Modul Praktikum

Alpro Lanjut

3

STRUCT

DASAR TEORI

Struct adalah pengelompokan variabel-variabel yang bernaung dalam satu nama yang sama. Berbeda dengan array yang berisi sekumpulan data yang bertipe sama dalam satu nama. Maka suatu struct dapat terdiri atas variabel-variabel yang berbeda tipenya dalam satu nama struct. Struct biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi sebuah kesatuan (dalam PASCAL struct disebut dengan record).

Variabel-variabel yang membentuk suatu struct, selanjutnya disebut sebagai elemen dari struct atau field. Dengan demikian dimungkinkan suatu struct dapat berisi elemen-elemen data berbeda tipe seperti char,int,float,double, dan lain-lain. Dalam program yang sederhana, jika kita menggunakan sedikit variable tentu tidak jadi masalah. Akan tetapi jika kita akan membuat sebuah program yang lebih kompleks, dengan berbagai macam nama dan tipe variabel dalam pendeklarasianya. Dengan struct, kita bisa mengelompokkan berbagai nama dan tipe variabel tersebut sesuai dengan kelompoknya. Hal ini tentunya bisa berguna untuk memudahkan dalam mengelompokkan sebuah variabel. Sebagai contoh umum, ada terdapat berbagai nama variabel: nama, nim, alamat, dll. Variabel – variabel tersebut dapat kita kelompokkan menjadi satu dengan nama data_mahasiswa.

1. Deklarasi Struct

Suatu struct didefinisikan dengan menggunakan kata kunci struct. Ada 2 cara pendefinisian dan deklarasi struct:

Cara pertama :

```
struct nama_struct {
    tipe_data1 elemen_struct;
};
```

Contoh :

```
struct Mahasiswa {
    int nim;
    string nama;
    float nilai;
};
```

Cara Kedua :

```
typedef struct {
   tipe_data1 elemen_struct;
}nama_struct;
```

Contoh :

```
typedef struct {
    int nim;
    string nama;
    float nilai;
} mahasiswa;
```

Mendefinisikan sebuah tipe data struct bernama mahasiswa yang memiliki tiga buah elemen (field) berupa :

- NIM
- Nama
- Nilai

Untuk mendeklarasikan sebuah variabel mhs yang bertipe struct mahasiswa pernyataan yang diperlukan adalah sebagai berikut :

```
Struct mahasiswa mhs;
```

atau

```
Mahasiswa mhs;
```

2. Mengakses Elemen Struct

Elemen dari suatu variabel struct dapat diakses dengan menyebutkan nama variabel struct diikuti dengan operator titik (“.”) dan nama dari elemen struct -nya. Cara penulisannya : **variable_struct.nama_field;**

Program berikut merupakan contoh yang melibatkan variabel struct. Mula-mula field dari struct diisi dengan suatu data, kemudian isinya ditampilkan. Contoh:

```
typedef struct{
    char nim[10];
    string nama;
    float nilai;
}mahasiswa;

mahasiswa mhs;
mhs.nim = ‘6054’ ;
mhs.nama = “Daffa”;
mhs.nilai = 60;

cout << "NIM : " << mhs.nim << endl;
cout << "Nama : " << mhs.nama << endl;
cout << "Nilai Akhir : " << mhs.nilai << endl;
```

3. Nested Struct

Nested struct merupakan suatu struct yang dapat digunakan didalam struct lainnya. Hal seperti ini dapat kita lihat pada contoh program dibawah ini :

```
typedef struct {
    char nim[10];
    string nama;
} datamhs;
struct datanilai {
    float nilai_uts;
    float nilai_uas;
};
typedef struct {
    datamhs mhs;
    struct datanilai nil;
} nilai;

nilai nilaimhs;
nilaimhs.mhs.nim = '6054';
nilaimhs.mhs.nama = "Daffa";
nilaimhs.nil.nilai_uts = 60;
nilaimhs.nil.nilai_uas = 60;

cout << "NIM : " << nilaimhs.mhs.nim << endl;
cout << "Nama : " << nilaimhs.mhs.nama << endl;
cout << "Nilai UTS : " << nilaimhs.nil.nilai_uts << endl;
cout << "Nilai UAS : " << nilaimhs.nil.nilai_uas << endl;
```

4. Array of Struct

Elemen-elemen dari suatu array juga dapat berbentuk sebuah struct. Misalnya array yang dipakai untuk menyimpan sejumlah data mahasiswa. Array struct berdimensi satu ini membentuk suatu tabel, dengan barisnya menunjukkan elemen dari struct. Dalam hal ini maka deklarasi yang dibutuhkan adalah sebagai berikut:

```
struct date {
    int month;
    int day;
    int year;
};
struct mahasiswa{
    string name;
    struct date birthday;
};

// deklarasi global dari variabel student
struct mahasiswa mhs[10];
```

Yang artinya mendeklarasikan array mhs yang memiliki elemen yang bertipe struct mahasiswa sebanyak 10. Contoh :

```
mhs[0].name = “Daffa”;
mhs[0].birthday.day = 30;
mhs[0].birthday.month = 1;
mhs[0].birthday.year = 2005;
```

6. Contoh CRUD Dengan Struct

Kita akan membuat program CRUD struct mahasiswa. Program ini akan meminta inputan dari user berupa biodata mahasiswa. Progam juga bisa nampilin, mengubah, dan menghapus data mahasiswa.

1. Membuat Menu Program

Kita akan buat menu program yang berisi pilihan-pilihan operasi yang bisa dilakukan pada data mahasiswa.

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

int main() {
  int pilihan, index;
  do {
    system("clear");
    cout << "Menu Program" << endl;
    cout << "1. Tampilkan Mahasiswa" << endl;
    cout << "2. Tambah Mahasiswa" << endl;
    cout << "3. Ubah Mahasiswa" << endl;
    cout << "4. Hapus Mahasiswa" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilihan: ";
    cin >> pilihan;

    switch (pilihan) {
    case 1:
      // Tampilkan Data Mahasiswa
      break;
    case 2:
      // Tambah Data Mahasiswa
      break;
    case 3:
      // Ubah Data Mahasiswa
      break;
    case 4:
      // Hapus Data Mahasiswa
      break;
```

```
    case 5:
      cout << "Program selesai" << endl;
      break;

    default:
      cout << "Pilihan tidak valid" << endl;
      break;
    }
  } while (pilihan != 5);

  return 0;
}
```

2. Membuat Menu Program

Kita akan membuat struct **Mahasiswa** yang berisi data mahasiswa.

```
struct Alamat {
  string jalan;
  int nomor;
  string kota;
};

struct Mahasiswa {
  string nama;
  int umur;
  string jurusan;
  Alamat alamat;
};
```

Setelah itu, kita akan deklarasikan array mahasiswa yang berisi data mahasiswa.

```
#define MAX_MAHASISWA 100
int panjang = 0;              // Jumlah elemen saat ini
mahasiswa mhs[MAX_MAHASISWA]; // Array dengan ukuran tetap
```

3. Menampilkan Data Mahasiswa

Kita lanjut ke menu pertama, yaitu nampilin data-data mahasiswa.

Taruh kode ini di case 1:

```
if (panjang == 0) {
    cout << "Belum ada mahasiswa" << endl;
} else {
    cout << "Daftar Mahasiswa" << endl;
    cout << "==================" << endl;
    for (int i = 0; i < panjang; i++) {
      cout << "Mahasiswa ke-" << i + 1 << endl;
      cout << "Nama: " << mhs[i].nama << endl;
      cout << "Umur: " << mhs[i].umur << endl;
      cout << "Jurusan: " << mhs[i].jurusan << endl;
      cout << "Alamat: " << mhs[i].alamat.jalan << " No. "
           << mhs[i].alamat.nomor << ", " << mhs[i].alamat.kota <<
endl;
      cout << endl;
    }
}
```

4. Menambah Data Mahasiswa

Kita akan lanjut ke menu kedua, yaitu menambah data mahasiswa.

Taruh kode ini di case 2:

```
if (panjang < MAX_MAHASISWA) {
    cout << "Masukkan nama mahasiswa: ";
    cin.ignore();
    getline(cin, mhs[panjang].nama);
    cout << "Masukkan umur mahasiswa: ";
    cin >> mhs[panjang].umur;
    cout << "Masukkan jurusan mahasiswa: ";
    cin.ignore();
    getline(cin, mhs[panjang].jurusan);
    cout << "Masukkan alamat mahasiswa: " << endl;
    cout << "Jalan: ";
    getline(cin, mhs[panjang].alamat.jalan);
    cout << "Nomor: ";
    cin >> mhs[panjang].alamat.nomor;
    cout << "Kota: ";
    cin.ignore();
    getline(cin, mhs[panjang].alamat.kota);
    panjang++;
```

```
    cout << "Mahasiswa berhasil ditambahkan" << endl;
} else {
    cout << "Kapasitas penuh! Tidak bisa menambah mahasiswa lagi." <<
endl;
  }
```

5. Mengubah Data Mahasiswa

Kita akan lanjut ke menu ketiga, yaitu mengubah data mahasiswa.

Taruh kode ini di case 3:

```
if (panjang == 0) {
    cout << "Belum ada mahasiswa untuk diubah." << endl;
} else {
    cout << "Daftar Mahasiswa" << endl;
    cout << "==================" << endl;
    for (int i = 0; i < panjang; i++) {
        cout << "Mahasiswa ke-" << i + 1 << endl;
        cout << "Nama: " << mhs[i].nama << endl;
        cout << endl;
    }
    cout << "Masukkan nomor mahasiswa yang akan diubah: ";
    cin >> index;

    if (index > 0 && index <= panjang) {
        cout << "Masukkan nama mahasiswa baru: ";
        cin.ignore();
        getline(cin, mhs[index - 1].nama);
        // tambahkan untuk field lainnya
        cout << "Mahasiswa berhasil diubah" << endl;
    } else {
        cout << "Nomor mahasiswa tidak valid" << endl;
    }
}
```

6. Menghapus Data Mahasiswa

Kita akan lanjut ke menu keempat, yaitu menghapus data mahasiswa.

Taruh kode ini di case 4:

```
if (panjang == 0) {
    cout << "Belum ada mahasiswa untuk dihapus." << endl;
```

```
} else {
    cout << "Daftar Mahasiswa" << endl;
    cout << "==================" << endl;
    for (int i = 0; i < panjang; i++) {
        cout << "Mahasiswa ke-" << i + 1 << endl;
        cout << "Nama: " << mhs[i].nama << endl;
        cout << endl;
    }
    cout << "Masukkan nomor mahasiswa yang akan dihapus: ";
    cin >> index;

    if (index > 0 && index <= panjang) {
        for (int i = index - 1; i < panjang - 1; i++) {
            mhs[i] = mhs[i + 1];
        }
        panjang--;
        cout << "Mahasiswa berhasil dihapus" << endl;
    } else {
        cout << "Nomor mahasiswa tidak valid" << endl;
    }
}
```

STUDI KASUS

1. Buatlah program dengan mengolala data mahasiswa. Setiap mahasiswa memiliki beberapa atribut, seperti NIM (Nomor Induk Mahasiswa), nama, dan nilai. Program akan menyimpan data beberapa mahasiswa dan menampilkan informasi tersebut.

2. Buatlah struct array untuk mengolala sebuah perpustakaan dengan menambahkan, menghapus, mengedit dan menampilkan data buku dengan atribut judul_buku, status_buku, tahun_terbit