

Nama : Andre Christian Manurung

NIM : 25/559730/TK/63191

Laporan Tugas Struct and Union

1. Exercises no. 6:

Consider the following statements:

```
struct nameType      struct courseType      struct studentType
{
    string first;
    string last;
};

{
    string name;
    int callNum;
    int credits;
    char grade;
};

{
    nameType name;
    double gpa;
    courseType course;
};

studentType student;
studentType classList[100];
courseType course;
nameType name;
```

Mark the following statements as valid or invalid. If a statement is invalid, explain why.

- a. `student.course.callNum = "CSC230";`
- b. `cin >> student.name;`
- c. `classList[0] = name;`
- d. `classList[1].gpa = 3.45;`
- e. `name = classList[15].name;`
- f. `student.name = name;`
- g. `cout << classList[10] << endl;`
- h. `for (int j = 0; j < 100; j++)`
 `classList[j].name = name;`
- i. `classList.course.credits = 3;`
- j. `course = studentType.course;`

- a. `student.course.callNum = "CSC230";` **Invalid**. Terjadi ketidakcocokan tipe data (type mismatch). Member `callNum` didefinisikan sebagai `int` (integer), tetapi pernyataan tersebut mencoba memberikan nilai `"CSC230"` yang merupakan string (teks).
- b. `cin >> student.name;` **Invalid**. Operator input `cin >>` tidak dapat digunakan untuk membaca nilai ke seluruh variabel struct (`nameType`) secara langsung. Input harus dilakukan per member, misalnya `cin >> student.name.first` dan `cin >> student.name.last`.
- c. `classList[0] = name;` **Invalid**. Terjadi ketidakcocokan tipe data. `classList[0]` adalah variabel bertipe `studentType`, sedangkan `name` adalah variabel bertipe `nameType`. Assignment tidak dapat dilakukan antara dua tipe data struct yang berbeda.
- d. `classList[1].gpa = 3.45;` **Valid**. Member `classList[1].gpa` bertipe `double`, dan nilai `3.45` juga merupakan `double`. Tipe data sudah sesuai.

- e. `name = classList[15].name;` **Valid**. Variabel `name` dan member `classList[15].name` keduanya bertipe `nameType`. Assignment dapat dilakukan karena kedua variabel memiliki tipe data yang sama.
 - f. `student.name = name;` **Valid**. Member `student.name` dan variabel `name` keduanya bertipe `nameType`. Tipe data sudah sesuai untuk assignment.
 - g. `cout << classList[10] << endl;` **Invalid**. Operator output `cout <<` tidak dapat digunakan untuk menampilkan seluruh isi variabel struct (`studentType`) secara langsung. Data harus ditampilkan per member, misalnya `cout << classList[10].name.first;`.
 - h. `for (int j = 0; j < 100; j++) classList[j].name = name;` **Valid**. Pernyataan di dalam loop (`classList[j].name = name;`) adalah assignment yang valid (tipe `nameType` ke `nameType`), dan sintaksis for loop itu sendiri sudah benar.
 - i. `classList.course.credits = 3;` **Invalid**. Pernyataan ini memiliki dua kesalahan: `classList` adalah sebuah array dan memerlukan indeks (misalnya `classList[0]`) untuk mengakses salah satu elemennya. Struct `courseType` (yang merupakan tipe dari `classList[...].course`) tidak memiliki member bernama `credits`.
 - j. `course = studentType.course;` **Invalid**. `studentType` adalah nama dari tipe data (definisi struct), bukan nama variabel. Tidak bisa mengakses member dari sebuah nama tipe data. Seharusnya menggunakan nama variabel yang sudah dideklarasikan, contohnya: `course = student.course;`.
2. Programming exercise no. 2 student test score :
- Write a program that reads students' names followed by their test scores. The program should output each student's name followed by the test scores and the relevant grade. It should also find and print the highest test score and the name of the students having the highest test score.
- Student data should be stored in a **struct** variable of type **studentType**, which has four components: **studentFName** and **studentLName** of type **string**, **testScore** of type **int** (**testScore** is between 0 and 100), and **grade** of type **char**. Suppose that the class has 20 students. Use an array of 20 components of type **studentType**.
- Your program must contain at least the following functions:
- a. A function to read the students' data into the array.
 - b. A function to assign the relevant grade to each student.
 - c. A function to find the highest test score.
 - d. A function to print the names of the students having the highest test score.

Penjelasan Kode:

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

digunakan sebagai bagian awal program untuk menyiapkan fungsi dasar C++.

```
struct Mahasiswa {
```

```
    string namaDepan;
```

```
    string namaBelakang;
```

```
    int nilaiTes;
```

```
    char nilaiHuruf;
```

```
};
```

Kode ini untuk mendefinisikan tipe data baru bernama Mahasiswa.

Di dalamnya terdapat empat variabel anggota:

namaDepan dan namaBelakang bertipe string untuk menyimpan nama mahasiswa,

nilaiTes bertipe int untuk menyimpan nilai ujian,

dan nilaiHuruf bertipe char untuk menyimpan nilai dalam bentuk huruf

```
void inputData(Mahasiswa m[], int jumlah) {
```

```
    for (int i = 0; i < jumlah; i++) {
```

```
        cout << "Masukkan nama depan, nama belakang, dan nilai: ";
```

```
        cin >> m[i].namaDepan >> m[i].namaBelakang >> m[i].nilaiTes;
```

```
    }
```

```
}
```

Fungsi inputData digunakan untuk memasukkan data mahasiswa ke dalam array m.

Melalui perulangan for, program meminta pengguna mengetik nama depan, nama belakang, dan nilai tes untuk setiap mahasiswa sebanyak jumlah yang ditentukan, lalu menyimpannya langsung ke dalam elemen array m[i].

```
void beriNilaiHuruf(Mahasiswa m[], int jumlah) {
```

```
    for (int i = 0; i < jumlah; i++) {
```

```
        if (m[i].nilaiTes >= 90) m[i].nilaiHuruf = 'A';
```

```
        else if (m[i].nilaiTes >= 80) m[i].nilaiHuruf = 'B';
```

```
        else if (m[i].nilaiTes >= 70) m[i].nilaiHuruf = 'C';
```

```
        else if (m[i].nilaiTes >= 60) m[i].nilaiHuruf = 'D';
```

```
        else m[i].nilaiHuruf = 'F';
```

```
    }
```

Fungsi beriNilaiHuruf digunakan untuk menentukan nilai huruf setiap mahasiswa berdasarkan nilai tes mereka. Di dalam perulangan for, program membandingkan nilai tiap mahasiswa dengan rentang nilai tertentu, lalu memberi huruf 'A' sampai 'F' sesuai kriteria. Nilai huruf ini kemudian disimpan dalam variabel nilaiHuruf milik masing-masing mahasiswa.

```
}
```

```
int cariNilaiTertinggi(Mahasiswa m[], int jumlah) {  
    int maxNilai = m[0].nilaiTes;  
    for (int i = 1; i < jumlah; i++)  
        if (m[i].nilaiTes > maxNilai)  
            maxNilai = m[i].nilaiTes;  
    return maxNilai;  
}
```

Fungsi cariNilaiTertinggi digunakan untuk mencari nilai tes paling tinggi dari seluruh mahasiswa. Nilai pertama (m[0].nilaiTes) dijadikan acuan awal, lalu program membandingkannya dengan nilai mahasiswa lain satu per satu menggunakan perulangan for. Jika ditemukan nilai yang lebih besar, maka nilai tersebut disimpan sebagai maxNilai, dan di akhir fungsi, nilai tertinggi ini dikembalikan (return) ke pemanggil fungsi.

```
void tampilkanTertinggi(Mahasiswa m[], int jumlah, int nilaiTertinggi) {  
    cout << "\nNilai tertinggi: " << nilaiTertinggi << endl;  
    cout << "Diperoleh oleh:\n";  
    for (int i = 0; i < jumlah; i++)  
        if (m[i].nilaiTes == nilaiTertinggi)  
            cout << m[i].namaDepan << " " << m[i].namaBelakang << endl;  
}
```

Fungsi tampilkanTertinggi digunakan untuk menampilkan nilai tertinggi dan nama mahasiswa yang mendapatkannya.

Pertama, program mencetak nilai tertinggi yang sudah ditemukan sebelumnya, lalu melalui perulangan for, program mengecek setiap data mahasiswa. Jika nilai tes mahasiswa sama dengan nilai tertinggi, maka nama depan dan nama belakang mahasiswa tersebut akan ditampilkan di layar.

```
int main() {  
    const int jumlah = 20;  
    Mahasiswa mhs[jumlah];  
    inputData(mhs, jumlah);  
    beriNilaiHuruf(mhs, jumlah);  
    int tertinggi = cariNilaiTertinggi(mhs, jumlah);  
    tampilkanTertinggi(mhs, jumlah, tertinggi);  
    cout << "\nData lengkap mahasiswa:\n";  
    for (int i = 0; i < jumlah; i++) {  
        cout << mhs[i].namaDepan << " "  
            << mhs[i].namaBelakang << " "  
            << mhs[i].nilaiTes << " "  
            << mhs[i].nilaiHuruf << endl;  
    }  
}
```

```
    return 0;  
}
```

Fungsi `main()` adalah bagian utama program tempat semua proses dijalankan. Pertama, program menentukan jumlah mahasiswa dan membuat array `mhs` untuk menampung datanya. Kemudian fungsi `inputData` digunakan untuk memasukkan nama dan nilai, dilanjutkan `beriNilaiHuruf` untuk memberi nilai huruf pada tiap mahasiswa. Setelah itu, `cariNilaiTertinggi` mencari nilai terbesar, dan `tampilkanTertinggi` menampilkan siapa yang mendapat nilai tersebut. Terakhir, program mencetak seluruh data mahasiswa beserta nilai hurufnya sebelum berakhir dengan `return 0;` yang menandakan program berjalan sukses.

Lampiran

Link GitHub : https://github.com/manurungandre1927-cloud/student_testScore

Hasil program, Input:

```
PS C:\Users\&re\OneDrive\Dokumen\00TUGAS\C> g++ student_testScore.cpp -o student_testScore
PS C:\Users\&re\OneDrive\Dokumen\00TUGAS\C> ./student_testScore.exe
Masukkan nama depan, nama belakang, dan nilai: Andre Christian 99
Masukkan nama depan, nama belakang, dan nilai: Budi Sibudi 76
Masukkan nama depan, nama belakang, dan nilai: Cecep Sicecep 66
Masukkan nama depan, nama belakang, dan nilai: Dani Sidani 98
Masukkan nama depan, nama belakang, dan nilai: Evan Sievan 51
Masukkan nama depan, nama belakang, dan nilai: Fadil Sifadil 77
Masukkan nama depan, nama belakang, dan nilai: Galang Sigalang 0
Masukkan nama depan, nama belakang, dan nilai: Harum Siharum 23
Masukkan nama depan, nama belakang, dan nilai: Ipul Siipul 73
Masukkan nama depan, nama belakang, dan nilai: Jamal Sijamal 83
Masukkan nama depan, nama belakang, dan nilai: Kevin Sikevin 93
Masukkan nama depan, nama belakang, dan nilai: Laura Silaura 86
Masukkan nama depan, nama belakang, dan nilai: Mamad Simamad 41
Masukkan nama depan, nama belakang, dan nilai: Opet Siopet 67
Masukkan nama depan, nama belakang, dan nilai: Petrus Sipetrus 88
Masukkan nama depan, nama belakang, dan nilai: Q Q 97
Masukkan nama depan, nama belakang, dan nilai: Rafi Sirafi 76
Masukkan nama depan, nama belakang, dan nilai: Steven Sisteven 87
Masukkan nama depan, nama belakang, dan nilai: Teman Teman 78
Masukkan nama depan, nama belakang, dan nilai: Ujang Siujang 84
```

Output:

```
Nilai tertinggi: 99
Diperoleh oleh:
Andre Christian

Data lengkap mahasiswa:
Andre Christian 99 A
Budi Sibudi 76 C
Cecep Sicecep 66 D
Dani Sidani 98 A
Evan Sievan 51 F
Fadil Sifadil 77 C
Galang Sigalang 0 F
Harum Siharum 23 F
Ipul Siipul 73 C
Jamal Sijamal 83 B
Kevin Sikevin 93 A
Laura Silaura 86 B
Mamad Simamad 41 F
Opet Siopet 67 D
Petrus Sipetrus 88 B
Q Q 97 A
Rafi Sirafi 76 C
Steven Sisteven 87 B
Teman Teman 78 C
Ujang Siujang 84 B
```