

MODUL

PERANCANGAN BERORIENTASI OBJEK

BAB I PENDAHULUAN

A. Konsep Dasar Pendekatan Objek

- Suatu teknik atau cara pendekatan baru dalam melihat permasalahan dari sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya).
- Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata.
- Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem.
- Saat mengabstraksikan dan memodelkan objek ini, data dan proses-proses yang dipunyai oleh objek akan dienkapsulasi (dibungkus) menjadi satu kesatuan.

Contoh:

Tinjau aktivitas kuliah pada suatu sistem akademik sebagai berikut:



Dari aktivitas kuliah tersebut, secara eksplisit ada 3 objek yang langsung dapat dikenali yaitu **Dosen** yang memberikan kuliah, **Mahasiswa** yang mengikuti kuliah, dan **Materi Kuliah** yang disampaikan. Secara implisit, ada 2 objek lain yang bisa dikenali lagi yaitu **Jadwal** kapan kuliah diadakan dan **Nilai** yang didapat mahasiswa dari kuliah yang sudah diikutinya. Abstraksi dan pemodelan untuk salah satu dari kelima objek tersebut, misalnya untuk objek Dosen adalah:

Objek Dosen



diabstraksikan
menjadi →

Kelas Dosen

Atribut

Kode_Dosen
Nama_Dosen
Pendidikan
...

Operasi

Rekam
Update
Retrieve
...

- Dalam rekayasa perangkat lunak, konsep pendekatan berorientasi objek dapat diterapkan pada tahap analisis, perancangan, pemrograman, dan pengujian perangkat lunak.
- Ada berbagai teknik yang dapat digunakan pada masing-masing tahap tersebut, dengan aturan dan alat bantu pemodelan tertentu.

B. Objek dan Kelas

Apakah yang disebut objek? Apakah yang disebut kelas? Adalah sangat penting untuk membedakan antar objek dengan kelas

Objek

- Objek adalah abstraksi dari sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status atau hal-hal lain yang bersifat abstrak.
- Suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya.
- Dalam konteks OOP, objek adalah instansiasi (yang dibentuk secara seketika) dari kelas pada saat eksekusi (seperti halnya deklarasi variabel pada pemograman prosedural). Jadi semua objek adalah instan dari kelas.
- Objek mempunyai siklus hidup: diciptakan, dimanipulasi, dan dihancurkan.

Kelas

- Kelas adalah kumpulan dari objek-objek dengan karakteristik yang sama.
- Kelas adalah definisi statik dari himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut.
- Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi), hubungan (*relationship*) dan arti.
- Suatu kelas dapat diturunkan dari kelas yang lain, dimana atribut dari kelas semula dapat diwariskan ke kelas yang baru.

Kesimpulan:

- Objek adalah model eksekusi, sementara kelas adalah deskripsi statik dari objek yang mungkin lahir pada saat eksekusi.
- Pada saat eksekusi yang kita punyai adalah objek, sementara dalam pemodelan (analisis dan perancangan) dan teks program yang kita lihat adalah kelas.

C. Property Objek

Sebuah objek pada dasarnya mempunyai property sebagai berikut:

Atribut

- Nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek.
- Merupakan ciri dari sebuah objek

- Dipunyai secara individual oleh sebuah objek.
- Contoh: berat, warna, jenis, nama, dan sebagainya.

Layanan (Service)

- Metode atau operasi yang berfungsi untuk memanipulasi objek itu sendiri.
- Fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek.
- Dapat berasal dari:
 - model objek
 - *event*
 - aktivitas atau aksi keadaan
 - fungsi
 - kelakuan dunia nyata
- Contoh: *Read, Write, Move, Copy* dan sebagainya.

Klasifikasi Objek

Menurut [BOO95] objek dapat menjadi:

- ADT (*Abstract Data Type*)
Definisi dari kelas dimana komponen *type* menjadi atribut dan fungsi primitif menjadi operasi/metode/layanan.
- Mesin
Objek pasif yang punya status yang akan diaktifkan oleh objek lain. Fungsi primitif pada mesin merupakan mekanisme transisi yang mengubah suatu status ke status lain.
- Proses
Objek aktif yang mempunyai “urutan kendali” (*thread of control*)

D. Sistem Berorientasi Objek

Definisi

- Sebuah sistem yang dibangun dengan berdasarkan metode berorientasi objek adalah sebuah sistem yang komponennya dibungkus (dienkapsulasi) menjadi kelompok data dan fungsi.
- Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dari komponen lainnya, dan dapat berinteraksi satu sama lainnya.

Karakteristik Sistem Berorientasi Objek

Karakteristik atau sifat-sifat yang dipunyai sebuah sistem berorientasi objek adalah:

- Abstraksi
Prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.
- Enkapsulasi

Pembungkusan atribut data dan layanan (operasi-operasi) yang dimiliki objek, untuk menyembunyikan implementasi dari objek sehingga objek lain tidak mengetahui cara kerjanya.

- **Pewarisan (*inheritance*)**
Mekanisme yang memungkinkan satu objek (baca: kelas) mewarisi sebagian atau seluruh definisi dari objek lain sebagai bagian dari dirinya.
- ***Reusability***
Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.
- **Generalisasi dan Spesialisasi**
Menunjukkan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.
- **Komunikasi antar Objek**
Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dari satu objek ke objek lainnya.
- ***Polymorphism***
Kemampuan suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

BAB II METODOLOGI BERORIENTASI OBJEK

A. Beberapa Prinsip Dasar

Pengertian Metodologi

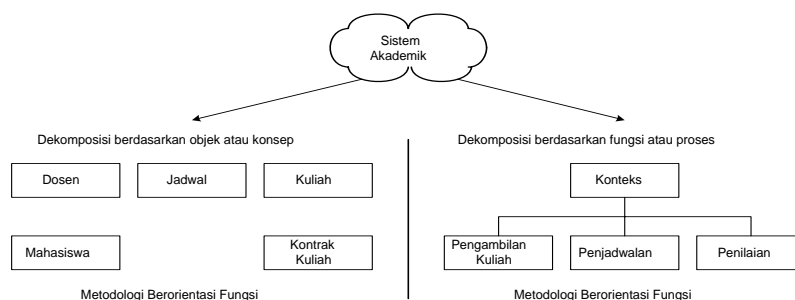
- Cara kerja yang sistematis untuk memudahkan pelaksanaan pembuatan perangkat lunak guna mencapai tujuan tertentu.
- Proses untuk menghasilkan perangkat lunak yang terorganisir dengan menggunakan sejumlah teknik dan konvensi notasi yang terdefinisi.

Pengertian Metodologi Berorientasi Objek

- Suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.
- Suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis.
- Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas.
- Metode berorientasi objek meliputi rangkaian aktivitas **analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.**
- Ada teknik yang digunakan, produk yang dihasilkan, prosedur verifikasi, dan kriteria untuk setiap aktivitas yang dikerjakan.
- Ada alat bantu untuk memodelkan (mendokumentasikan) hasil dari setiap aktivitas.

Metodologi Berorientasi Objek vs Fungsi

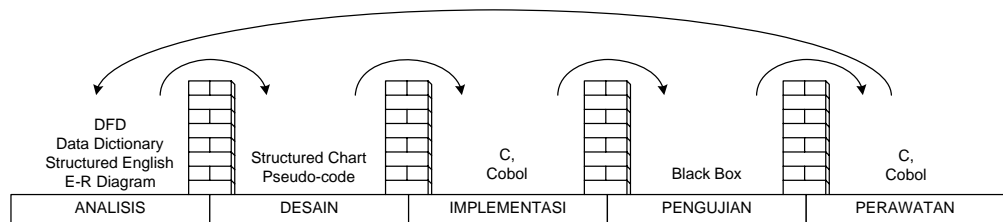
- Strategi utama untuk menangani kompleksitas pembangunan perangkat lunak adalah dekomposisi permasalahan menjadi bagian-bagian kecil yang dapat dikelola.
- Pada metode berorientasi fungsi atau aliran data (DFD), dekomposisi permasalahan dilakukan berdasarkan fungsi atau proses secara hirarki, mulai dari konteks sampai proses-proses yang paling kecil.
- Sementara pada metode berorientasi objek, dekomposisi permasalahan dilakukan berdasarkan objek-objek yang ada dalam sistem.
- Perbandingan kedua metode tersebut, misalnya untuk masalah penilaian kuliah pada suatu sistem akademik, ditunjukkan oleh gambar berikut ini.



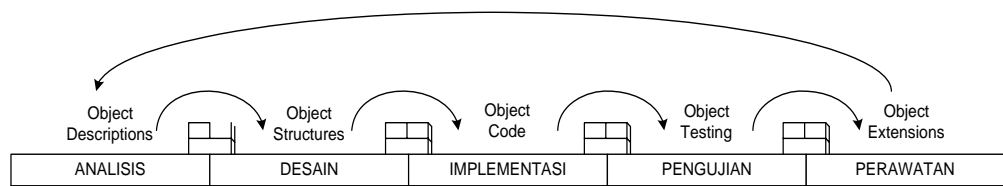
Mengapa Metodologi Berorientasi Objek ?

Metode berorientasi objek banyak dipilih karena :

- Metodologi lama banyak menimbulkan masalah
Adanya kesulitan pada saat mentransformasi hasil dari suatu tahap pengembangan ke tahap berikutnya, misalnya pada metode *Structured Analysis and Design*



Bandingkan dengan pendekatan objek sebagai berikut :



- Jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan pada saat ini sangat beragam (aplikasi bisnis, real-time, utility dan sebagainya) dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan Metodologi Berorientasi Objek

Pengembangan sistem dengan metode berorientasi objek dapat memberikan keuntungan-keuntungan sebagai berikut, walaupun beberapa buku (referensi) menunjukkan “pitfall” dari metodologi ini.

- Meningkatkan produktivitas
Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable*).
- Kecepatan pengembangan
Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.
- Kemudahan pemeliharaan
Karena dengan objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dari pola-pola yang mungkin sering berubah.

- Adanya konsistensi
Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan, maupun pengkodean.
- Meningkatkan kualitas perangkat lunak
Karena pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

B. Beberapa Metode Berorientasi Objek

Ada beberapa metode pengembangan perangkat lunak berorientasi objek yang sudah dikenal dan diantaranya adalah :

- *Object Oriented Analysis* (OOA) dan *Object Oriented Design* (OOD) dari Peter Coad dan Edward Yourdan [1990].
- *Object Modeling Technique* (OMT) dari James Rumbaugh, Michael Blaha, William Premerlan, Frederick Eddy dan William Lorenson [1991].
- *Object Oriented Software Engineering* (OOSE) dari Ivar Jacobson [1992].
- *Booch Method* dari Grady Booch [1994].
- *Syntropy* dari Steve Cook dan John Daniels [1994].
- UML (*Unified Modeling Language*) dari James Rumbaugh, Grady Booch dan Ivar Jacobson [1997].

BAB III

ANALISIS BERORIENTASI OBJEK

A. Pengertian Dasar

Analisis

- Penguraian suatu pokok atas berbagai bagiannya dan penelaahan bagian itu sendiri serta hubungan antar bagian untuk memperoleh pengertian yang tepat dan pemahaman arti keseluruhan.
- Studi dari suatu permasalahan dengan cara memilah-milah permasalahan tersebut sehingga dapat dipahami dan dievaluasi, sebelum diambil tindakan-tindakan tertentu.

Analisis Berorientasi Objek

- Investigasi masalah untuk menemukan (mengidentifikasi) dan mendefinisikan objek-objek atau konsep-konsep yang ada di ruang masalah.
- Proses untuk menentukan objek-objek potensial yang ada dalam sistem dan mendeskripsikan karakteristik dan hubungannya dalam sebuah notasi formal.
- Aplikasi konsep berorientasi objek untuk memodelkan permasalahan dan sistem, baik untuk lingkup perangkat lunak maupun non-perangkat lunak.

Tujuan Analisis

- Memahami permasalahan secara menyeluruh.
- Mengungkapkan apa yang harus dikerjakan oleh sistem untuk memenuhi kebutuhan pemakai.
- Mengetahui ruang lingkup produk (*product space*) dan pemakai yang akan menggunakan produk tersebut.

Tahap Analisis

- Mempelajari permasalahan
- Menentukan kebutuhan pemakai
- Mengubah kebutuhan yang belum terstruktur menjadi model-model atau gambar-gambar dengan memanfaatkan metode dan teknik analisis tertentu.
- Mendokumentasikan hasil analisis, misalnya *Software Requirement Specification* (SRS).

B. Metode Analisis Berorientasi Objek

Pengertian

- Cara kerja yang sistematis untuk mengerjakan tahap analisis berdasarkan pendekatan objek.
- Ada kumpulan aturan-aturan tertentu yang harus diikuti untuk menyelesaikan pekerjaan analisis tersebut.

- Mempunyai urutan aktivitas, teknik, dan alat bantu (*tools*) tertentu untuk memodelkan (mendokumentasikan) hasil dari setiap aktivitas.
- Ada beberapa metode yang dapat digunakan untuk melakukan analisis berorientasi objek, dan diantaranya adalah sebagai berikut.

Metode Coad & Yourdan

- Diperkenalkan oleh Peter Coad dan Edward Yourdan pada tahun 1990.
- Disebut juga dengan nama *Object Oriented Analysis* (OOA), dan dipandang sebagai salah satu teknik yang mudah untuk dipelajari.
- Notasi model relatif sederhana karena didasarkan pada struktur fisik dunia nyata, dan petunjuk untuk melakukan analisis cukup jelas.
- Tahap atau skema pelaksanaan:
 - Identifikasi kelas dan objek
 - Identifikasi struktur
 - Struktur "*generalization-specification*"
 - Struktur "*whole-part*" atau "*a-part-of*"
- Identifikasi subjek
- Definisikan atribut
 - Atribut implisi objek
 - Koneksi instan (*instance connection*)
- Definisikan layanan
 - Layanan implisit objek
 - Layanan yang berasosiasi dengan atribut
 - Layanan yang berasosiasi dengan "*message-connection*"

Metode Rumbaugh

- Diperkenalkan oleh James Rumbaugh, Michael Blaha, William Premerlan, Frederick Eddy dan William Lorenzen pada tahun 1991.
- Lebih dikenal dengan *Object Modeling Technique* (OMT) yang dapat digunakan baik untuk analisis maupun desain.
- Selain model-model fisik dari objek, pendekatan analisis dilakukan juga untuk model-model dinamik dan model fungsional.
- Tahap atau skema pelaksanaan:
 - Tentukan ruang lingkup masalah
 - Buat model objek
 - Identifikasi kelas yang relevan dengan permasalahan
 - Definisikan atribut dan asosiasi
 - Definisikan keterkaitan (*link*) antar kelas dan objek
 - Organisasikan kelas objek dengan menggunakan pewarisan
 - Buat model dinamik
 - Siapkan skenario
 - Definisikan kejadian (*event*) dan buat penelusurannya untuk setiap skenario
 - Bangun diagram aliran kejadian (*event flow diagram*)
 - Buat diagram keadaan (*state diagram*)

- Buat model fungsional sistem
 - Identifikasikan masukan dan keluaran
 - Gunakan diagram aliran data untuk merepresentasikan aliran transformasi
 - Buat spesifikasi proses untuk setiap fungsi

Metode Jacobson

- Diperkenalkan oleh Ivar Jacobson dengan nama *Object Oriented Software Engineering* (OOSE) pada tahun 1992.
- Merupakan versi yang juga sederhana dari metode berorientasi objek.
- Sudut pandang atau fokus analisis ditekankan pada “*use case*”, yaitu deskripsi atau skenario yang menggambarkan bagaimana pemakai berinteraksi dengan produk atau sistem yang akan dikembangkan.
- Tahap atau skema pelaksanaan:
 - Identifikasi pemakai sistem dan semua tanggung jawabnya
 - Buat model kebutuhan
 - Definisikan aktor dan tanggung jawabnya
 - Identifikasi use-case untuk setiap aktor
 - Inisialisasi gambaran sistem objek dan hubungannya
 - Buat model analisis
 - Identifikasi antarmuka objek
 - Buat gambaran struktural dari antarmuka objek
 - Representasikan perilaku objek
 - Isolasi sub-sistem dan buat masing-masing modelnya

Metode Booch

- Diperkenalkan oleh Grady Booch pada tahun 1994.
- Meliputi proses pengembangan makro dan mikro, dengan anggapan bahwa analisis dan desain merupakan rangkaian kesatuan aktivitas yang tidak dipisahkan.
- Tahap atau skema pelaksanaan:
 - Identifikasi kelas dan objek
 - Identifikasi kandidat objek
 - Identifikasi skenario yang relevan
 - Definisikan atribut dan layanan untuk setiap kelas
 - Identifikasi Semantik dari kelas dan objek
 - Pilih skenario kemudian analisis
 - Pilih objek dan daftar peran serta tanggung jawabnya
 - Cari kolaborasi diantara objek-objek
 - Identifikasi hubungan diantara kelas dan objek
 - Definisikan ketergantungan yang ada diantara objek
 - Jelaskan peran dari setiap objek
 - Validasi berdasarkan skenario
- Buat diagram yang berhubungan dengan langkah-langkah di atas
- Implementasikan kelas dan objek

C. Metode Analisis Secara Umum

- Pada prinsipnya semua metode analisis berorientasi objek adalah sama, perbedaan hanya terletak pada sudut pandang dan teknis pelaksanaannya.
- Secara umum, metode analisis berorientasi objek mencakup representasi kelas dan hirarki kelas, model hubungan objek, dan model perilaku objek.
- Tahap atau skema pelaksanaan analisis berorientasi objek :
 - Tentukan kebutuhan pemakai untuk sistem berorientasi objek
 - Identifikasi kelas dan objek
 - Identifikasi atribut dan layanan untuk setiap objek
 - Definisikan struktur dan hirarki
 - Buat model hubungan objek
 - Buat model perilaku objek

Menentukan Kebutuhan Pemakai untuk Sistem Berorientasi Objek

- Mengidentifikasi proses-proses bisnis dan kebutuhan pemakai dan mengekspresikan dengan ‘*use-case*’.
- Sebenarnya bukan merupakan aktivitas analisis berorientasi objek, karena tidak membicarakan pembahasan tentang objek.
- Diperlukan karena dapat menjelaskan aktivitas-aktivitas apa saja yang harus dikerjakan oleh sistem, dan menjelaskan juga perilaku dari komponen-komponen sistem.
- Ada diagram tertentu yang dapat merepresentasikan model kebutuhan dari “*use-case*” yang diperoleh.

Identifikasi Kelas dan Objek

- Mengidentifikasi kelas-kelas dan objek-objek yang ada dalam lingkup aplikasi:
 - eksplisit pada pernyataan masalah
 - implisit pada lingkup aplikasi atau pengetahuan atas lingkup aplikasi
- Kelas dan objek dapat diidentifikasi dari:
 - **entitas** eksternal yang memproduksi dan memakai informasi yang akan digunakan oleh sistem berbasis komputer
 - sesuatu yang merupakan bagian dari wilayah informasi dari permasalahan
 - kejadian, misalnya prosedur operasional, yang muncul dalam lingkup operasional sistem
 - peran yang dimainkan oleh orang-orang yang berinteraksi dengan sistem
 - unit organisasi yang relevan dengan aplikasi
 - tempat yang menentukan ruang lingkup masalah dan seluruh fungsi dari sistem
 - struktur yang mendefinisikan kelas dari objek atau yang menghubungkan kelas-kelas objek.
- Abaikan kelas dan objek yang tidak tepat karena:
 - redunden
 - tidak relevan
 - lebih tepat berupa atribut

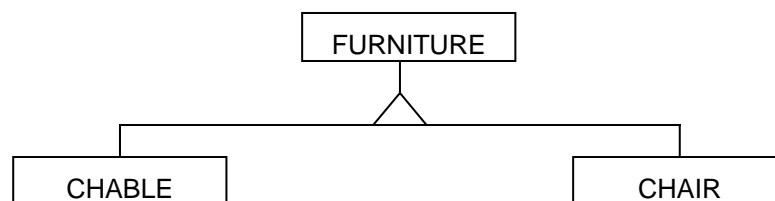
- lebih tepat berupa operasi
- lebih tepat berupa peran
- lebih merupakan konstruksi implementasi.

Identifikasi Atribut dan Layanan

- Mengidentifikasi atribut dan layanan yang terkait untuk setiap atribut tersebut.
- Atribut diidentifikasi dari elemen-elemen data yang dapat menggambarkan (mencirikan) sebuah objek secara utuh.
- Layanan diidentifikasi dari perilaku spesifik yang dapat menunjukkan peran dan tanggung jawab suatu objek.
- Abaikan atribut yang tidak tepat karena:
 - berupa objek
 - berupa qualifier
 - berupa nama
 - berupa identifier pada implementasi
 - menyatakan status internal objek
 - merupakan atribut yang sangat kecil (minor)
 - bertentangan dengan atribut lain

Definisi Struktur dan Hirarki

- Mendefinisikan struktur dan hirarki dari objek yang akan mengorganisasikan kelas objek.
- Mengatur dan menyederhanakan objek-objek menjadi kelas-kelas objek melalui konsep agregasi dan pewarisan.
- Struktur dan hirarki yang mungkin didefinisikan:
 - Struktur “*generalization-specification*”



- Struktur “*whole-part*” atau “*a-part-of*”



Buat Model Hubungan Objek

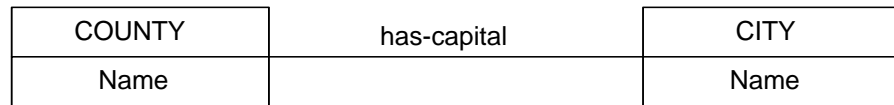
- Mendefinisikan hubungan (asosiasi atau koneksi) antar kelas, yaitu ketergantungan antar satu kelas atau lebih dengan kelas lainnya.

- Asosiasi dapat berbentuk:
 - lokasi fisik atau penempatan (next, to, part, of contained in)
 - aksi terarah (drive)
 - komunikasi (transmit to, acquires from)
 - kepemilikan (incorporated by, is composed of)
 - pemenuhan kondisi (*manages, coordinates, controls*)

- Jenis-jenis asosiasi:

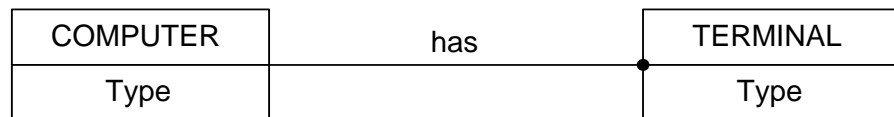
- Asosiasi 1 – 1 (*one-to-one association*)

Contoh:



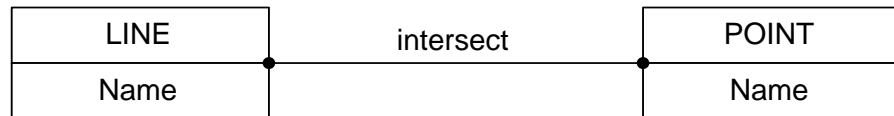
- Asosiasi 1 – n (*one-to-many association*)

Contoh:



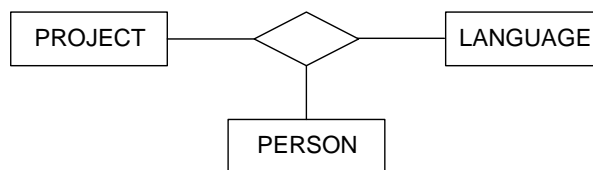
- Asosiasi n – n (*many-to-many association*)

Contoh:



- Ternary Association

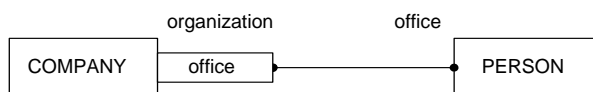
Contoh:



- Kualifikasi

Hubungan asosiatif berkualifikasi antara 2 kelas objek.

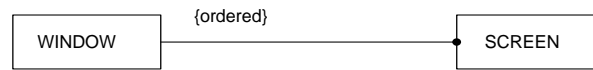
Contoh:



- Ordering

Hubungan berdasarkan urutan kejadian.

Contoh:



- Nama hubungan dan garis atau anak panah digunakan untuk menyatakan hubungan antar kelas-kelas tersebut.
- Abaikan asosiasi yang tidak tepat karena:
 - asosiasi antara kelas yang diabaikan
 - asosiasi implementatif atau tidak relevan
 - asosiasi yang berupa aksi
 - asosiasi ternary
 - asosiasi turunan

Buat Model Perilaku Objek

- Menyatakan bagaimana sistem berorientasi objek akan menanggapi kejadian atau stimuli eksternal (memunculkan sifat dinamis objek).
- Tahap-tahap untuk membuat model perilaku objek:
 - evaluasi semua “use-case” untuk memahami urutan interaksi yang ada dalam sistem
 - identifikasi kejadian yang menggerakkan urutan interaksi, dan pahami bagaimana kejadian-kejadian tersebut berhubungan dengan objek tertentu
 - buat penelusuran kejadian untuk setiap “use-case”
 - buat diagram transisi keadaan untuk sistem
 - tinjau ulang model perilaku objek untuk verifikasi keakuratan dan konsistensi

Contoh Hasil Analisis Berorientasi Objek dengan Metode Coad-Yourdon

- Identifikasi Objek, Atribut, dan *Service*

KOLEKSI
Id_Koleksi Judul Tahun Tipe Status Kelas Subkelas
Didaftar Dipinjam Dikembalikan Dipesan Dihapus

BUKU
Pengarang Edisi

TA/THESIS
Pembuat Edisi

JURNAL
Bulan Penerbit Volume

PELAYANAN
Tanggal Id_Koleksi Tipe Id_Anggota
Create

PEMINJAMAN
Tanggal_Kembali
Simpan_Peminjaman Update_Peminjaman

PEMESANAN
Tanggal_Pengambilan
Simpan_pemesanan

PENCARIAN
Status_Pencarian

ANGGOTA
Nama Tanggal_Daftar
DaftarAnggota_Baru Hapus_Anggota

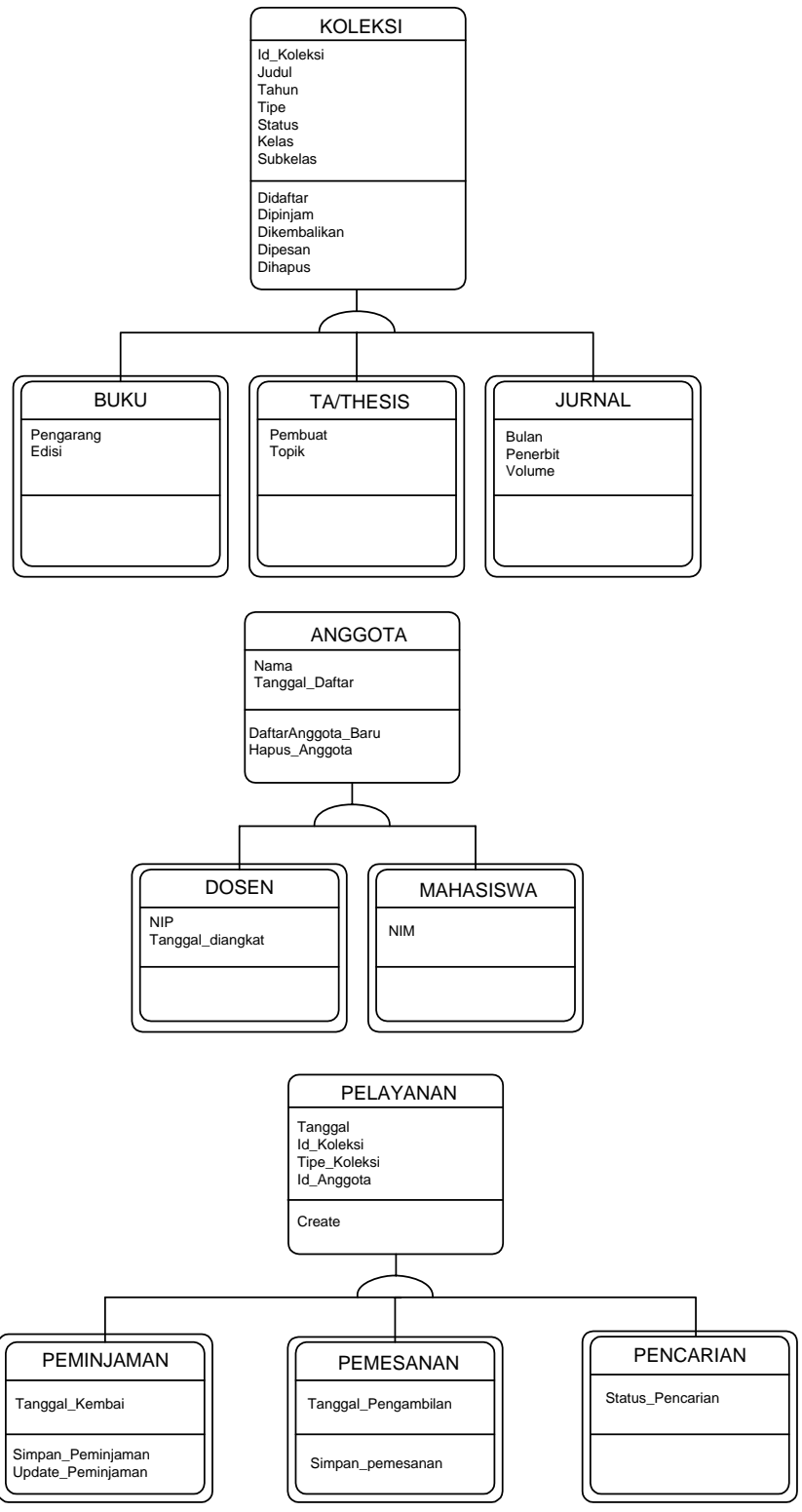
DOSEN
NIP Tanggal_diangkat

MAHASISWA
NIM

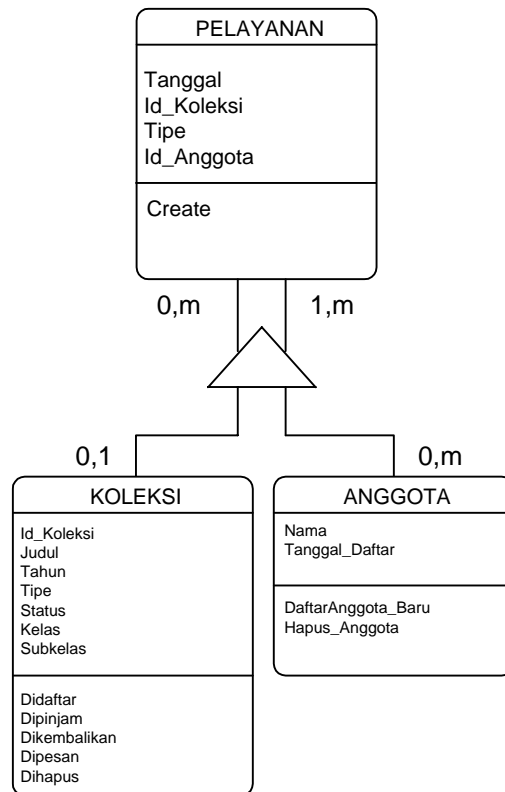
REGISTRASI
Tanggal
Pendaftaran Penghapusan

PEMINJAMAN
Id_Rak Kapasitas Status
Didaftar Diisi Dihapus

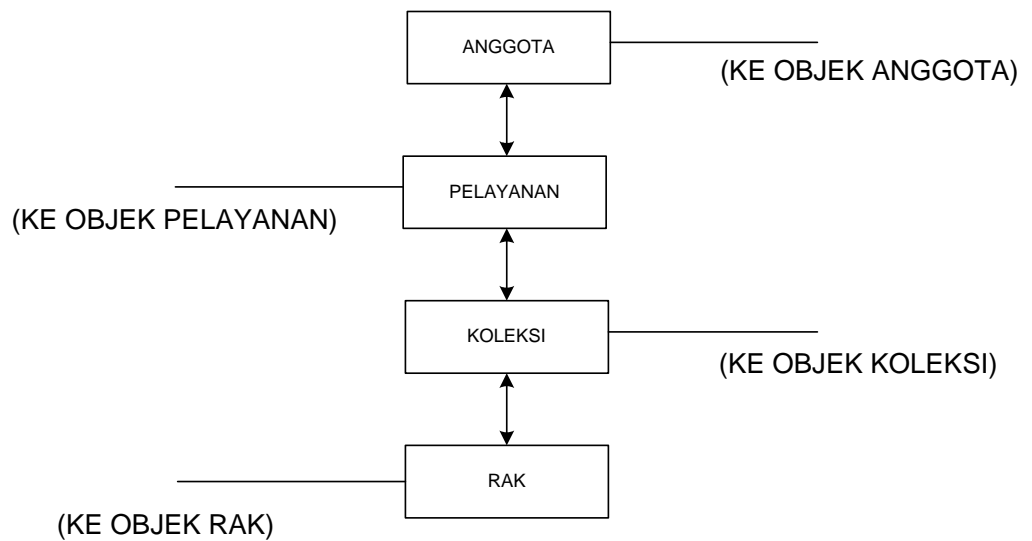
- **Identifikasi Struktur**
Struktur Generalisasi – Spesialisasi



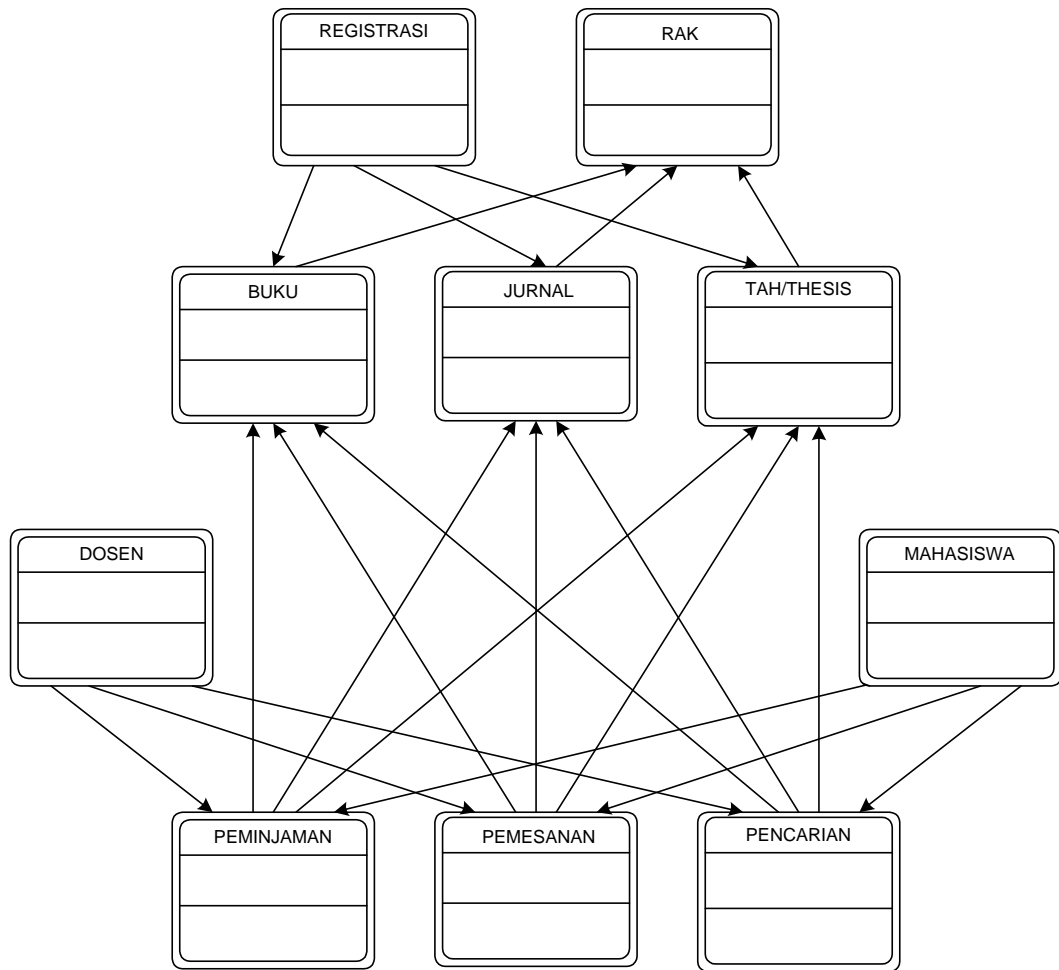
Struktur *Whole – Part*



- **Identifikasi Subjek**



- Pengiriman Message Antar Objek



BAB IV

PERANCANGAN BERORIENTASI OBJEK

Pengertian Dasar

Perancangan

- Proses untuk merencanakan atau mengatur segala sesuatu menurut tahapan tertentu, sebelum bertindak, mengerjakan, atau melakukan sesuatu tersebut [KBB1].
- Proses untuk mengaplikasikan berbagai macam teknik dan prinsip untuk tujuan pendefinisian secara rinci suatu perangkat, proses atau sistem agar dapat direalisasikan dalam suatu bentuk fisik [PRE97].
- Membuat solusi logika atau cara bagaimana kebutuhan-kebutuhan yang ada dipenuhi (diselesaikan) oleh sistem [LAR98].
- Pendefinisian arsitektur perangkat lunak, komponen, modul, antarmuka, pendekatan pengujian, dan data untuk memenuhi kebutuhan-kebutuhan yang sudah ditentukan sebelumnya [IEEE].

Perancangan Berorientasi Objek

- Proses untuk menerjemahkan model analisis hasil OOA menjadi model implementasi yang spesifik yang dapat direalisasi menjadi sebuah perangkat lunak [PRE97].
- Spesifikasi dari solusi perangkat lunak secara logika dalam kerangka objek-objek perangkat lunak, seperti kelas, atribut, metode dan hubungan antar kelas [LAR98].
- Proses pembangunan arsitektur sistem melalui konsep berorientasi objek [MEY97].

Tujuan Perancangan

- Secara umum, tujuan perancangan adalah menghasilkan suatu model atau penggambaran dari suatu entitas yang akan dibangun kemudian.
- Dalam konteks perancangan berorientasi objek (OOD), tujuan perancangan adalah menurunkan objek-objek dari setiap kelas dan bagaimana mengimplementasikan hubungan, perilaku dan komunikasi antar objek-objek tersebut [PRE97].

Proses Perancangan

Merupakan proses kreatif dalam pembangunan perangkat lunak untuk memecahkan suatu persoalan. Model dari proses perancangan secara garis besar terdiri dari empat tahap proses:

- Mengemukakan suatu solusi
- Membangun model dari solusi tersebut
- Evaluasi model terhadap spesifikasi kebutuhan yang telah ada
- Menjabarkan rincian spesifikasi dari solusi tersebut

Perancangan yang baik:

- Melaksanakan semua kebutuhan dan persyaratan yang tercantum pada dokumen SRS.
- Merupakan acuan yang dapat dibaca, dimengerti oleh pembuat program dan penguji perangkat lunak.
- Menyediakan gambaran lengkap dari perangkat lunak mencakup data, fungsi, dan tanggapan, dalam perspektif pelaksanaan pembuatan perangkat lunak.
- Menghasilkan model atau representasi dari perangkat lunak untuk digunakan dalam proses implementasi atau *coding*.

Tahap Perancangan

Dari sudut pandang manajemen proyek, perancangan terdiri dari dua bagian, yaitu:

- Perancangan awal (*preliminary design*)
Menentukan arsitektur perangkat lunak secara keseluruhan (*preliminary design*).
 - Bagaimanakah lingkungan programnya?
 - Bagaimana bentuk penyimpanan datanya?
 - Bagaimana bentuk antarmukanya?
- Perancangan rinci (*detailed design*)
Menentukan modul program (prosedural) yang harus dibuat

Adapun dari sudut pandang teknis, kegiatan perancangan terdiri dari aktivitas:

- Perancangan arsitektural program
 - arsitektural logika
 - arsitektural fisik
- Perancangan modul program (prosedural)
- Perancangan data
 - struktur data internal
 - struktur data fisik
- Perancangan antarmuka
 - Perancangan antarmuka antar modul-modul
 - Perancangan antarmuka antar perangkat lunak dengan non-humanity (external entity)
 - Perancangan antarmuka pemakai

Metode Perancangan Berorientasi Objek

Pengertian

- Cara kerja yang sistematis untuk mengerjakan tahap perancangan berdasarkan pendekatan objek.
- Seperti halnya analisis, perancangan berorientasi objek mempunyai urutan-urutan aktivitas, teknik, dan alat bantu (tools) tertentu untuk memodelkan hasil dari setiap aktivitasnya.
- Beberapa metode yang dapat digunakan untuk melakukan perancangan berorientasi objek diantaranya adalah sebagai berikut.

Metode Coad & Yourdon

- *Problem domain component*
- *Human interaction component*
- *Task management component*
- *Data Management component*

Metode Rumbaugh

- *Perform design system*
- *Conduct object design*
- *Implement control mechanisms defined in system design*
- *Adjust class structure to strengthen inheritance*
- *Design messaging to implement the object relationship (associations)*
- *Package classes and associations into modules*

Metode Jacobson

- *Consider adaptations to make the idealized analysis model fit the real world environment*
- *Create blocks as the primary design object*
- *Create an interaction diagram shows how stimuli are passed between blocks*
- *Organize blocks into subsystems*
- *Review the design work*

Metode Booch

- *Architectural planning*
- *Tactical design*
- *Release planning*

Metode Perancangan Secara Umum

- Pada prinsipnya semua metode perancangan berorientasi objek adalah sama.
- Tahap pelaksanaan perancangan berorientasi objek secara umum:
 - Perbaiki dan lengkapi model objek hasil analisis
 - Perancangan objek
 - Rancang setiap operasi pada level prosedural
 - Definisikan kelas-kelas internal
 - Rancang struktur data internal untuk setiap atribut kelas
 - Rancang model pesan berdasarkan kerjasama (kolaborasi) dan hubungan antar objek
 - Rancang antarmuka pemakai
 - Kaji ulang model perancangan dan ulangi sesuai kebutuhan.