

Bahasa Pemrograman

Kuliah 9 :: Implementasi Class
Julio Adisantoso

Review OOP :: Prinsip Dasar

- Membungkus prosedur dan data dalam satu obyek → **Encapsulation**.
- OOP memodelkan obyek yang ada di dunia nyata ke dalam software obyek dalam pemrograman.
- Implementasi dalam bentuk **Class**.
- Berfungsi sebagai ADT (**Abstract Data Type**).

Review OOP :: Implementasi Class

- Struktur Class dalam C++:

```
class <class-name> {
    // data members
    ...
    // member functions
    ...
}
```

- Akses terhadap komponen class : private, protected, public. By default, akses komponen adalah private.

Akses terhadap anggota class

- **private**
hanya dapat diakses di dalam class itu sendiri
- **protected**
dapat diakses oleh class itu sendiri beserta turunannya
- **public**
dapat diakses di luar class

Contoh Kasus :: Class Person

- Sebagai implementasi software dari obyek "orang" yang memiliki atribut nama, usia, tinggi, berat.
- Obyek "orang" memiliki *member functions*:
 - Dapat ditentukan nilainya dengan fungsi setPerson
 - Dapat dicetak fungsi print
 - Dapat dihitung nilai indeks massa tubuh (imt)=berat/tinggi²

Class Person

```
class Person {
    string nama;
    int usia, tinggi;
    double berat;

public:
    Person() {nama=""; usia=tinggi=0; berat=0.0;}
    void setPerson(string n, int u, int b, double b) {
        nama=n; usia=u; tinggi=t; berat=b; }
    void print() {
        cout << nama << " " << usia << " " << tinggi
            << " " << berat << endl; }
    double getIMT() {
        double t=(double)tinggi/100.0;
        return berat/(t*t); }
}
```

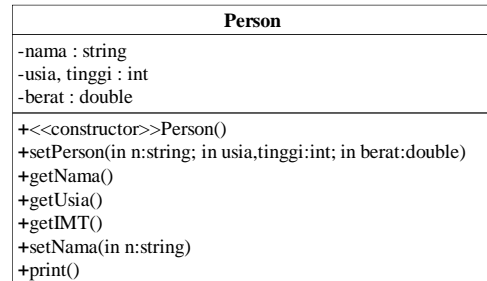
Mutator dan Accessor

- **Mutator** merupakan fungsi yang dapat memberi atau mengubah nilai atribut/data dalam class. Fungsi ini diperlukan terutama bagi data yang memiliki akses private.
- **Accessor** merupakan fungsi yang digunakan untuk membaca nilai atribut/data pada class. Fungsi ini diperlukan karena data yang akan diakses merupakan private atau protected.
- Pada contoh class Person, mana yang mutator dan mana yang accessor?

Slide 6

UML Presentation -- Person

- Unified Modeling Language
- Menyajikan class diagram



Slide 7

Standard Template Library STL

Slide 8

STL

- Pustaka C++ yang menyediakan struktur dan fungsi yang sering digunakan di dalam pemrograman.
- Terdiri atas empat komponen:
 - Algorithms : prosedur-prosedur
 - Containers : struktur data
 - Functors : function operator
 - Iterators : obyek untuk iterasi
- STL Containers: vector, deque, list, stack, queue, priority_queue, set, multiset, map, multimap, bitset

Slide 9

STL : Vector

Template:

```
template < class T, class Allocator = allocator<T> >  
class vector;
```

Member functions

(constructor), (destructor), operator=

Iterator:

begin, end, rbegin, rend

Capacity:

size, max_size, resize, capacity, empty, reserve

Element access:

operator[], at, front, back

Modifiers:

assign, push_back, pop_back, insert, swap, erase, clear

Slide 10

Contoh

```
#include <iostream>  
#include <vector>  
using namespace std;  
  
int main () {  
    unsigned int i;  
  
    vector<int> v1;           // empty vector  
    vector<int> v2 (4,100);  // 4 nilai 100  
    vector<int> v4 (v2);     // copy dari v2  
  
    // mengisi vector dari array tertentu  
    int myints[] = {16,2,77,29};  
    vector<int> v5  
        (myints, myints + sizeof(myints) / sizeof(int) );
```

Slide 11

Contoh

```
// mengisi 5 nilai vector dari standard input
vector<int> v6;           // empty vector
int t;
for (i=0; i<5; i++) {
    cin >> t;
    v6.push_back(t); }

cout << "Isi vector v5\n";
for (i=0; i<v5.size(); i++)
    cout << " " << v5[i];

cout << "\n\nIsi vector v6\n";
for (i=0; i < v6.size(); i++) cout << " " << v6[i];
cout << endl;

return 0;
}
```

5/10