

PRAKTIKUM STRUKTUR DATA

TUGAS PENDAHULUAN 3



Nama :

Aulia Ahmad Ghaus Adzam (2311104028)

Dosen :

Yudha Islami Sulistya, S.
Kom.,M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

B. Soal Tugas Pendahuluan

1. Jelaskan Apa Yang Dimaksud Dengan Pointer

Pointer adalah sebuah variable, dan variable itu menyimpan suatu Alamat variable lain, dan hal ini membuat penggunaan memori menjadi lebih efisien, namun kalo salah digunakan juga dapat menyebabkan kebocoran memori.

2. Bagaimana cara menampilkan Alamat memori dari suatu variable dalam program C++? Berikan Contoh!

Untuk menampilkan Alamat memori dari suatu variable kita bisa menggunakan `&` dengan menggunakan itu kita bisa tau berada di Alamat mana suatu variable kek gini contohnya:



```
SOAL_02.cpp
#include <iostream>

using namespace std;
// Cara Tau Alamat Suatu Variabel

int main(){
    int a = 10;

    cout << "Alamat Dari Variabel A: " << &a << endl;

    return 0;
}
```

Output:

```
Alamat Dari Variabel A: 0x5ffe9c
```

3. Bagaimana Cara Menggunakan Pointer Dalam Program C++? Berikan Contoh!

A screenshot of a code editor window titled 'SOAL_03.cpp'. The code is in C++ and demonstrates pointer usage. It includes the <iostream> header, uses the std namespace, and defines a main function. Inside main, an integer 'a' is initialized to 10, and a pointer 'ptr' of type 'int*' is initialized to point to 'a'. The program then prints the value of 'a', the address of 'ptr', and the value pointed to by 'ptr' (*ptr). After setting *ptr to 50, it prints the value of 'a' again. The output shows that 'a' remains 10, while *ptr becomes 50.

```
#include <iostream>

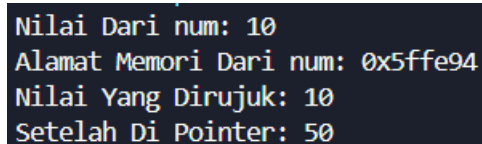
using namespace std;
// Contoh Penggunaan Pointer and Friends

int main(){
    int a = 10;
    int* ptr = &a;

    cout << "Nilai Dari num: " << a << endl;
    cout << "Alamat Memori Dari num: " << ptr << endl;
    cout << "Nilai Yang Dirujuk: " << *ptr << endl;
    *ptr = 50;
    cout << "Setelah Di Pointer: " << a << endl;

    return 0;
}
```

Untuk memakai pointer mudah hanya kita tambahkan *di awal variable pada variable yang telah dirujuk contoh pada gambar di atas variable a nilai nya 10 lalu kita buat variable int* ptr yang berisi kan Alamat dari variable a lalu karena telah terhubung kedua variable ini maka kita bisa mengakses isi value variable a melalui pointer *ptr.

A screenshot of the program's output in a terminal window. It shows four lines of text: 'Nilai Dari num: 10', 'Alamat Memori Dari num: 0x5ffe94', 'Nilai Yang Dirujuk: 10', and 'Setelah Di Pointer: 50'. The output demonstrates that the pointer 'ptr' correctly points to the memory address of 'a' and that changing the value through the pointer (*ptr) does not affect the original variable 'a'.

```
Nilai Dari num: 10
Alamat Memori Dari num: 0x5ffe94
Nilai Yang Dirujuk: 10
Setelah Di Pointer: 50
```

4. Jelaskan Apa Yang dimaksud dengan Abstract Data Type (ADT)

ADT adalah pengerjaan sebuah program melalui tahap yang fleksibel dan reusable dan sangat menghindari metode *DRY (Don't Repeat your code)* dengan cara membuat banyak class pengerjaan secara terpisah terpisah namun program itu dapat berjalan dengan sempurna.

5. Berikan Contoh ilustrasi sederhana di dalam dunia nyata, tetapi di luar konteks pemrograman!

Aplikasi Dana Yang user itu bisa topup game terus bayar tagihan Listrik, transfer bank sini sana dan masih Banyak lagi yang bisa digunakan oleh pengguna tanpa mereka perlu tahu gimana cara kerja di balik layar. Fokusnya adalah pada fungsi dan perilaku yang ditawarkan, bukan implementasi detailnya, persis seperti prinsip ADT dalam pemrograman.

6. Tuliskan ADT dari bangun ruang kerucut dalam Bahasa C++!

```
SOAL_06.cpp

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

class Kerucut {
private:
    double radius;
    double tinggi;

public:
    Kerucut(double r, double t) {
        radius = r;
        tinggi = t;
    }

    double hitungVolume() {
        return (1.0 / 3.0) * M_PI * radius * radius * tinggi;
    }

    double hitungLuasPermukaan() {
        double s = sqrt((radius * radius) + (tinggi * tinggi));
        return M_PI * radius * (radius + s);
    }

    double getRadius() {
        return radius;
    }

    double getTinggi() {
        return tinggi;
    }

    void setRadius(double r) {
        radius = r;
    }

    void setTinggi(double t) {
        tinggi = t;
    }
};

int main() {
    Kerucut kerucut1(5, 10);

    cout << "Volume kerucut: " << kerucut1.hitungVolume() << endl;

    cout << "Luas permukaan kerucut: " << kerucut1.hitungLuasPermukaan() << endl;

    return 0;
}
```

Volume kerucut: 261.799
Luas permukaan kerucut: 254.16

C. Latihan Unguided

1. Buat program yang dapat menyimpan data mahasiswa (max. 10) ke dalam sebuah array dengan field nama, nim, uts, uas, tugas, dan nilai akhir. Nilai akhir diperoleh dari FUNGSI dengan rumus $0.3*uts + 0.4*uas + 0.3*tugas$.

```
LATIHAN_01.cpp

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

struct Mahasiswa
{
    string nama;
    string nim;
    float uts;
    float uas;
    float tugas;
    float nilaiAkhir;
};

float hitungNilaiAkhir(float uts, float uas, float tugas) {
    return (0.3*uts, 0.4*uas, 0.3*tugas);
}

int main(){
    const int MAKS = 10;
    Mahasiswa mahasiswa[MAKS];
    int jumlahMahasiswa;

    cout << "Masukkan Jumlah Mahasiswa : ";
    cin >> jumlahMahasiswa;

    if(jumlahMahasiswa > MAKS)
    {
        cout << "Jumlah Maksimal 10" << endl;
        return 1;
    }

    for(int i = 0; i < jumlahMahasiswa; i++) {
        cout << "\n Masukan Mahasiswa ke " << i+1 << endl;
        cout << "Nama: ";
        cin.ignore();
        getline(cin, mahasiswa[i].nama);
        cout << "NIM: ";
        cin >> mahasiswa[i].nim;
        cout << "Nilai Tugas: ";
        cin >> mahasiswa[i].tugas;
        cout << "Nilai UTS: ";
        cin >> mahasiswa[i].uts;
        cout << "Nilai UAS: ";
        cin >> mahasiswa[i].uas;

        mahasiswa[i].nilaiAkhir = hitungNilaiAkhir(mahasiswa[i].uts, mahasiswa[i].uas, mahasiswa[i].tugas);
    }

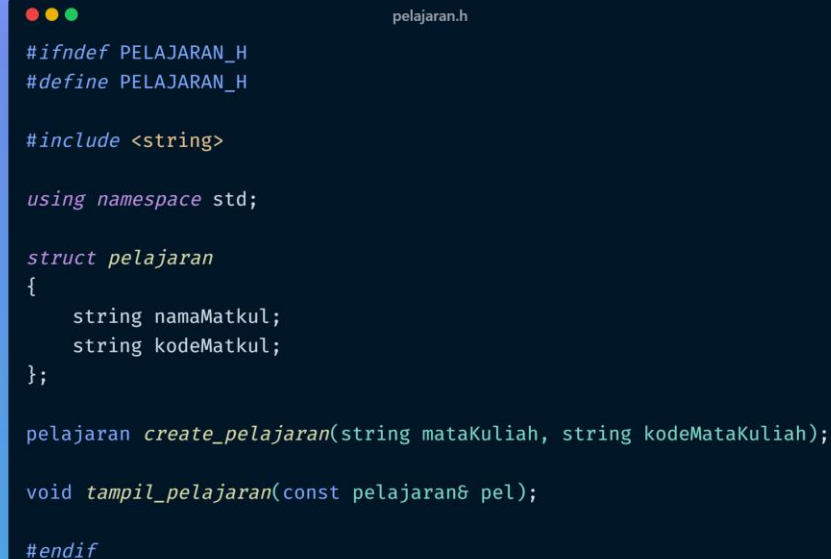
    cout << "\nHasil Akhir Nilai Mahasiwa\n";
    for (int i = 0; i < jumlahMahasiswa; i++)
    {
        cout << "Mahasiswa Ke- " << i+1 << endl;
        cout << "Nama: " << mahasiswa[i].nama << endl;
        cout << "NIM: " << mahasiswa[i].nim << endl;
        cout << "Nilai Tugas: " << mahasiswa[i].tugas << endl;
        cout << "Nilai UTS: " << mahasiswa[i].uts << endl;
        cout << "Nilai UAS: " << mahasiswa[i].uas << endl;
        cout << "Nilai Akhir: " << mahasiswa[i].nilaiAkhir << endl;
    }

    return 0;
}
```

Penjelasan pertama kita buat dulu data struct mahasiswa yang terdiri dari nama, nim, nilai tugas, uts, uas. Setelah itu kita buat function hitung nilai tugas akhir parameter nya isian nya variable uts, uas, dan tugas, dan parameter menyimpan rumus $0.3 * \text{uts}$, $0.4 * \text{uas}$, $0.3 * \text{tugas}$. Lalu kita buat input data2 mahasiswa nya seperti nama, nim, nilai tugas, uts dan uas lalu kita masukan ake array mahasiswa[i] yang bertipe konstanta 10 lalu kita hubungkan dengan function nilai akhir dan berisikan hasil input user untuk nilai tugas, uts, dan uas. Maka output nya akan seperti ini kurang lebih:

```
Hasil Akhir Nilai Mahasiswa
Mahasiswa Ke- 1
Nama: Aulia Ahmad
NIM: 2311104028
Nilai Tugas: 98
Nilai UTS: 78
Nilai UAS: 85
Nilai Akhir: 29.4
```

2. Hasil Jawaban no2 :



```
pelajaran.h

#ifndef PELAJARAN_H
#define PELAJARAN_H

#include <string>

using namespace std;

struct pelajaran
{
    string namaMatkul;
    string kodeMatkul;
};

pelajaran create_pelajaran(string mataKuliah, string kodeMataKuliah);

void tampil_pelajaran(const pelajaran& pel);

#endif
```

File – pelajaran.h

```

pelajaran.cpp

#include "pelajaran.h"
#include <iostream>

pelajaran create_pelajaran(string mataKuliah, string kodeMataKuliah) {
    pelajaran pel;
    pel.namaMatkul = mataKuliah;
    pel.kodeMatkul = kodeMataKuliah;
    return pel;
}

void tampil_pelajaran(const pelajaran& pel) {
    cout << "Nama Mata Kuliah: " << pel.namaMatkul << endl;
    cout << "Kode Mata Kuliah: " << pel.kodeMatkul << endl;
}

```

File – Pelajaran.cpp

```

main.cpp

#include <iostream>
#include "pelajaran.h"

using namespace std;

int main() {
    string mataKuliah = "Struktur Data";
    string kodeMataKuliah = "STD";

    pelajaran pel = create_pelajaran(mataKuliah, kodeMataKuliah);

    tampil_pelajaran(pel);

    return 0;
}

```

File – main.cpp

pelajaran.h: File header yang mendefinisikan tipe data ADT pelajaran serta deklarasi fungsi dan prosedur terkait. pelajaran.cpp: Implementasi dari fungsi `create_pelajaran` untuk membuat instance pelajaran, dan `tampil_pelajaran` untuk menampilkan isi dari

pelajaran. main.cpp: Program utama yang menguji implementasi ADT pelajaran dengan menggunakan dua string, yaitu namapel dan kodepel.

```
nama pelajaran : Struktur Data
nilai : STD
```

3. Buatlah program dengan ketentuan :
 - 2 buah array 2D integer berukuran 3x3 dan 2 buah pointer integer
 - fungsi/prosedur yang menampilkan isi sebuah array integer 2D
 - fungsi/prosedur yang akan menukarkan isi dari 2 array integer 2D pada posisi tertentu
 - fungsi/prosedur yang akan menukarkan isi dari variabel yang ditunjuk oleh 2 buah pointer

```
LATIHAN_03.cpp
#include <iostream>
using namespace std;

void tampilkanArray(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 tukarArrayPadaPosisi(int arr1[3][3], int arr2[3][3], int row, int col) {
    int temp = arr1[row][col];
    arr1[row][col] = arr2[row][col];
    arr2[row][col] = temp;
}

void tukarPointer(int* ptr1, int* ptr2) {
    int temp = *ptr1;
    *ptr1 = *ptr2;
    *ptr2 = temp;
}

int main() {
    int array1[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int array2[3][3] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};

    int a = 10, b = 20;
    int* ptr1 = &a;
    int* ptr2 = &b;

    cout << "Array 1 sebelum pertukaran:" << endl;
    tampilkanArray(array1);
    cout << "\nArray 2 sebelum pertukaran:" << endl;
    tampilkanArray(array2);
    tukarArrayPadaPosisi(array1, array2, 1, 1);

    cout << "\nArray 1 setelah pertukaran pada posisi [1][1]: " << endl;
    tampilkanArray(array1);
    cout << "\nArray 2 setelah pertukaran pada posisi [1][1]: " << endl;
    tampilkanArray(array2);

    cout << "\nNilai sebelum pertukaran pointer:" << endl;
    cout << "ptr1 menunjuk ke: " << *ptr1 << endl;
    cout << "ptr2 menunjuk ke: " << *ptr2 << endl;
    tukarPointer(ptr1, ptr2);

    cout << "\nNilai setelah pertukaran pointer:" << endl;
    cout << "ptr1 menunjuk ke: " << *ptr1 << endl;
    cout << "ptr2 menunjuk ke: " << *ptr2 << endl;

    return 0;
}
```


Fungsi tampilkanArray: Menampilkan isi array 2D. Fungsi tukarArrayPadaPosisi: Menukar elemen pada posisi tertentu antara dua array 2D (misalnya pada posisi [1][1]). Fungsi tukarPointer: Menukar nilai yang ditunjuk oleh dua pointer.
Output:

```
Nilai sebelum pertukaran pointer:  
ptr1 menunjuk ke: 10  
ptr2 menunjuk ke: 20  
  
Nilai setelah pertukaran pointer:  
ptr1 menunjuk ke: 20  
ptr2 menunjuk ke: 10
```