

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



**Disusun oleh:
FATTAH RIZQY ADHIPRATAMA
NIM: 2311102019**

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

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

BAB I

TUJUAN PRAKTIKUM

1. Mahasiswa dapat mempelajari tipe data primitif, abstrak, dan kolektif.
2. Mahasiswa dapat memahami pengaplikasian pada tools yang digunakan.
3. Mahasiswa mengaplikasikan berbagai tipe data pada bahasa pemrograman yang telah ditentukan
4. Mahasiswa dapat mengembangkan keterampilan pemrograman melalui latihan dan proyek-praktek yang melibatkan tipe data.
5. Mahasiswa dapat menjelaskan apa itu tipe data dan peranannya dalam pemrograman.

BAB II

DASAR TEORI

Tipe data adalah adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Adapun tipe data yang akan dipelajari, sebagai berikut :

1. Tipe data Primitif
2. Tipe data Abstrak
3. Tipe data Koleksi

Tipe data Primitif

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem, tipe data primitif ini disediakan oleh banyak bahasa pemrograman, perbedaannya terletak pada jumlah bit yang dialokasikan untuk setiap bit pada tipe data primitif tergantung pada bahasa pemrograman, compiler dan sistem operasinya. Contoh tipe data primitif adalah :

- a. Int : adalah tipe data yang digunakan untuk menyimpan bilangan bulat seperti 12, 1, 4, dan sebagainya.
- b. Float : tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.5, 2.1, 3.14, dan sebagainya.
- c. Char : berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya digunakan untuk simbol seperti A, B, C dan seterusnya
- d. Boolean : tipe data ini digunakan untuk menyimpan nilai boolean yang hanya memiliki dua nilai yaitu true dan false.

Tipe Data Abstrak

Tipe data abstrak atau yang biasa disebut Abstrak Data Tipe(ADT) merupakan tipe data yang dibentuk oleh programer 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](https://learn.microsoft.com/en-us/cpp/faq/struct-vs-class) perbedaan antara Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class bersifat private.

Tipe Data Koleksi

Tipe data koleksi (Collection Data Type) adalah 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 elemenelemen 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.

BAB III

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. +, -, *, /
    cin >> op;
    // It allow user to enter the operands
    cin >> num1 >> num2;
    // Switch statement begins
    switch (op)
    {
        // If user enter +
        case '+':
            cout << num1 + num2;
            break;
        // If user enter -
        case '-':
            cout << num1 - num2;
            break;
        // If user enter *
        case '*':
            cout << num1 * num2;
            break;
        // If user enter /
        case '/':
            cout << num1 / num2;
```

```

        break;

// If the operator is other than +, -, * or /,
// error message will display
default:
    cout << "Error! operator is not correct";
} // switch statement ends
return 0;
}

```

Screenshoot program

```

Run ... < -> Strukdat
e Guided1.cpp X
e Guided1.cpp > main()
1 #include <iostream>
2 using namespace std;
3 // Main program
4 int main()
5 {
6     char op;
7     float num1, num2;
8     // It allows user to enter operator i.e. +, -, *, /
9     cin >> op;
10    // It allow user to enter the operands
11    cin >> num1 >> num2;
12    // Switch statement begins
13    switch (op)
14    {
15        // If user enter +
16        case '+':
17            cout << num1 + num2;
18            break;
19        // If user enter -
20        case '-':
21            cout << num1 - num2;
22            break;
23        // If user enter *
24        case '*':
25            cout << num1 * num2;
26            break;
27        // If user enter /
28        case '/':
29            cout << num1 / num2;
30            break;
31    }
32    return 0;
33 }

```

```

PS C:\Users\Fatta\Documents\Strukdat> cd "C:\Users\Fatta\Documents\Strukdat\" ; if ($?) { g++ Guided1.cpp -o Guided1 } ; if ($?) { .\Guided1 }
*
5
4
20
PS C:\Users\Fatta\Documents\Strukdat>

```

Deskripsi program

Program diatas menerima input operator dan dua bilangan float dari pengguna. Program menggunakan switch statement untuk menentukan operator yang dimasukkan oleh pengguna dan menghasilkan output berupa hasil operasi yang sesuai dengan operator tersebut.

2. Guided 2

Source Code

```

#include <stdio.h>

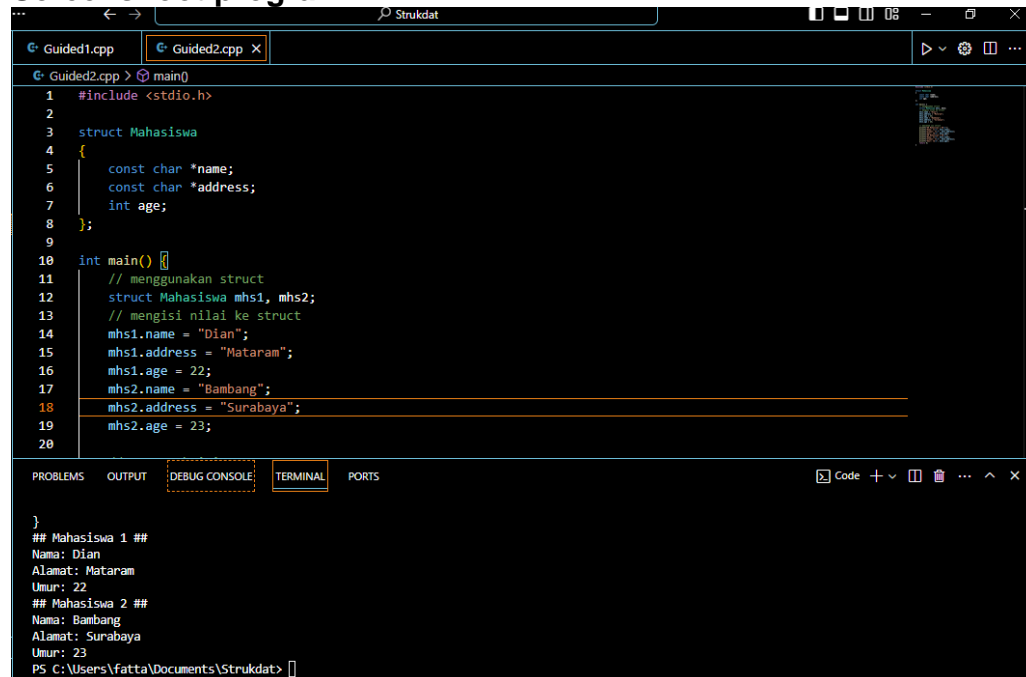
struct Mahasiswa
{
    const char *name;

```

```
    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;
}
```

Screenshoot program



```
1 #include <stdio.h>
2
3 struct Mahasiswa
4 {
5     const char *name;
6     const char *address;
7     int age;
8 };
9
10 int main() {
11     // menggunakan struct
12     struct Mahasiswa mhs1, mhs2;
13     // mengisi nilai ke struct
14     mhs1.name = "Dian";
15     mhs1.address = "Mataram";
16     mhs1.age = 22;
17     mhs2.name = "Bambang";
18     mhs2.address = "Surabaya";
19     mhs2.age = 23;
20
21     printf("## Mahasiswa 1 ##\n");
22     printf("Nama: Dian\n");
23     printf("Alamat: Mataram\n");
24     printf("Umur: 22\n");
25     printf("## Mahasiswa 2 ##\n");
26     printf("Nama: Bambang\n");
27     printf("Alamat: Surabaya\n");
28     printf("Umur: 23\n");
29
30     return 0;
31 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Code + - - - ^ x

Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\Users\Fatta\Documents\Strukdat>

Deskripsi Program

Program ini merupakan program yang menggunakan struct Mahasiswa untuk menyimpan informasi tentang mahasiswa. Program memiliki beberapa variabel struct Mahasiswa, yaitu mhs1 dan mhs2, yang digunakan untuk menyimpan informasi mahasiswa.

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;  
}
```

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

```
#include <iostream>  
using namespace std;  
  
int pengurangan(int a, int b){  
    return a-b;  
}  
  
float tambah (float c, float d){  
    return c+d;  
}  
  
int main (){  
    int a = 16;  
    int b = 7;  
    float c = 23;  
    float d = 17;  
    cout << "Hasil pengurangan dari 16-7 = " << pengurangan(a,b)  
<< endl;  
    cout << "Hasil tambah dari 23+17 = " << tambah(c,d) << endl;  
    return 0;  
}
```

Screenshoot program



```
1 #include <iostream>
2 using namespace std;
3
4 int pengurangan(int a, int b){
5     return a-b;
6 }
7
8 float tambah (float c, float d){
9     return c+d;
10 }
11
12 int main ()
13 {
14     int a = 16;
15     int b = 7;
16     float c = 23;
17     float d = 17;
18     cout << "Hasil pengurangan dari 16-7 = " << pengurangan(a,b) << endl;
19     cout << "Hasil tambah dari 23+17 = " << tambah(c,d) << endl;
20     return 0;
21 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1> cd "d:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1\" ; if ($?) { g++ unguided.cpp -o unguided } ; if ($?) { .\unguided }
Hasil pengurangan dari 16-7 = 9
Hasil tambah dari 23+17 = 40
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1>
```

Deskripsi program

Program ini adalah program yang menggunakan bahasa C++. Program ini memiliki dua fungsi, yaitu pengurangan dan tambah. Fungsi pengurangan menerima dua parameter berupa int dan mengembalikan hasil pengurangan dari kedua parameter tersebut. Fungsi tambah juga menerima dua parameter berupa float dan mengembalikan hasil penjumlahan dari kedua parameter tersebut.

2. Unguided 2

Source Code

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

struct Mobil {
    string merek;
    string model;
    string tipe;
    int harga;
};
```

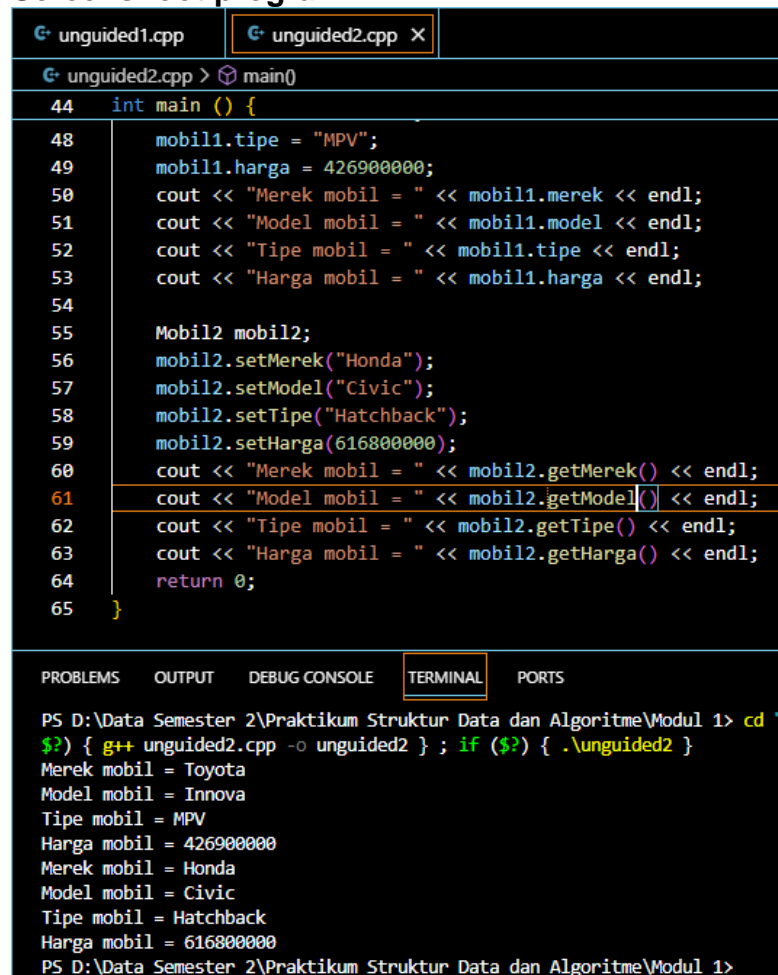
```
class Mobil2 {
    private:
        string merek;
        string model;
        string tipe;
        int harga;
    public:
        void setMerek(string merek) {
            this->merek = merek;
        }
        void setModel(string model) {
            this->model = model;
        }
        void setType(string tipe) {
            this->tipe = tipe;
        }
        void setHarga(int harga) {
            this->harga = harga;
        }
        string getMerek() {
            return this->merek;
        }
        string getModel() {
            return this->model;
        }
        string getType() {
            return this->tipe;
        }
        int getHarga() {
            return this->harga;
        }
};

int main () {
    Mobil mobil1;
```

```
    mobil1.merek = "Toyota";
    mobil1.model = "Innova";
    mobil1.tipe = "MPV";
    mobil1.harga = 426900000;
    cout << "Merek mobil = " << mobil1.merek << endl;
    cout << "Model mobil = " << mobil1.model << endl;
    cout << "Tipe mobil = " << mobil1.tipe << endl;
    cout << "Harga mobil = " << mobil1.harga << endl;

    Mobil2 mobil2;
    mobil2.setMerek("Honda");
    mobil2.setModel("Civic");
    mobil2.setTipe("Hatchback");
    mobil2.setHarga(616800000);
    cout << "Merek mobil = " << mobil2.getMerek() << endl;
    cout << "Model mobil = " << mobil2.getModel() << endl;
    cout << "Tipe mobil = " << mobil2.getTipe() << endl;
    cout << "Harga mobil = " << mobil2.getHarga() << endl;
    return 0;
}
```

Screenshoot program



```
unguided1.cpp | unguided2.cpp X
unguided2.cpp > main()
44 int main () {
48     mobil1.tipe = "MPV";
49     mobil1.harga = 4269000000;
50     cout << "Merek mobil = " << mobil1.merek << endl;
51     cout << "Model mobil = " << mobil1.model << endl;
52     cout << "Tipe mobil = " << mobil1.tipe << endl;
53     cout << "Harga mobil = " << mobil1.harga << endl;
54
55     Mobil2 mobil2;
56     mobil2.setMerek("Honda");
57     mobil2.setModel("Civic");
58     mobil2.setTipe("Hatchback");
59     mobil2.setHarga(6168000000);
60     cout << "Merek mobil = " << mobil2.getMerek() << endl;
61     cout << "Model mobil = " << mobil2.getModel() << endl;
62     cout << "Tipe mobil = " << mobil2.getTipe() << endl;
63     cout << "Harga mobil = " << mobil2.getHarga() << endl;
64     return 0;
65 }
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1> cd ".
$?) { g++ unguided2.cpp -o unguided2 } ; if ($?) { .\unguided2 }
Merek mobil = Toyota
Model mobil = Innova
Tipe mobil = MPV
Harga mobil = 4269000000
Merek mobil = Honda
Model mobil = Civic
Tipe mobil = Hatchback
Harga mobil = 6168000000
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1>
```

Deskripsi Program

Program ini memiliki dua konstruksi data, yaitu struct Mobil dan class Mobil2, yang digunakan untuk mengelompokkan data mengenai merek mobil, model mobil, tipe mobil, dan harga mobil. Program ini akan menghasilkan output berupa merek mobil, model mobil, tipe mobil, dan harga mobil yang telah diisi oleh kode.

3. Unguided 3 Source Code

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

int main() {
    map<string, string> brand;
    brand["Toyota"] = "Corolla, Innova, Fortuner";
    brand["Honda"] = "Civic, Brio, HRV";
```

```

        cout << "Contoh nama merek mobil Toyota adalah " <<
brand["Toyota"] << endl;
        cout << "Contoh nama merek mobil Honda adalah " <<
brand["Honda"] << endl;
        return 0;
}

```

Screenshoot program

The screenshot shows a C++ program in an IDE. The code defines a map named 'brand' that maps car brands to their models. It then prints the models for Toyota and Honda. The terminal output shows the execution of the program, displaying the models for Toyota (Corolla, Innova, Fortuner) and Honda (Civic, Brio, HRV).

```

C: unguided3.cpp > main()
1  #include <iostream>
2  #include <map>
3  using namespace std;
4
5  int main() {
6      map<string, string> brand;
7      brand["Toyota"] = "Corolla, Innova, Fortuner";
8      brand["Honda"] = "Civic, Brio, HRV";
9
10     cout << "Contoh nama merek mobil Toyota adalah " << brand["Toyota"] << endl;
11     cout << "Contoh nama merek mobil Honda adalah " << brand["Honda"] << endl;
12     return 0;
13 }

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```

PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1> cd "d:\Data Semester 2\Prak
$?" { g++ unguided3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Contoh nama merek mobil Toyota adalah Corolla, Innova, Fortuner
Contoh nama merek mobil Honda adalah Civic, Brio, HRV
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 1>

```

Deskripsi Program

Dalam contoh kode di atas, kami menggunakan fungsi map untuk menyimpan data nama merek mobil dan nama model mobil. Kami juga dapat menggunakan fungsi map untuk melakukan operasi lain, seperti menghapus data, mengubah data, dan lain-lain.

BAB IV

KESIMPULAN

Kesimpulannya Tipe data adalah klasifikasi untuk menentukan jenis nilai yang dimiliki variabel. Tipe data berfungsi menempatkan variabel mana saja yang dapat diaplikasikan menggunakan operasi-operasi tertentu. Dalam pemrograman, tipe data diperlukan agar compiler tahu operasi apa yang valid dan seberapa banyak memori yang diperlukan oleh sebuah nilai.

Tipe data adalah satu bagian dari pemrograman yang sangat penting, karena ia membantu pemrogram untuk memahami dan menangani data yang akan digunakan dalam program. Dengan tipe data, pemrogram dapat memahami jenis data yang akan digunakan, serta mengatur operasi yang dapat dilakukan terhadap data tersebut.