



IPB University
— Bogor Indonesia —

KOM20C -- PEMROGRAMAN

Object Oriented Programming

Object and Class

Tim Pengajar Pemrograman IPB University

Review: Prinsip Dasar OOP

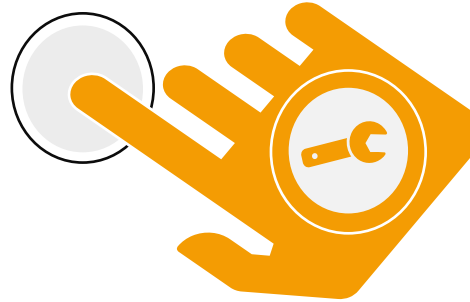
POLYMORPHISM

Tiap objek tahu siapa dirinya



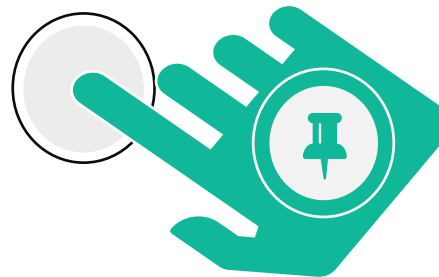
ENCAPSULATION

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



INHERITANCE

- Pewarisan sifat objek
- Mengembangkan class baru dari class yang sudah ada.



Review: Implementasi Class

Struktur class dalam C++:

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

Akses terhadap elemen class (*access modifier*) : **private**, **protected**, **public**. By default, akses komponen adalah private.

Access Modifier



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: objek "person"



Sebagai implementasi software dari objek "person" atau "orang".

Memiliki atribut (hanya contoh, sebagian dari banyak atribut): **nama**, **usia**, **tinggi**, **berat**.

Memiliki behaviour (perilaku, methods, etc) antara lain (hanya contoh):

- Dapat diberi nilai setiap atribut (mutator)
- Dapat diakses nilai setiap atribut (accessor)
- Dapat dihitung nilai indeks massa tubuh (IMT) = $\text{berat} / \text{tinggi}^2$.

Desain Class Person

Sebagai implementasi software dari objek "person" atau "orang".

Memiliki atribut (hanya contoh, sebagian dari banyak atribut): **nama**, **usia**, **tinggi**, **berat**.

Memiliki behaviour (perilaku, methods, etc) antara lain (hanya contoh):

- Dapat diberi nilai atribut (mutator)
- Dapat diakses nilai atribut (accessor)
- Dapat dihitung nilai indeks massa tubuh (IMT) = $\text{berat} / \text{tinggi}^2$.

Access modifier

+ : public

- : private

: protected

Person

```
-nama : string  
-usia, tinggi : int  
-berat : double
```

```
+<<constructor>>Person()  
+setPerson(in n:string;  
           in usia,tinggi:int;  
           in berat:double)  
+getNama()  
+getUsia()  
+getIMT()  
+setNama(in n:string)  
+print()
```

Presentasi berupa UML
(Unified Modeling Language)

Constructor

Constructor

adalah fungsi khusus dalam class yang akan dieksekusi terlebih dahulu saat pembuatan *instance (object)*.

Constructor biasanya digunakan untuk insialisasi atribut. Misalnya memberi nilai 0 untuk usia, tinggi, dan berat.

Karakteristik Constructor

- Memiliki identitas (nama fungsi) yang sama dengan nama class.
- Tidak memiliki return type.
- Tidak dapat dipanggil dari luar class.

Implementasi dalam C++

```
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); }  
}
```

Deklarasi kelas

Deklarasi atribut dari kelas

Constructor

Mutator

Tahapan OOP



Object Design

Merancang objek (*attribute and behaviour*)

Menyusun hubungan antar objek (inheritance, polymorphism)



Create Class

Implementasi objek dalam pemrograman



Driver

Menyusun instruksi untuk memecahkan persoalan dengan menggunakan class yang sudah dibuat. Disini tempat untuk membuat instance (object).



Testing

Menguji program dengan menggunakan kasus data.

Contoh Driver

Contoh kasus:

Class Person digunakan untuk membaca N data orang sesuai atribut yang ada, dan menyimpan ke dalam array.

Contoh masukan:

5

Noh Jong Hyun,17,165,60.5

Yoon Ji Ho,25,170,48.7

Nam Se Hee,30,180,60.2

Ri Jung Hyuk,35,182,62.8

Yoon Se Ri,30,170,58.9

Kerangka Program C++

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

// definisi class
class Person
{
    // elemen class
};

// driver
int main()
{
    // elemen program sesuai problem yang diberikan

    return 0;
}
```

Latihan

Revisi program yang ada sesuai kebutuhan (tanpa mengubah atribut class), agar dapat menampilkan banyaknya orang yang memiliki kategori obesitas ($IMT \geq 30$), berat badan berlebih ($25 \leq IMT < 30$), berat badan normal ($18.5 \leq IMT < 25$), dan berat badan kurang ($IMT < 18.5$)

Contoh output:

```
Obesitas: ..  
Berat badan berlebih: ..  
Berat badan normal: ..  
Berat badan kurang: ..
```

Fungsi getline()

Sintaks:

```
istream& getline (istream& is, string& str, char delim);  
istream& getline (istream& is, string& str);
```

Contoh:

```
string line;  
getline(cin, line);
```

Split and Convert String

Gunakan stringstream

```
istringstream ss(line);  
string token;  
getline(ss, token, ',');
```

Convert string to number:

```
int a=0;  
stringstream degree(token);  
degree>>a;
```