

LAPORAN PRAKTIKUM
PERTEMUAN 3
MODUL 3 STD



Nama :

Farrel I'zaz Yuwono (2311104014)

Dosen :

YUDHA ISLAMI SULISTYA

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Tugas Pendahuluan

1. Pointer adalah variabel yang menyimpan alamat memori dari variabel lain. Dalam bahasa C++, pointer memungkinkan akses langsung ke lokasi memori, yang memungkinkan manipulasi variabel secara lebih fleksibel.

2.

```
int main() {  
    int a = 10; cout << "Alamat memori dari variabel a adalah: " << &a << endl;  
    return 0;  
}
```

3.

```
int a = 20;  
int *ptr = &a;  
cout << "Nilai dari a: " << *ptr << endl;  
return 0;
```

4. Abstract Data Type (ADT) adalah konsep matematika yang mendefinisikan tipe data berdasarkan operasi yang didukung dan sifat-sifatnya, tanpa memperhatikan implementasi. ADT fokus pada bagaimana data dapat diakses dan dioperasikan, bukan bagaimana data disimpan secara fisik dalam memori.

5. Pada antrian di bank yang Dimana siapa yang lebih dulu datang, akan di layani lebih dulu juga.

6.

```
class Kerucut {  
private:  
    double radius;  
    double tinggi;  
  
public:  
    Kerucut(double r, double t) : radius(r), tinggi(t) {}
```

```
double volume() {  
    return (1.0 / 3) * M_PI * radius * radius * tinggi;  
}  
  
double luasPermukaan() {  
    double s = sqrt(radius * radius + tinggi * tinggi);  
    return M_PI * radius * (radius + s);  
}  
};  
  
int main() {  
    Kerucut k(5.0, 12.0);  
    cout << "Volume kerucut: " << k.volume() << endl;  
    cout << "Luas permukaan kerucut: " << k.luasPermukaan() << endl;  
    return 0;  
}
```

LATIHAN

1.

```
#include <iostream>

#include <string>

using namespace std;

struct Mahasiswa {

    string nama;

    string nim;

    float uts, uas, tugas, nilaiAkhir;

};

// Fungsi untuk menghitung nilai akhir

float hitungNilaiAkhir(float uts, float uas, float tugas) {

    return (0.3 * uts) + (0.4 * uas) + (0.3 * tugas);

}

// Fungsi untuk input data mahasiswa

void inputMahasiswa(Mahasiswa &mhs) {

    cout << "Masukkan nama: ";

    cin >> mhs.nama;

    cout << "Masukkan NIM: ";

    cin >> mhs.nim;

    cout << "Masukkan nilai UTS: ";

    cin >> mhs.uts;

    cout << "Masukkan nilai UAS: ";

    cin >> mhs.uas;

    cout << "Masukkan nilai Tugas: ";

    cin >> mhs.tugas;

    mhs.nilaiAkhir = hitungNilaiAkhir(mhs.uts, mhs.uas, mhs.tugas);

}
```

```
// Fungsi untuk menampilkan data mahasiswa
void tampilMahasiswa(Mahasiswa mhs) {
    cout << "Nama: " << mhs.nama << endl;
    cout << "NIM: " << mhs.nim << endl;
    cout << "Nilai UTS: " << mhs.uts << endl;
    cout << "Nilai UAS: " << mhs.uas << endl;
    cout << "Nilai Tugas: " << mhs.tugas << endl;
    cout << "Nilai Akhir: " << mhs.nilaiAkhir << endl;
}

int main() {
    Mahasiswa mhs[10];
    int jumlahMahasiswa;

    cout << "Masukkan jumlah mahasiswa (maks. 10): ";
    cin >> jumlahMahasiswa;

    for (int i = 0; i < jumlahMahasiswa; i++) {
        cout << "\nData Mahasiswa ke-" << i+1 << endl;

        inputMahasiswa(mhs[i]);
    }

    cout << "\nData Mahasiswa yang sudah diinput: " << endl;

    for (int i = 0; i < jumlahMahasiswa; i++) {
        cout << "\nMahasiswa ke-" << i+1 << endl;

        tampilMahasiswa(mhs[i]);
    }

    return 0;
}
```

Data Mahasiswa yang sudah diinput:

```
Mahasiswa ke-1
Nama: Izaz
NIM: 23
Nilai UTS: 80
Nilai UAS: 90
Nilai Tugas: 76
Nilai Akhir: 82.8
```

```
Mahasiswa ke-2
Nama: farrel
NIM: 24
Nilai UTS: 67
Nilai UAS: 89
Nilai Tugas: 99
Nilai Akhir: 85.4
```

2.

Pelajaran.h

```
#ifndef PELAJARAN_H_INCLUDED
#define PELAJARAN_H_INCLUDED

#include <string>

using namespace std;

struct Pelajaran {

    string namaMapel;

    string kodeMapel;

};

Pelajaran create_pelajaran(string namaMapel,
string kodeMapel);

void tampil_pelajaran(Pelajaran pel);

#endif
```

Pelajaran.cpp

```
#include "pelajaran.h"

#include <iostream>

using namespace std;

Pelajaran create_pelajaran(string namaMapel, string
kodeMapel) {

    Pelajaran pel;

    pel.namaMapel = namaMapel;

    pel.kodeMapel = kodeMapel;

    return pel;

}

void tampil_pelajaran(Pelajaran pel) {

    cout << "Nama Mata Pelajaran: " << pel.namaMapel <<
endl;

    cout << "Kode Mata Pelajaran: " << pel.kodeMapel <<
endl;

}
```

main.cpp

```
#include <iostream>

#include "pelajaran.h"

using namespace std;

int main() {

    string namapel = "Struktur Data";

    string kodepel = "STD";

    Pelajaran pel = create_pelajaran(namapel, kodepel);

    tampil_pelajaran(pel);

    return 0;

}
```

```
PS D:\kuliah\strukturdat\tugas kuliah\03_Abstract_Data_Type> Set-Location "D:\kuliah\strukturdat\tugas kuliah\03_Abstract_Data_Type"
PS D:\kuliah\strukturdat\tugas kuliah\03_Abstract_Data_Type> g++ main.cpp pelajaran.cpp -o program
PS D:\kuliah\strukturdat\tugas kuliah\03_Abstract_Data_Type> ./program
Nama Mata Pelajaran: Struktur Data
Kode Mata Pelajaran: STD
PS D:\kuliah\strukturdat\tugas kuliah\03_Abstract_Data_Type>
```

3.

```
#include <iostream>

using namespace std;

void tampilArray(int arr[3][3])
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            cout << arr[i][j] << " ";
        }

        cout << endl;
    }
}

void tukarArray(int arr1[3][3], int arr2[3][3], int x, int y)
{
    int temp = arr1[x][y];
    arr1[x][y] = arr2[x][y];
    arr2[x][y] = temp;
}

void tukarPointer(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

int main()
{
    int arr1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int arr2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};

    cout << "Array 1 sebelum pertukaran:" << endl;
    tampilArray(arr1);

    cout << "Array 2 sebelum pertukaran:" << endl;
    tampilArray(arr2);

    tukarArray(arr1, arr2, 1, 1);
}
```

```

cout << "\nArray 1 setelah pertukaran:" << endl;

tampilArray(arr1);

cout << "Array 2 setelah pertukaran:" << endl;

tampilArray(arr2);


int a = 10, b = 20;

int *ptrA = &a;

int *ptrB = &b;


cout << "\nSebelum pertukaran pointer: a = " << *ptrA << ", b = " << *ptrB << endl;

tukarPointer(ptrA, ptrB);

cout << "Setelah pertukaran pointer: a = " << *ptrA << ", b = " << *ptrB << endl;


return 0;

}

```

Penjelasan Tugas Latihan :

1. Struct Mahasiswa: Struktur data yang digunakan untuk menyimpan informasi tentang seorang mahasiswa, termasuk nama, NIM, nilai UTS, UAS, tugas, dan nilai akhir.

Fungsi hitungNilaiAkhir: Fungsi ini digunakan untuk menghitung nilai akhir mahasiswa berdasarkan bobot nilai UTS (30%), UAS (40%), dan tugas (30%).

Fungsi inputMahasiswa: Fungsi ini meminta pengguna untuk memasukkan data mahasiswa, termasuk nama, NIM, dan nilai-nilai (UTS, UAS, tugas). Setelah data diinput, fungsi ini langsung menghitung nilai akhir menggunakan hitungNilaiAkhir.

Fungsi tampilMahasiswa: Fungsi ini digunakan untuk menampilkan informasi lengkap tentang seorang mahasiswa, termasuk nilai-nilainya dan nilai akhir.

Fungsi main:

- Meminta jumlah mahasiswa yang akan diinput (maksimal 10).
- Menggunakan loop untuk menginput data mahasiswa satu per satu.
- Setelah semua data mahasiswa dimasukkan, program akan menampilkan seluruh data mahasiswa yang sudah diinput.

2. File pelajaran.cpp

Berisi implementasi fungsi `create_pelajaran` yang digunakan untuk membuat objek mata pelajaran dengan nama dan kode.

Fungsi `tampil_pelajaran` digunakan untuk menampilkan informasi mata pelajaran ke layar (pelajaran).

File pelajaran.h

Mendefinisikan struktur `Pelajaran` yang berisi dua atribut: `namaMapel` (nama mata pelajaran) dan `kodeMapel` (kode mata pelajaran).

Deklarasi fungsi `create_pelajaran` dan `tampil_pelajaran` juga ada di sini sebagai antarmuka dari file header (pelajaran).

File main.cpp

Program utama yang menggunakan fungsi dari file header dan file implementasi.

Membuat objek mata pelajaran dengan nama "Struktur Data" dan kode "STD", kemudian menampilkannya menggunakan fungsi yang telah dibuat (main).

3. Fungsi `tampilArray`: Menampilkan elemen-elemen dari array 3x3 ke layar dalam bentuk matriks.

Fungsi `tukarArray`: Menukar nilai elemen tertentu antara dua array 3x3 berdasarkan posisi yang diberikan.

Fungsi `tukarPointer`: Menukar nilai dari dua variabel menggunakan pointer.

Fungsi main:

Mendefinisikan dua array 3x3 dan menampilkan nilainya sebelum dan sesudah pertukaran.

Menggunakan fungsi `tukarArray` untuk menukar elemen array pada posisi tertentu.

Menukar nilai dua variabel menggunakan pointer dan menampilkan hasilnya.