

Class

Oleh : Fiftin Noviyanto

Class Overview

- Saat membuat kelas dengan C++ kita akan selalu menemukan kata kunci yang umum digunakan seperti public, private dan protected.
- Kata kunci ini merupakan tingkatan akses yang digunakan dalam mengakses data-data yang ada dalam program yang kita buat dengan C++.
- Dengan menggunakan kata kunci ini kita dapat memberikan pembatasan akses terhadap data-data yang ada dalam kelas hingga kita dapat menentukan data mana saja yang dapat diakses oleh kelas lain dan data mana saja tidak dapat (tidak boleh) diakses.

Tingkatan Akses Class

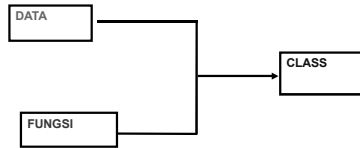
- **Public**
Tingkatan akses ini memberikan hak akses secara umum kepada kelas-kelas turunannya maupun kepada lingkungan diluar program.
- **Private**
Tingkatan akses ini hanya akan memberikan hak akses data kepada kelas yang bersangkutan saja. Data yang menggunakan tingkat akses ini tidak dapat diakses oleh kelas turunannya apalagi lingkungan diluar program. Setiap variabel atau fungsi yang tidak didefinisikan tingkat aksesnya secara otomatis akan ditetapkan sebagai data dengan tingkat akses private.
- **Protected**
Tingkatan akses ini memberikan hak akses kepada kelas turunannya tetapi tetap tidak mengizinkan lingkungan luar program untuk mengakses data ini.

Keuntungan menggunakan class

- **Ketika program-program menjadi lebih besar, satu daftar instruksi menjadi berat.**
 - Sedikit pemrogram dapat memahami program lebih dari beberapa ratus pernyataan kecuali unit bawah rusak dan menjadi lebih kecil. Untuk alasan ini *fungsi* diadopsi sebagai cara untuk membuat program lebih dapat di pahami oleh programmer. (Istilah *fungsi* yang digunakan dalam C + + dan C. Bahasa lain konsep yang sama mungkin akan sebagai subroutine, sebuah subprogram, atau prosedural program dibagi ke dalam fungsi, dan (idealnya, setidaknya) fungsi masing-masing memiliki tujuan yang jelas dan jelas ke berbagai fungsi lain dalam program ini.
 - Ide yang melanggar fungsi ke dalam sebuah program dapat lebih diperpanjang oleh pengelompokan beberapa fungsi kedalam entitas yang lebih besar yang dapat memanggil sebuah modul (yang sering disebut file), namun prinsipnya lebih mirip: sebuah kelompok dari komponen yang menjalankan daftar instruksi.
 - Dividing program dalam fungsi dan modul merupakan salah satu dari cornerstones struktur programming, yang kurang lebih didefinisikan disiplin yang dipengaruhi program organisasi selama beberapa dekade sebelum kedatangan dari pemrograman berorientasi objek.

Sintak dasar class

Dalam C++ class merupakan organisasi data dan fungsi :



Encapsulation : Data & Fungsi dalam organisasi yang sama, yakni class.

Member class : data & fungsi.

Sintak dasar Class (2)

Syntaxnya sbb:

```
class nama_class {
    permission_label_1:
        member1;
    permission_label_2:
        member2;
    ...
}
```

```
} nama_object;
```

atau

```
class nama_class {
    permission_label_1:
        member1;
    permission_label_2:
        member2;
    ...
};
```

Contoh program sederhana (1)

```
// classes example
#include <iostream.h>
class CRectangle {
    int x, y;
public:
    void set_values (int,int);
    int area (void) {return (x*y);}
};
void CRectangle::set_values (int a, int b) {
    x = a;
    y = b;
}
int main () {
    CRectangle rect;
    rect.set_values (3,4);
    cout << "Area: " << rect.area();
}
```

Penjelasan contoh(1)

Ada 2 permission label dalam class

CRectangle yakni:

- private membeinya berupa data x dan y
- public membeinya adalah fungsi set_values() dan fungsi area()
 - Fungsi set_value() didefinisikan diluar class sehingga harus menggunakan tanda scope ::
 - Fungsi area() dideklarasikan di dalam class

Contoh Program sederhana (2)

```
// class example
#include <iostream.h>
class CRectangle {
    int x, y;
public:
    void set_values (int,int);
    int area (void) {return (x*y);}
};
void CRectangle::set_values (int a, int b) {
    x = a;
    y = b;
}
int main () {
    CRectangle rect, rectb;
    rect.set_values (3,4);
    rectb.set_values (5,6);
    cout << "Rect area: " << rect.area() << endl;
    cout << "Rectb area: " << rectb.area() << endl;
}
```

Penjelasan contoh (2)

- Sama dengan contoh 1. hanya ditambahkan pembuatan beberapa objek sekaligus

Contoh program sederhana(3)

```
#include <iostream.h>
#include <conio.h>
class p_panjang {
public:
    int p,l;
    int luas()
    {return (p*l);}
};
main() {
    p_panjang a;
    a.p=5;
    a.l=9;
    cout<<"Luas = "<<a.luas();
    getch();
}
```

Modifikasi program sederhana(3)

```
1.Agar anda dapat menginputkan p dan l dari keyboard ganti instruksi :
a.p=5;
a.l=9;
dengan instruksi :
cout<<"Panjang = ";cin>>a.p;
cout<<"Luas = ";cin>>a.l;
sehingga program menjadi :
#include <iostream.h>
#include <conio.h>
class p_panjang {
public:
    int p,l;
    int luas()
    {return (p*l);}
};
main() {
    p_panjang a;
    cout<<"Panjang = ";cin>>a.p;
    cout<<"Lebar = ";cin>>a.l;
    cout<<"Luas = "<<a.luas();
    getch();
}

Jalankan program dan amati hasilnya.
```

Modifikasi program sederhana(3)

2. Ganti permission label dari **public** menjadi **private** sehingga program menjadi:

```
#include <iostream.h>
#include <conio.h>
class p_panjang {
private:
    int p,l;
    int luas()
    {return (p*l);}
};
main() {
    p_panjang a;
    cout<<"Panjang = ";cin>>a.p;
    cout<<"Luas = ";cin>>a.l;
    cout<<"Luas = "<<a.luas();
    getch();
}
```

Jalankan program amati pesan kesalahan yang muncul. Mengapa kesalahan tersebut muncul?

Note: Member private tidak boleh diakses dari luar class

Modifikasi program sederhana (3)

3. Ganti permission label untuk fungsi **luas()** dari **public** menjadi **private** sehingga program menjadi:

```
#include <iostream.h>
#include <conio.h>
class p_panjang {
private:
    int p,l;
public:
    int luas()
    {return (p*l);}
};
main() {
    p_panjang a;
    cout<<"Panjang = ";cin>>a.p;
    cout<<"Luas = ";cin>>a.l;
    cout<<"Luas = "<<a.luas();
    getch();
}
```

Jalankan program amati pesan kesalahan yang muncul.

Note: Pesan kesalahan muncul karena masih terdapat pengaksesan member private diluar class.

Modifikasi program sederhana(3)

4. Tambahkan fungsi **beri_nilai(int, int)** pada **public** member dan ubah instruksi pada fungsi **main()** untuk input data sehingga program menjadi:

```
#include <iostream.h>
#include <conio.h>
class p_panjang {
private:
    int p,l;
public:
    int luas()
    {return (p*l);}
    void beri_nilai (int p1, int l1)
    {p=p1; l=l1;}
};
main() {
    p_panjang a,b;
    int x,y;
    cout<<"Panjang = ";cin>>x;
    cout<<"Luas = ";cin>>y;
    a.beri_nilai(x,y);
    b.beri_nilai(10,15);
    cout<<"\nLuas a = "<<a.luas();
    cout<<"\nLuas b = "<<b.luas();
    getch();
}
```

Jalankan program dan amati outputnya.

Note: Penambahan fungsi **beri_nilai(int,int)** bertujuan agar member p dan l tetap bersifat private, tetapi nilainya bisa didapatkan dari luar class.