

LAPORAN PRAKTIKUM
Modul 03
“ABSTRACT DATA TYPE (ADT)”



Disusun Oleh:
Ganesha Rahman Gibran -2211104058
Kelas S1SE-07-02

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

Tujuan

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

Landasan Teori

ABSTRACT DATA TYPE (ADT)

Abstract Data Type (ADT) adalah sebuah konsep dalam pemrograman yang mendefinisikan tipe data beserta operasi-operasi dasar yang dapat dilakukan terhadap tipe tersebut. ADT menekankan pada definisi statis, di mana struktur dan perilaku dari tipe data ditentukan tanpa memperhatikan implementasi spesifiknya

Definisi dan Komponen ADT

1. Type : Merupakan definisi dari tipe data yang ingin dibuat.
2. Primitif : Sekumpulan operasi dasar yang dapat dilakukan pada tipe tersebut
Termasuk :
 - Konstruktor/Kreator : membentuk nilai dari tipe. Biasanya diawali dengan 'Make'.
 - Selector : Mengakses komponen dari tipe, biasanya diawali dengan 'Get'.
 - Pengubah nilai : Mengubah nilai komponen, biasanya diawali dengan 'Set'.
 - Validator : Memastikan bahwa nilai yang dimasukkan sesuai dengan batasan tipe.
 - Destruktor/Dealokator : Menghapus objek dari memori.
 - Operasi Aritmatika dan Relasional : mendukung operasi matematis dan perbandingan antar objek.

Implementasi Contoh Waktu

```
TYPE Waktu = <HH: Jam, MM: Menit, SS: Detik>
```

Definisi Primitif untuk waktu

- Konstruktor

```
FUNCTION MakeWaktu(HH: integer, MM: integer, SS: integer) -> Waktu
```

- Selector

```
FUNCTION Jam(W: Waktu) -> integer  
FUNCTION Menit(W: Waktu) -> integer  
FUNCTION Detik(W: Waktu) -> integer
```

- Pengubah Nilai

```
PROCEDURE SetJam(VAR W: Waktu; HH: integer)
```

Modul ADT

1. Header File (.h) : Mendefinisikan struktur tipe dan deklarasi fungsi.
2. Body File (.c) : Berisi implementasi dari fungsi-fungsi yang dideklarasikan.

<pre>//Header File #ifndef WAKTU_H_INCLUDED #define WAKTU_H_INCLUDED struct Waktu { int jam; int menit; int detik; }; Waktu MakeWaktu(int HH, int MM, int SS); int Jam(Waktu w); int Menit(Waktu w); int Detik(Waktu w); void SetJam(Waktu &w, int HH); void TulisWaktu(Waktu w); #endif // WAKTU_H_INCLUDED</pre>	<pre>//main.cpp #include <iostream> #include "waktu.h" int main() { Waktu waktu = MakeWaktu(12, 30, 45); TulisWaktu(waktu); SetJam(waktu, 13); TulisWaktu(waktu); return 0; }</pre>
<pre>// Body File #include "waktu.h" #include <iostream> Waktu MakeWaktu(int HH, int MM, int SS) { Waktu w; w.jam = HH; w.menit = MM; w.detik = SS; return w; } int Jam(Waktu w) { return w.jam; }</pre>	

```
int Menit(Waktu w) {  
    return w.menit;  
}  
  
int Detik(Waktu w) {  
    return w.detik;  
}  
  
void SetJam(Waktu &w, int HH) {  
    w.jam = HH;  
}  
  
void TulisWaktu(Waktu w) {  
    std::cout << w.jam << ":" << w.menit  
    << ":" << w.detik << std::endl;  
}
```

Guided

Input :

```
Guided > G+ modul3_1.cpp > ...  
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 int main(){  
13     mahasiswa mhs;  
14     inputMhs(mhs);  
15     cout << "rata-rata = " << rata2(mhs);  
16     return 0;  
17 }  
18  
19  
20 void inputMhs(mahasiswa &m){  
21     cout << "Input nim= ";  
22     cin >> (m).nim;  
23     cout << "Input nilai= ";  
24     cin >> (m).nilai1;  
25     cout << "Input nilai= ";  
26     cin >> (m).nilai2;  
27 }  
28  
29 float rata2(mahasiswa m){  
30     return(m.nilai1 + m.nilai2)/2;  
31 }
```

Output :

```
ran_SE-06-02\03_Abstract_Data_Type\Guided\output'  
PS E:\Struktur Data\2211104058_Ganesha_Rahman_Gibran_SE-06-02\03_Abstract_Data_Type\Guided\output> & .\'  
Input nim= 2211104058  
Input nilai= 90  
Input nilai= 100  
rata-rata = 95  
PS E:\Struktur Data\2211104058_Ganesha_Rahman_Gibran_SE-06-02\03_Abstract_Data_Type\Guided\output> |
```

Unguided

1. Buat program yang dapat menyimpan data mahasiswa (max. 10) ke dalam sebuah array dengan field nama, nim, uts, uas, tugas, dan nilai akhir. Nilai akhir diperoleh dari FUNGSI dengan rumus $0.3 \times \text{uts} + 0.4 \times \text{uas} + 0.3 \times \text{tugas}$

Input :

```
#include <iostream>  
#include <string>  
using namespace std;  
  
struct Mahasiswa {  
    string nama;  
    string nim;  
    double uts;  
    double uas;  
    double tugas;  
    double nilai_akhir;  
};  
  
double hitungNilaiAkhir(double uts, double uas, double tugas) {  
    return 0.3 * uts + 0.4 * uas + 0.3 * tugas;  
}  
  
int main() {  
    const int MAX_MAHASISWA = 10;  
    Mahasiswa data_mahasiswa[MAX_MAHASISWA];  
  
    for (int i = 0; i < MAX_MAHASISWA; ++i) {  
        cout << "Masukkan data mahasiswa ke-" << i+1 << ":" << endl;  
        cout << "Nama: ";  
        cin >> data_mahasiswa[i].nama;  
        cout << "NIM: ";  
        cin >> data_mahasiswa[i].nim;  
        cout << "UTS: ";  
        cin >> data_mahasiswa[i].uts;  
        cout << "UAS: ";  
        cin >> data_mahasiswa[i].uas;  
        cout << "Tugas: ";  
        cin >> data_mahasiswa[i].tugas;
```

```

        data_mahasiswa[i].nilai_akhir = hitungNilaiAkhir(data_mahasiswa[i].uts, data_mahasiswa[i].uas, data_mahasiswa[i].tugas);
        cout << endl;
    }

    cout << "Data Mahasiswa:" << endl;
    for (int i = 0; i < MAX_MAHASISWA; ++i) {
        cout << "Mahasiswa ke-" << i+1 << ":" << endl;
        cout << "Nama: " << data_mahasiswa[i].nama << endl;
        cout << "NIM: " << data_mahasiswa[i].nim << endl;
        cout << "UTS: " << data_mahasiswa[i].uts << endl;
        cout << "UAS: " << data_mahasiswa[i].uas << endl;
        cout << "Tugas: " << data_mahasiswa[i].tugas << endl;
        cout << "Nilai Akhir: " << data_mahasiswa[i].nilai_akhir << endl;
        cout << endl;
    }

    return 0;
}

```

Output :

```

PS E:\Struktur Data\2211104058_Ganesha_Rahman_Gibran_SE-06-02\03_Abstract_Data_Type>
ran_SE-06-02\03_Abstract_Data_Type\Unguided\output'
PS E:\Struktur Data\2211104058_Ganesha_Rahman_Gibran_SE-06-02\03_Abstract_Data_Type>
Masukkan data mahasiswa ke-1:
Nama: Ganesha_Rahman_Gibran
NIM: 2211104058
UTS: 90
UAS: 100
Tugas: 95

Masukkan data mahasiswa ke-2:
Nama: 

```

2. Buatlah ADT pelajaran sebagai berikut di dalam file “pelajaran.h”:

```

tipe pelajaran <
    namaMapel : string
    kodeMapel : string
>
fungsi create_pelajaran( namapel : string, kodepel :
    string ) →
    pelajaran
prosedur tampil_pelajaran( pel : pelajaran )

```

Buatlah implementasi ADT pelajaran pada file “pelajaran.cpp”

Cobalah hasil implementasi ADT pada file “main.cpp”

```

using namespace std;
int main(){
    string namapel = "Struktur Data";
    string kodepel = "STD";
    pelajaran pel = create_pelajaran(namapel,kodepel);
    tampil_pelajaran(pel);
    return 0;
}

```

Contoh Output :

```
nama pelajaran : Struktur Data
nilai : STD
```

Input :

```
main.cpp U x pelajaran.h U pelajaran.cpp U
03_Abstract_Data_Type > Unguided > main.cpp > ...
1  #include "pelajaran.h"
2  #include <iostream>
3  using namespace std;
4
5  int main() {
6      string namapel = "Struktur Data";
7      string kodepel = "STD";
8
9      pelajaran pel = create_pelajaran(namapel, kodepel);
10
11     tampil_pelajaran(pel);
12
13     return 0;
14 }
15
```

```
main.cpp U pelajaran.h U x pelajaran.cpp U
03_Abstract_Data_Type > Unguided > pelajaran.h > ...
1  #ifndef PELAJARAN_H
2  #define PELAJARAN_H
3
4  #include <string>
5  using namespace std;
6
7  struct pelajaran {
8      string namaMapel;
9      string kodeMapel;
10 };
11
12 pelajaran create_pelajaran(string namapel, string kodepel);
13
14 void tampil_pelajaran(pelajaran pel);
15
16 #endif
```

```
main.cpp U pelajaran.h U pelajaran.cpp U x
03_Abstract_Data_Type > Unguided > pelajaran.cpp > ...
1  #include "pelajaran.h"
2  #include <iostream>
3
4  pelajaran create_pelajaran(string namapel, string kodepel) {
5      pelajaran pel;
6      pel.namaMapel = namapel;
7      pel.kodeMapel = kodepel;
8      return pel;
9  }
10
11 void tampil_pelajaran(pelajaran pel) {
12     cout << "Nama Mata Pelajaran: " << pel.namaMapel << endl;
13     cout << "Kode Mata Pelajaran: " << pel.kodeMapel << endl;
14 }
```

Output :

Nama Mata Pelajaran: **Struktur Data**
Kode Mata Pelajaran: **STD**

3. Buatlah program dengan ketentuan :

- 2 buah array 2D integer berukuran 3x3 dan 2 buah pointer integer
- Fungsi/prosedur yang menampilkan isi sebuah array integer 2D
- Fungsi/prosedur yang akan menukarkan isi dari 2 array integer 2D pada posisi tertentu STRUKTUR DATA 46
- Fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah pointer

Input :

```
#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 tukarIsiArray(int arr1[3][3], int arr2[3][3], int row, int col) {
    int temp = arr1[row][col];
    arr1[row][col] = arr2[row][col];
    arr2[row][col] = temp;
}

void tukarIsiPointer(int* ptr1, int* ptr2) {
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}

int main() {
    int array1[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };
    int array2[3][3] = { {9, 8, 7}, {6, 5, 4}, {3, 2, 1} };

    int a = 10;
    int b = 20;
    int* ptr1 = &a;
    int* ptr2 = &b;

    cout << "Array 1 sebelum ditukar:" << endl;
```



```
tampilArray(array1);
cout << "Array 2 sebelum ditukar:" << endl;
tampilArray(array2);

tukarIsiArray(array1, array2, 1, 1);

cout << "\nArray 1 setelah ditukar pada posisi [1][1]:" << endl;
tampilArray(array1);
cout << "Array 2 setelah ditukar pada posisi [1][1]:" << endl;
tampilArray(array2);

cout << "\nNilai sebelum pointer ditukar:" << endl;
cout << "Nilai ptr1 (a): " << *ptr1 << endl;
cout << "Nilai ptr2 (b): " << *ptr2 << endl;

tukarIsiPointer(ptr1, ptr2);

cout << "\nNilai setelah pointer ditukar:" << endl;
cout << "Nilai ptr1 (a): " << *ptr1 << endl;
cout << "Nilai ptr2 (b): " << *ptr2 << endl;

return 0;
```

Output :

```
'--interpreter=mi'
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 pada posisi [1][1]:
1 2 3
4 5 6
7 8 9
Array 2 setelah ditukar pada posisi [1][1]:
9 8 7
6 5 4
3 2 1

Nilai sebelum pointer ditukar:
Nilai ptr1 (a): 10
Nilai ptr2 (b): 20

Nilai setelah pointer ditukar:
Nilai ptr1 (a): 20
Nilai ptr2 (b): 10
PS E:\Struktur Data\2211104058_Ganesha_Rahman_Gibran_SE-06-02>
```

Kesimpulan

ADT memberikan kerangka kerja yang jelas untuk mendefinisikan tipe data dan operasi yang dapat dilakukan terhadapnya. Dengan memisahkan definisi tipe dan implementasinya ke dalam modul-modul terpisah, kita dapat meningkatkan keterbacaan dan pemeliharaan kode