

# **LAPORAN PRAKTIKUM**

## **MODUL I TIPE DATA**



**Disusun oleh:**  
**Andika Indra Prastawa**  
**NIM: 2311102033**

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

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
PURWOKERTO  
2023**

# **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 pemograman yang telah ditentukan.

## **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 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

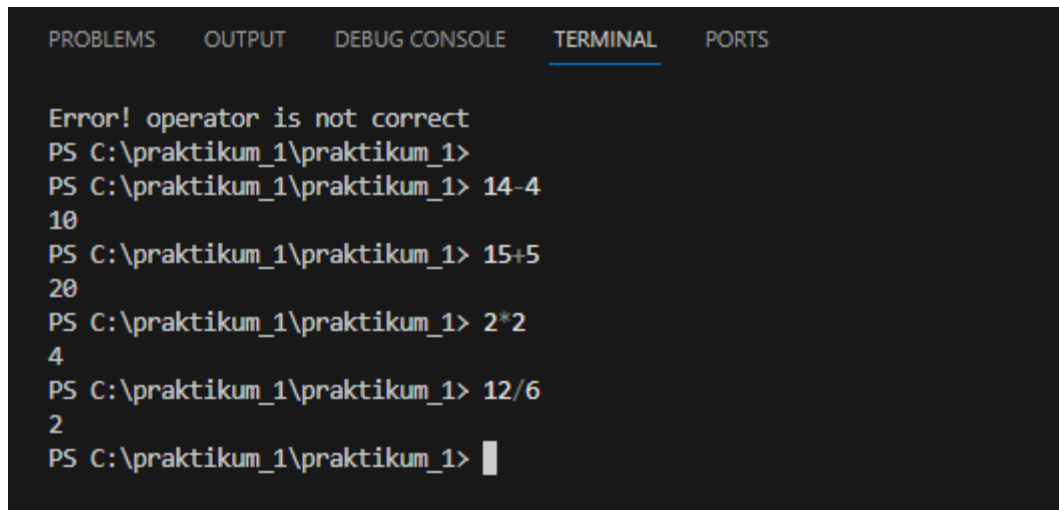
##### Source code

```
#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



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Error! operator is not correct
PS C:\praktikum_1\praktikum_1>
PS C:\praktikum_1\praktikum_1> 14-4
10
PS C:\praktikum_1\praktikum_1> 15+5
20
PS C:\praktikum_1\praktikum_1> 2*2
4
PS C:\praktikum_1\praktikum_1> 12/6
2
PS C:\praktikum_1\praktikum_1> 
```

### Deskripsi program

Sebuah program yang mengimplementasikan operator aritmatika (+, -, \*, /) pada dua angka float. Program ini menggunakan konsep "switch-case" untuk mengevaluasi operator yang diberikan oleh pengguna dan melakukan operasi aritmatika yang sesuai. Program akan meminta pengguna untuk memasukkan operator aritmatika (yaitu +, -, \*, atau /) dan dua angka float. kemudian Program akan mengevaluasi operator yang diberikan oleh pengguna menggunakan konsep "switch-case". Jika operator yang diberikan oleh pengguna adalah +, -, \*, atau /, program akan melakukan operasi aritmatika yang sesuai dan menampilkan hasilnya. Jika operator yang diberikan oleh pengguna bukan merupakan salah satu dari operator aritmatika yang valid (+, -, \*, /), program akan menampilkan pesan kesalahan.

## 2. Guided 2

### Source code

```
#include <stdio.h>

//Struct
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

```

PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1" ; if ($?) { g++ praktik
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\praktikum_1\praktikum_1> 

```

### Deskripsi program

Sebuah program yang menggunakan struct untuk menyimpan informasi tentang mahasiswa. Dengan menggunakan struct, kita dapat menyimpan beberapa

variabel yang berhubungan dengan satu objek (dalam kasus ini, informasi tentang mahasiswa) dalam satu variabel. Hal ini dapat membantu kita untuk mengatur kode yang lebih rapi dan mudah dibaca. Selain itu, program ini juga menunjukkan bagaimana cara menggunakan pointer ke string dan integer pada struct.

- Program menginclude file header `stdio.h` untuk menggunakan fungsi `printf()`.
- Program mendefinisikan struct Mahasiswa dengan tiga elemen, yaitu name (pointer ke string), address (pointer ke string), dan age (integer).
- Pada blok `main()`, program menggunakan dua variabel struct `mhs1` dan `mhs2`.
- Program mengisi nilai ke variabel struct `mhs1` dan `mhs2` dengan nama, alamat, dan umur mahasiswa.
- Program mencetak isi variabel struct `mhs1` dan `mhs2` dengan menggunakan fungsi `printf()`.
- Program mengembalikan nilai 0 untuk menandakan bahwa program berjalan dengan baik.

### 3. Guided 3

#### Source code

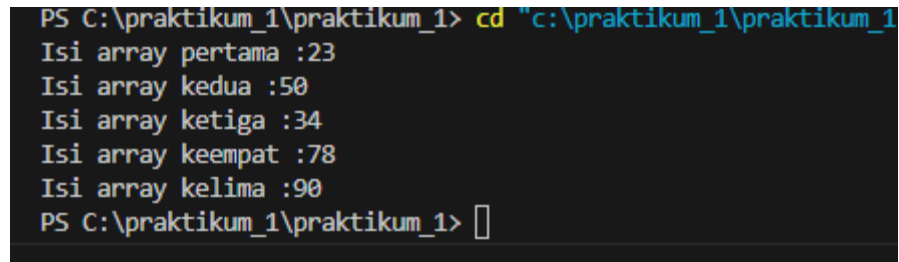
```
#include <iostream>
using namespace std;
int main()
{
    //deklarasi dan inisialisasi array
    int nilai[5];
    nilai[0] = 23;
    nilai[1] = 50;
    nilai[2] = 34;
    nilai[3] = 78;
    nilai[4] = 90;

    //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;  
}
```

### Screenshoot program



```
PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1"  
Isi array pertama :23  
Isi array kedua :50  
Isi array ketiga :34  
Isi array keempat :78  
Isi array kelima :90  
PS C:\praktikum_1\praktikum_1> 
```

### Deskripsi program

Program menggunakan array untuk menyimpan beberapa nilai integer. Dengan menggunakan array, kita dapat menyimpan beberapa nilai yang memiliki tipe data yang sama dalam satu variabel. Hal ini dapat membantu kita untuk mengatur kode yang lebih rapi dan mudah dibaca. Selain itu, program ini juga menunjukkan bagaimana cara menggunakan notasi indeks pada array.

- Program menginclude file header iostream untuk menggunakan fungsi cout.
- Program mendeklarasikan dan menginisialisasi array nilai dengan kapasitas 5 elemen.
- Program menulis nilai-nilai ke array nilai dengan menggunakan notasi indeks.
- Program mencetak isi array nilai dengan menggunakan fungsi cout.
- Program mengembalikan nilai 0 untuk menandakan bahwa program berjalan dengan baik.

## LATIHAN KELAS - UNGUIDED

### 1. Unguided 1

#### Source code

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

int tambah_2311102033 (int a, int d ){
    return a + d ;
}

float kurang_2311102033 (float i, float p){
    return i - p;
}

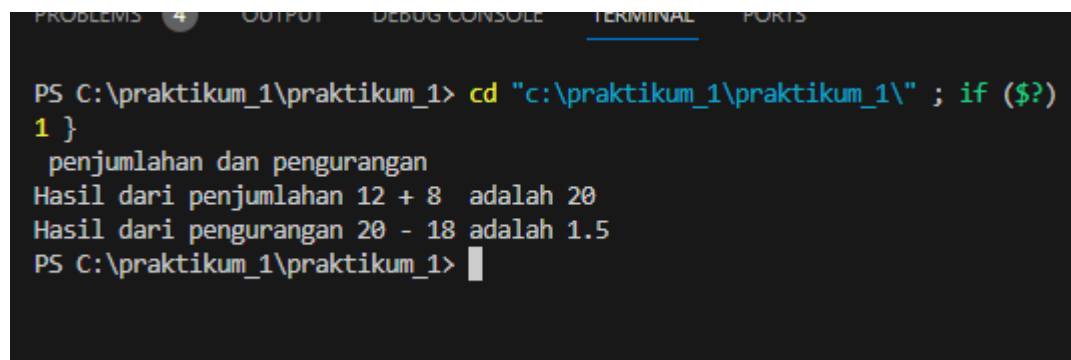
int main(){

    int a= 12;
    int d= 8;
    float i= 19.5;
    float p= 18;

    cout<<" penjumlahan dan pengurangan "<<endl;
    cout<<"Hasil    dari    penjumlahan    12    +    8    adalah
"<<tambah_2311102033    (a,d)<<endl;
    cout<<"Hasil    dari    pengurangan    20    -    18    adalah
"<<kurang_2311102033    (i,p)<<endl;

}
```

#### Screenshoot program



```
PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1\" ; if ($?)
1 }
    penjumlahan dan pengurangan
Hasil dari penjumlahan 12 + 8  adalah 20
Hasil dari pengurangan 20 - 18 adalah 1.5
PS C:\praktikum_1\praktikum_1>
```

#### Deskripsi program

Program di atas adalah program sederhana dalam bahasa C++ yang mendefinisikan dua fungsi untuk melakukan penjumlahan dan pengurangan. Fungsi tambah\_2311102033 memiliki dua parameter bertipe integer a dan d,

dan mengembalikan hasil penjumlahan dari kedua parameter tersebut. Fungsi kurang\_2311102033 memiliki dua parameter bertipe float i dan p, dan mengembalikan hasil pengurangan dari kedua parameter tersebut. Di dalam fungsi main, terdapat empat variabel: dua integer a dan d, dan dua float 'l' dan 'p', yang masing-masing diberi nilai. Kemudian, hasil penjumlahan dari 'a' dan 'd' dicetak menggunakan fungsi tambah\_2311102033, sedangkan hasil pengurangan dari i dan p dicetak menggunakan fungsi kurang\_2311102033. Hasil cetakan dari kedua operasi matematika tersebut ditampilkan di layar.

## Kesimpulan

Tipe data primitif ialah tipe data yang ditentukan oleh sebuah sistem, dan jumlah bit yang digunakan untuk setiap tipe data primitif tergantung pada bahasa pemrograman, compiler, dan sistem operasinya. Contoh tipe data primitif adalah int, float, char, boolean, dan lain-lain.

## 2. Unguided 2

### Source code

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

struct mahasiswa{
    string nama;
    string jurusan;
    string Alamat;
    float umur;
    float ipk;
};

int main(){

    mahasiswa mhs;

    mhs.nama="Andika Indra Prastawa";
```

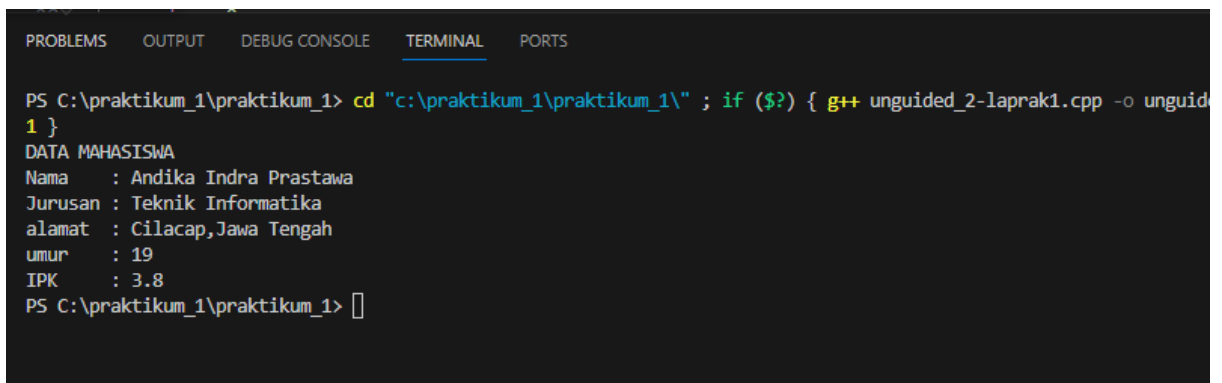
```

    mhs.jurusan="Teknik Informatika";
    mhs.Alat="Cilacap,Jawa Tengah";
    mhs.umur =19;
    mhs.ipk=3.8;

    cout<<"DATA MAHASISWA"<<endl;
    cout<<"Nama      : "<<mhs.nama<<endl;
    cout<<"Jurusan  : "<<mhs.jurusan<<endl;
    cout<<"alamat   : "<<mhs.Alat<<endl;
    cout<<"umur     : "<<mhs.umur<<endl;
    cout<<"IPK      : "<<mhs.ipk<<endl;
    return 0;
}

```

### Screenshoot program



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1\" ; if ($?) { g++ unguided_2-laprak1.cpp -o unguid
1 }
DATA MAHASISWA
Nama      : Andika Indra Prastawa
Jurusan   : Teknik Informatika
alamat    : Cilacap,Jawa Tengah
umur      : 19
IPK       : 3.8
PS C:\praktikum_1\praktikum_1>

```

### Deskripsi program

Program di atas adalah program C++ yang mendefinisikan sebuah struct bernama mahasiswa. Struct ini memiliki lima variabel, yaitu nama, jurusan, Alamat, umur, dan ipk. Pada blok main program, variabel mhs dideklarasikan sebagai instans dari struct mahasiswa. Kemudian, nilai-nilai dari variabel-variabel tersebut diinisialisasi. Selanjutnya, program mencetak atribut-atribut mahasiswa ke layar menggunakan operator <<. Hasil dari program akan menampilkan data mahasiswa, yaitu nama, jurusan, alamat, umur, dan IPK. Secara sederhana, program ini digunakan untuk menyimpan dan mencetak data mahasiswa.

### Fungsi Struct dan Class

Struct adalah sebuah tipe data yang digunakan untuk grup beberapa variabel, yang mungkin memiliki tipe data berbeda, menjadi satu kesatuan.

Secara default, akses ke anggota struct adalah public, kecuali jika diubah secara eksplisit.

Sementara itu, Class adalah tipe data yang lebih kompleks dari struct. Selain menyimpan data, class dapat menyertakan fungsi-fungsi atau metode yang beroperasi pada data tersebut. Class memiliki kontrol akses yang lebih ketat di antara private, protected, dan public, memungkinkan pembatasan akses ke anggota class. Class menyediakan konsep encapsulation, yaitu menyembunyikan detail implementasi dan hanya mengekspos operasi yang diperlukan melalui antarmuka publik

### 3. Unguided 3

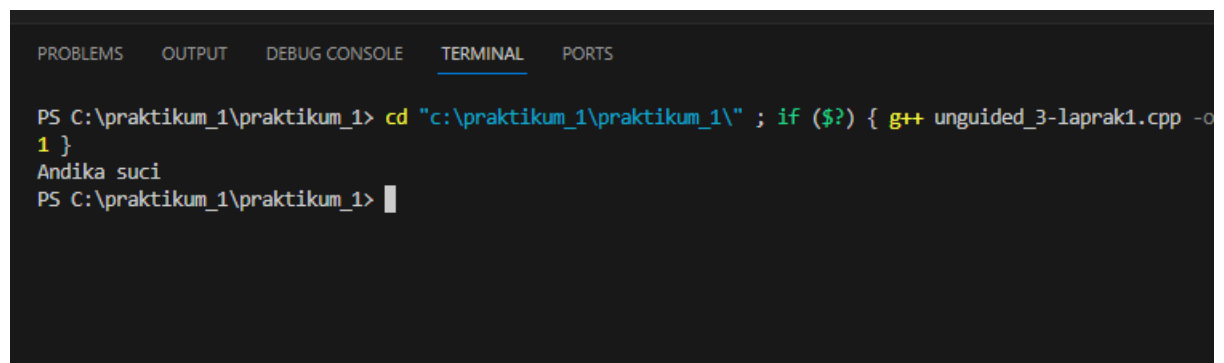
#### Source code

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

int main() {
    map<int, string> nama_mahasiswa { { 1, "Andika"}, { 2, "tiyas" }, { 3, "suci " }, { 4, "asep" }, { 5, "aziz" } };

    cout << nama_mahasiswa[1] << " " << nama_mahasiswa[3]<<endl;
}
```

#### Screenshoot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1\" ; if ($?) { g++ unguided_3-laprak1.cpp -o 1 }
Andika suci
PS C:\praktikum_1\praktikum_1> 
```

#### Deskripsi program

Program di atas adalah program yang menggunakan map container. Map adalah container yang menyimpan pasangan key-value. Dalam program ini,

terdapat variabel `nama_mahasiswa` yang dideklarasikan sebagai map dengan tipe key `int` dan tipe value `string`. Variabel `nama_mahasiswa` diinisialisasi dengan beberapa pasangan key-value. pada blok main, program mencetak nilai dari key 1 dan 3 dari map `nama_mahasiswa`. Hasil dari program akan menampilkan nama mahasiswa dengan key 1 dan 3, yaitu "Andika" dan "suci". Secara sederhana, program ini digunakan untuk menyimpan dan mencetak nama-nama mahasiswa.

### **Perbedaan Array dan Map**

Array adalah sebuah struktur data yang tersusun secara linear dan terdiri dari sejumlah elemen dengan tipe data yang sama. Elemen-elemen tersebut ditempatkan secara berurutan di dalam memori dan dapat diakses menggunakan indeks numerik yang dimulai dari nol. Ukuran array harus ditentukan saat deklarasi dan tidak bisa diubah selama program berjalan. Pengaksesan dan pemodifikasian elemen array dilakukan dengan cepat menggunakan indeks numerik.

Sementara itu, Map adalah struktur data yang asosiatif, yang menghubungkan kunci unik dengan nilai tertentu. Setiap kunci dalam map memiliki satu nilai yang berkaitan. Pengaksesan elemen dalam map dilakukan berdasarkan kunci, bukan indeks numerik. Map dapat tumbuh atau menyusut secara dinamis sesuai dengan penambahan atau penghapusan elemen. Meskipun operasi penyisipan, penghapusan, dan pencarian elemen dalam map memerlukan kompleksitas waktu yang lebih tinggi daripada array, namun map memungkinkan pengelompokan data yang lebih fleksibel.

## **BAB IV**

### **KESIMPULAN**

Dalam pemrograman, terdapat beragam tipe data yang dapat digunakan, mulai dari tipe data primitif yang telah disediakan oleh bahasa pemrograman seperti integer, float, char, dan bool, hingga tipe data abstrak yang dapat dibuat oleh pengguna seperti array, list, stack, queue, dan tree. Selain itu, terdapat juga koleksi seperti array, vector, set, map, dan hash table yang merupakan bagian penting dari tipe data dalam pemrograman.