Modul
Pemrograman Terstruktur

06.03.2023

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung
2023

Struct

Structure (struktur) adalah kumpulan elemen data yang digabungkan menjadi satu kesatuan. Dengan kata lain, structure merupakan bentuk struktur data yang dapat menyimpan variabel-variabel dalam satu nama.

Capaian Pembelajaran

1. Mahasiswa dapat memahami konsep Struct dan cara menggunakannya
2. Mahasiswa dapat mengimplementasikan Struct ke dalam program sesuai dengan kebutuhan Program

Materi

Struktur Struct

```
struct nama_struct
{
    <tipe_data> nama_field_1;
    <tipe_data> nama_field_2;
    ...
    <tipe_data> nama_field_n;
}
```

Dalam pembuatan Struct keyword yang dibutuhkan adalah **“struct”.**

Atribut dalam struct dapat berisi fungsi, void ataupun variabel.

Contoh

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

struct Mahasiswa{
    string nama, npm;
};

int main(){
    Mahasiswa Mhs;

    cin>> Mhs.nama;
    cin>> Mhs.npm;

    cout<< Mhs.nama << " " << Mhs.npm;
}
```

Pada contoh diatas, kita membuat sebuah Struct bernama Mahasiswa yang memiliki 2 buah Atribut didalamnya yaitu nama dan npm dengan tipe data string. Kemudian pada fungsi main kita membuat sebuah object dengan nama Mhs yang bertipe Struct(Mahasiswa).
Pemanggilan atribut dilakukan dengan menuliskan nama **"object.atribut".**

Note: Secara Default Access Specifiers pada Struct adalah Public. Kita akan membahas access specifiers lebih lanjut pada materi class.

Class

Class adalah salah satu dari konsep OOP yang digunakan untuk membungkus data abstraksi *prosedural* sebagai deskripsi tergeneralisir atau rancangan dari sebuah *object* untuk mendefinisikan atau menggambarkan isi dan tingkah laku sebagai entitas dari *object*.

Capaian Pembelajaran

1. Mahasiswa dapat memahami konsep Struct dan cara menggunakannya
2. Mahasiswa dapat mengimplementasikan Struct ke dalam program sesuai dengan kebutuhan Program

Materi

Struktur Class

```
class nama_class{
    akses_specifier :
        data_member;
        …
    akses_specifier2 :
        data_member;
        …
}nama_object;
```

Keterangan

1. **nama_class** : tempat dimana anda dapat memberikan nama pada *class* tersebut.
2. **akses_specifier** : tempat dimana anda dapat mendefinisikan hak akses kepada anggota yang ada dibawahnya. Ada 3 macam access specifier yaitu publik, private dan juga protected. Berbeda dengan struct, pada class default access specifier nya bersifat private
3. **Data_member / Atribut** : tempat dimana anda dapat mendirikan sebuah variabel atau function sebagai anggota dari class.
4. **nama_object** : tempat dimana anda dapat mendirikan object-object dengan menggunakan class tersebut, hal ini merupakan opsional tapi jika class tidak diberi nama class maka diwajibkan mendirikan object, karena kita tidak akan bisa membuat object dengan class di luar dari deklarasi class tersebut.

Contoh dengan Akses Publik

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

class Mahasiswa{
    public:
        string nama, npm;
};

int main(){
  Mahasiswa Mhs;

  cin>> Mhs.nama;
  cin>> Mhs.npm;

  cout<< Mhs.nama << " " << Mhs.npm;
}

```

Pada contoh diatas, kita membuat sebuah class Mahasiswa yang memiliki Akses specifier publik, Sehingga dapat diakses dari luar kelas itu sendiri.

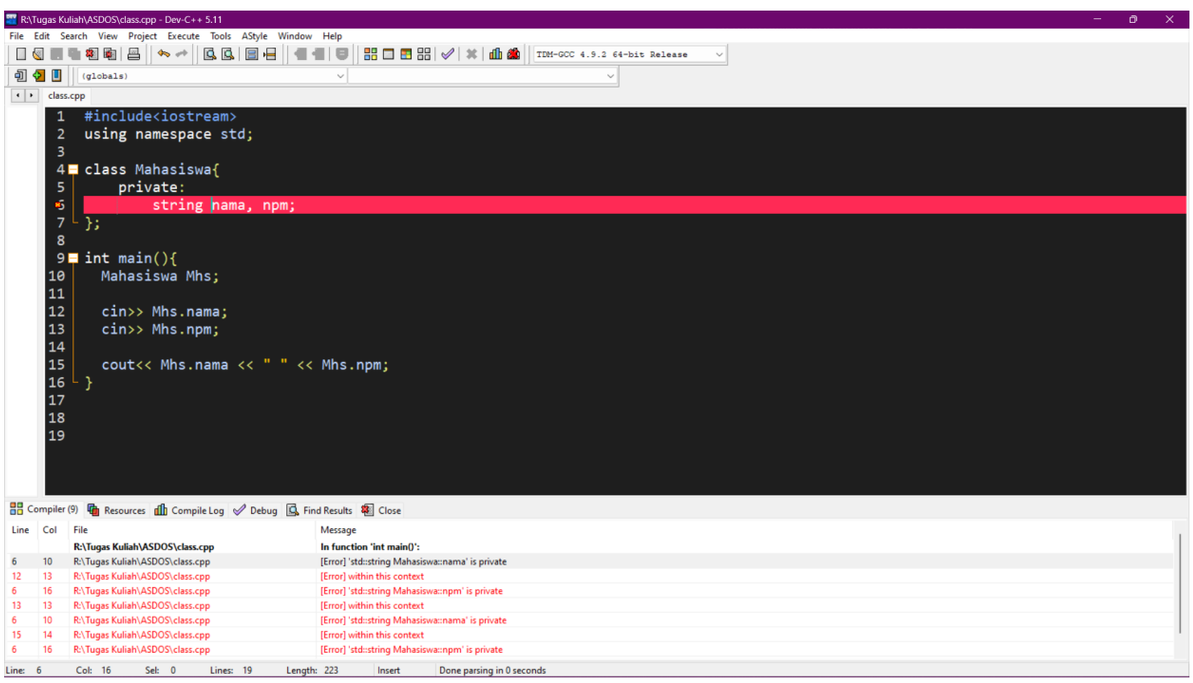
Sebagai contoh kita mengakses class mahasiswa pada fungsi main, dengan membuat object baru bernama Mhs.

Contoh Lain dengan menggunakan sebuah method:

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

class Mahasiswa{
    public:
        string nama, npm;

        void Perkenalan(){
            cout<< "Nama: " << nama << endl;
            cout<< "Npm: " << npm;
        }
};

int main(){
   Mahasiswa Mhs;

   cin>> Mhs.nama >> Mhs.npm;

   Mhs.Perkenalan();
}
```

Pada contoh diatas, class tidak hanya dapat menampung sebuah atribut variabel saja, namun sebuah class juga dapat menampung sebuah method didalamnya

Contoh dengan Akses Private



Pada contoh diatas, kita mengganti akses yang sebelumnya publik menjadi akses private. Kemudian kita membuat object baru bernama Mhs pada fungsi main dan memanggil atribut milik class yaitu nama dan npm, pada saat di jalankan compiler akan menunjukkan hasil error dikarenakan akses pada class adalah private.

Penjelasan Access Specifier

1. **Private:** data member atau atribut tidak dapat di akses atau dilihat dari luar kelas itu sendiri.
2. **Public:** data member atau atribut dapat diakses atau dilihat dari luar kelas itu sendiri
3. **Protected:** data member atau Atribut tidak dapat diakses dari luar class, namun mereka dapat diakses pada class turunannya atau inherited class nya.

Constructor

method spesial yang berfungsi untuk inisialisasi ketika pembuatan objek. Konstruktor dipanggil segera setelah objek baru dibuat. Ciri dari konstruktor pada C++ adalah nama method sama persis dengan nama kelasnya.

Constructor tanpa parameter

```
1  #include<iostream>
2  using namespace std;
3
4  class Mahasiswa{
5      public:
6          Mahasiswa(){ // CONSTRUCTOR
7              cout<<"Hello World";
8          }
9  };
10
11 int main(){
12    Mahasiswa Mhs;
13 }
14
```

Constructor dengan parameter

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

class Mahasiswa{
    public:
        string nama, npm;

        Mahasiswa(string nama, string b){
            this->nama = nama;
            npm = b;
        }
};

int main(){
   Mahasiswa Mhs("Reza","2117");

   cout<< "Nama: " << Mhs.nama << endl;
   cout<< "Npm: " << Mhs.npm;
}

```

Note: Secara Default sebuah class akan memiliki sebuah constructor tanpa parameter saat pertama kali class dibuat.

Setter dan Getter pada C++

Setter adalah member function atau method yang dipakai untuk memberikan nilai ke dalam sebuah data member. Sedangkan **Getter** adalah member function yang dipakai untuk menampilkan nilai data member.

Setter

```
1  #include<iostream>
2  using namespace std;
3
4  class Mahasiswa{
5      private:
6          string nama;
7
8      public:
9          void setNama(string nama){
10             this->nama = nama;
11         }
12
13 };
14
15 int main(){
16    Mahasiswa Mhs;
17
18    Mhs.setNama("Reza")
19 }
20
```

Pada contoh diatas kita memiliki sebuah atribut nama dengan akses private. Seperti yang kita ketahui jika sebuah atribut memiliki akses private, maka tidak dapat diakses diluar class itu sendiri. Bagaimana cara memberi nilai pada atribut tersebut?

Kita dapat memanfaatkan sebuah fungsi yang disebut **Setter**, dengan membuat sebuah fungsi setter dengan nama setNama dengan akses publik, setter biasanya menggunakan void atau prosedur dan memiliki satu parameter didalamnya. Lalu pada fungsi main kita akan memanggil fungsi setter tersebut untuk diberikan nilai.

Getter

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

class Mahasiswa{
    private:
        string nama;

    public:
        void setNama(string nama){
            this->nama = nama;
        }

        string getNama(){
            return nama;
        }
};

int main(){
    Mahasiswa Mhs;

    Mhs.setNama("Reza");
    cout<< Mhs.getNama();
}
```

Setelah kita berikan sebuah nilai pada atribut nama, bagaimana cara kita melihat hasil output nya?

Kita dapat memanfaatkan fungsi yang disebut **Getter,** dengan membuat sebuah fungsi getter dengan nama getNama. Tipe fungsi dari Getter selalu sama dengan tipe data atribut yang dituju. Seperti contoh diatas, kita memiliki atribut nama yang bertipe data string, maka tipe fungsi getter juga akan mengikuti yaitu string.

Langkah Praktikum

Membuat Sebuah Project Baru

1. Buka dev C++
2. Buat file baru dengan format **“nama_materiClass.cpp”**

Studi Kasus

Kack John adalah siswa kelas 1 SD, ia mendapatkan tugas dari guru nya untuk menghitung luas dari persegi panjang. Karena ia malas untuk menghitung, maka ia membuat program menggunakan c++

Membuat Program dengan Setter dan Getter

1. Buatlah sebuah class dengan nama **“class PersegiPanjang”**

```
1   #include<iostream>
2   using namespace std;
3
4   class PersegiPanjang{
5
6   };
```

2. Isi class tersebut dengan atribut **“int panjang dan int lebar”** yang diberi akses **Private**

```
1   #include<iostream>
2   using namespace std;
3
4   class PersegiPanjang{
5       private:
6           int panjang;
7           int lebar;
8   };
```

3. Lalu buat sebuah fungsi **Setter** dan **Getter** untuk masing-masing atribut, dan juga buat sebuah fungsi untuk menghitung luas persegi panjang.

```
    public:
        void setPanjang(int panjang){
            this->panjang = panjang;
        }

        void setLebar(int lebar){
            this->lebar = lebar;
        }

        int getPanjang(){
            return panjang;
        }

        int getLebar(){
            return lebar;
        }

        int Luas(){
            return panjang * lebar;
        }
```

4. Lalu buat sebuah objek baru pada fungsi main dengan nama "**PersegiPanjang psg",** panggil fungsi setter untuk mengisi nilai atribut, getter dan juga fungsi luas .

```
int main(){
    PersegiPanjang psg;

    psg.setLebar(10);
    psg.setPanjang(10);
    cout<< "Panjang: " << psg.getPanjang() << endl;
    cout<< "Lebar: " << psg.getLebar() << endl;
    cout<< "Luas: " << psg.Luas();

}
```

Keseluruhan Program

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

class PersegiPanjang{
    private:
        int panjang;
        int lebar;

    public:
        void setPanjang(int panjang){
            this->panjang = panjang;
        }

        void setLebar(int lebar){
            this->lebar = lebar;
        }

        int getPanjang(){
            return panjang;
        }

        int getLebar(){
            return lebar;
        }

        int Luas(){
            return panjang * lebar;
        }
};

int main(){
    PersegiPanjang psg;

    psg.setLebar(10);
    psg.setPanjang(10);
    cout<< "Panjang: " << psg.getPanjang() << endl;
    cout<< "Lebar: " << psg.getLebar() << endl;
    cout<< "Luas: " << psg.Luas();

}
```

Membuat Program dengan Constructor

1. Pada program sebelumnya, tambahkan sebuah constructor dengan parameter didalamnya

```
public:
    PersegiPanjang(int panjang, int lebar){
        this->panjang = panjang;
        this->lebar = lebar;
    }

    void setPanjang(int panjang){
        this->panjang = panjang;
    }

    void setLebar(int lebar){
        this->lebar = lebar;
    }

    int getPanjang(){
        return panjang;
    }

    int getLebar(){
        return lebar;
    }

    int Luas(){
        return panjang * lebar;
    }
```

2. Jalankan program dengan mengisi constructor pada fungsi main

```
36 int main(){
37     PersegiPanjang psg(10,10);
38     cout<< "Panjang: " << psg.getPanjang() << endl;
39     cout<< "Lebar: " << psg.getLebar() << endl;
40     cout<< "Luas: " << psg.Luas();
41
42 }
```

Keseluruhan Program

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

class PersegiPanjang{
    private:
        int panjang;
        int lebar;

    public:
        PersegiPanjang(int panjang, int lebar){
            this->panjang = panjang;
            this->lebar = lebar;
        }

        void setPanjang(int panjang){
            this->panjang = panjang;
        }

        void setLebar(int lebar){
            this->lebar = lebar;
        }

        int getPanjang(){
            return panjang;
        }

        int getLebar(){
            return lebar;
        }

        int Luas(){
            return panjang * lebar;
        }
};

int main(){
    PersegiPanjang psg(10,10);
    cout<< "Panjang: " << psg.getPanjang() << endl;
    cout<< "Lebar: " << psg.getLebar() << endl;
    cout<< "Luas: " << psg.Luas();

}
```