

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



**Disusun oleh:
Nofita Fitriyani
NIM: 2311102001**

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.

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 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](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

Tipe Data Primitif :

Source code

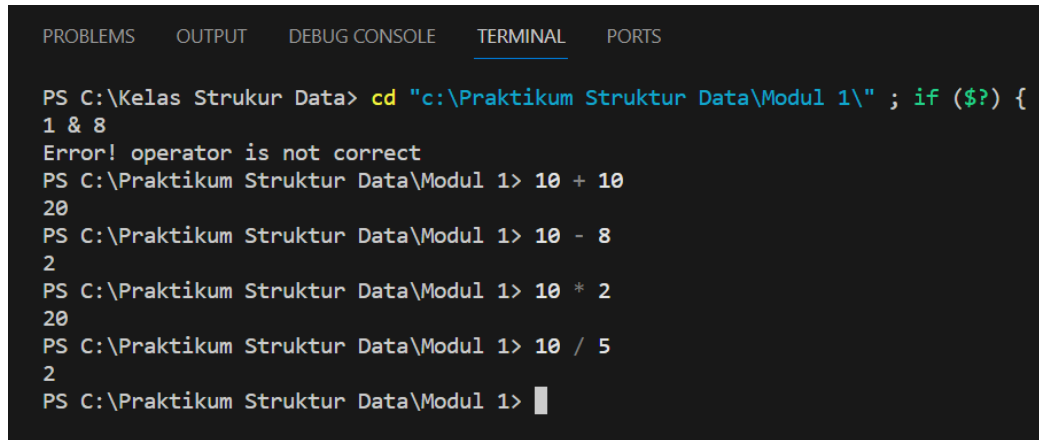
```
int main ()
{
    char op;
    float num1, num2;

    cin>>op;
    cin>>num1>>num2;
    switch (op)
    {
        case '+' :
            cout << num1 + num2;
            break;
        case '-':
            cout << num1 - num2;
            break;
        case '*':
            cout << num1 * num2;
            break;
        case '/':
            cout << num1 / num2;
            break;

        default:
            cout << "Error! operator is not correct";

    }
    return 0;
}
```

Screenshoot program



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Kelas Strukur Data> cd "c:\Praktikum Struktur Data\Modul 1\" ; if ($?) {
1 & 8
Error! operator is not correct
PS C:\Praktikum Struktur Data\Modul 1> 10 + 10
20
PS C:\Praktikum Struktur Data\Modul 1> 10 - 8
2
PS C:\Praktikum Struktur Data\Modul 1> 10 * 2
20
PS C:\Praktikum Struktur Data\Modul 1> 10 / 5
2
PS C:\Praktikum Struktur Data\Modul 1> █
```

Deskripsi program

Program tersebut menggunakan tipe data primitif ‘op’ bertipe char atau character yang biasa digunakan untuk huruf (A, B, C, dst) sedangkan untuk ‘num1’ dan ‘num2’ bertipe float yang digunakan untuk bilangan desimal. Program tersebut menggunakan switch case untuk mengecek nilai dari variabel ‘op’ dengan cara ini, program ini memberikan fungsionalitas dasar kalkulator sederhana yang memungkinkan pengguna untuk melakukan operasi aritmetika dasar.

2. Guided 2

Tipe Data Abstrak :

Source code

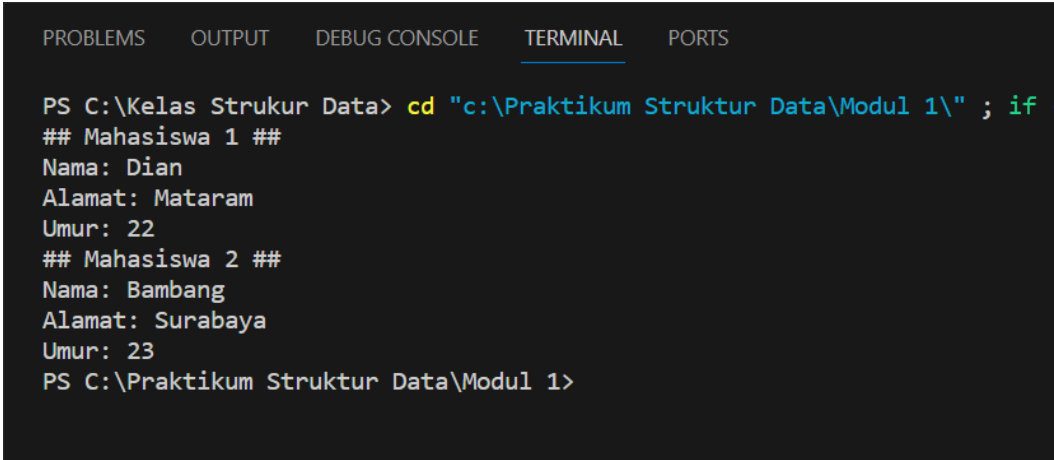
```
#include <stdio.h>

struct Mahasiswa
{
    const char *name;
    const char *address;
    int age;
};

int main()
```

```
{  
  
    struct Mahasiswa mhs1, mhs2;  
  
    mhs1.name = "Dian";  
    mhs1.address = "Mataram";  
    mhs1.age = 22;  
    mhs2.name = "Bambang";  
    mhs2.address = "Surabaya";  
    mhs2.age = 23;  
  
    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;  
}
```

Screenshot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
PS C:\Kelas Strukur Data> cd "c:\Praktikum Struktur Data\Modul 1\" ; if  
## Mahasiswa 1 ##  
Nama: Dian  
Alamat: Mataram  
Umur: 22  
## Mahasiswa 2 ##  
Nama: Bambang  
Alamat: Surabaya  
Umur: 23  
PS C:\Praktikum Struktur Data\Modul 1>
```

Deskripsi program

Program mendefinisikan struct Mahasiswa memiliki 3 anggota yaitu : name, address, dan age. Dalam program tersebut juga mendeklarasikan variable bertipe struct Mahasiswa yaitu mhs1 dan mhs2. Program mencetak informasi mahasiswa ke layar menggunakan fungsi printf.

Informasi yang dicetak meliputi nama, alamat, dan usia untuk masing-masing mahasiswa. Dengan cara ini, program ini menggunakan struktur untuk menyimpan dan mengelola informasi mahasiswa, dan kemudian mencetak informasi tersebut ke layar. Program ini memberikan contoh penggunaan struktur dalam bahasa C++ untuk merepresentasikan data terstruktur.

3. Guided 3

Tipe Data Koleksi :

Source code

```
#include <iostream>
using namespace std;
int main ()
{

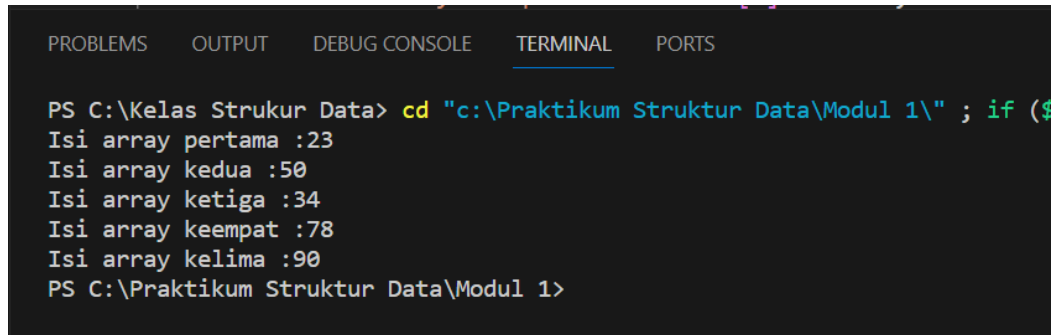
    int nilai [5];
    nilai [0] = 23;
    nilai [1] = 50;
    nilai [2] = 34;
    nilai [3] = 78;
    nilai [4] = 90;

    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;  
}
```

Screenshot program



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
  
PS C:\Kelas Strukur Data> cd "c:\Praktikum Struktur Data\Modul 1\" ; if ($?) {  
Isi array pertama :23  
Isi array kedua :50  
Isi array ketiga :34  
Isi array keempat :78  
Isi array kelima :90  
PS C:\Praktikum Struktur Data\Modul 1>
```

Deskripsi program

Program yang diberikan adalah program sederhana dalam bahasa C++ yang menggunakan array untuk menyimpan dan menampilkan sejumlah nilai integer. Dalam program ini, sebuah array dengan nama `nilai` dideklarasikan dan diinisialisasi dengan lima nilai. Setelah itu, setiap elemen dari array tersebut ditampilkan ke layar menggunakan pernyataan `cout`. Program ini memberikan contoh penggunaan array untuk menyimpan dan mengakses sejumlah nilai dalam bahasa C++, dan hasilnya adalah mencetak isi array ke layar.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Buatlah program menggunakan tipe data primitif minimal dua fungsi dan bebas. Menampilkan program, jelaskan program tersebut dan ambil kesimpulan dari materi tipe data primitif!

Source code

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

void judulProgram_2311102001() {
    cout << "PROGRAM PENENTU HARI" << endl;
    cout << "-----" << endl;
}

int main ()
{
    judulProgram_2311102001();
    int hari;

    cout << "Masukan angka 1-7 :";
    cin>>hari;

    switch (hari)
    {
        case 1 :
            cout << "Hari ke-1 yaitu Senin";
            break;
        case 2:
            cout << "Hari ke-2 yaitu Selasa";
            break;
        case 3:
            cout << "Hari ke-3 yaitu Rabu";
```

```

        break;
        case 4 :
            cout << "Hari ke-4 yaitu Kamis";
            break;
        case 5 :
            cout << "Hari ke-5 yaitu Jumat";
            break;
        case 6 :
            cout << "Hari ke-6 yaitu Sabtu";
            break;
        case 7 :
            cout << "Hari ke-7 yaitu Minggu" ;
            break;

        default:
            cout << "Error! operator is not correct";

    }
    return 0;
}

```

Screenshoot program

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Praktikum Struktur Data\Modul 1> cd "c:\Praktikum Struktur Data\Modul 1\"
unguided1 }
PROGRAM PENENTU HARI
-----
Masukan angka 1-7 :1
Hari ke-1 yaitu Senin
PS C:\Praktikum Struktur Data\Modul 1>

```

Deskripsi program

Program C++ ini menggunakan dua fungsi, yaitu `judulProgram_2311102001()` dan `main()`. Fungsi `judulProgram_2311102001()` bertujuan untuk mencetak judul program dan garis pemisah ke layar. Di sisi lain, fungsi `main()` berfungsi

sebagai fungsi utama. Setelah memanggil fungsi judul, program meminta pengguna untuk memasukkan angka 1-7. Menggunakan struktur switch case, program menentukan dan mencetak nama hari yang sesuai dengan input. Jika input tidak sesuai dengan angka yang diharapkan, program menampilkan pesan kesalahan. Dengan menggunakan fungsi, program menjadi lebih terstruktur dan modular, mempermudah pemahaman dan pemeliharaan kode.

Kesimpulannya bahwa tipe data primitif adalah tipe data yang telah ditentukan oleh sistem dan disediakan oleh banyak bahasa pemrograman. Perbedaan dalam alokasi bit untuk setiap tipe data primitif tergantung pada bahasa pemrograman, compiler, dan sistem operasi yang digunakan. Contoh tipe data primitif meliputi `int` untuk menyimpan bilangan bulat, `float` untuk menyimpan bilangan desimal, `char` untuk menyimpan huruf atau simbol, dan `boolean` untuk menyimpan nilai boolean (true atau false). Kesimpulannya, tipe data primitif menyediakan dasar untuk menyimpan berbagai jenis nilai dalam pemrograman, dan pemahaman tentang perbedaan alokasi bit dapat bervariasi tergantung pada lingkungan pemrograman yang digunakan.

2. Unguided 2

Jelaskan fungsi dari class dan struct secara detail dan berikan contoh programnya.

Source code class :

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

//2311102001
class Buku {
private:
    string judul;
    string penulis;
    int tahunTerbit;
```

```

public:
    Buku(string j, string p, int t) : judul(j), penulis(p),
    tahunTerbit(t) {}

    void displayInfo() {
        cout << "Judul: " << judul << endl;
        cout << "Penulis: " << penulis << endl;
        cout << "Tahun Terbit: " << tahunTerbit << endl;
    }
};

int main() {
    Buku bukul("Harry Potter", "J.K. Rowling", 1997);

    cout << "Informasi Buku\n";
    cout << "*****\n";

    bukul.displayInfo();
    return 0;
}

```

Source code struct :

```

#include <iostream>
#include <string>
using namespace std;
//2311102001

struct Buku {
    string judul;
    string penulis;
    int tahunTerbit;
};

```

```

void displayInfo(const Buku& buku) {
    cout << "Judul: " << buku.judul << endl;
    cout << "Penulis: " << buku.penulis << endl;
    cout << "Tahun Terbit: " << buku.tahunTerbit << endl;
}

int main() {
    Buku bukul = {"Harry Potter", "J.K. Rowling", 1997};

    cout << "Informasi Buku\n";
    cout << "*****\n";

    displayInfo(bukul);

    return 0;
}

```

Screenshot program

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

if ($?) { .\unguided2_class }
Informasi Buku
*****
Judul: Harry Potter
Penulis: J.K. Rowling
Tahun Terbit: 1997
PS C:\Praktikum Struktur Data\Modul 1> cd "c:\Praktikum Struktur Data\Modul 1\"
; if ($?) { .\unguided2_struct }
Informasi Buku
*****
Judul: Harry Potter
Penulis: J.K. Rowling
Tahun Terbit: 1997

```

Deskripsi Program

Kedua program tersebut adalah contoh penggunaan struktur data untuk merepresentasikan informasi sebuah buku dalam bahasa pemrograman C++. Pada program pertama, kita menggunakan `class` untuk membuat entitas `Buku` yang memiliki tiga data member private, yaitu `judul`, `penulis`, dan `tahunTerbit`.

Program tersebut juga memiliki sebuah constructor untuk menginisialisasi nilai data member saat objek dibuat, serta sebuah member function public `displayInfo()` yang digunakan untuk menampilkan informasi buku ke layar. Fungsi `main()` membuat objek `buku1` dari class `Buku` dan menggunakan member function `displayInfo()` untuk menampilkan informasi buku tersebut.

Sementara pada program kedua, kita menggunakan `struct` untuk mencapai hasil yang serupa. Struct `Buku` memiliki tiga data member, dan tidak memerlukan constructor karena struktur secara otomatis memberikan inisialisasi default. Fungsi `displayInfo()` pada program ini adalah fungsi biasa di luar struct, yang menerima objek `Buku` sebagai parameter dan menampilkan informasinya ke layar. Fungsi `main()` membuat objek `buku1` dari struct `Buku` dan menggunakan fungsi `displayInfo()` untuk menampilkan informasi buku tersebut.

Kedua program tersebut menunjukkan bahwa baik `class` maupun `struct` dapat digunakan untuk membuat tipe data baru yang menggabungkan data dan fungsi. Pemilihan antara keduanya tergantung pada kebutuhan dan preferensi programmer, dengan `class` memberikan lebih banyak kontrol terhadap hak akses dan member function, sedangkan `struct` memberikan sifat default yang bersifat public untuk anggotanya.

3. Unguided 3

Buat dan jelaskan program menggunakan fungsi map dan jelaskan perbedaan dari array dengan map

Source code

```
#include <iostream>
#include <map>
//2311102001
using namespace std;
```

```
int main() {  
    map<string, int> skincareBrands;  
  
    skincareBrands["Wardah"] = 10;  
    skincareBrands["Emina"] = 5;  
    skincareBrands["Scarlett"] = 8;  
  
    cout << "Isi map (Skincare Brands):\n";  
    for (auto it = skincareBrands.begin(); it !=  
        skincareBrands.end(); ++it) {  
        cout << it->first << " => " << it->second << "\n";  
    }  
  
    return 0;  
}
```

Deskripsi Program

Program C++ di atas menggunakan STL (Standard Template Library) `map` untuk membuat dan mengelola data tentang jumlah stok produk dari beberapa merek skincare. Program ini mendeklarasikan sebuah `map` dengan tipe kunci berupa string (nama merek skincare) dan nilai berupa integer (jumlah stok). Tiga merek skincare, yaitu "Wardah", "Emina", dan "Scarlett", beserta jumlah stoknya diinisialisasi ke dalam map.

Selanjutnya, program menggunakan loop `for` untuk mengiterasi melalui seluruh elemen map dan mencetak nama merek serta jumlah stok ke layar. Dalam konteks ini, setiap elemen map memiliki dua bagian: `it->first` yang menyimpan kunci (nama merek) dan `it->second` yang menyimpan nilai (jumlah stok).

Hasilnya, program ini akan mencetak informasi jumlah stok skincare brands ke layar, memberikan pemahaman tentang cara menggunakan struktur data map untuk menyimpan dan mengakses data terkait merek dan stok produk.

Array dan `map` adalah dua struktur data yang berbeda. Array adalah struktur data yang menyimpan elemen-elemen dengan tipe data yang sama dalam urutan berindeks, diakses menggunakan indeks numerik. Ukuran array ditentukan pada saat deklarasi dan tetap sepanjang program. Sebaliknya, `map` adalah struktur data asosiatif yang memetakan kunci unik ke nilai tertentu. Elemen-elemen diatur berdasarkan urutan kunci dan dapat diakses menggunakan kunci tersebut. Ukuran `map` dapat berubah dinamis saat elemen ditambahkan atau dihapus. Perbedaan utama antara keduanya adalah cara akses elemen: array menggunakan indeks numerik, sementara `map` menggunakan kunci. Selain itu, array cocok untuk data homogen dengan tipe data yang sama, sementara `map` lebih sesuai untuk data heterogen dengan tipe data yang berbeda. Pemilihan antara keduanya tergantung pada kebutuhan program dan jenis data yang akan disimpan, di mana array memberikan efisiensi untuk operasi sederhana, sedangkan `map` memberikan fleksibilitas dan kemudahan manajemen untuk data terkait kunci dan nilai.

BAB IV

KESIMPULAN

Program-program yang telah dibahas menunjukkan penggunaan tipe data dan struktur data dalam bahasa pemrograman C++. Dalam program kalkulator sederhana, kita menggunakan tipe data primitif seperti char dan float untuk mengimplementasikan operasi aritmetika dasar. Pada contoh program struktur mahasiswa, struktur digunakan untuk merepresentasikan informasi mahasiswa dengan nama, alamat, dan usia. Program dengan array dan vector menunjukkan cara menyimpan dan mengelola sejumlah nilai dengan struktur data yang sesuai.

Pemahaman tipe data primitif membantu dalam menyimpan nilai dengan format yang tepat. Penggunaan struktur data, baik dengan class atau struct, memberikan cara untuk mengorganisir dan merepresentasikan informasi terstruktur. Dalam penggunaan array, vector, dan map, kita dapat menyimpan dan mengakses sejumlah data dengan cara yang terstruktur dan fleksibel sesuai dengan kebutuhan.

Program terakhir yang menggunakan map STL menunjukkan bagaimana kita bisa mengelola informasi terkait merek skincare dan stok produknya dengan mudah. Perbandingan antara array dan map memberikan wawasan tentang penggunaan yang tepat berdasarkan sifat data yang akan disimpan.

Secara keseluruhan, kesimpulan tersebut memperkuat ide bahwa pemahaman tipe data dan pemilihan struktur data yang tepat sangat penting dalam pengembangan program. Dengan pemilihan yang bijak, program dapat menjadi lebih terstruktur, mudah dipahami, dan efisien dalam mengelola informasi.