataMhs[10];
9      int jumlahMhs;
10     cout << "Masukan jumlah mahasiswa (1-10): " << endl;
11     cin >> jumlahMhs;
12     if (jumlahMhs < 1 || jumlahMhs > 10) {
13         cout << "Jumlah mahasiswa harus antara 1 sampai 10." << endl;
14         return 1;
15     }
16
17     cin.ignore();
18     inputMahasiswa(dataMhs, jumlahMhs);
19     cout << "\n=== DATA MAHASISWA ===\n";
20     for (int i = 0; i < jumlahMhs; i++) {
21         cout << "\nNama: " << dataMhs[i].nama << endl;
22         cout << "NIM: " << dataMhs[i].nim << endl;
23         cout << "Nilai UTS: " << dataMhs[i].nilaiUts << endl;
24         cout << "Nilai UAS: " << dataMhs[i].nilaiUas << endl;
25         cout << "Nilai Tugas: " << dataMhs[i].nilaiTugas << endl;
26         cout << "Nilai Akhir: " << dataMhs[i].nilaiAkhir << endl;
27     }
28 }
```

*Unti...
File Edit Format View Help
103112430133
Faris Walid Awwal Aidi
Windows (CRLF) UTF-8

Screenshots Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS D:\C++\modul3_9okt25> cd .\unguided1\
PS D:\C++\modul3_9okt25\unguided1> .\mahasiswa.exe
Masukan jumlah mahasiswa (1-10):
2
Masukkan Nama: Faris Walid Awwal Aidi
Masukkan NIM: 103112430133
Masukkan Nilai UTS: 99
Masukkan Nilai UAS: 100
Masukkan Nilai Tugas: 89

Masukkan Nama: Raihan Dzaky Muflih
Masukkan NIM: 103112430133
Masukkan Nilai UTS: 96
Masukkan Nilai UAS: 89
Masukkan Nilai Tugas: 79

=== DATA MAHASISWA ===

Nama: Faris Walid Awwal Aidi
NIM: 103112430133
Nilai UTS: 99
Nilai UAS: 100
Nilai Tugas: 89
Nilai Akhir: 96.4

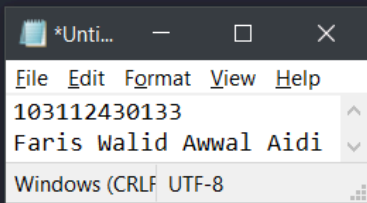
Nama: Raihan Dzaky Muflih
NIM: 103112430133
Nilai UTS: 96
Nilai UAS: 89
Nilai Tugas: 79
Nilai Akhir: 88.1
PS D:\C++\modul3_9okt25\unguided1> 
```

Deskripsi:

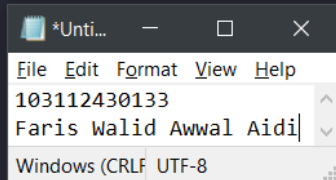
Program ini mengimplementasikan ADT untuk mengelola data beberapa mahasiswa. Dengan menggunakan array of struct, program meminta input nama, NIM, serta nilai UTS, UAS, dan Tugas untuk setiap mahasiswa. Selanjutnya, program secara otomatis menghitung nilai akhir berdasarkan bobot yang ditentukan dan menampilkan seluruh data lengkap dari setiap mahasiswa yang telah diinput.

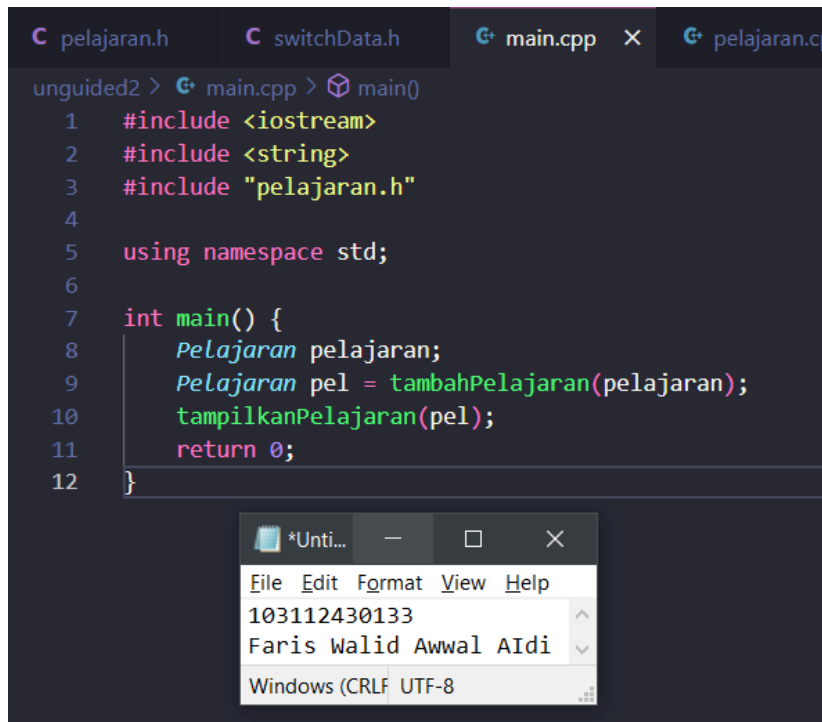
2. Unguided II

```
pelajaran.h X main.cpp pelajaran.cpp
unguided2 > C pelajaran.h > tampilkanPelajaran(Pelajaran)
1  #ifndef PELAJARAN_H_INCLUDED
2  #define PELAJARAN_H_INCLUDED
3  #include <string>
4
5  using namespace std;
6
7  struct Pelajaran {
8      string namaMapel, kodeMapel;
9  };
10
11  Pelajaran tambahPelajaran(Pelajaran inputPelajaran);
12  void tampilkanPelajaran(Pelajaran inputPelajaran);
13  #endif
```



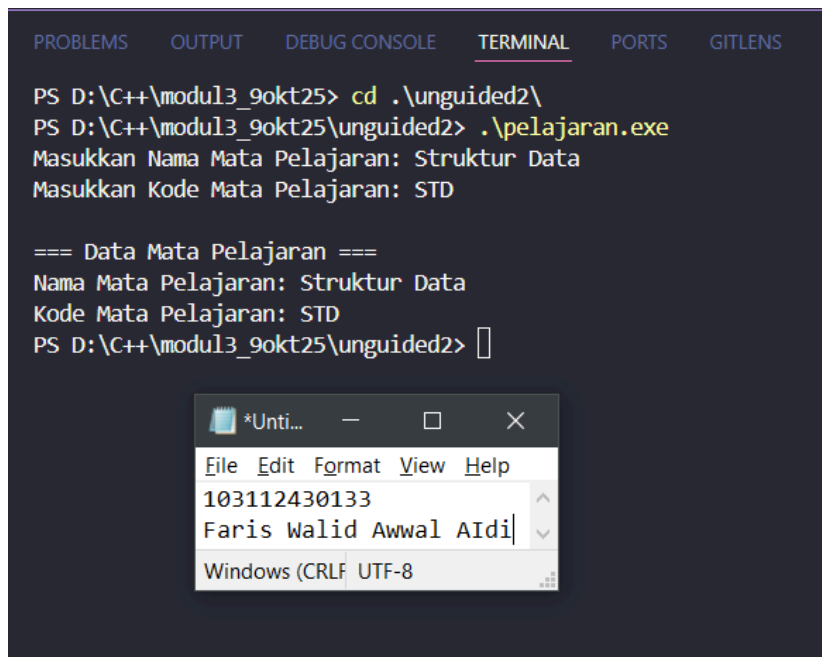
```
pelajaran.h main.cpp pelajaran.cpp X
unguided2 > pelajaran.cpp > tampilkanPelajaran(Pelajaran)
1  #include "pelajaran.h"
2  #include <iostream>
3  #include <string>
4
5  using namespace std;
6
7  Pelajaran tambahPelajaran(Pelajaran inputPelajaran) {
8      cout << "Masukkan Nama Mata Pelajaran: ";
9      getline(cin, inputPelajaran.namaMapel);
10     cout << "Masukkan Kode Mata Pelajaran: ";
11     getline(cin, inputPelajaran.kodeMapel);
12     return inputPelajaran;
13 }
14
15 void tampilkanPelajaran(Pelajaran inputPelajaran) {
16     cout << endl;
17     cout << "=== Data Mata Pelajaran ===" << endl;
18     cout << "Nama Mata Pelajaran: " << inputPelajaran.namaMapel << endl;
19     cout << "Kode Mata Pelajaran: " << inputPelajaran.kodeMapel << endl;
20 }
21
```





```
unguided2 > C main.cpp > main()
1  #include <iostream>
2  #include <string>
3  #include "pelajaran.h"
4
5  using namespace std;
6
7  int main() {
8      Pelajaran pelajaran;
9      Pelajaran pel = tambahPelajaran(pelajaran);
10     tampilkanPelajaran(pel);
11     return 0;
12 }
```

Screenshot Output



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

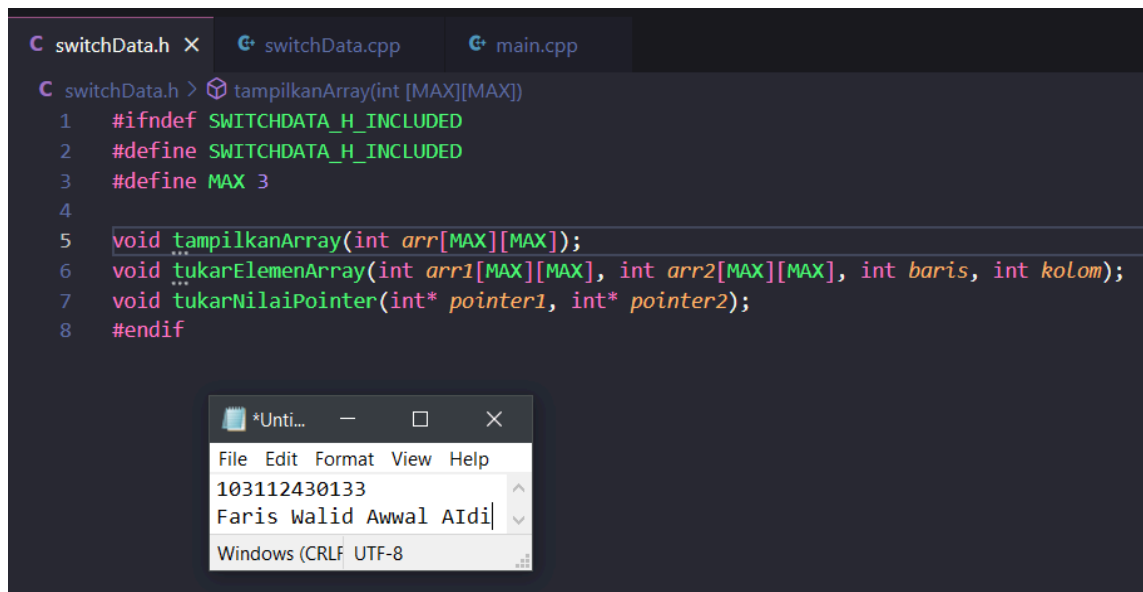
PS D:\C++\modul3_9okt25> cd .\unguided2\
PS D:\C++\modul3_9okt25\unguided2> .\pelajaran.exe
Masukkan Nama Mata Pelajaran: Struktur Data
Masukkan Kode Mata Pelajaran: STD

=== Data Mata Pelajaran ===
Nama Mata Pelajaran: Struktur Data
Kode Mata Pelajaran: STD
PS D:\C++\modul3_9okt25\unguided2> 
```

Deskripsi:

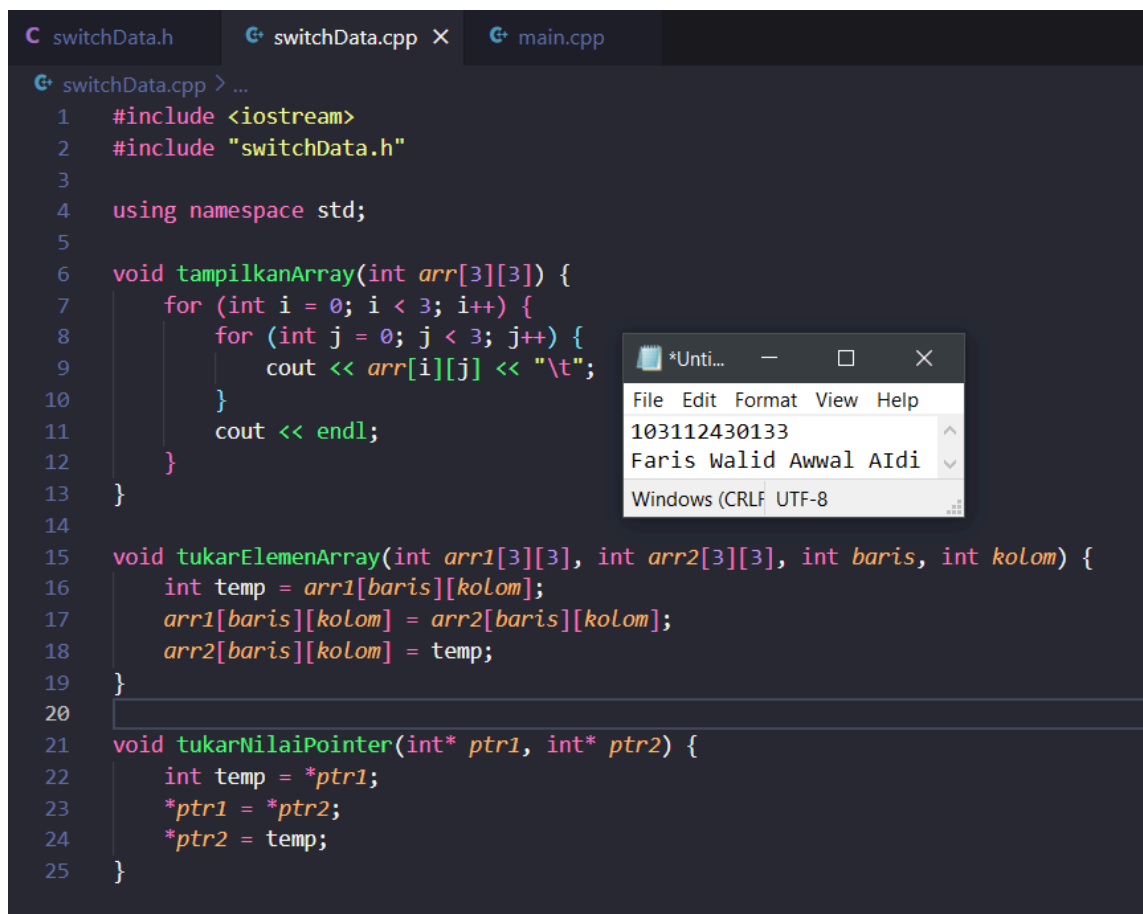
Program ini menerapkan konsep ADT untuk mengelola entitas mata pelajaran. Program memisahkan definisi struct dan deklarasi fungsi (.h) dari implementasinya (.cpp). Fungsi utamanya adalah untuk menginput nama dan kode mata pelajaran dari pengguna, yang kemudian datanya akan ditampilkan kembali ke layar menggunakan prosedur terpisah.

3. Unguided III



```
C switchData.h X switchData.cpp main.cpp
C switchData.h > tampilkanArray(int [MAX][MAX])
1  #ifndef SWITCHDATA_H_INCLUDED
2  #define SWITCHDATA_H_INCLUDED
3  #define MAX 3
4
5  void tampilkanArray(int arr[MAX][MAX]);
6  void tukarElemenArray(int arr1[MAX][MAX], int arr2[MAX][MAX], int baris, int kolom);
7  void tukarNilaiPointer(int* pointer1, int* pointer2);
8  #endif
```

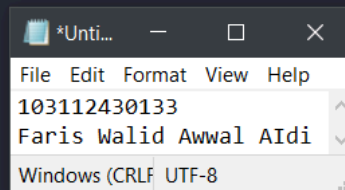
*Unti...
File Edit Format View Help
103112430133
Faris Walid Awwal AIdi
Windows (CRLF UTF-8



```
C switchData.h switchData.cpp X main.cpp
G switchData.cpp > ...
1  #include <iostream>
2  #include "switchData.h"
3
4  using namespace std;
5
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] << "\t";
10             }
11             cout << endl;
12         }
13     }
14
15     void tukarElemenArray(int arr1[3][3], int arr2[3][3], int baris, int kolom) {
16         int temp = arr1[baris][kolom];
17         arr1[baris][kolom] = arr2[baris][kolom];
18         arr2[baris][kolom] = temp;
19     }
20
21     void tukarNilaiPointer(int* ptr1, int* ptr2) {
22         int temp = *ptr1;
23         *ptr1 = *ptr2;
24         *ptr2 = temp;
25     }
```

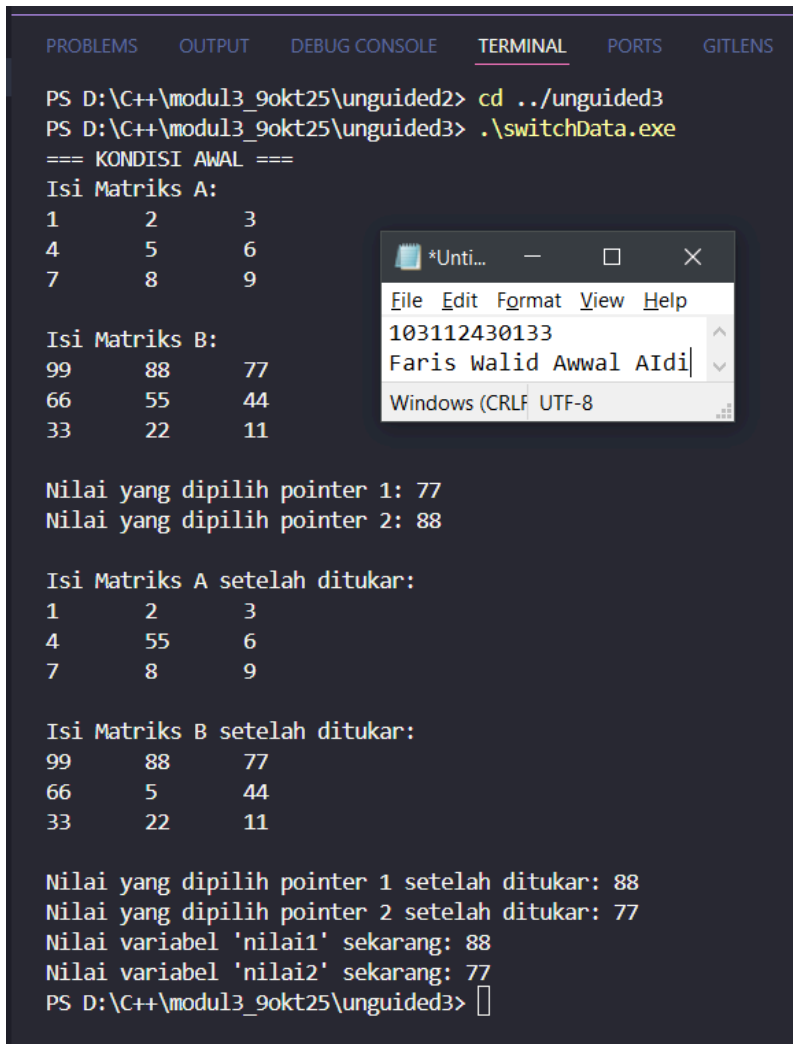
*Unti...
File Edit Format View Help
103112430133
Faris Walid Awwal AIdi
Windows (CRLF UTF-8

```
C switchData.h  G switchData.cpp  G main.cpp X
G main.cpp > ...
1  #include <iostream>
2  #include "switchData.h"
3  using namespace std;
4
5  int main() {
6      int matriksA[3][3] = {
7          {1, 2, 3},
8          {4, 5, 6},
9          {7, 8, 9}
10     };
11
12     int matriksB[3][3] = {
13         {99, 88, 77},
14         {66, 55, 44},
15         {33, 22, 11}
16     };
17
18     int nilai1 = 77;
19     int nilai2 = 88;
20     int* pointer1 = &nilai1;
21     int* pointer2 = &nilai2;
22
23     cout << "=== KONDISI AWAL ===" << endl;
24     cout << "Isi Matriks A:" << endl;
25     tampilkanArray(matriksA);
26     cout << "\nIsi Matriks B:" << endl;
27     tampilkanArray(matriksB);
28
29     cout << "\nNilai yang dipilih pointer 1: " << *pointer1 << endl;
30     cout << "Nilai yang dipilih pointer 2: " << *pointer2 << endl;
31
32     tukarElemenArray(matriksA, matriksB, 1, 1);
33
34     cout << "\nIsi Matriks A setelah ditukar:" << endl;
35     tampilkanArray(matriksA);
36     cout << "\nIsi Matriks B setelah ditukar:" << endl;
37     tampilkanArray(matriksB);
38
39     tukarNilaiPointer(pointer1, pointer2);
40
41     cout << "\nNilai yang dipilih pointer 1 setelah ditukar: " << *pointer1 << endl;
42     cout << "Nilai yang dipilih pointer 2 setelah ditukar: " << *pointer2 << endl;
43     cout << "Nilai variabel 'nilai1' sekarang: " << nilai1 << endl;
44     cout << "Nilai variabel 'nilai2' sekarang: " << nilai2 << endl;
45     return 0;
46 }
```



```
38
39     tukarNilaiPointer(pointer1, pointer2);
40
41     cout << "\nNilai yang dipilih pointer 1 setelah ditukar: " << *pointer1 << endl;
42     cout << "Nilai yang dipilih pointer 2 setelah ditukar: " << *pointer2 << endl;
43     cout << "Nilai variabel 'nilai1' sekarang: " << nilai1 << endl;
44     cout << "Nilai variabel 'nilai2' sekarang: " << nilai2 << endl;
45     return 0;
46 }
```

Screenshots Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

PS D:\C++\modul3_9okt25\unguided2> cd ../unguided3
PS D:\C++\modul3_9okt25\unguided3> .\switchData.exe
=== KONDISI AWAL ===
Isi Matriks A:
1      2      3
4      5      6
7      8      9

Isi Matriks B:
99     88     77
66     55     44
33     22     11

Nilai yang dipilih pointer 1: 77
Nilai yang dipilih pointer 2: 88

Isi Matriks A setelah ditukar:
1      2      3
4      55     6
7      8      9

Isi Matriks B setelah ditukar:
99     88     77
66     5      44
33     22     11

Nilai yang dipilih pointer 1 setelah ditukar: 88
Nilai yang dipilih pointer 2 setelah ditukar: 77
Nilai variabel 'nilai1' sekarang: 88
Nilai variabel 'nilai2' sekarang: 77
PS D:\C++\modul3_9okt25\unguided3> 
```

Deskripsi:

Program ini menerapkan ADT untuk melakukan serangkaian operasi manipulasi data. Program mendefinisikan dan mengimplementasikan beberapa prosedur, yaitu untuk menampilkan isi array 2D, menukar elemen pada posisi tertentu antara dua array, serta menukar nilai variabel yang ditunjuk oleh dua pointer. Program utama mendemonstrasikan penggunaan setiap fungsi tersebut secara berurutan.

D. Kesimpulan

Dari keseluruhan program yang telah dibuat, dapat disimpulkan bahwa penerapan Abstract Data Type (ADT) secara fundamental mengubah cara penulisan program menjadi lebih modular dan terstruktur. Dengan memisahkan spesifikasi berupa definisi struct dan deklarasi fungsi dalam file header (.h) dari implementasi kode dalam file sumber (.cpp), program menjadi lebih mudah dikelola dan dipahami. Praktikum ini menunjukkan bahwa baik untuk data sederhana maupun struktur yang lebih kompleks seperti data mahasiswa atau mata pelajaran, ADT memungkinkan program utama (driver) berinteraksi dengan data melalui operasi yang jelas tanpa perlu mengetahui detail internalnya.

Referensi

- Raharjo, Budi. 2025. *Buku Pemrograman C++ Mudah dan Cepat Menjadi Master C*.
- Wikipedia contributors. (2024, 8 Mei). C++. Wikipedia, Ensiklopedia Bebas. Diakses pada 2 Oktober 2025, dari <https://id.wikipedia.org/wiki/C%2B%2B>