

PERTEMUAN 7

ORIENTASI OBJEK

A. TUJUAN PEMBELAJARAN

Pada pertemuan ini dijelaskan tentang konsep perancangan berorientasi objek, konteks sistem dan model penggunaan, serta perancangan arsitektural. Dari pertemuan diharapkan mahasiswa mampu memahami konsep pendekatan perancangan berorientasi objek.

B. URAIAN MATERI

1. Konsep Perancangan Berorientasi Objek

Perancangan berorientasi objek adalah teknik atau metode baru untuk menemukan masalah dan sistem (sistem perangkat lunak, sistem informasi atau sistem lain). Pendekatan berorientasi objek akan memperlakukan sistem yang akan dikembangkan sebagai kumpulan objek yang sesuai dengan objek dunia nyata. Konsep "berorientasi objek" berarti kita mengatur perangkat lunak sebagai kumpulan objek tertentu dengan struktur data dan perilaku.

a. Karakteristik Dari Objek

Dalam desain berorientasi objek, objek itu sendiri memiliki banyak karakteristik. Berikut beberapa ciri dari benda tersebut:

Objek ialah Identitas artinya data yang diukur memiliki nilai tertentu, yang dapat membedakan entitas yang disebut objek. Objek bisa spesifik, seperti file dalam sistem, atau konseptual, seperti strategi penjadwalan dalam multiprocessing di sistem operasi. Setiap benda memiliki atribut yang melekat pada identitasnya. Meskipun semua nilai atribut sama, dua objek bisa berbeda.

Kelas Objek ialah Tinjauan sekumpulan objek yang dibagi menjadi atribut, operasi, metode, hubungan, dan arti yang sama. Mengumpulkan data (atribut) dan perilaku (operasi) dengan struktur data yang sama ke dalam sekumpulan aktivitas. Kelas objek adalah wadah untuk objek. Dapat

digunakan untuk membuat objek. Objek mewakili deskripsi fakta / kelas.

b. Metodologi Berorientasi Objek

Metodologi pengembangan sistem berorientasi objek mempunyai 3 karakteristik utama:

Encapsulation merupakan Batasi cakupan program pada basis data yang diproses. Data dan prosedur atau fungsi dikemas dalam suatu objek sehingga prosedur atau fungsi eksternal lainnya tidak dapat mengaksesnya. Data tidak dilindungi oleh proses atau objek lain, kecuali proses dalam objek itu sendiri.

Inheritance adalah Ditunjukkan bahwa anak-anak dari objek akan secara langsung mewarisi data / atribut dan metode dari induknya. Properti dan metode objek di objek induk akan diteruskan ke objek turunan, dan seterusnya. Pewarisan berarti bahwa atribut dan operasi yang dibagi antar kelas memiliki hubungan hierarki. Anda biasanya dapat mengidentifikasi kelas dan kemudian menetapkan sebagai subkelas. Setiap subclass memiliki hubungan atau mewarisi semua properti yang dimiliki oleh kelas induknya, dan menambahkan properti uniknya. Properti dan layanan kelas objek dapat ditentukan dari kelas Objek lainnya. Pewarisan menggambarkan generalisasi kelas.

Polimorfisme yaitu tunjukkan konsep bahwa benda yang sama dapat memiliki bentuk dan perilaku yang berbeda. Polimorfisme berarti bahwa operasi yang sama mungkin berbeda dalam kategori yang berbeda. Kemampuan objek yang berbeda untuk melakukan metode yang sesuai dalam menanggapi pesan yang sama. Pilihan metode yang sesuai bergantung pada kelas di mana Objek harus dibuat.

c. Kelebihan dan Kekurangan Perancangan Berorientasi Objek

Berikut kelebihan dan kekurangan perancangan berorientasi objek sebagai berikut :

1) Kelebihan:

- a) Dibandingkan dengan metode SSAD, OOAD lebih mudah digunakan dalam konstruksi sistem.
- b) Tidak ada pemisahan antara tahap desain dan analisis, sehingga komunikasi antara pengguna dan pengembang ditingkatkan dari

awal hingga akhir pengembangan sistem..

- c) Analis dan pemrogram tidak dibatasi oleh batasan implementasi sistem, sehingga mereka dapat mengonfirmasi desain melalui berbagai lingkungan eksekusi.
- d) Encapsulation data dan method agar dapat digunakan kembali dalam proyek lain, yang akan memudahkan proses desain, pemrograman, dan pengurangan harga.
- e) OOAD memungkinkan standarisasi objek, yang akan memudahkan pemahaman desain dan mengurangi risiko implementasi proyek.
- f) Dekomposisi obyek, Memungkinkan analis untuk menguraikan masalah menjadi bagian-bagian masalah dan bagian-bagian yang dikelola secara terpisah. Kode program dapat bekerja sama. Metode ini memungkinkan pengembangan perangkat lunak yang cepat, sehingga dapat segera dicantumkan dan kompetitif. Sistem akhir sangat fleksibel dan mudah dirawat.

2) Kekurangan:

- a) Pada tahap awal desain OOAD, sistemnya mungkin sangat sederhana.
- b) OOAD lebih fokus pada pengkodean daripada SSAD.
- c) OOAD tidak menekankan kinerja tim seperti SSAD.
- d) Di OOAD, tidak mudah untuk mendefinisikan kelas dan objek yang dibutuhkan oleh sistem.
- e) Umumnya, pemrograman berorientasi objek digunakan untuk menganalisis fungsi sistem, dan metode OOAD tidak didasarkan pada fungsi sistem.

2. Konteks Sistem dan Model Penggunaan

Pemodelan berorientasi objek dibagi menjadi dua bentuk, yaitu sebagai berikut:

a. Pemodelan Sebagai Teknik Desain

Teknik pemodelan objek menggunakan tiga model untuk menggambarkan sistem:

1) Model Objek

- a) Model objek menggambarkan struktur statis dan hubungan objek

dalam sistem.

- b) Model objek berisi diagram objek. Grafik objek adalah grafik di mana node adalah kelas-kelas yang memiliki hubungan antar kelas.

2) Model Dinamik

- a) Model dinamik menggambarkan aspek sistem yang berubah seiring waktu.
- b) Model dinamik Digunakan untuk mengekspresikan aspek kontrol dari sistem.
- c) Model dinamik terdapat state diagram. State diagram merupakan graph yang nodenya adalah state dan arc adalah pergatian antara state yang disebabkan oleh event.

3) Model Fungsional

- a) Model fungsional Menjelaskan konversi nilai data dalam sistem.
- b) Model fungsional Berisi diagram aliran data. DFD adalah grafik, di mana node menggambarkan proses, dan busur adalah aliran data.

b. Model Berorientasi Objek

Model objek menangkap struktur statis sistem dengan mendeskripsikan objek-objek dalam sistem, hubungan antar objek, serta karakteristik dan atribut dari setiap kelas dan objek. Model berorientasi objek lebih mendekati keadaan sebenarnya, dan dilengkapi dengan representasi grafis dari sistem, yang sangat berguna untuk berkomunikasi dengan pengguna dan mendokumentasikan struktur sistem..

1) Objek

- a) Objek didefinisikan sebagai konsep, abstrak atau objek dengan batasan dan makna masalah.
- b) Semua benda memiliki identitas yang berbeda dengan benda lainnya. Istilah "identitas" berarti bahwa objek dibedakan berdasarkan atribut yang melekat daripada dengan deskripsi atributnya. Misalnya, meskipun kembar identik terlihat mirip, mereka adalah dua orang yang berbeda.
- c) Terkadang objek mewakili item, jadi istilah "instance objek" dan "kelas objek" digunakan untuk menunjukkan sekelompok item yang identik.

2) Kelas

- a) Kelas objek mendeskripsikan sekumpulan objek dengan atribut (atribut), perilaku umum (operasi), hubungan umum dengan objek lain, dan semantik umum. Contoh: orang, perusahaan, hewan, proses adalah objek. Setiap orang memiliki usia, IQ, dan mungkin pekerjaan. Setiap proses memiliki pemilik, prioritas, dan daftar sumber daya yang diperlukan.
- b) Objek kelas dan objek biasanya sama dengan objek dalam deskripsi masalah.

3. Diagram Objek

Diagram objek adalah pelengkap simbol grafis dan digunakan untuk memodelkan objek, kelasnya, dan hubungannya dengan objek lain. Diagram objek berguna untuk pemodelan dan pemrograman abstrak.

a. Kelas dan Objek

Konsep dasar dari analisis berorientasi objek adalah objek itu sendiri. Objek adalah entitas yang berisi data dan metode. Kelas adalah satu atau lebih objek dengan properti dan metode yang sama, dan kelas - & - objek adalah kelas dengan satu atau lebih objek di dalamnya. Nama kelas adalah kata benda tunggal, atau kata sifat dan kata benda. Nama kelas - & - objek harus menggambarkan satu objek kelas.

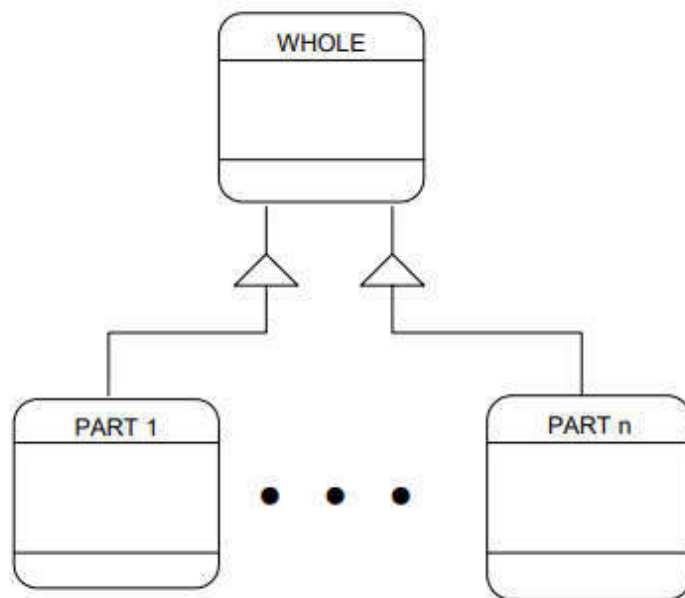


Gambar 20. Notasi untuk kelas dan kelas-&-objek

b. Struktur Objek dan Hirarki Kelas

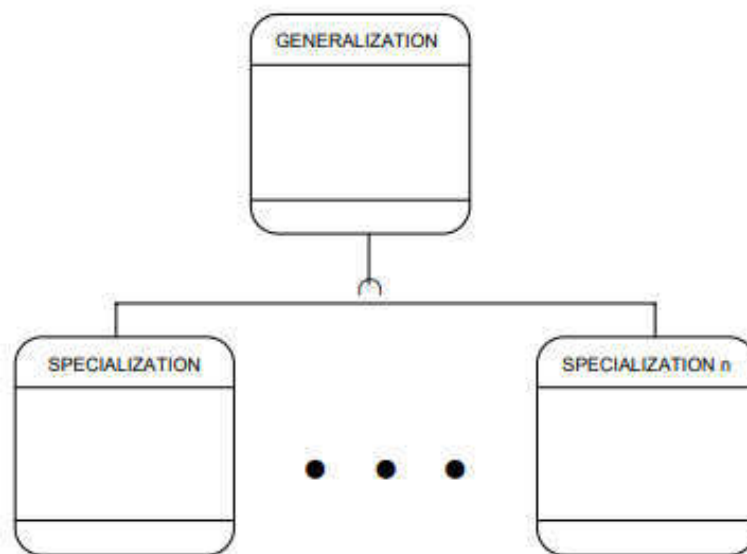
Struktur kelas dibagi menjadi dua jenis, yaitu **Whole-Part Structure** dan **Gen-Spec Structure**.

Whole-Part Structure Menampilkan struktur hierarki dari satu kelas sebagai komponen kelas lain. Kelas ini juga disebut sub-objek. Misalnya, kelas Mobil adalah Utuh, dan komponennya Mesin, Bingkai, dll. Adalah Part1, Part 2, ..., Partn.



Gambar 21. Notasi untuk whole-part structure

Gen-Spec Structure Tunjukkan kelas tersebut sebagai kursus khusus dari kelas di atas. Kelas dengan karakteristik umum disebut Generalisasi, Superclass atau Topclass, dan kelas dengan atribut khusus disebut Spesialisasi.

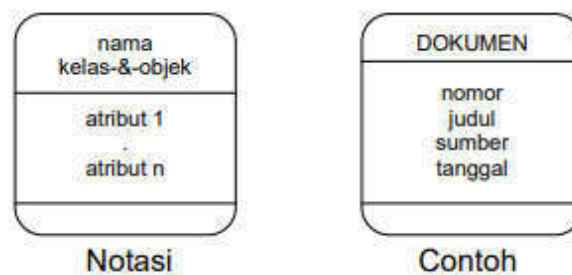


Gambar 22. Notasi untuk gen-spec structure

Misalnya, kategori "mobil" adalah "tujuan umum", sedangkan "mobil", "truk", "mobil penumpang kecil", dll. Adalah "spesialisasi 1", "spesialisasi 2", ..., "spesialisasi". Kategori fitur.

Atribut

Data yang dideskripsikan oleh atribut dapat memberikan informasi tentang kelas atau objek di mana atribut tersebut berada.

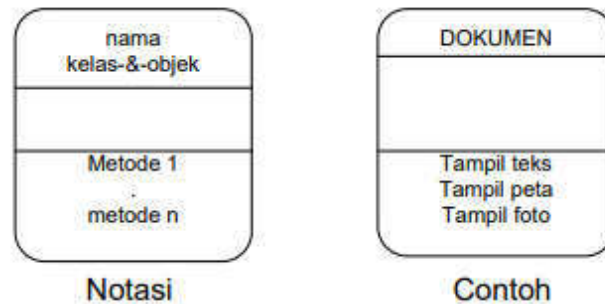


Gambar 23. Notasi untuk atribut

Metode

Metode (metode), juga disebut layanan atau operator, adalah prosedur atau fungsi yang biasanya ditemukan dalam bahasa Pascal, tetapi cara kerjanya sedikit berbeda. Metode adalah subrutin, yang menggabungkan objek dan

atribut. Metode ini digunakan untuk mengakses data yang terdapat dalam objek tersebut.



Gambar 24. Notasi untuk metode

Pesan (Message)

Message Ini adalah metode menghubungkan satu objek dengan objek lainnya. Pesan yang dikirim oleh objek ke objek tertentu dapat dijelaskan dengan panah.



Gambar 25. Notasi untuk message

4. Perancangan Arsitektural

Arsitektur bias bergantung pada sudut pandang tertentu, itu bisa berarti berbagai hal. Dari perspektif PBO, arsitektur standar UML merupakan aspek integral dari organisasi sistem. Aspek-aspek tersebut merupakan bagian dari arsitektur sistem, yang terdiri dari fungsi, struktur dan proses (fungsi, statik dan perilaku). Selain itu, arsitektur juga mengatasi kendala kinerja, skalabilitas, penggunaan kembali, serta ekonomi dan teknis..

Secara garis besar, arsitektur dapat dilihat dari sisi eksternal (pengguna), yaitu dalam bentuk fungsional. Selain itu, Anda juga dapat melihat bentuk

properti, metode, dan kelas dari dalam. Arsitektur juga melibatkan hubungan antar objek, objek ini berisi pesan untuk menjalankan fungsi tertentu.

Setelah interaksi antara sistem perangkat lunak yang dirancang dan lingkungan sistem didefinisikan sehingga dapat menggunakan informasi ini sebagai dasar untuk merancang arsitektur sistem..

- a. Lapisan antarmuka, tangani semua komunikasi eksternal.
- b. Lapisan pengumpulan data mengumpulkan dan merangkum semua data yang dibutuhkan oleh sistem
- c. Lapisan instrumen, merangkum semua instrumen untuk mengumpulkan data mentah

Ada dua desain arsitektur, sebagai berikut:

a. Arsitektural Logika

- 1) Arsitektur database yang menyediakan entitas dan hubungan tingkat rendah
- 2) Lapisan kedua meliputi proses bisnis inti dan logika bisnis yang diproses oleh sistem
- 3) Lapisan ketiga menyiapkan informasi rinci tentang aplikasi yang mendukung berbagai fungsi bisnis yang dibangun dari sistem ERP
- 4) Pengguna akhir:
 - a) Tingkat pertama dan kedua tidak pernah terlihat, karena mereka biasanya berinteraksi di tingkat aplikasi untuk menyediakan akses ke aplikasi fungsional
 - b) Hanya mengakses lapisan ketiga (aplikasi antarmuka klien-pengguna). Mereka memberikan layanan kepada pengguna akhir dalam bentuk akses fungsional, termasuk setiap fungsi dan modul yang dicakup oleh sistem ERP.

b. Arsitektural Fisik

- 1) Fokus pada efisiensi sumber daya sistem.
- 2) Sumber daya yang terlibat: biaya, waktu, respons terhadap jumlah perangkat, dll..
- 3) Jadikan seluruh sistem lebih skalabel dan kurangi sumber daya yang diperlukan.

C. SOAL LATIHAN/TUGAS

1. Sebutkan dan jelaskan model pada orientasi objek!
2. Sebutkan dan jelaskan Karakteristik pada Objek!
3. Menurut anda apa keuntungan dan kekurangan menggunakan orientasi objek!
4. Sebutkan dan jelaskan metodologi berorientasi Objek!
5. Jelaskan yang dimaksud Whole-Part Structure dan Gen-Spec Structure!

D. REFERESI

Rumbaugh James, et all, 1999, "The Unified Modeling Language Reference Manual".

Hoffer, A.Jeffrey and George, F, Joey, *Modern System Analysis and Design*, Prentice Hall-Inc, 2002

McGraw Hill., *UML*, 1999

Edward V. Berard, *Origins Objects Oriented Technology*

GLOSARIUM

User adalah pengguna pada layanan atau perangkat dalam sistem teknologi informasi.

Methods atau metode adalah kumpulan pernyataan yang dikelompokkan bersama-sama untuk melakukan operasi.

Class atau Kelas adalah cetak biru untuk membuat objek.

Reuse berarti 'menggunakan kembali'. Langkah ini mengajak kita untuk menggunakan kembali produk yang sudah dipakai.

+Enkapsulasi (encapsulation) adalah sebuah metoda untuk mengatur struktur class dengan cara menyembunyikan alur kerja dari class tersebut.