**PENJELASAN UTS NO.1**

Salsabila Nur Aqmarina Aulia

2209076041

UTS Struktur Data

Soal ini meminta kita untuk membuat sebuah program dalam bahasa pemrograman C atau C++ untuk mengelola data mahasiswa menggunakan *array* dan *pointer*. Berikut ini adalah penjelasan langkah-langkah darri penyelesaian program seperti pada soal :

**1. *Struct* Mahasiswa:**

Untuk membuat `*struct*` yang menyimpan data mahasiswa dengan tiga anggota:

- `NIM` (Nomor Induk Mahasiswa) bertipe `string` atau `char[]`

- `nama` bertipe `string` atau `char[]`

- `IPK` bertipe `float`

Pada C++ dibuat seperti :

struct Mahasiswa {

std::string NIM;

std::string nama;

float IPK;

};

```

**2. *Array* *of* *Pointers* untuk Menyimpan Data Mahasiswa:**

Untuk menggunakan array yang menyimpan pointer ke struct `Mahasiswa` untuk 10 mahasiswa. Kita bisa mendeklarasikan array pointer ini sebagai berikut:

Pada C++ dibuat seperti :

Mahasiswa\* dataMahasiswa[10];

```

Dengan *array pointer*, kita bisa mengelola data mahasiswa secara dinamis, sehingga ketika data dihapus, tidak perlu memindahkan seluruh data secara manual.

**3. Fungsi Menambah Data Mahasiswa :**

Fungsi ini menambahkan data mahasiswa baru ke dalam *array* jika masih ada ruang. Fungsi ini perlu membuat instance `Mahasiswa` baru, mengisi nilainya, dan menambahkannya ke *array pointer*.

Pada C++ dibuat seperti :

void tambahMahasiswa(Mahasiswa\* data[], int &jumlah, const std::string &NIM, const std::string &nama, float IPK) {

if (jumlah < 10) {

data[jumlah] = new Mahasiswa{NIM, nama, IPK};

jumlah++;

} else {

std::cout << "Data penuh!\n";

}

}

```

**4. Fungsi Menghapus Data Mahasiswa :**

Fungsi ini menghapus data mahasiswa berdasarkan NIM. Setelah menemukan mahasiswa dengan NIM yang cocok, fungsi akan menghapusnya dan menggeser elemen setelahnya.

Pada C++ dibuatkan seperti :

void hapusMahasiswa(Mahasiswa\* data[], int &jumlah, const std::string &NIM) {

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

if (data[i]->NIM == NIM) {

delete data[i];

for (int j = i; j < jumlah - 1; ++j) {

data[j] = data[j + 1];

}

data[jumlah - 1] = nullptr;

jumlah--;

std::cout << "Data mahasiswa dengan NIM " << NIM << " telah dihapus.\n";

return;

}

}

std::cout << "Data tidak ditemukan!\n";

}

```

**5. Fungsi Menampilkan Data Mahasiswa :**

Fungsi ini menampilkan seluruh data mahasiswa yang ada di dalam array.

Pada C++ dibuat Seperti :

void tampilkanMahasiswa(Mahasiswa\* data[], int jumlah) {

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

std::cout << "NIM: " << data[i]->NIM << ", Nama: " << data[i]->nama << ", IPK: " << data[i]->IPK << '\n';

}

}

```

**6. Fungsi Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort) :**

Fungsi ini akan mengurutkan mahasiswa berdasarkan IPK dari yang tertinggi ke terendah menggunakan algoritma bubble sort.

Pada C++ dibuat seperti :

void urutkanMahasiswa(Mahasiswa\* data[], int jumlah) {

for (int i = 0; i < jumlah - 1; ++i) {

for (int j = 0; j < jumlah - i - 1; ++j) {

if (data[j]->IPK < data[j + 1]->IPK) {

Mahasiswa\* temp = data[j];

data[j] = data[j + 1];

data[j + 1] = temp;

}

}

}

}

```

**7. Main Program :**

kita dapat menambahkan kode untuk menguji seluruh fungsi di atas dan membuat program interaktif dengan menambahkan pilihan input untuk pengguna.

Contoh Struktur Program Utama :

int main() {

Mahasiswa\* dataMahasiswa[10] = {nullptr};

int jumlahMahasiswa = 0;

// Tambah data, hapus, tampilkan, atau urutkan menggunakan fungsi yang telah dibuat

// Contoh penggunaan tambahMahasiswa:

tambahMahasiswa(dataMahasiswa, jumlahMahasiswa, "12345", "Ali", 3.9);

tampilkanMahasiswa(dataMahasiswa, jumlahMahasiswa);

urutkanMahasiswa(dataMahasiswa, jumlahMahasiswa);

// Jangan lupa membersihkan memory setelah selesai

for (int i = 0; i < jumlahMahasiswa; ++i) {

delete dataMahasiswa[i];

}

return 0;

}

```