

PERANCANGAN BERORIENTASI OBJEK

1. PENDAHULUAN

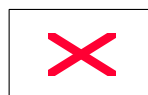
Analisis dan disain berorientasi objek adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep sekitar dunia nyata. Dasar pembuatan adalah objek, yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas.

Pengertian "**berorientasi objek**" berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya.

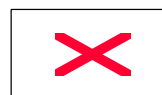
1.1. Karakteristik dari Objek

Objek

- Identitas berarti bahwa data diukur mempunyai nilai tertentu yang membedakan entitas disebut Objek.
- Objek dapat kongkrit, seperti halnya arsip dalam sistem, atau konseptual seperti kebijakan penjadwalan dalam multiprocessing pada sistem operasi.
- Setiap objek mempunyai sifat yang melekat pada identitasnya.
- Dua objek dapat berbeda walaupun bila semua nilai atributnya identik.



Mobil



Singa

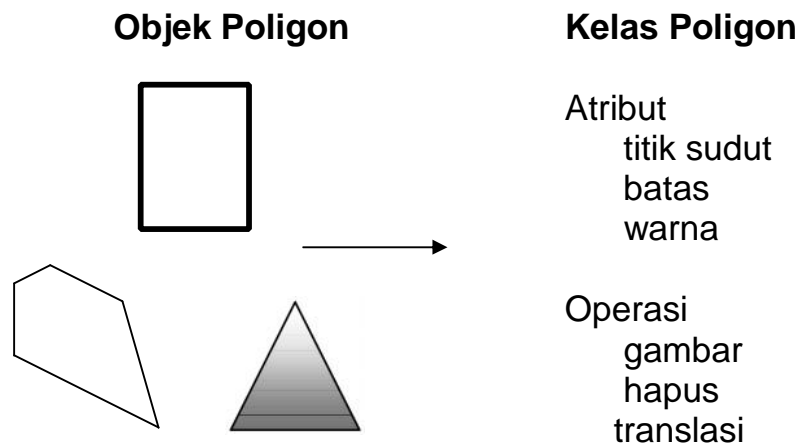
NoPeg	Nama
96001	Susan
96002	David
97001	Shila

Tabel

Gambar 1. Macam-macam Objek

Kelas Objek

- Kelas merupakan gambaran sekumpulan Objek yang terbagi dalam atribut, operasi, metode, hubungan, dan makna yang sama.
- Suatu kegiatan mengumpulkan data (atribut) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup.
- Kelas Objek merupakan wadah bagi Objek. Dapat digunakan untuk menciptakan Objek.
- Objek mewakili fakta/keterangan dari sebuah kelas.



Gambar 2. Kelas dan Objek

Istilah-istilah Objek

- Atribut : Data item yang menegaskan Objek
- Operasi : Fungsi di dalam kelas yang dikombinasikan ke bentuk tingkah laku kelas
- Metode : Pelaksanaan prosedur (badan dari kode yang mengeksekusi respon terhadap permintaan objek lain di dalam sistem).

1.2. Karakteritik Metodologi Berorientasi Objek

Metodologi pengembangan sistem berorientasi objek mempunyai tiga karakteristik utama :

- **Encapsulation**
- **Inheritance**
- **Polymorphism**

A. ENCAPSULATION (PENKAPSULAN)

- Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
- Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.
- Data terlindung dari prosedur atau objek lain, kecuali prosedur yang berada dalam objek itu sendiri.

B. INHERITANCE (PEWARISAN)

- Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya langsung. Atribut dan metode dari objek dari objek induk diturunkan kepada anak objek, demikian seterusnya.
- Inheritance mempunyai arti bahwa atribut dan operasi yang dimiliki bersama di antara kelas yang mempunyai hubungan secara hirarki.
- Suatu kelas dapat ditentukan secara umum, kemudian ditentukan spesifik menjadi subkelas. Setiap subkelas mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh kelas induknya, dan ditambah dengan sifat unik yang dimilikinya.
- Kelas Objek dapat didefinisikan atribut dan service dari kelas Objek lainnya.
- Inheritance menggambarkan generalisasi sebuah kelas

Contoh :

- *Sedan dan Sepeda Motor adalah subkelas dari Kendaraan Bermotor.*
- *Kedua subkelas mewarisi sifat yang dimiliki oleh Kendaraan Bermotor, yaitu mempunyai mesin dan dapat berjalan.*
- *Kedua subkelas mempunyai sifat masing-masing yang berbeda, misalnya jumlah roda, dan kemampuan untuk berjalan mundur yang tidak dimiliki oleh sepeda motor.*

C. POLYMORPHISM (POLIMORFISME)

- Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda.
- Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda.

- Kemampuan objek-objek yang berbeda untuk melakukan metode yang pantas dalam merespon message yang sama.
- Seleksi dari metode yang sesuai bergantung pada kelas yang seharusnya menciptakan Objek.

2. PEMODELAN BERORIENTASI OBJEK

2.1 . Pemodelan Sebagai Teknik Desain

Teknik pemodelan objek menggunakan tiga macam model untuk menggambarkan sistem :

- A. Model Objek
- B. Model Dinamik
- C. Model Fungsional

A. Model Objek :

- Model objek Menggambarkan struktur statis dari suatu objek dalam sistem dan relasinya
- Model objek berisi diagram objek. Diagram objek adalah *graph* dimana *nodenya* adalah kelas yang mempunyai relasi antar kelas.

B. Model Dinamik

- Model dinamik menggambarkan aspek dari sistem yang berubah setiap saat.
- Model dinamik dipergunakan untuk menyatakan aspek kontrol dari sistem.
- Model dinamik berisi *state diagram*. *State diagram* adalah *graph* dimana *nodenya* adalah *state* dan *arc* adalah tarnsisi antara *state* yang disebabkan oleh *event*.

C. Model Fungsional

- Model fungsional menggambarkan transformasi nilai data di dalam sistem.
- Model fungsional berisi data flow diagram. DFD adalah suatu *graph* dimana *nodenya* menyatakan proses dan *arcnya* adalah aliran data.

2.2. Model Berorientasi Objek

Sebuah model objek menangkap struktur statis dari sistem dengan menggambarkan objek dalam sistem, hubungan antara objek, serta

atribut dan operasi yang merupakan karakteristik setiap kelas dan objek.

Model berorientasi objek lebih mendekati keadaan nyata, dan dilengkapi dengan penyajian grafis dari sistem yang sangat bermanfaat untuk komunikasi dengan user dan pembuatan dokumentasi struktur dari sistem.

A. Objek dan Kelas

Objek

- Objek didefinisikan sebagai konsep, abstraksi atau benda dengan batasan dan arti untuk suatu masalah.
- Semua objek mempunyai identitas yang berbeda dengan lainnya. Istilah identitas berarti bahwa objek dibedakan oleh sifat yang melekat dan bukan dengan uraian sifat yang dimilikinya. Contohnya, *kembar identik*, walaupun mereka nampak seperti sama, tetapi merupakan dua orang yang berbeda.
- Kadang-kadang objek berarti suatu barang, maka digunakan istilah *object instance*, dan *object class* untuk menunjukkan satu grup dari barang yang sama.

Kelas

- Suatu *object class* menggambarkan kumpulan dari objek yang mempunyai sifat (atribut), perilaku umum (operasi), relasi umum dengan objek lain dan semantik umum.
Contoh : Orang, perusahaan, binatang, proses adalah *objek*. Setiap orang mempunyai umur, IQ, dan mungkin pekerjaan. Setiap proses mempunyai pemilik, prioritas, list dari sumber daya yang dibutuhkan.
- Objek dan object class sering sama sebagai benda dalam deskripsi masalah.

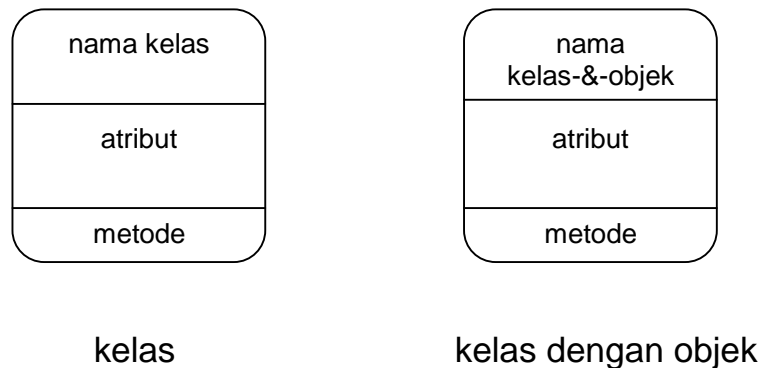
B. Diagram Objek

Diagram objek melengkapi notasi grafik untuk pemodelan objek, kelas dan relasinya dengan yang lain. Diagram objek bermanfaat untuk pemodelan abstrak dan membuat perancangan program.

Kelas dan Objek

Konsep fundamental dalam analisis berorientasi objek adalah objek itu sendiri. Sebuah objek adalah sebuah entitas yang mencakup data dan metode.

Kelas merupakan satu atau lebih objek dengan persamaan atribut dan metode, sedangkan kelas-&-objek adalah kelas dengan satu atau lebih objek di dalamnya. **Nama kelas** adalah **kata benda tunggal**, atau **kata sifat dan kata benda**. Nama dari kelas-&-objek harus dapat menjelaskan objek tunggal dari suatu kelas.



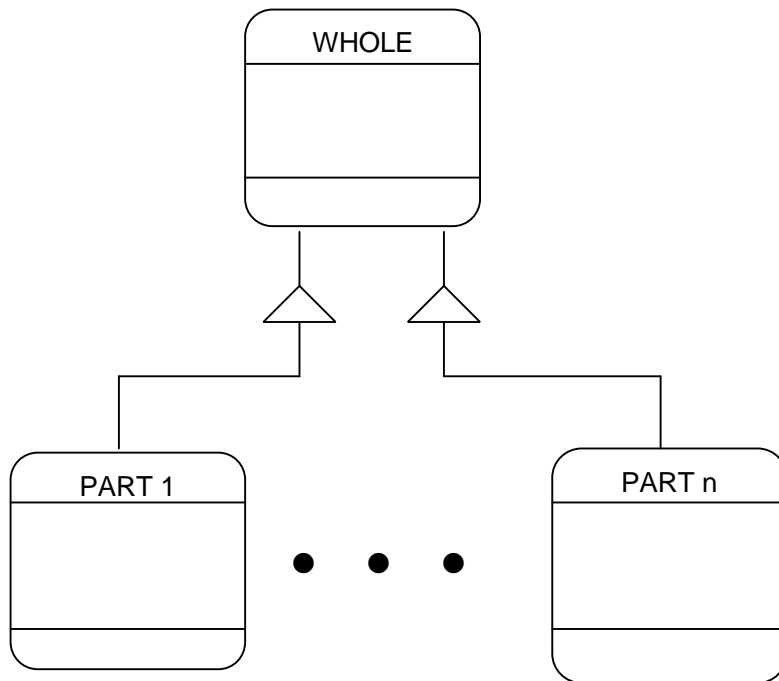
Gambar 3. Notasi untuk kelas dan kelas-&-objek

Struktur Objek dan Hirarki Kelas

Struktur kelas dibagi dua macam, yaitu **Whole-Part Structure** dan **Gen-Spec Structure**.

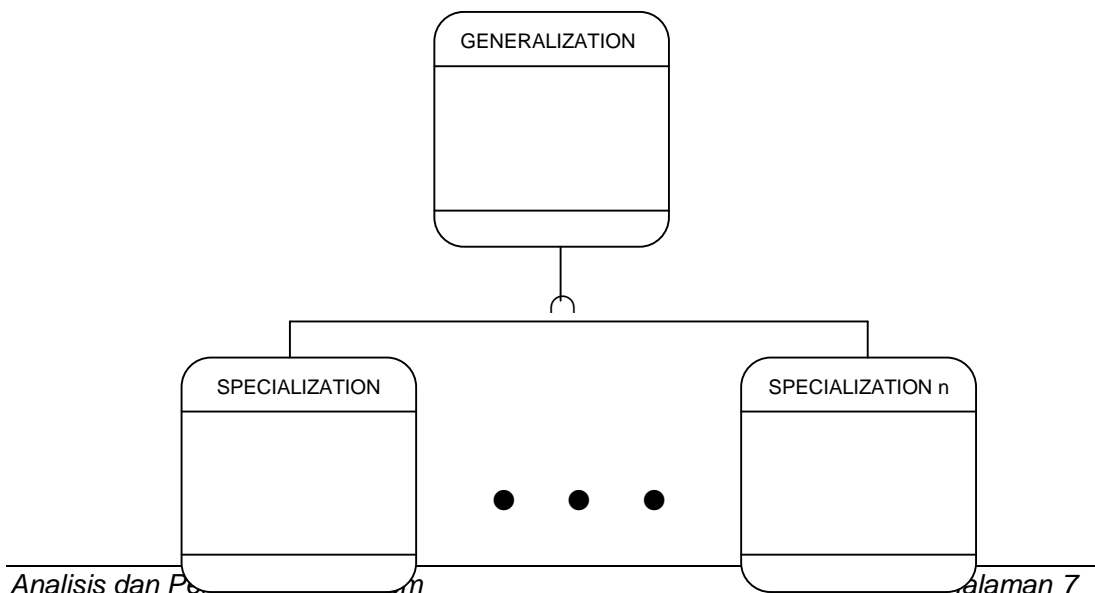
Whole-Part Structure memperlihatkan hirarki dari suatu kelas sebagai komponen dari kelas lain yang disebut juga sub objek.

Contohnya, kelas Mobil adalah *Whole* dan komponennya Mesin, Rangka, dll merupakan *Part1, Part 2, ..., Partn*.



Gambar 4. Notasi untuk whole-part structure

Gen-Spec Structure memperlihatkan kelas sebagai spesialisasi dari kelas di atasnya. Kelas yang mempunyai sifat umum disebut *Generalization*, *Superclass* atau *Topclass*, sedangkan kelas yang mempunyai sifat khusus disebut *Specialization*.

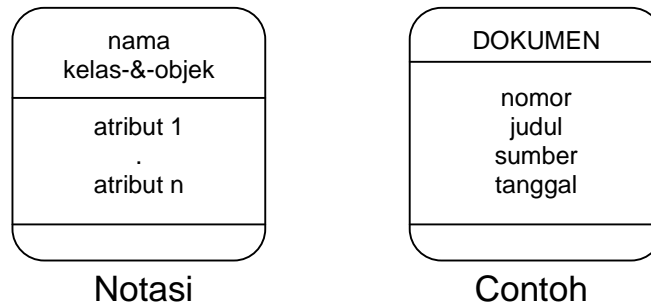


Gambar 5. Notasi untuk gen-spec structure

Contohnya, kelas Mobil adalah Generalization, sedangkan Sedan, Truk, Minibus, dll merupakan Specizlization1, Specialization2, ..., Specializationn, yaitu kelas yang mempunyai sifat khusus.

Atribut

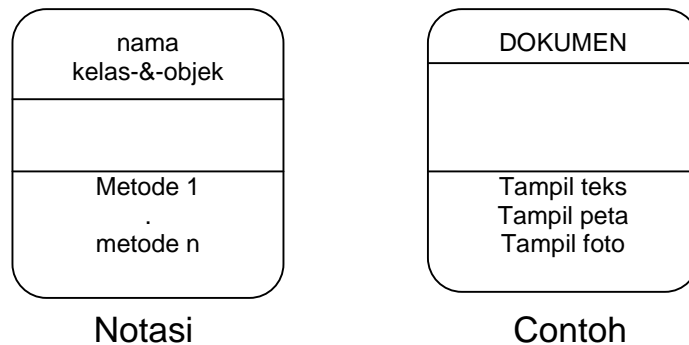
Atribut menggambarkan data yang dapat memberikan informasi mengenai kelas atau objek dimana atribut tersebut berada.



Gambar 6. Notasi untuk atribut

Metode

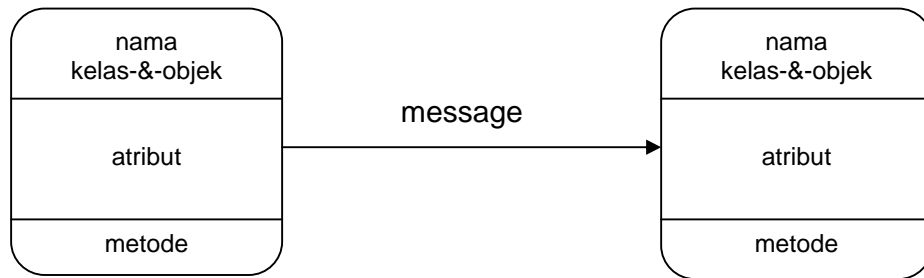
Metode (method) disebut juga *service* atau *operator* adalah prosedur atau fungsi seperti yang terdapat dalam bahasa Pascal pada umumnya, tetapi cara kerjanya agak berlainan. Metode adalah subprogram yang tergabung dalam objek bersama-sama dengan atribut. Metode dipergunakan untuk pengaksesan terhadap data yang terdapat dalam objek tersebut.



Gambar 7. Notasi untuk metode

Pesan (Message)

Message merupakan cara untuk berhubungan antara satu objek dengan objek lain. Suatu pesan dikirimkan oleh suatu objek kepada objek tertentu dapat digambarkan dengan anak panah.



Gambar 8. Notasi untuk message

3. ANALISIS BERORIENTASI OBJEK

3.1. Pendekatan Berorientasi Objek

A. Dasar

Metode berorientasi objek dapat digambarkan sebagai berikut :

Berorientasi objek = Kelas dan Objek
+ Inheritance
+ Komunikasi dengan message

B. Prinsip

Analisis Berorientasi Objek berdasarkan pada aplikasi yang mempunyai bentuk sama untuk mengatur kompleksitas :

1. Abstraksi
 - a. Prosedural
 - b. Data
2. Pengkapsulan
3. Pewarisan
4. Asosiasi
5. Hubungan dengan message
6. Meliputi metode dari organisasi
 - a. Objek
 - b. Whole – part
 - c. Kelas dan anggota, dan perbedaan diantaranya
7. Skala
8. Kategori dari perilaku

- a. Penyebab yang mendadak
- b. Perubahan dari waktu ke waktu
- c. Kesamaan fungsi

C. Pendekatan

Analisis Berorientasi Objek memiliki lima aktivitas utama dalam pendekatannya, yaitu :

1. Menentukan Kelas-&-Objek
2. Identifikasi Struktur
3. Identifikasi Subyek
4. Menentukan Atribut
5. Menentukan Metode

D. Tujuan dan Keuntungan

Terdapat tujuh macam tujuan dan keuntungan dari Analisis Berorientasi Objek, yaitu

3.2. Contoh Kasus

OBJECT ORIENTED ANALYSIS AND DESIGN (OOAD)

Object Oriented analysis adalah metode analisis yang memeriksa *requirements* (syarat/keperluan yang harus dipenuhi suatu sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup permasalahan.

Object Oriented Design adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek-objek sistem atau subsistem.

Beberapa Konsep Dasar dalam OOAD

Objek

Objek (*object*) adalah “benda”, secara fisik atau konseptual.

Contoh : hardware, software, dokumen, manusia, dan konsep.

Untuk kepentingan memodelkan perusahaannya, seorang kepala eksekutif akan melihat karyawan, gedung, divisi, dokumen, dan keuntungan sebagai objek.

Sebuah objek memiliki keadaan sesaat (*state*) dan perilaku (*behavior*).

State dari sebuah objek adalah kondisi objek tersebut atau himpunan dari keadaan yang menggambarkan objek tersebut. Sebagai contoh, bola lampu adalah objek, dan salah satu keadaan nyala atau tidak adalah *state* dari objek bola lampu tersebut.

State dinyatakan dengan nilai atribut objeknya. **Atribut** adalah nilai internal suatu objek yang mencerminkan antara lain karakteristik objek, kondisi sesaat, koneksi dengan objek lain, dan identitas. Perubahan state dicerminkan oleh perilaku (*behavior*) objek tersebut.

Behavior suatu objek mendefinisikan bagaimana sebuah objek bertindak (beraksi) dan memberi reaksi. *Behavior* ditentukan oleh himpunan semua atau beberapa operasi yang dapat dilakukan dalam objek itu sendiri. *Behavior* dari sebuah objek dicerminkan oleh *interface*, *service*, dan metode dari objek tersebut.

Interface adalah pintu untuk mengakses *service* objek. **Service** adalah fungsi yang bisa diemban objek. **Metode** adalah mekanisme internal objek yang mencerminkan perilaku (*behavior*) objek tersebut. Sebagai contoh, jika printer merupakan sebuah objek, maka perilaku (*behavior*) atau *service*-nya mencetak apapun yang dia terima.

Kelas

Kelas adalah definisi umum (pola, *template* atau cetak biru) untuk himpunan objek sejenis. Kelas menetapkan spesifikasi perilaku (*behavior*) dan atribut objek-objek tersebut. Kelas adalah keniscalaan (abstraksi) dari entitas dalam dunia nyata. Objek adalah “contoh” (*instance*) dari sebuah kelas. Sebagai contoh, atribut untuk kelas binatang adalah berkaki empat dan memiliki ekor. Perilakunya adalah tidur dan makan. “Contoh” (*instance*) yang mungkin dari kelas binatang ini adalah kucing, gajah, dan kuda.

Keunggulan Object Oriented

- Programmer dapat mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata.
- Proses pembuatan software dapat dilakukan dengan lebih cepat karena software dibangun dari objek-objek standar, dapat menggunakan ulang model yang ada, dan dapat membuat model dengan cepat melalui metodologi. Kualitas yang tinggi dari software dapat dicapai karena adanya *tested components*.
- Lebih mudah dalam maintenance karena perbaikan kode hanya diperlukan pada satu tempat (bukan diurut dari awal).
- Mudah dalam membangun sistem yang besar karena subsistem dapat dibuat dan diuji secara terpisah. Mengubah sistem yang sudah ada tidak memerlukan membangun ulang keseluruhan sistem.

Bahasa Pemrograman yang Menggunakan Object Oriented

Terdapat berbagai bahasa pemrograman yang menggunakan OO sekarang ini, diantaranya yang terkenal adalah SmallTalk, C++, Visual Basic, dan Java.