



PEMROGRAMAN LANJUT

01-b | Dasar OOP

Procedural vs OOP

No	Pemrograman Terstruktur	Pemrograman Berorientasi Objek
1	Memecah program dalam fungsi dan data	Menggabungkan fungsi dan data dalam kelas-kelas atau objek-objek
2	Memiliki ciri Sequence (berurutan), Selection (pemilihan) dan Repetition (perulangan)	Memiliki ciri Encapsulation (pengemasan), Inheritance (penurunan sifat), dan Polymorphism (perbedaan bentuk dan perilaku)
3	Struktur program rumit karena berupa urutan proses dan fungsi-fungsi	Struktur program ringkas, cukup dengan membuat Objek dan class lalu bekerja berdasarkan object dan class tersebut.
4	Re-use kode program kurang	Kode program sangat re-usable. object dan class dapat digunakan berkali-kali, sehingga dapat menghemat space memori.
5	Efektif digunakan untuk menyelesaikan masalah kecil dan tidak cocok untuk menyelesaikan masalah yang rumit, karena nantinya akan kesulitan menemukan solusi permasalahan ketika terjadi error	Efektif digunakan untuk menyelesaikan masalah besar, karena OOP terdiri dari class-class yang memisahkan setiap kode program menjadi kelompok-kelompok kecil, sesuai dengan fungsinya
6	Mudah diawal, namun kompleks diproses selanjutnya	Sulit diawal (karena harus membuat class) namun selanjutnya akan terasa mudah dan cepat
7	Eksekusi lebih lambat karena setiap perintah dikerjakan berurutan	Eksekusi lebih cepat karena dieksekusi bersamaan, program hanya mengatur Objek, properties dan method-nya saja

Apakah OBJEK ?

- Semua benda yang ada di dunia nyata dapat dianggap sebagai objek
- Contoh : rumah, mobil, sepeda motor, gelas, komputer, meja, dll

Karakteristik Objek

- Setiap objek memiliki atribut sebagai suatu status (state)
- Setiap objek memiliki tingkah laku (behavior)
- Penggambaran pemrograman berorientasi objek = penggambaran pada dunia nyata
- Pada pemrograman berorientasi objek :
 - State disimpan pada -> variable
 - Tingkah laku disimpan pada -> method

Apakah CLASS ?

- Class merupakan template yang digunakan untuk membuat objek
- Class merupakan prototipe /blue-prints yang mendefinisikan variabel-variable (atribut) dan method-method (behavior) secara umum
- Objek merupakan hasil instansiasi dari suatu kelas
- Proses pembentukan objek dari suatu class disebut *instantiation*
- Objek disebut juga *instances*

Apakah Attribute ?

- Atribut adalah data yang membedakan antara objek satu dengan yang lain
- Contoh : Mobil
mempunyai atribut sebagai berikut :
 - pabrikan
 - type
 - warna
- Dalam class atribut disebut sebagai **VARIABLE**

Attribute

- **Instance variable** : atribut tiap objek yang memiliki class yang sama
- Tiap objek mempunyai dan menyimpan nilai atributnya sendiri
- Tiap objek dari class yang sama boleh mempunyai nilai yang sama atau berbeda
- **Class variable** : atribut semua objek yang dibuat dari class yang sama
- Semua objek mempunyai nilai atribut yang sama
- Semua objek dari class yang sama mempunyai hanya satu nilai yang nilainya sama

Behavior (Tingka h Laku)

- Behavior/ tingkah laku adalah hal-hal yang bisa dilakukan oleh objek dari suatu class
- Behavior dapat digunakan untuk mengubah nilai atribut suatu objek,
- menerima informasi dari objek lain dan mengirim informasi ke objek lain untuk melakukan suatu pekerjaan
- Contoh :Mobil
- Bergerak maju
- Bergerak mundur
- Parkir
- Dalam class, behavior disebut juga sebagai **METHOD**
- Method : serangkaian statement dalam suatu class yang menangani suatu pekerjaan tertentu

Ilustrasi Instansiasi dan Abstraksi Ketika Membuat Class

Object nyata



- Mobil
 - State:
 - Pabrikan: Toyota
 - Type: avanza
 - Warna: perak
 - Behavior:
 - Bergerak maju
 - Bergerak mundur
 - Parkir

Abstraksi



Class

Mobil
Pabrikan: _____
Type: _____
Warna: _____
Bergerak maju
Bergerak mundur
Parkir

Object instance dari class Mobil

- Mobil I
 - State:
 - Pabrikan: Honda
 - Type: CR-Z
 - Warna: perak metalik
 - Behavior:
 - Bergerak maju
 - Bergerak mundur
 - Parkir

- Mobil II
 - State:
 - Pabrikan: Toyota
 - Type: Rush
 - Warna: Hijau
 - Behavior:
 - Bergerak maju
 - Bergerak mundur
 - Parkir

Instansiasi

Instansiasi

Class Diagram



<ul style="list-style-type: none">- artinya private+ artinya public# artinya protected(kosong) artinya default

Aturan Penulisan Identifier

- Variabel dan method:
 1. diawali dengan **huruf** atau **\$** atau **_**, tapi diusahakan untuk tidak menggunakan **\$** atau **_**
 2. setelah karakter pertama, boleh diikuti dengan **angka**
 3. **tidak** boleh ada **spasi**
 4. jika terdiri dari 1 kata, maka semuanya ditulis dengan **huruf kecil**, namun jika ada lebih dari 1 kata, maka **kata ke dua** dan seterusnya diawali dengan **huruf capital**
 5. Contoh: nama, namaAwal, tanggalLahir

- Nama class:
 1. **sama** dengan nomor **1** sampai **3** dari aturan variabel
 2. setiap kata diawali dengan **huruf kapital**
 3. Contoh : Mobil, MobilBalap, MobilBalapKarung

Konsep OOP

- ☐ Enkapsulasi (Encapsulation)
- ☐ Penurunan sifat (Inheritance)
- ☐ Polimorfisme (Polymorphism)

Enkapsulasi

- Enkapsulasi : Pembungkusan variabel dan method dalam sebuah objek yang terlindungi serta menyediakan interface untuk mengakses variabel tersebut
- Variabel dan method yang dipunyai suatu objek, bisa ditentukan hak aksesnya
- Contoh : Jam Tangan
- Digunakan untuk mengetahui waktu
- Cara jam mencatat waktu baik yang analog maupun digital tidak penting kita ketahui
- Dalam OOP, konsep enkapsulasi sebenarnya perluasan dari struktur dalam bahasa C

Pewarisan Sifat

- Pewarisan sifat merupakan pewarisan atribut dan method dari sebuah class ke class lainnya
 - Class yang mewarisi -> superclass
 - Class yang diwarisi -> sub class
 - Sub class bisa berlaku sebagai superclass bagi class lainnya -> multilevel inheritance
- Contoh : terdapat class sepeda dan sepeda gunung
- Sepeda -> superclass
- Sepeda gunung -> sub class
- Sepeda gunung memiliki variabel dan method yang dimiliki oleh sepeda

Polimorfisme

- Polimorfisme adalah kemampuan suatu objek untuk mempunyai lebih dari satu bentuk
- Dengan polimorfisme, suatu interface yang sama untuk memerintah objek agar melakukan aksi atau tindakan yang mungkin secara prinsip sama namun secara proses berbeda
- Contoh : Penyanyi melakukan aktivitasnya dengan bernyanyi. Kita baru mengetahui penyanyi tersebut adalah penyanyi jazz setelah mendengarkannya menyanyikan lagu jazz, demikian pula dengan penyanyi dangdut atau penyanyi pop. Namun demikian kesemuanya memiliki aktivitas yang sama, yaitu bernyanyi.

Bahasa Pemrograman Berorientasi Objek

- ☐ Visual Foxpro
- ☐ Java
- ☐ C++
- ☐ Visual Basic

Kelebihan OOP

- Kebebasan pengembangan
- Peningkatan kualitas
- Kemudahan pemeliharaan
- Kemampuan yang tinggi dalam modifikasi
- Peningkatan penggunaan kembali software

Tugas

- Buatlah sebuah program untuk menghitung luas lingkaran
- Buatlah sebuah program untuk menghitung keliling, luas, dan panjang diagonal dari sebuah segiempat