

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN ALGORITMA**

**MODUL I
“TIPE DATA”**



Disusun Oleh :

Nama : Fitri Kusumaningtyas
NIM : 2311102068
Kelas : IF 11 B

DOSEN:

WAHYU ANDI SAPUTRA, S.Pd. , M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

A. Dasar Teori

Tipe data ialah sebuah tipe yang menandakan jenis data apa yang ingin disimpan dengan menyesuaikan kegunaan memori penyimpanan, karena data bisa bermacam bentuk isinya seperti huruf, angka, karakter, dan lain-lain. Maka dari itu tipe data sangat dibutuhkan dalam pemrograman. Umumnya tipe data digunakan untuk membuat variabel yang nanti akan mewakili data dari macam-macam jenis tipe data di dalam pemrograman C++. Berikut ini beberapa tipe data yang ada dalam Bahasa pemrograman C++.

1. Void (kosong)

Tipe data yang tidak memiliki nilai dan juga tidak mengembalikan nilai, biasanya digunakan untuk tipe data kosong seperti membuat fungsi.

2. Int (integer)

Tipe data ini berisikan angka yang berupa bilangan bulat.

3. Char (karakter)

Tipe data ini digunakan untuk karakter yang sering digunakan untuk tipe data yang berupa huruf atau angka, tapi lebih sering berupa huruf. Cont : 'L'.

4. Float (floating point)

Tipe data numerik yang digunakan untuk data berupa angka pecahan (decimal).

5. Double (double floating point)

Tipe data numerik yang digunakan untuk data berupa angka pecahan yang desimalnya lebih banyak maupun bilangan bulat.

6. Bool (Boolean)

Tipe data yang digunakan untuk menentukan true atau false, tipe data ini sama seperti bilangan biner hanya dua angka saja yaitu 0 dan 1.

7. Wchar_t

Tipe data yang hampir sama seperti char akan tetapi biasanya digunakan untuk menyimpan karakter yang tidak ada huruf abjad misal nya huruf jepang, china, korea, dll.

Tipe data abstrak atau yang biasa disebut Abstrak Data Tipe(ADT)

merupakan tipe data yang dibentuk oleh programmer itu sendiri. Pada tipe data abstrak bisa berisi banyak tipe data, jadi nilainya bisa lebih dari satu dan beragam tipe data. Fitur Class adalah fitur Object Oriented Program(OPP) pada bahasa C++ yang mirip dengan fitur data structures Struct pada bahasa C. Keduanya berfungsi untuk membungkus tipe data di dalamnya sebagai anggota. menurut learn.microsoft.com perbedaan antara Praktikum Struktur Data dan Algoritma 2 Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class bersifat private.

Tipe Data Koleksi Tipe data koleksi (Collection Data Type)

tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan. Tipe data koleksi memungkinkan Anda menyimpan, mengelola, dan mengakses sejumlah besar data dengan cara yang terstruktur. Ada beberapa tipe data koleksi yang umum digunakan dalam pemrograman, dan di antaranya adalah:

a. Array

Array adalah struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama. Elemen-elemen tersebut dapat diakses dengan menggunakan indeks. Array memiliki ukuran tetap yang ditentukan saat deklarasi.

b. Vector

Vector adalah Standard Template Library (STL) jika di dalam C/C++ memiliki bentuk `std::vector` . Umumnya, vector mirip seperti array yang memiliki kemampuan untuk menyimpan data dalam bentuk elemen-elemen yang alokasi memorinya dilakukan otomatis dan bersebelahan. Kemampuan vector bukan hanya pada jumlah elemen yang dinamis, vector pada C/C++ juga dilengkapi dengan fitur-fitur pelengkap seperti element access, iterators, capacity, modifiers

c. Map

Map terasa mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer. Pada map, indeks tersebut diberi nama

“key”. Pada `std::map` digunakan Self-Balancing Tree khususnya Red-Black Tree.

B. Guided

1. Guided 1

Source code :

```
#include <iostream>
using namespace std;
//main program
int main(){
    char op;
    float num1, num2;

    //it allows user to enter operator i,e,+, -,*,/
    cout << "Masukkan operator (+,-,*,/) : ";
    cin >> op;
    //it allow user to enter the operands
    cout << "Masukkan angka pertama : ";
    cin >> num1 ;
    cout << "Masukkan angka kedua   : ";
    cin >> num2 ;

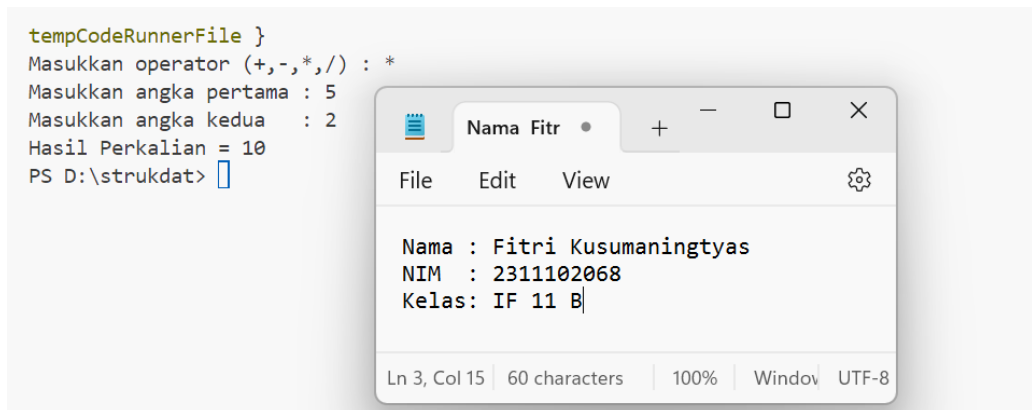
    switch (op){
        //if user enter+
        case '+' :
            cout << "Hasil Penjumlahan = " << num1+ num2;
            break;
        //if user enter
        case '-':
            cout << "Hasil Pengurangan = " << num1- num2;
            break;
        //if user enter *
        case '*':
            cout << "Hasil Perkalian = " << num1*num2;
            break;
        //if user enter /
        case '/':
            if (num2 !=0)
                cout << "Hasil Pembagian = " << num1/num2;
            else
                cout << "Error! Tidak dapat melakukan Pembagian dengan nol.
" ;
            break;
        //if the operator is other than +,-,*, or /
        //error message will diplay
        default:
```

```

        cout << "Error! operator tidak benar." ;
    }
    return 0;
}

```

Output :



Deskripsi program :

Program ini bertujuan untuk melakukan operasi bilangan sederhana. Dengan menggunakan variabel “op” dalam bentuk tipe data “char” dan variabel “num 1”, “num2” dalam bentuk tipe data “integer”. Kode mengizinkan user untuk memasukkan operator matematika (+,-,*,/) dan dua bilangan bulat. . kode ini menggunakan pernyataan “switch”. Untuk menentukan operasi bilangan apa yang harus dilakukan itu tergantung pada operator yang dimasukkan pengguna. Setelah peng-inputan operator, pengguna diminta utk memasukkan nilai untuk 2 bilangan bulat. Setelah kode berjalan maka nanti akan menghasilkan nilai dari operasi bilangan ke layar.

2. Guided 2

Source code :

```

#include <stdio.h>

//struk
struct Mahasiswa{
    const char *name;

```

```

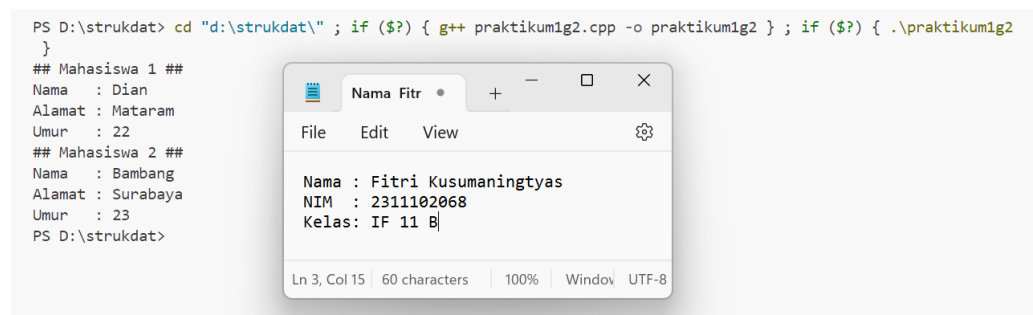
    const char *address;
    int age;
};

int main(){
    //menggunakan struct
    struct Mahasiswa mhs1, mhs2;
    // mengisi nilai ke struct
    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;
    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;

    //mencetak isi struct
    printf ("## Mahasiswa 1 ##\n");
    printf ("Nama   : %s\n", mhs1.name);
    printf ("Alamat  : %s\n", mhs1.address);
    printf ("Umur    : %d\n", mhs1.age);
    printf ("## Mahasiswa 2 ##\n");
    printf ("Nama   : %s\n", mhs2.name);
    printf ("Alamat  : %s\n", mhs2.address);
    printf ("Umur    : %d\n", mhs2.age);
    return 0;
}

```

Output :



```

PS D:\struktat> cd "d:\struktat\" ; if ($?) { g++ praktikum1g2.cpp -o praktikum1g2 } ; if ($?) { .\praktikum1g2 }
## Mahasiswa 1 ##
Nama   : Dian
Alamat : Mataram
Umur   : 22
## Mahasiswa 2 ##
Nama   : Bambang
Alamat : Surabaya
Umur   : 23
PS D:\struktat>

```

Deskripsi program :

Program diatas menggunakan fungsi “struct”. Struct ini bernama “mahasiswa” yang memiliki tiga struktur “nama”, “alamat”, dan “umur” . dalam fungsi utama

,kode ini mendeklarasikan dua variabel “mhs1” dan “mhs2”. Kode ini mengisi nilai ke “mhs1” dan “mhs2”. Kemudian kode ini mencetak informasi tentang “mhs1” dan “mhs2” menggunakan fungsi “printf ()”.

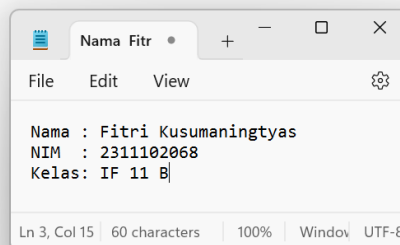
3. Guided 3

Source code

```
#include <iostream>
using namespace std;
int main()
{
    //deklarasi dan inisialisasi array
    int nilai[5];
    nilai[0] = 23;
    nilai[1] = 50;
    nilai[2] = 34;
    nilai[3] = 78;
    nilai[4] = 90;
    //mencetak array
    cout << "Isi array pertama :" << nilai[0] << endl;
    cout << "Isi array kedua :" << nilai[1] << endl;
    cout << "Isi array ketiga :" << nilai[2] << endl;
    cout << "Isi array keempat :" << nilai[3] << endl;
    cout << "Isi array kelima :" << nilai[4] << endl;
    return 0;
}
```

Output :

```
PS D:\strukdat> cd "d:\strukdat\" ; if ($?) { g++ praktikum1g3.cpp -o praktikum1g3 } ; if ($?) { .\praktikum1g3
}
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS D:\strukdat> 
```



Deskripsi program :

Program ini menggunakan array dengan mendeklarsikan dan menginisialisasi sebuah array dalam bentuk “integer”, kemudian mencetak nilai dari setiap elemen dalam array ke layar.

C. Unguided

1. Unguided 1

Source code :

```
#include <iostream>
using namespace std;

int luaspersegi panjang (int p , int l){
    return p * l;
}

double luaslingkaran ( int r){
    double pi = 3.14;
    return pi * (r*r);
}

int main (){

    int p;
    int l;
    int r;

    cout << "Masukkan nilai p : " ;
    cin >> p;
    cout << "Masukkan nilai l : " ;
    cin >> l;
    cout << "Masukkan nilai r : " ;
    cin >> r;

    cout << "LUAS PERMUKAAN TABUNG = " << luaspersegi panjang
(p,l) + (2 * luaslingkaran (r)) << " cm" << endl;

    return 0;
}
```

Output :

```

PS D:\strukdat> cd "d:\strukdat\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\
tempCodeRunnerFile }
Masukkan nilai p : 8
Masukkan nilai l : 4
Masukkan nilai r : 5
LUAS PERMUKAAN TABUNG = 189 cm
PS D:\strukdat>

```

Nama : Fitri Kusumaningtyas
NIM : 2311102068
Kelas: IF 11 B

Deskripsi program :

Program ini menggunakan 2 fungsi dengan menggunakan tipe data primitive yaitu “integer” dan “double” . Program ini berfungsi menghitung “luas permukaan tabung” dengan memanfaatkan inputan dari user untuk setiap variabel “p” , “l” , “r” yang dibutuhkan dalam setiap fungsi. Program akan menjalankan kode lalu menampilkan hasil dari penghitungan tersebut ke layar.

2. Unguided 2

Perbedaan struct dan class yaitu :

Struct	Class
Anggota struct bersifat public secara default	Anggota class bersifat pribadi secara default tetapi tidak semua Bahasa pemograman memiliki perilaku default ini, misalnya java dll
Sebuah instance dari struct disebut “variabel struktur”.	Sebuah instance dari suatu kelas disebut “objek”
Biasanya digunakan untuk pengelompokkan data	Biasanya digunakan untuk abstraksi data dan pewarisan lebih lanjut.
Dinyatakan menggunakan kata kunci “struct”	Dinyatakan menggunakan kata kunci “class”
Nilai NULL tidak dimungkinkan di struct	Nilai NULL dimungkinkan di class

Dibawah ini adalah contoh penggunaan dari “struct” dan “class”

a. Struct

Source code :

```
#include <iostream>
using namespace std;

//struct
struct buah {
    string namabuah;
    double berat;
    double harga;
};

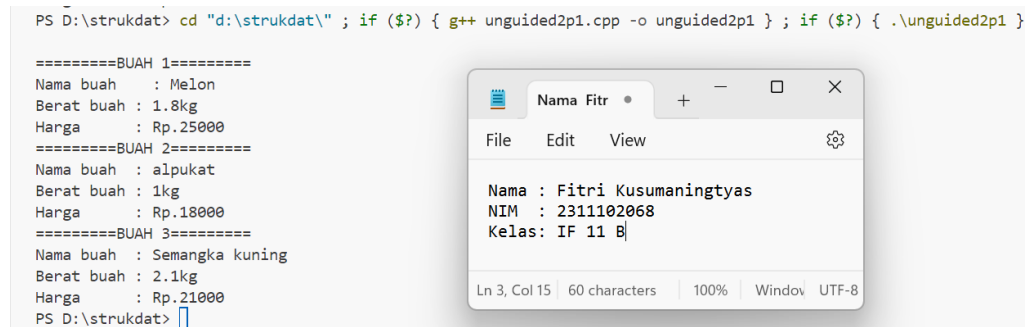
int main (){
    struct buah buah1,buah2,buah3;

    buah1.namabuah = "Melon";
    buah1.berat = 1.8;
    buah1.harga = 25000;
    buah2.namabuah = "alpukat";
    buah2.berat = 1;
    buah2.harga = 18000;
    buah3.namabuah= "Semangka kuning";
    buah3.berat= 2.1;
    buah3.harga= 21000;

    cout << "\n=====BUAH 1======" << endl;
    cout << "Nama buah      : " << buah1.namabuah;
    cout << "\nBerat buah : " << buah1.berat << "kg";
    cout << "\nHarga        : Rp." << buah1.harga;
    cout << "\n=====BUAH 2======" << endl;
    cout << "Nama buah : " << buah2.namabuah ;
    cout << "\nBerat buah : " << buah2.berat << "kg";
    cout << "\nHarga        : Rp." << buah2.harga;
    cout << "\n=====BUAH 3======" << endl;
    cout << "Nama buah : " << buah3.namabuah;
    cout << "\nBerat buah : " << buah3.berat << "kg";
    cout << "\nHarga        : Rp." << buah3.harga;

    return 0;
}
```

Output :



The screenshot shows two windows. The terminal window on the left displays the execution of a C++ program. It runs the command `cd "d:\strukdat\" ; if ($?) { g++ unguided2p1.cpp -o unguided2p1 } ; if ($?) { .\unguided2p1 }`. The output lists details for three fruit objects: BUAH 1 (Melon, 1.8kg, Rp.25000), BUAH 2 (alpukat, 1kg, Rp.18000), and BUAH 3 (Semangka kuning, 2.1kg, Rp.21000). The Notepad++ window on the right, titled 'Nama Fitr', shows the source code of the program, which defines a `hewan` class with `nama` and `berat` attributes and `getnama` and `getberat` methods.

```
PS D:\strukdat> cd "d:\strukdat\" ; if ($?) { g++ unguided2p1.cpp -o unguided2p1 } ; if ($?) { .\unguided2p1 }

=====BUAH 1=====
Nama buah   : Melon
Berat buah  : 1.8kg
Harga       : Rp.25000
=====BUAH 2=====
Nama buah   : alpukat
Berat buah  : 1kg
Harga       : Rp.18000
=====BUAH 3=====
Nama buah   : Semangka kuning
Berat buah  : 2.1kg
Harga       : Rp.21000
PS D:\strukdat>
```

```
#include <iostream>
using namespace std;

//class
class hewan {
private :
    string nama;
    double berat;

public:
    hewan (string a,double b){
        nama  = a;
        berat = b;
    }
    string getnama (){
        return nama;
    }
    double getberat(){
        return berat;
    }
}
```

Deskripsi program :

Program ini menggambarkan fungsi “struct”. Dalam program ini “struct” didefinisikan dengan “buah”, dengan 3 variabel yaitu “namabuah”, “ berat”, dan “harga”. Di dalam fungsi utama, tiga onjek dari “struct buah” dideklarasikan menjadi “buah1”,”buah2”,”buah3”. Kemudian variabel dari setiap objek di inisialisasi dengan nilai. Program kemudian mencetak detail dari setiap objek ke layar.

b. Class

Source code :

```
#include <iostream>
using namespace std;

//class
class hewan {
private :
    string nama;
    double berat;

public:
    hewan (string a,double b){
        nama  = a;
        berat = b;
    }
    string getnama (){
        return nama;
    }
    double getberat(){
        return berat;
    }
}
```

```

    }

    void display (){
        cout << "Nama hewan : "<< nama<<endl;
        cout << "Berat      : "<< berat<< "kg"<<endl;
    }
};

int main (){
    hewan kucing ("kucing", 3.8);
    kucing.display ();

    return 0;
}

```

Output :

The screenshot shows a Windows command prompt window with the following text:

```

PS D:\strukdat> cd "d:\strukdat\" ; if ($?) { g++ unguided3p1.cpp -o unguided
3p1 } ; if ($?) { .\unguided3p1 }
Nama hewan : kucing
Berat      : 3.8kg
PS D:\strukdat>

```

Overlaid on the command prompt is a Notepad++ window titled "Nama Fitr". The window contains the following text:

```

Nama : Fitri Kusumaningtyas
NIM  : 2311102068
Kelas: IF 11 B

```

The Notepad++ window also shows a status bar at the bottom indicating "Ln 3, Col 15 | 60 characters | 100% | Window UTF-8".

Deskripsi program :

Program di atas menggunakan fungsi class dengan private “nama” dan “berat”, dan juga dengan fungsi public “getnama () :”, “getberat () :”, “display () :”. Di dalam fungsi utama, kita membuat sebuah objek bernama “hewan” dengan nama “kucing” dan variabel “nama” dan “berat”. Kemudian kita memanggil “display()” untuk menampilkan nama dan berat hewan ke layar.

3. Unguided 3

Source code :

```

#include <iostream>
#include <map>
using namespace std;

int main() {
    // Buat sebuah map dengan kunci string dan nilai integer
    map<string, int> myMap;

    // Masukkan beberapa pasangan kunci-nilai ke dalam map
    myMap.insert(make_pair("apel", 10));
    myMap.insert(make_pair("pisang", 20));
    myMap.insert(make_pair("ceri", 30));

    // Akses nilai menggunakan operator []
    cout << "Buah: " << "apel" << ", Jumlah: " << myMap["apel"]
<< endl;
    cout << "Buah: " << "pisang" << ", Jumlah: " <<
myMap["pisang"] << endl;
    cout << "Buah: " << "ceri" << ", Jumlah: " << myMap["ceri"]
<< endl;

    // Cek apakah kunci ada di dalam map
    if (myMap.find("apel") != myMap.end()) {
        cout << "Kunci 'apel' ada dalam map" << endl;
    }

    // Melakukan iterasi over map menggunakan iterator
    for (map<string, int>::iterator it = myMap.begin(); it !=
myMap.end(); ++it) {
        cout << "Kunci: " << it->first << ", Nilai: " << it-
>second << endl;
    }

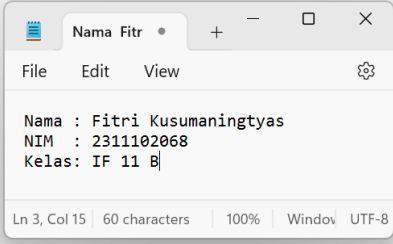
    // Melakukan iterasi over map menggunakan range-based for
loop
    for (const auto& pair : myMap) {
        cout << "Kunci: " << pair.first << ", Nilai: " <<
pair.second << endl;
    }

    return 0;
}

```

Output:

```
PS D:\strukdat> cd "d:\strukdat\" ; if ($?) { g++ unguided3p1.cpp -o unguided3p1 } ; if ($?) { .\unguided3p1 }
Buah: apel, Jumlah: 10
Buah: pisang, Jumlah: 20
Buah: ceri, Jumlah: 30
Kunci 'apel' ada dalam map
Kunci: apel, Nilai: 10
Kunci: ceri, Nilai: 30
Kunci: pisang, Nilai: 20
Kunci: apel, Nilai: 10
Kunci: ceri, Nilai: 30
Kunci: pisang, Nilai: 20
PS D:\strukdat>
```



Deskripsi program :

Dalam program ini kita membuat sebuah “map” dengan kunci “string” dan nilai dalam “integer”. Kita memasukkan beberapa pasangan kunci dan nilai kedalam map menggunakan “insert()” dan helper fungsi “make_pair()”. Lalu kita mengakses nilai menggunakan “[]”, memeriksa apakah kunci di dalam map melakukan iterasi over map menggunakan iterator dan range-based for loop.

Perbedaan “Array” dan “Map” yaitu array hanya bisa menyimpan data dalam satu tipe data saja sedangkan map bisa menyimpan data dalam berbagai tipe data dan lebih banyak memiliki ruang jika dibandingkan dengan array.

D. Kesimpulan

Tipe data adalah hal yang sangat dibutuhkan dalam semua bahasa pemograman. Pemahaman mendalam tentang macam-macam tipe data seperti tipe data primitice, abstrak, dan koleksi dengan tepat sangat penting. Pemilihan tipe data yang tepat sangat berperan dalam mengoptimalkan penggunaan memori dan meningkatkan kinerja program. Dengan memanfaatkan tipe data dengan bijak, program dapat dibangun dengan lebih efisien , aman, dan dapat dipelajari dengan mudah.

E. Referensi jurnal

- Huda, A., Ardi, N., & Muabi, A. (2021). *Pengantar Coding Berbasis C/C++*. UNP PRESS.
- Hanief, S., Jepriana, I. W., & Kom, S. (2020). *Konsep Algoritme dan Aplikasinya dalam Bahasa Pemrograman C++*. Penerbit Andi.
- Putra, M. T. D., Pradeka, D., & Yuniarti, A. R. (2022). BELAJAR DASAR PEMROGRAMAN DENGAN C++.
- Satrio, E., & Pakpahan, S. (2019). *Pemrograman C++ Untuk Pembelajaran Mandiri* (Vol. 1). Penerbit Cahaya INFORMATIKA.
- Kadir, A. (2019). *Logika Pemrograman Menggunakan C++*. Elex Media Komputindo.