

Kriteria Aplikasi Rancangan Object

Tujuan :

- Menjelaskan kopling, kohesi, dan penggunaan ulang sebagai cara mengukur kualitas desain

Kopling Interaksi

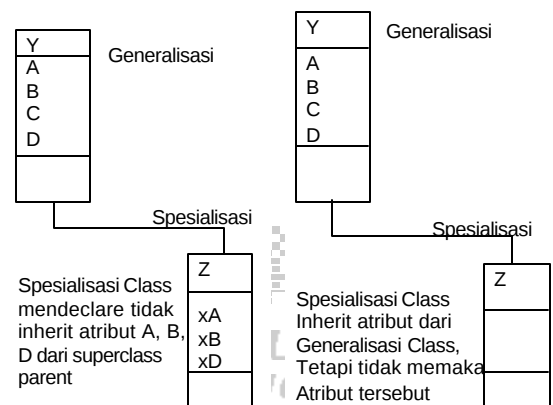
- Kopling interaksi yang rendah sangat diinginkan.
- Bila beberapa object yang saling terhubung melalui message yang kompleks akan terkopel terlampau ketat.
- Perubahan pada satu object akan berakibat adanya perubahan pada object lainnya.
- Untuk meminimumkan kompleksitas dari MessageConnection, jumlah message yang dikirim dan diterima oleh object harus diperkecil.

Kopling Inheritance

- Kopling inheritance yang tinggi sangat diinginkan, dimana inheritance merupakan bentuk kopling antara generalisasi Class dan spesialisasi Class.
- Untuk mencapai kopling inheritance tinggi, setiap spesialisasi Class harus merupakan spesialisasi dari generalisasi Class-nya.

Kopling

- Kopling menjelaskan tingkat saling ketergantungan atau hubungan antara komponen OOD.
- Kopling merupakan hal penting ketika mengevaluasi suatu desain.
- Perubahan pada satu bagian harus berakibat yang minimum pada bagian lainnya.
- Pada OOD, pengujian dan pengertian dari satu komponen harus mempunyai probabilitas minimum dalam usaha mengerti komponen lain.
- Tingkat kekuatan kopling diantara dua komponen diukur dengan jumlah dan kompleksitas dari informasi yang ditransmisikan antara dua komponen.



Kohesi

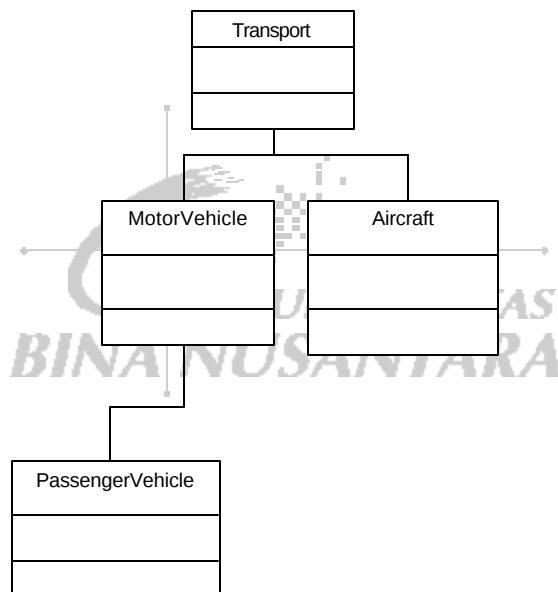
- Kohesi menjelaskan level saling keterhubungan dalam suatu grup komponen OOD, atau, suatu derajat dimana elemen dari bagian suatu desain berkontribusi untuk melakukan suatu tujuan tunggal yang terdefinisikan secara baik.

Kohesi Fungsi

- Fungsi mengerjakan satu, dan hanya satu fungsi. Bila lebih dari satu atau kurang dari satu tidak diinginkan.
- Cara lain untuk mengukur level kohesi adalah dengan cara memberi nama fungsi dengan kata yang menjelaskan fungsinya.

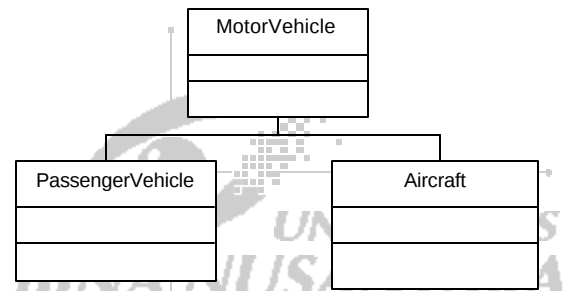
Kohesi Class

- Atribut dan Fungsi harus mempunyai tingkat kohesi tinggi, dan tidak ada atribut dan fungsi yang tidak terpakai.



Kohesi Generalisasi-Spesialisasi

- Bentuk kohesi Generalisasi-Spesialisasi digambarkan sebagai berikut:



- Kohesi Gen-Spec di atas kurang baik, karena "Aircraft" sebenarnya bukan "a kind of" MotorVehicle. Aircraft seharusnya ditempatkan lebih tinggi pada Struktur.
- Dalam hal ini, Transport, yang mempunyai spesialisasi pada MotorVehicle dan Aircraft, merupakan aplikasi generalisasi-spesialisasi yang lebih baik seperti terlihat pada gambar berikut:

Penggunaan Ulang (Reuse)

1. Mengapa menggunakan ulang?

Keuntungan utama adalah produktivitas tinggi, walaupun memerlukan:

- investasi untuk menciptakan reusable component saat pertama kali
- investasi untuk menghasilkan quality assurance tinggi
- investasi untuk memelihara library, browser dan fasilitas lain sehingga software engineer dapat menemukan komponen saat diperlukan

Selain itu reusability software memerlukan quality-assurance yang lebih tinggi daripada non-reusability software.

2. Derajat Reusability

Penggunaan ulang dari code

- Cut dan paste source code, bentuk paling primitif dari reusability
- Source-level includes, fasilitas untuk mengikutsertakan code dari library ke dalam program

- Inheritance, OOPL mempunyai fasilitas inheritance tunggal dan jamak
- Binary links, aktifitas yang dilakukan setelah proses kompilasi dengan cara menggabungkan modul object yang telah dikompilasi sebelumnya.
- Execution-time invocation, binding (pengikatan) antara program dengan reused komponen library-nya berlangsung saat program dieksekusi.

Penggunaan ulang hasil desain

Menggunakan OOD untuk memfasilitasi reporting aplikasi terhadap platform software maupun hardware.

CASE tool dapat dipakai untuk penggunaan ulang desain lama untuk membuat desain yang berbeda.

Penggunaan ulang hasil analisis

Level reuse yang lebih tinggi adalah pemakaian ulang spesifikasi, yang berisi model OOA. Berguna bila sistem diubah dari memakai teknologi hardware lama ke teknologi yang lebih baru.