

Capaian Pembelajaran

Mata kuliah : PSBO
Sks : 3 sks

CAPAIAN PEMBELAJARAN :

Mahasiswa mengenal dan memahami serta dapat membuat model dengan diagram UML



PERTEMUAN 2

PEMODELAN BERORIENTASI OBJEK




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

Analisa dan Desain Berorientasi Object :

- Memahami domain permasalahan dan merumuskan solusinya dengan menggunakan pendekatan object(benda, konsep, atau entitas)
- Memudahkan pemanfaatan ulang code dan arsitektur
- Lebih mencerminkan dunia nyata (lebih tepat dalam menggambarkan entitas perusahaan, dekomposisi berdasarkan pembagian yang natural, lebih mudah untuk dipahami dan dirawat)
- Kestabilan (perubahan kecil dalam requirement tidak berarti perubahan yang signifikan dalam system yang sedang dikembangkan) Lebih mudah disesuaikan dengan perubahan



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



Teknologi Objek

- Teknologi perangkat lunak yang mendefinisikan sistem dalam hal obyek yang mengkonsolidasikan data dan tingkah laku (ke dalam obyek).
- Obyek menjadi dapat digunakan kembali dan komponen yang dapat diperluas untuk developer perangkat Lunak



OBJEK

- Objek adalah kombinasi antara struktur data dan perilaku dalam suatu entitas dan mempunyai nilai tertentu.
- objek, dapat digambarkan sebagai benda, orang, tempat dan sebagainya yang memiliki atribut dan metode.
- 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.



Contoh Objek

- obyek konkrit : rumah, sekolah, dosen, mahasiswa, dll.
- obyek abstrak : mata kuliah, penjadwalan, dll.



Karakteristik dari Objek

- Identitas berarti bahwa data yg diukur mempunyai nilai tertentu yang berbeda. Entitas disebut Objek.
- Setiap objek mempunyai sifat yang melekat pada identitasnya.
- Dua objek dapat berbeda walaupun semua nilai atributnya identik.
- Memiliki behavior (perilaku/method)
Perilaku suatu objek dinyatakan dalam operation.
- Perilaku suatu objek adalah mendefinisikan bagaimana objek tersebut bertindak dan memberikan reaksi.

CONTOH OBYEK



Atribut :

Nomor Polisi

Ban, Stir, Pedal

Kopling/Gas/Rem,

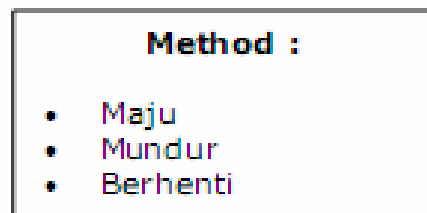
Warna, Tahun Produksi

Behavior/Tingkah Laku :

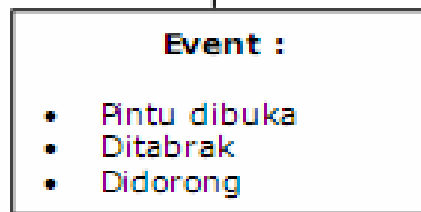
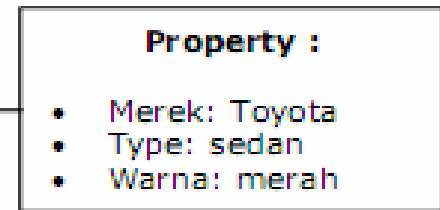
Cara Menghidupkan Mesin

Cara Manjalankan Mobil

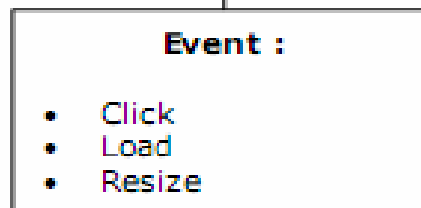
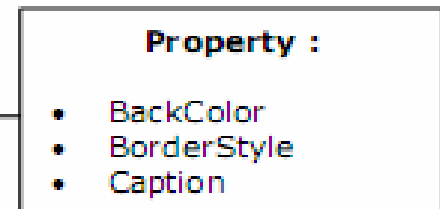
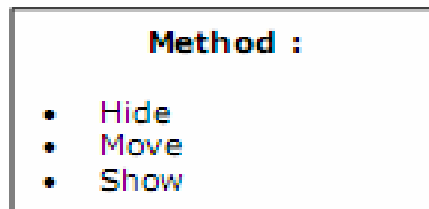
Cara Memundurkan Mobil



Object : Mobil



Object : Form





Istilah-istilah Objek

- **Atribut** : Data item yang menegaskan Objek, data yang menyatakan sifat/identitas dari obyek, karakteristik yang dimiliki obyek
- **Metode** : aksi yang dapat dilakukan oleh obyek, Pelaksanaan prosedur (bagian dari kode yang mengeksekusi respon terhadap permintaan objek lain didalam sistem), Fungsi di dalam kelas yang dikombinasikan ke bentuk tingkah laku kelas
- **Event** : kejadian yang dapat di alami oleh obyek



Kelas Objek

- Kelas merupakan gambaran sekumpulan Objek yang mempunyai sifat (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



Asosiasi

- Merupakan relasi atau hubungan obyek satu dengan lainnya.
- Sebuah obyek dapat berelasi dengan obyek yang lain



Metodologi Berorientasi Objek

- Encapsulation (pembungkusan)
- Inheritance (pewarisan)
- Polymorphism (polimorfisme)
- Abstraction (Abstraksi)
- Aggregation (Agregasi)



ENCAPSULATION ***(PENGKAPSULAN)***

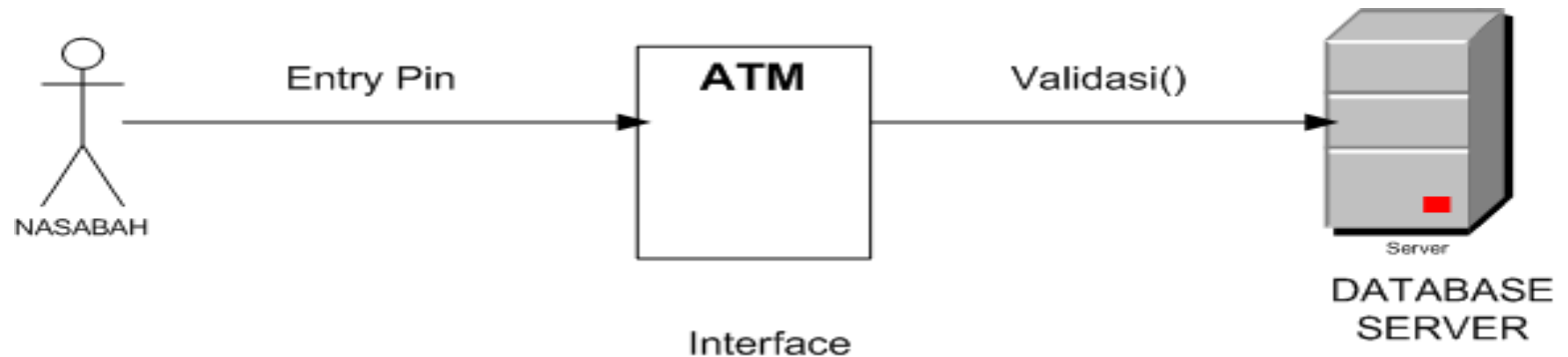
- Encapsulation merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses.
- Data dan prosedur atau fungsi dikemas bersama-sama dalam suatu objek, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya.
- Data terlindung dari prosedur atau objek lain, kecuali prosedur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



encapsulation

- **Adalah**
pembungkusan/pengemasan data dan fungsi dalam wadah bernama obyek.
- **atau...**Menyembunyikan operasi-operasi dari dunia luar dan dari obyek-obyek lain

CONTOH ENKAPSULASI



Disini terjadi penyembunyian informasi tentang bagaimana cara kerja pengecekan validitas kartu, kecocokan pin yang dimasukkan, koneksi ke database server, dll, dimana hal-hal tersebut tidak perlu diketahui oleh pengguna tentang bagaimana cara kerjanya.



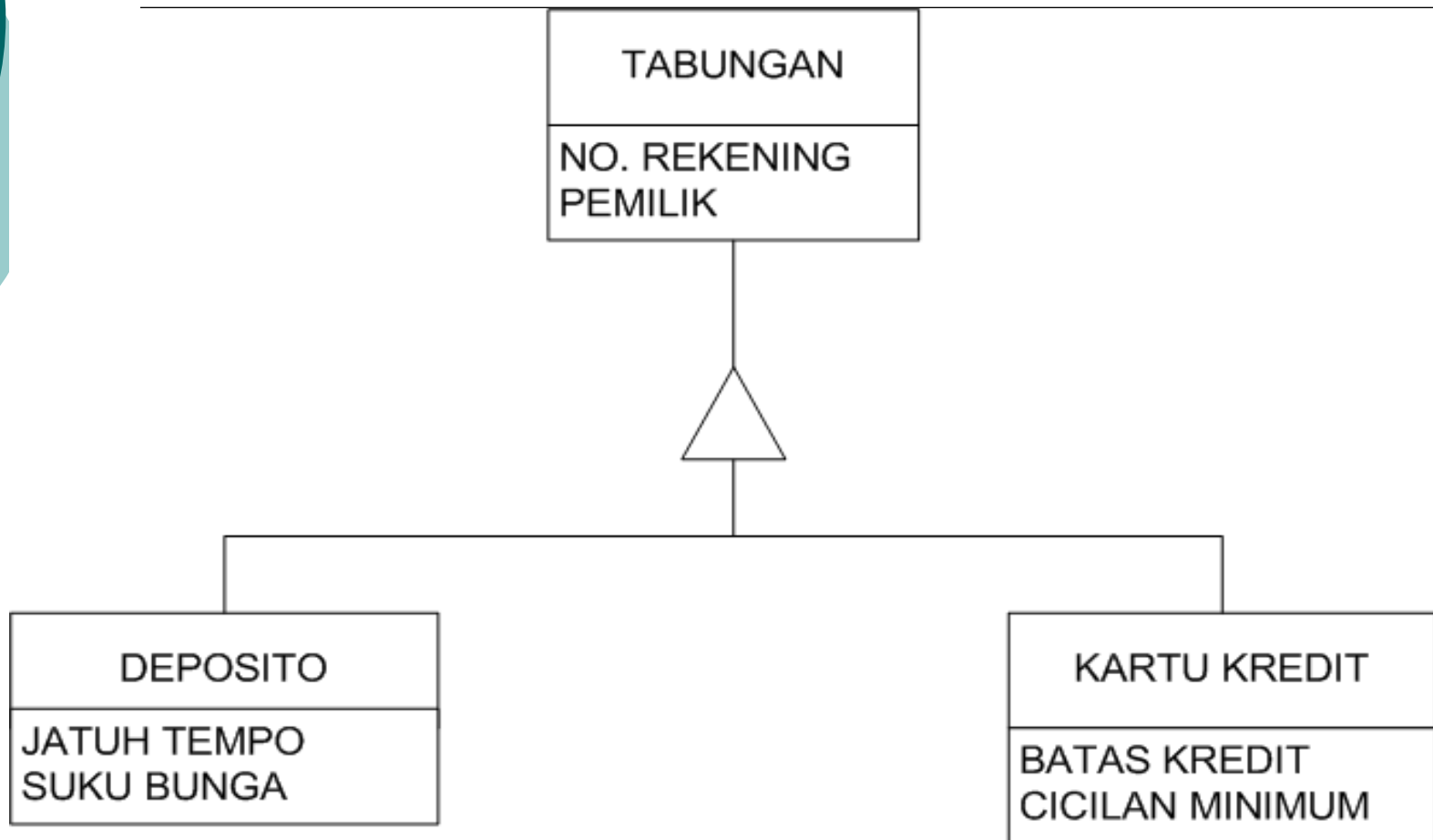
INHERITANCE (PEWARISAN)

- Inheritance adalah teknik yang menyatakan bahwa anak dari objek akan mewarisi data/atribut dan metode dari induknya. Atribut dan metode dari objek induk diturunkan kepada anak objek, demikian seterusnya.
- Inheritance mempunyai arti bahwa atribut dan operasi akan 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.
- 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

Contoh Inheritance





GENERALISASI

- Generalisasi adalah relasi antara beberapa subclass dengan superclass di atasnya.
- Kelas yang lebih rendah mewarisi semua atribut yang dimiliki oleh kelas yang lebih tinggi dan juga memiliki atribut yang membedakannya dengan kelas-kelas lain yang sederajat.
- Kelas yang mewarisi disebut superclass, kelas yang diwarisi disebut subclass

Contoh GENERALISASI

Generalization



**Inheritable
Attributes
and
Behaviors**

Specialization



+



+

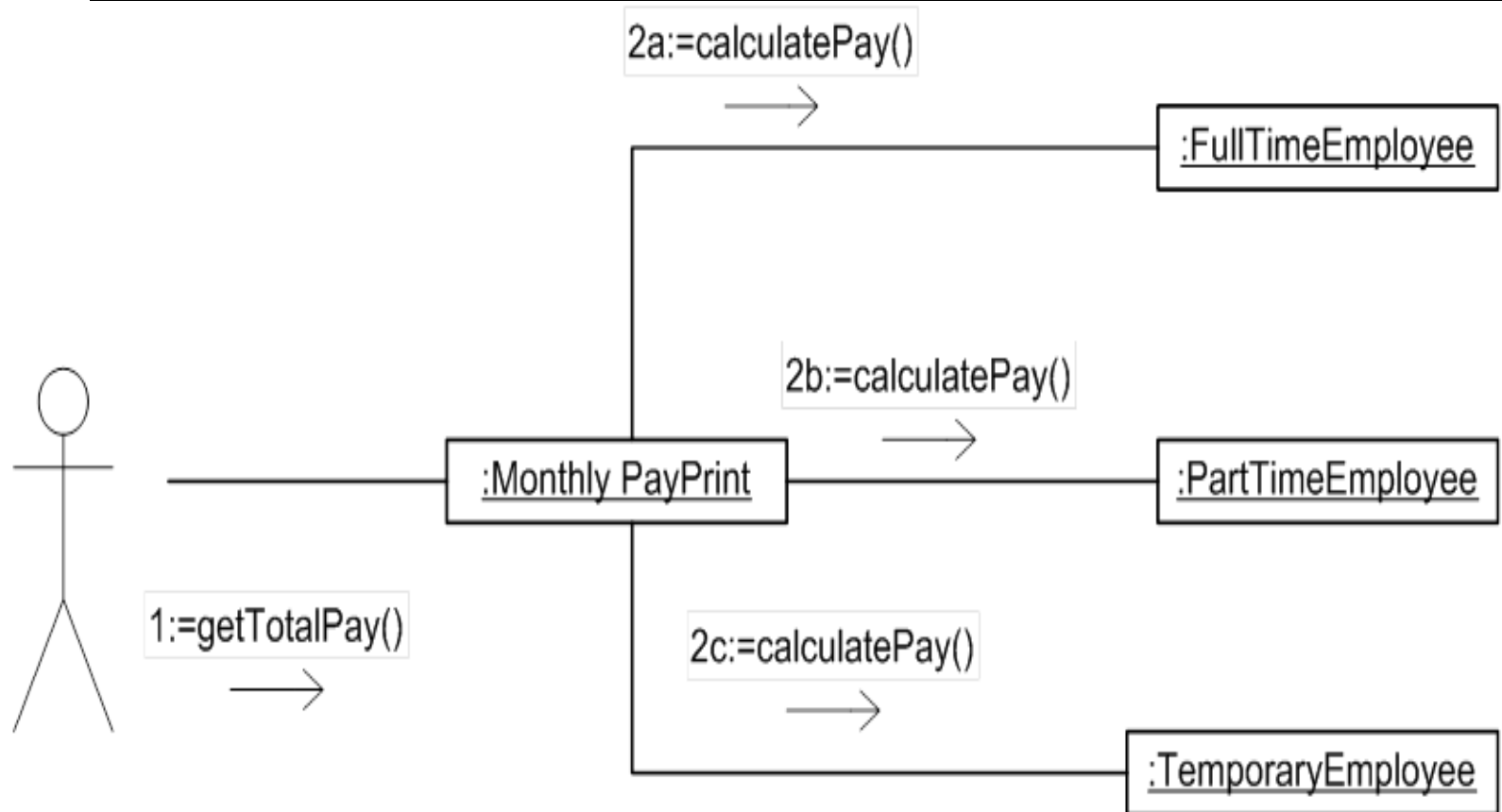




POLYMORPHISM

- 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 sesuai dalam merespon pesan yang sama.
- Seleksi dari metode yang sesuai, bergantung pada kelas yang seharusnya menciptakan Objek.

Contoh POLYMORPHISM





abstraksi

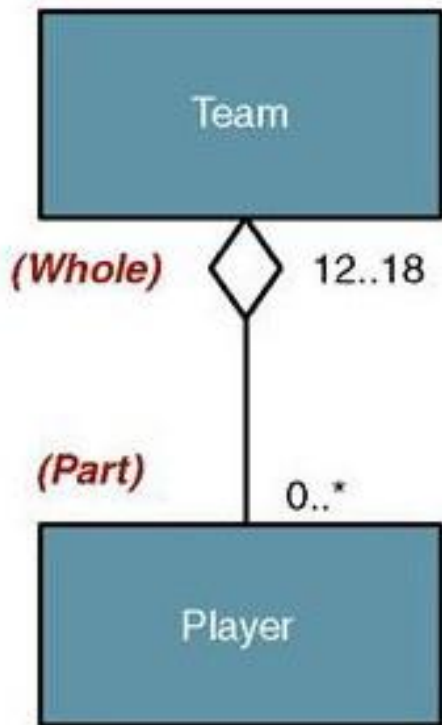
- Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
- Secara sederhana dikatakan membuang atribut obyek dan operasi yang hanya sampai pada yang benar-benar diperlukan.



Contoh Abstraksi

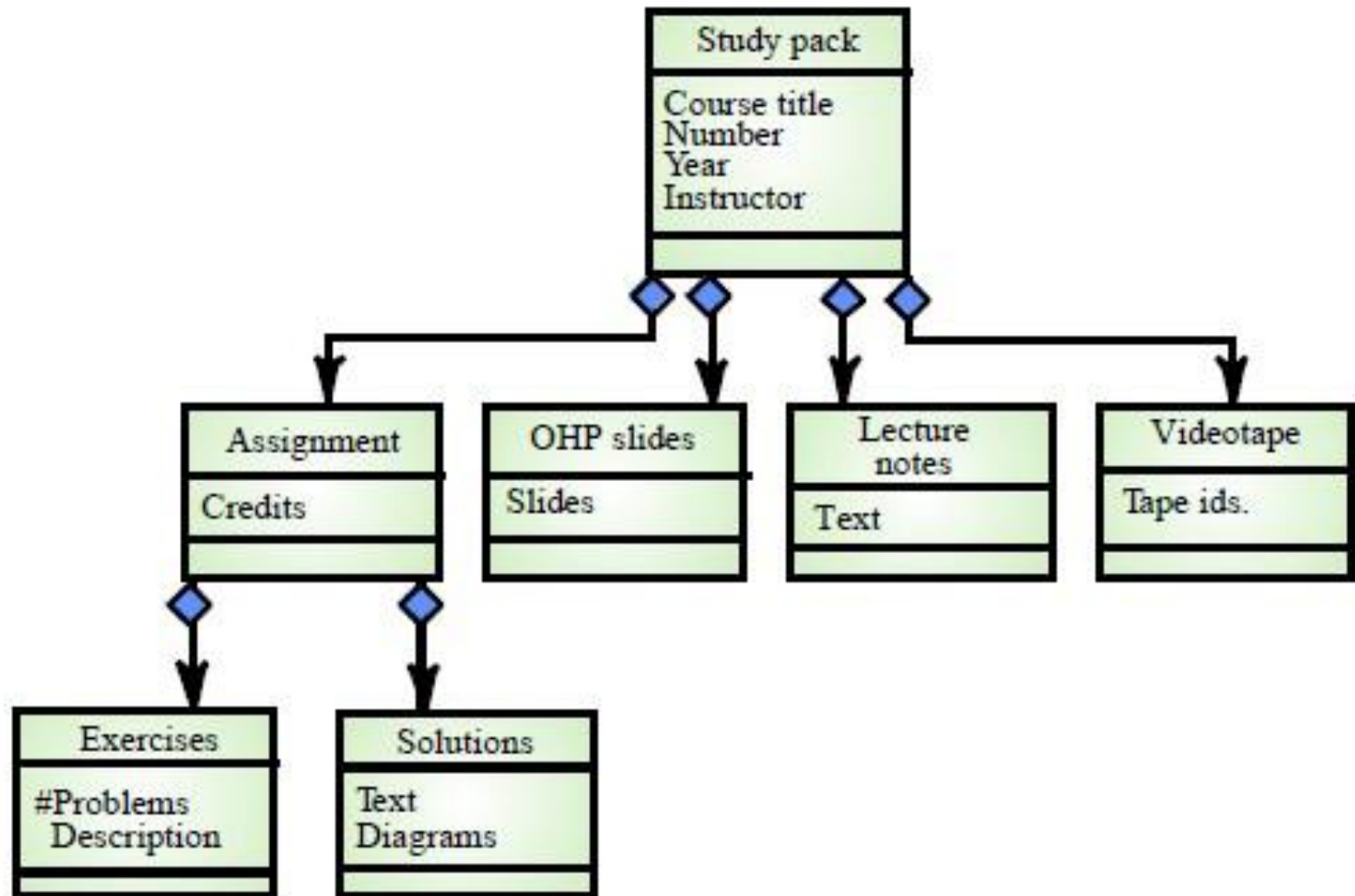
- Program mesin cuci → tidak butuh serial number
- Program transaksi laundry → butuh serial number
- Obyek sama-sama mesin cuci

Agregasi



- **Agregasi** – sebuah relasi yang menyatakan bahwa satu kelas “utuh (*whole*)” yang lebih besar memuat satu atau lebih kelas “bagian (*part*)” yang lebih kecil. Sebaliknya, kelas “bagian” adalah bagian dari kelas “utuh”
- Dalam UML 2.0 notasi agregasi sudah tidak dipakai lagi

Contoh Agregasi



Contoh Agregasi

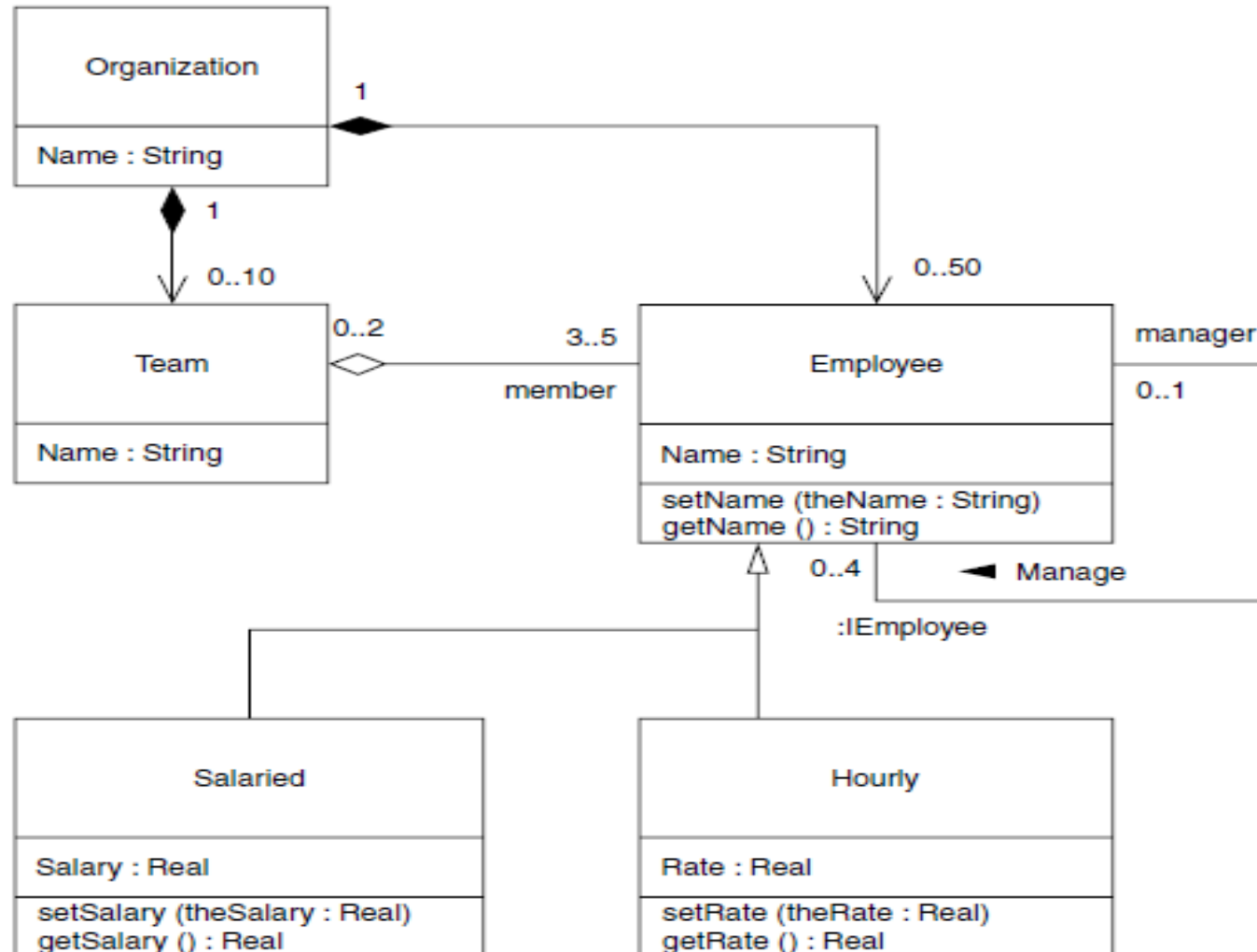


FIGURE 3.10. Aggregations, compositions, and generalizations between classes.



PEMODELAN BERORIENTASI OBJEK

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. Yang berupa *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*. Yang berbentuk graph dimana *nodenya* adalah *state* dan *arc* adalah transisi antara *state* yang disebabkan oleh *event*.

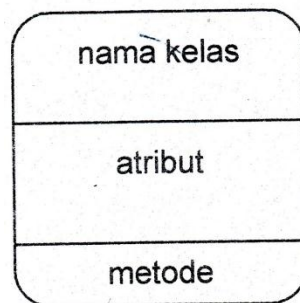


C. Model Fungsional

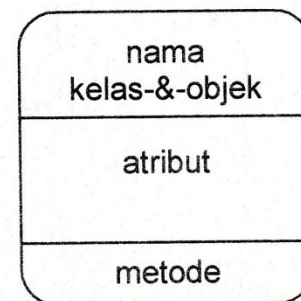
- Model fungsional menggambarkan transformasi nilai data di dalam sistem.
- Model fungsional berisi data flow diagram. DFD adalah suatu *graph* dimana *node*nya menyatakan proses dan *arc*nya adalah aliran data.

Diagram 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



kelas



kelas dengan objek



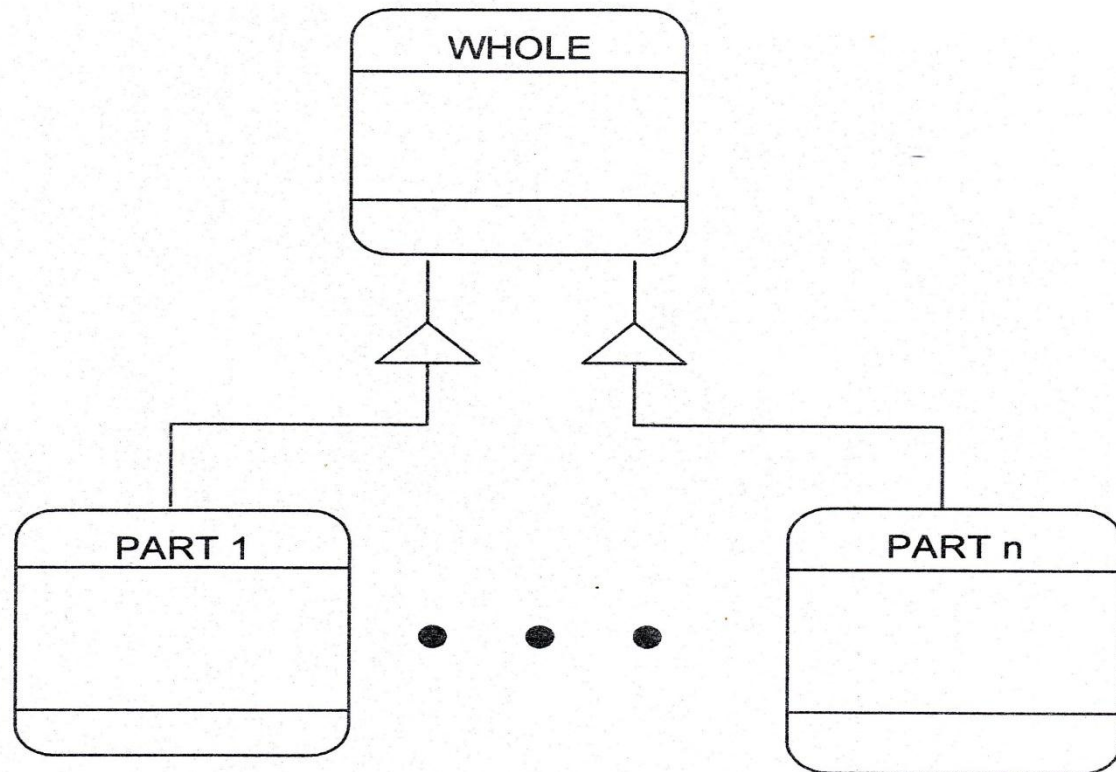
Struktur Objek dan Hirarki Kelas

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*.

Whole-Part Structure

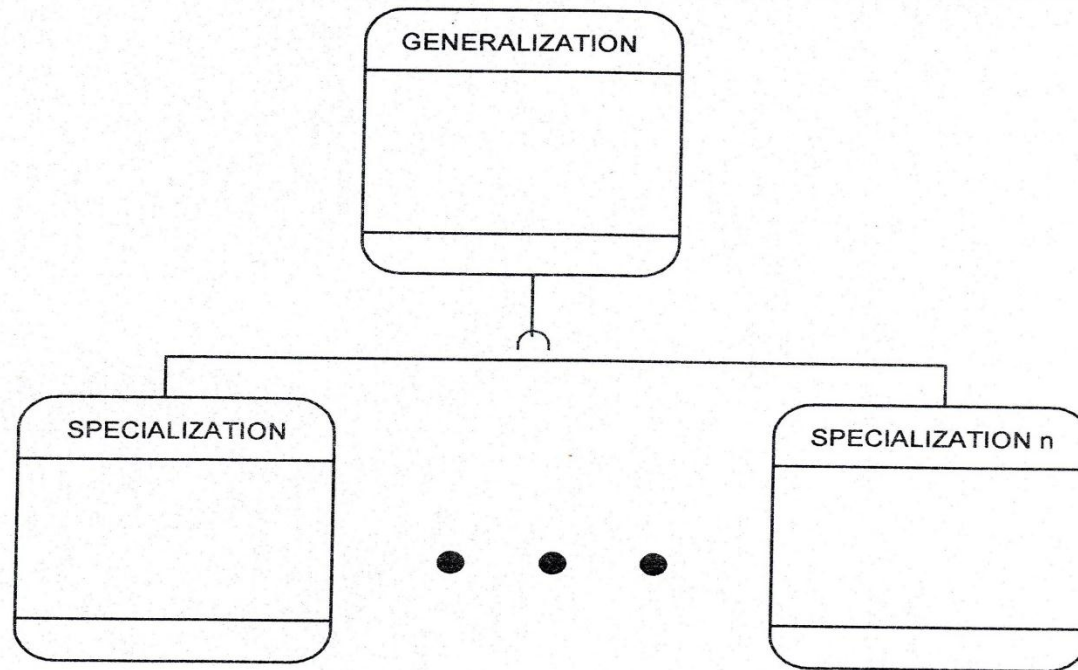




Gen-Spec Structure.

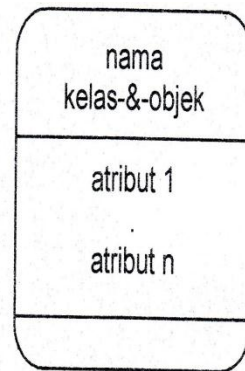
- ***Gen-Spec Structure*** memperlihatkan kelas sebagai spesialisasi dari kelas di atasnya. Kelas yang mempunyai sifat umum disebut *Generalization*, *Superclass* atau *Topclass*, sedangkan kelas yang mempunyai sifat khusus disebut *Specialization*.

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

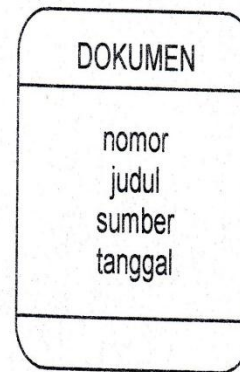


Atribut

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



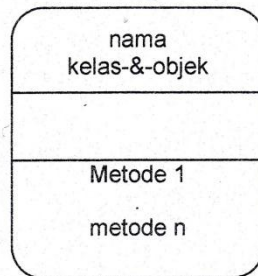
Notasi



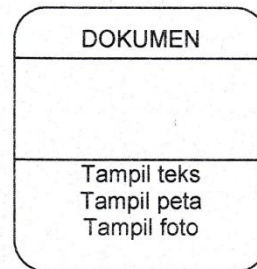
Contoh

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



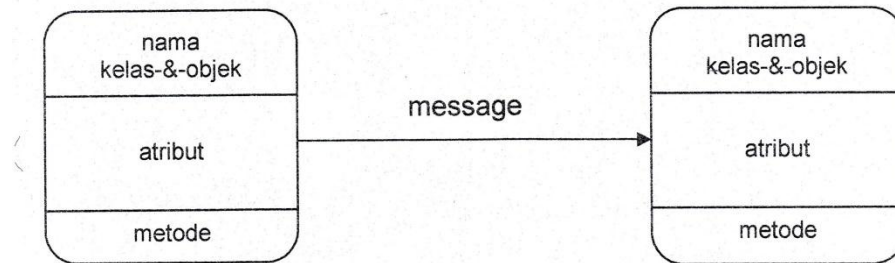
Notasi



Contoh

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





Structured vs Object-Oriented

- **Structured**

Pendekatan masalah berorientasi pada aksi **atau** data

- **Object-Oriented**

Pendekatan masalah berorientasi pada aksi **dan** data