

LAPORAN PRAKTIKUM

MODUL I TIPE DATA



Disusun oleh:
Nabila Shasya Sabrina
NIM: 2311102039

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 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

Tipe data Primitif adalah tipe data dasar yang dapat digunakan untuk menyimpan dan handle data berupa bilangan, karakter, dan boolean. Contoh tipe data primitif dalam C++ adalah *int*, *char*, dan *bool*.

2. Tipe data Abstrak

Tipe data abstrak adalah tipe data yang didefinisikan oleh programmer, yang memiliki atribut dan operasi yang dapat digunakan untuk menyelesaikan masalah tertentu. Contoh tipe data abstrak dalam C++ adalah *struct* dan *class*.

3. Tipe data Koleksi

Tipe data koleksi adalah tipe data yang dapat menyimpan koleksi data, seperti array, list, dan map. Contoh tipe data koleksi dalam C++ adalah *array*, *vector*, dan *map*.

BAB III

GUIDED

1. Guided 1

Source code

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

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

    cin >> op;
    cin >> num1 >> num2;

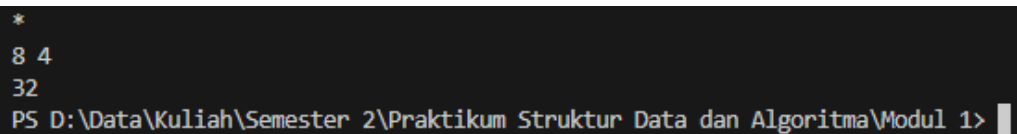
    switch (op)
    {
        case '+':
            cout << num1 + num2;
            break;

        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



```
*
8 4
32
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1>
```

Deskripsi program

Program tersebut merupakan program yang menerima dua bilangan dan satu operator matematika. Program akan melakukan perhitungan hasil operasi matematika pada program berdasarkan operator yang diberikan dan menampilkan hasilnya. Jika operator yang diberikan tidak sama dengan operator yang ada dalam program, program akan menampilkan pesan “Error! Operator is not correct”.

1. Baris 4: Membuat variabel op bertipe char untuk menyimpan operator matematika (+, -, *, /).
2. Baris 5-6: Membuat variabel num1 dan num2 bertipe float untuk menyimpan dua angka yang akan dioperasikan.
3. Baris 8: Meminta pengguna untuk memasukkan operator matematika.
4. Baris 9: Meminta pengguna untuk memasukkan dua angka yang akan dioperasikan.
5. Baris 11-28: Menggunakan struktur switch-case untuk memeriksa nilai variabel op (operator matematika) yang dimasukkan pengguna. Jika operator yang dimasukkan adalah +, -, *, atau /, maka program akan melakukan operasi yang sesuai dan menampilkan hasilnya. Jika operator tidak valid, program akan menampilkan pesan kesalahan.
6. Baris 29: Mengakhiri program dengan nilai kembali 0, menandakan bahwa program berakhir dengan sukses.

2. Guided 2

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;
}

```

Screenshoot program:

```

## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1>

```

Deskripsi Program:

Program tersebut adalah sebuah program yang mendemonstrasikan penggunaan struktur (struct) untuk menyimpan informasi tentang mahasiswa. Setiap mahasiswa memiliki atribut nama, alamat, dan umur.

1. Baris 4-7: Mendefinisikan sebuah struktur (struct) bernama Mahasiswa yang memiliki tiga anggota (atribut): name (nama), address (alamat), dan age (umur). Atribut name dan address bertipe pointer ke char (string), sedangkan atribut age bertipe int.

2. Baris 9-19: Di dalam fungsi `main()` , membuat dua variabel `mhs1` dan `mhs2` yang bertipe `struct Mahasiswa` untuk menyimpan data dua mahasiswa.
3. Baris 21-26: Menginisialisasi data untuk masing-masing mahasiswa. `Mhs1` memiliki nama "Dian", alamat "Mataram", dan umur 22 tahun. `Mhs2` memiliki nama "Bambang", alamat "Surabaya", dan umur 23 tahun.
4. Baris 28-33: Mencetak informasi tentang masing-masing mahasiswa ke layar menggunakan `printf()` . Setiap informasi dipisahkan dengan baris baru (`\n`).
5. Baris 35: Mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

3. Guided 3

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;
}
```

Screenshoot program:

```
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1> █
```

Deskripsi Program:

Program di atas adalah sebuah program yang mendemonstrasikan penggunaan array untuk menyimpan beberapa nilai bilangan bulat (`int`) dan kemudian mencetak nilai-nilai tersebut ke layar.

1. Baris 4: Mendeklarasikan sebuah array bernama `nilai` yang dapat menyimpan 5 nilai bulat.
2. Baris 5-9: Menginisialisasi nilai-nilai dalam array dengan menggunakan indeks array. Setiap nilai diberikan ke indeks tertentu dalam array.
3. Baris 11-15: Mencetak isi dari setiap elemen array ke layar menggunakan operator `<<` dari objek `cout`. Setiap nilai dipisahkan dengan baris baru (`\n`).
4. Baris 16: Mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

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>

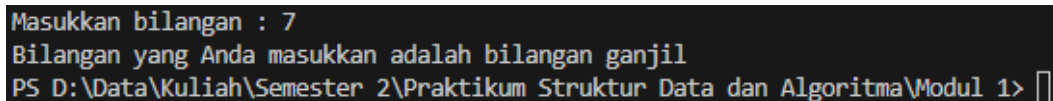
using namespace std;

int bilangan;
string ganjilGenap (int bilangan) {
    if (bilangan % 2 == 0) {
        return "genap";
    } else {
        return "ganjil";
    }
}

int main(){
    cout << "Masukkan bilangan : ";
    cin >> bilangan;
    string hasil = ganjilGenap(bilangan);
    cout << "Bilangan yang Anda masukkan adalah bilangan " <<
    hasil << endl;

    return 0;
}
```

Screenshoot program



```
Masukkan bilangan : 7
Bilangan yang Anda masukkan adalah bilangan ganjil
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1> █
```

Deskripsi program

Program di atas adalah program sederhana yang meminta pengguna memasukkan sebuah bilangan dan kemudian menentukan apakah bilangan tersebut genap atau ganjil. Berikut penjelasan dari program tersebut:

1. Program meminta pengguna untuk memasukkan sebuah bilangan bulat.

2. Bilangan yang dimasukkan oleh pengguna kemudian disimpan dalam variabel `bilangan`.
3. Program memanggil fungsi `ganjilGenap(bilangan)` dengan argumen `bilangan` yang dimasukkan pengguna.
4. Fungsi `ganjilGenap(int bilangan)` memeriksa apakah bilangan tersebut genap atau ganjil dengan menggunakan operator modulo (%). Jika bilangan tersebut habis dibagi dua, maka bilangan tersebut genap, dan jika tidak, maka bilangan tersebut ganjil.
5. Hasil dari fungsi `ganjilGenap()` disimpan dalam variabel `hasil`.
6. Fungsi `main()` merupakan titik masuk program.
7. Program menampilkan hasilnya ke layar dengan pesan "Bilangan yang Anda masukkan adalah bilangan [hasil]" di mana [hasil] adalah "genap" jika bilangan tersebut genap dan "ganjil" jika bilangan tersebut ganjil.

Kesimpulan:

Tipe data primitif adalah tipe data dasar yang dapat digunakan untuk menyimpan dan menghandle data berupa bilangan bulat, bilangan pecahan, karakter, dan boolean. Tipe data primitif berfungsi untuk memudahkan program dalam mendeteksi berbagai jenis informasi dan memastikan sistem operasional dapat berjalan dan meminimalkan kesalahan. Contoh tipe data primitif dalam C++ antara lain `short`, `long`, `byte`, `int`, `float`, `double`, `boolean`, dan `char`.

2. Unguided 2

Jelaskan fungsi dari `class` dan `struct` secara detail dan berikan contoh programnya!

1. Class

`Class` adalah struktur data yang dapat mengelompokkan data anggota (variabel) dan fungsi anggota (metode) ke dalam satu unit tunggal.

Data anggota biasanya bersifat privat dan hanya dapat diakses oleh metode dalam class tersebut. Namun, dapat ditentukan sebagai public, protected, atau private menggunakan akses kontrol.

Class menyediakan konsep enkapsulasi, di mana data dan fungsi yang berhubungan dikemas bersama, sehingga mengurangi kompleksitas dan meningkatkan keamanan dan fleksibilitas dalam pengembangan perangkat lunak.

Source Code:

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

class Person {
private:
    string name;
    int age;

public:
    void setName(string n) {
        name = n;
    }

    void setAge(int a) {
        age = a;
    }

    void display() {
        cout << "Name: " << name << endl;
        cout << "Age: " << age << endl;
    }
};

int main() {
    Person p;
    p.setName("Luffy");
    p.setAge(20);
    p.display();

    return 0;
}
```

Screenshot Program:

```
Name: Luffy
Age: 20
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1>
```

2. Struct

Struct (singkatan dari "structure") adalah kumpulan data yang dibuat untuk menyimpan beberapa tipe data yang berbeda dalam satu unit.

Secara umum, anggota dalam struct secara default bersifat public, yang berarti dapat diakses dari luar struktur. Namun, dalam beberapa bahasa pemrograman, seperti C++, bisa juga diberi hak akses lainnya.

Struct biasanya digunakan untuk menyimpan sekumpulan data sederhana yang saling terkait, tanpa menyediakan metode atau operasi tambahan di atas data tersebut.

Source Code:

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

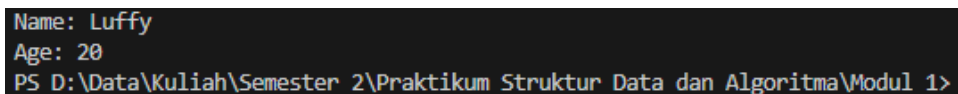
struct Person {
    string name;
    int age;
};

int main() {
    Person p;
    p.name = "Luffy";
    p.age = 20;

    cout << "Name: " << p.name << endl;
    cout << "Age: " << p.age << endl;

    return 0;
}
```

Screenshot Program:



```
Name: Luffy
Age: 20
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1>
```

3. Unguided 3

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

Source code:

```
#include <iostream>
#include <map>

using namespace std;

string ganjilGenap(int num) {
    return (num % 2 == 0) ? "genap" : "ganjil";
}

int main() {
    map<int, string> kategoriBilangan;

    for (int i = 1; i <= 10; ++i) {
        kategoriBilangan[i] = ganjilGenap(i);
    }

    for (const auto& pair : kategoriBilangan) {
        cout << pair.first << " adalah bilangan " << pair.second
        << endl;
    }

    return 0;
}
```

Screenshot Program:

```
1 adalah bilangan ganjil
2 adalah bilangan genap
3 adalah bilangan ganjil
4 adalah bilangan genap
5 adalah bilangan ganjil
6 adalah bilangan genap
7 adalah bilangan ganjil
8 adalah bilangan genap
9 adalah bilangan ganjil
10 adalah bilangan genap
PS D:\Data\Kuliah\Semester 2\Praktikum Struktur Data dan Algoritma\Modul 1>
```

Deskripsi Program:

1. Program membuat sebuah map dengan tipe kunci (key) `int` dan tipe nilai (value) `string` untuk menyimpan kategori setiap bilangan.
2. Program menggunakan loop `for` untuk mengisi map dengan bilangan dari 1 hingga 10, serta menentukan kategori masing-masing bilangan menggunakan fungsi `ganjilGenap` untuk mengecek apakah bilangan tersebut ganjil atau genap.

3. Setelah itu, program menampilkan kategori dari setiap bilangan yang telah dimasukkan ke dalam map.

Perbedaan antara array dan map:

1. Struktur Data: Array adalah struktur data yang menyimpan elemen-elemen berurutan dengan indeks numerik, sedangkan map adalah struktur data yang menyimpan pasangan nilai dan kunci.
2. Indeks: Pada array, elemen-elemen diakses menggunakan indeks numerik, sedangkan pada map, nilai diakses menggunakan kunci yang unik.
3. Keterbatasan pada Kunci: Pada array, kunci (indeks) harus berupa bilangan bulat non-negatif yang berurutan, sedangkan pada map, kunci bisa berupa tipe data apapun yang mendukung operasi perbandingan.
4. Penggunaan: Array biasanya digunakan ketika jumlah elemen yang akan disimpan diketahui sebelumnya dan ketika akses ke elemen harus dilakukan secara efisien berdasarkan indeks numerik. Sedangkan map biasanya digunakan ketika data tidak tersusun secara teratur dan ketika akses ke elemen berdasarkan kunci unik.

DAFTAR PUSTAKA

Asisten Praktikum. 2024. “*MODUL I TIPE DATA*”. Learning Management System.

Andre. 2020. “*Tutorial Belajar C++ Part 11: Jenis-jenis Tipe Data dalam Bahasa C++*”. Diakses 9 Maret 2024. <https://www.duniaikom.com/tutorial-belajar-c-plus-plus-jenis-jenis-tipe-data-dalam-bahasa-c-plus-plus/>

Dicoding Intern. 2020. “*Macam-Macam Tipe Data Pemrograman Beserta Fungsinya*”. Diakses 9 Maret 2024. <https://www.dicoding.com/blog/macam-macam-tipe-data/>