BERORIENTASI PADA OBJEK

ANALISIS DAN DESAIN

Pemodelan Objek—Diagram Kelas

TUJUAN PEMBELAJARAN

Setelah mempelajari bagian ini, Anda seharusnya dapat:

- Definisikan secara ringkas masing-masing istilah pemodelan data utama berikut: objek, status, perilaku, kelas objek, diagram kelas, operasi, enkapsulasi, peran asosiasi, kelas abstrak, polimorfisme, agregasi, dan komposisi.
- Gambarkan diagram kelas untuk menggambarkan situasi bisnis umum.
- Jelaskan kemampuan unik diagram kelas dibandingkan dengan diagram ER untuk pemodelan data.

PERKENALAN

Di bagian ini, kami menunjukkan cara mengembangkan diagram kelas, notasi pemodelan data berorientasi objek. Kami menjelaskan konsep dan teknik utama yang terlibat dalam pemodelan objek, termasuk objek dan kelas, enkapsulasi atribut dan operasi, hubungan agregasi, polimorfisme, dan pewarisan. Kami menunjukkan cara mengembangkan diagram kelas, menggunakan notasi UML, untuk memberikan pandangan konseptual tentang sistem yang sedang dimodelkan. Untuk cakupan pemodelan objek yang lebih menyeluruh, lihat George et al. (2007).

MEWAKILI OBJEK DAN KELAS

Dengan pendekatan berorientasi objek, kita memodelkan dunia dalam bentuk objek. Sebelum menerapkan pendekatan ini pada masalah dunia nyata, kita perlu memahami apa itu objek sebenarnya. Mirip dengan instance entitas, **obyek**memiliki peran yang terdefinisi dengan baik dalam domain aplikasi, dan memiliki karakteristik status (data), perilaku, dan identitas. Objek adalah satu kemunculan dari suatu kelas, yang kami definisikan di bawah ini.

Suatu objek memiliki status dan menunjukkan perilaku melalui operasi yang dapat memeriksa atau memengaruhi statusnya.**negara**dari suatu objek mencakup propertinya (atribut dan hubungan) dan nilai-nilai properti tersebut.**perilaku**menggambarkan bagaimana sebuah objek bertindak dan bereaksi (Booch, 1994). Keadaan sebuah objek ditentukan oleh nilai atributnya dan hubungannya dengan objek lain. Perilaku sebuah objek bergantung pada keadaannya dan operasi yang dilakukan. Operasi hanyalah tindakan yang dilakukan satu objek terhadap objek lain untuk mendapatkan respons. Anda dapat menganggap operasi sebagai layanan yang disediakan oleh sebuah objek (pemasok) kepada kliennya. Klien mengirimkan

Obyek
Suatu entitas yang memiliki peran yang terdefinisi dengan baik dalam domain aplikasi, dan memiliki status, perilaku, dan identitas karakteristik.

Negara
Meliputi properti suatu objek (atribut dan hubungan) dan nilai properti tersebut.

Perilaku
Menggambarkan bagaimana suatu objek bertindak dan bereaksi.

Kelas objek
Disebut juga kelas. Pengelompokan logis dari objek yang memiliki kesamaan (atau serupa) atribut, hubungan, dan perilaku.

Diagram kelas
Menunjukkan struktur statis dari model berorientasi objek, kelas objek, struktur internalnya, dan hubungan di mana mereka berpartisipasi.

Operasi
Suatu fungsi atau layanan yang disediakan oleh semua instansi suatu kelas.

pesan kepada pemasok, yang memberikan layanan yang diinginkan dengan menjalankan operasi terkait.

Pertimbangkan contoh seorang siswa, Mary Jones, yang direpresentasikan sebagai sebuah objek. Keadaan objek ini dicirikan oleh atribut-atributnya, misalnya, nama, tanggal lahir, tahun, alamat, dan telepon, serta nilai-nilai yang dimiliki atribut-atribut ini saat ini; misalnya, nama adalah "Mary Jones," tahun adalah "junior," dan seterusnya. Perilakunya diekspresikan melalui operasi seperti calc-gpa, yang digunakan untuk menghitung nilai rata-rata siswa saat ini. Oleh karena itu, objek Mary Jones mengemas keadaan dan perilakunya secara bersamaan.

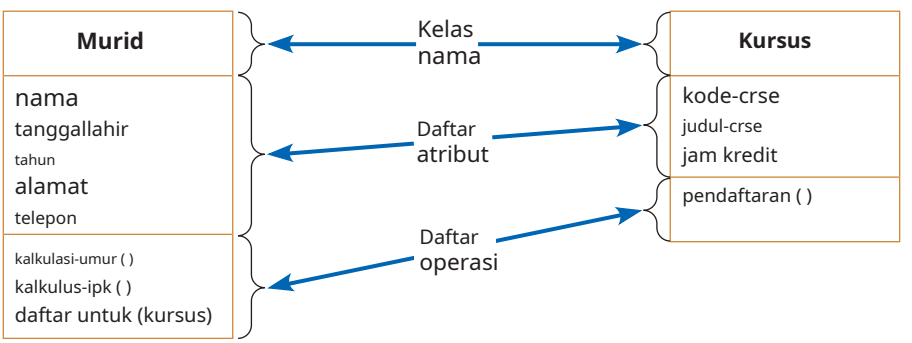
Semua objek memiliki identitas; artinya, tidak ada dua objek yang sama. Misalnya, jika ada dua contoh Siswa dengan nama dan tanggal lahir yang sama (atau bahkan semua atribut), keduanya pada dasarnya adalah dua objek yang berbeda. Sebuah objek mempertahankan identitasnya sendiri selama masa hidupnya. Misalnya, jika Mary Jones menikah dan mengubah nama, alamat, dan teleponnya, ia akan tetap diwakili oleh objek yang sama. Konsep identitas inheren ini berbeda dari konsep pengenal yang kita lihat sebelumnya untuk pemodelan ER.

Kami menggunakan istilah**kelas objek**(atau cukup kelas) untuk merujuk pada pengelompokan logis objek yang memiliki atribut, hubungan, dan perilaku (metode) yang sama (atau serupa) (seperti yang kita gunakan pada tipe entitas dan instans entitas sebelumnya dalam bab ini). Oleh karena itu, dalam contoh kita, Mary Jones adalah instans objek, sedangkan Student adalah kelas objek (seperti Student adalah tipe entitas dalam diagram ER).

Kelas dapat digambarkan secara grafis dalam diagram kelas, seperti yang ditunjukkan pada Gambar 8-26.**diagram kelas**menunjukkan struktur statis dari model berorientasi objek: kelas objek, struktur internalnya, dan hubungan yang terlibat di dalamnya. Dalam UML, kelas direpresentasikan oleh persegi panjang dengan tiga kompartemen yang dipisahkan oleh garis horizontal. Nama kelas muncul di kompartemen atas, daftar atribut di kompartemen tengah, dan daftar operasi di kompartemen bawah persegi panjang. Diagram pada Gambar 8-26 menunjukkan dua kelas, Student dan Course, beserta atribut dan operasinya.

Kelas Student adalah sekelompok objek Student yang memiliki struktur dan perilaku yang sama. Setiap objek mengetahui kelas tempat objek tersebut berada; misalnya, objek Mary Jones mengetahui bahwa objek tersebut termasuk dalam kelas Student. Objek yang termasuk dalam kelas yang sama juga dapat berpartisipasi dalam hubungan yang serupa dengan objek lain; misalnya, semua siswa mendaftar untuk mata kuliah dan, oleh karena itu, kelas Student dapat berpartisipasi dalam hubungan yang disebut "registers-for" dengan kelas lain yang disebut Course (lihat bagian Representing Associations).

Sebuah**operasi**, seperti calc-gpa di Student (lihat Gambar 8-26), adalah fungsi atau layanan yang disediakan oleh semua instans kelas untuk memanggil perilaku dalam suatu objek dengan meneruskan pesan. Hanya melalui operasi seperti itu objek lain dapat mengakses atau memanipulasi informasi yang disimpan dalam suatu objek. Oleh karena itu, operasi tersebut menyediakan antarmuka eksternal ke suatu kelas; antarmuka menyajikan tampilan luar kelas tanpa menunjukkan struktur internalnya atau bagaimana operasinya diimplementasikan. Teknik menyembunyikan detail implementasi internal suatu objek dari



GAMBAR 8-26
Diagram kelas UML yang menunjukkan dua kelas

pandangan luar dikenal sebagai**enkapsulasi**, atau penyembunyian informasi (Booch, 1994; Rumbaugh et al., 1991). Jadi, sementara kita menyediakan abstraksi perilaku yang umum untuk semua contoh kelas dalam antarmukanya, kita merangkum dalam kelas strukturnya dan rahasia perilaku yang diinginkan.

Enkapsulasi
Teknik menyembunyikan detail implementasi internal suatu objek dari pandangan luarnya.

JENIS OPERASI

Operasi dapat diklasifikasikan menjadi tiga jenis, tergantung pada jenis layanan yang diminta oleh klien: (1) konstruktor, (2) kueri, dan (3) pembaruan (*Panduan Notasi UML,* 1997). Sebuah **operasi konstruktor**menciptakan contoh baru dari suatu kelas. Misalnya, Anda dapat memiliki operasi bernama create-student di dalam Student yang menciptakan siswa baru dan menginisialisasi statusnya. Operasi konstruktor tersebut tersedia untuk semua kelas dan, oleh karena itu, tidak ditampilkan secara eksplisit dalam diagram kelas.

Operasi konstruktor
Suatu operasi yang menciptakan contoh baru suatu kelas.

A**operasi kueri**adalah operasi yang tidak memiliki efek samping apa pun; operasi ini mengakses status objek tetapi tidak mengubah status tersebut (Fowler, 2000; Rumbaugh et al., 1991). Misalnya, kelas Student dapat memiliki operasi yang disebut get-year (tidak ditampilkan) yang hanya mengambil tahun (mahasiswa baru, mahasiswa tahun kedua, mahasiswa tahun ketiga, atau mahasiswa tahun keempat) dari objek Student yang ditentukan dalam kueri. Perhatikan bahwa kueri seperti get-year tidak perlu ditampilkan secara eksplisit dalam diagram kelas karena kueri tersebut mengambil nilai dari atribut dasar yang independen. Namun, pertimbangkan operasi calc-age dalam Student. Ini juga merupakan operasi kueri karena tidak memiliki efek samping apa pun. Perhatikan bahwa satu-satunya argumen untuk kueri ini adalah objek Student yang menjadi target. Kueri tersebut dapat direpresentasikan sebagai atribut turunan (Rumbaugh et al., 1991); misalnya, kita dapat merepresentasikan "age" sebagai atribut turunan dari Student. Karena objek target selalu merupakan argumen implisit dari suatu operasi, maka tidak perlu ditampilkan secara eksplisit dalam deklarasi operasi.

Operasi kueri
Suatu operasi yang mengakses keadaan suatu objek tetapi tidak mengubah keadaan tersebut.

Sebuah**operasi pembaruan**memiliki efek samping; mengubah status objek. Misalnya, perhatikan operasi Student yang disebut promote-student (tidak ditampilkan). Operasi tersebut mempromosikan siswa ke tahun ajaran baru, misalnya, dari junior ke senior, sehingga mengubah status objek Student (nilai atribut tahun). Contoh lain dari operasi pembaruan adalah register-for(course), yang, ketika dipanggil, memiliki efek membuat koneksi dari objek Student ke objek Course tertentu. Perhatikan bahwa, selain memiliki objek Student target sebagai argumen implisit, operasi tersebut memiliki argumen eksplisit yang disebut "course" yang menentukan kursus yang ingin diikuti siswa. Argumen eksplisit ditampilkan dalam tanda kurung.

Operasi pembaruan
Suatu operasi yang mengubah keadaan suatu objek.

A**operasi lingkup kelas**adalah operasi yang berlaku untuk kelas, bukan untuk instans objek. Misalnya, avg-gpa untuk kelas Student (tidak ditampilkan dengan operasi lain untuk kelas ini pada Gambar 8-26) menghitung rata-rata gpa untuk semua siswa (nama operasi digarisbawahi untuk menunjukkan bahwa itu adalah operasi cakupan).

Operasi lingkup kelas
Suatu operasi yang diterapkan pada suatu kelas dan bukan pada suatu instansi objek.

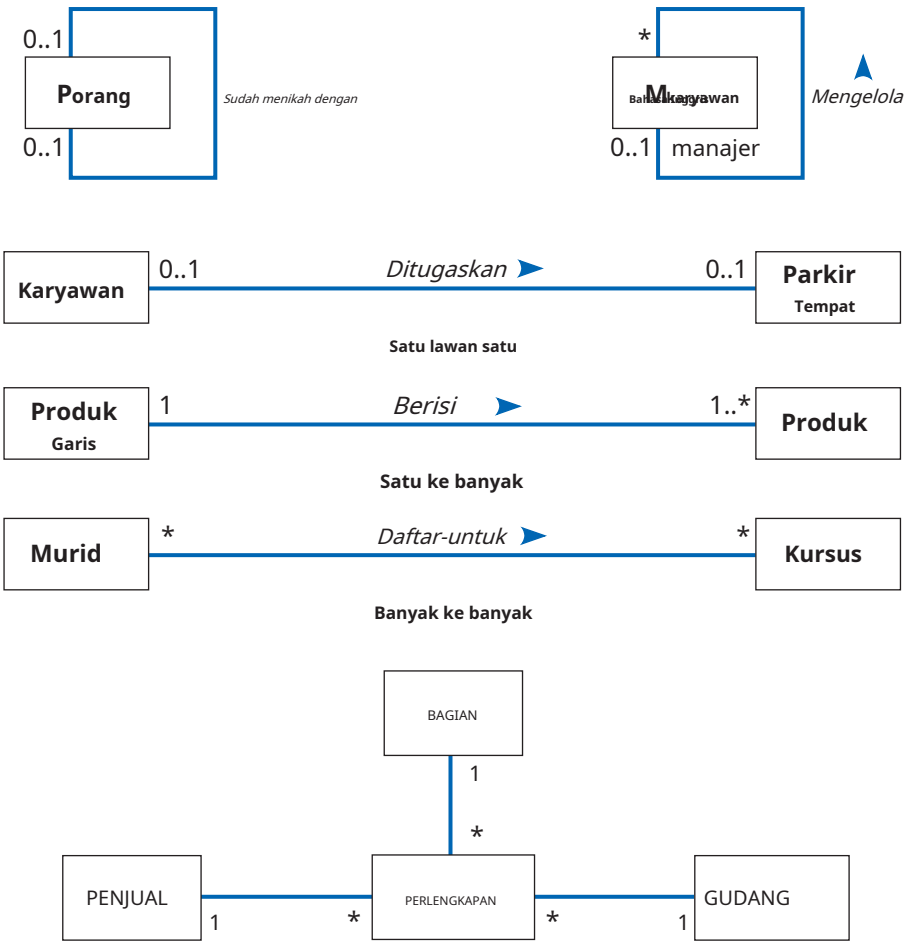
MEWAKILI ASOSIASI

Sejalan dengan definisi hubungan untuk model ER,**asosiasi**adalah hubungan antara contoh kelas objek. Seperti dalam model ER, tingkat hubungan asosiasi bisa satu (unary), dua (binary), tiga (ternary), atau lebih tinggi (*N*-ary). Pada Gambar 8-27, kami mengilustrasikan bagaimana model berorientasi objek dapat digunakan untuk merepresentasikan hubungan asosiasi dengan derajat yang berbeda. Sebuah asosiasi ditunjukkan sebagai garis padat antara kelas yang berpartisipasi. Nama yang diberikan pada akhir asosiasi tempat asosiasi tersebut terhubung ke kelas disebut **peran asosiasi**(*Panduan Notasi UML,* 1997). Setiap asosiasi memiliki dua atau lebih peran. Peran dapat diberi nama secara eksplisit dengan label di dekat akhir asosiasi (lihat peran "manajer" pada Gambar 8-27). Nama peran menunjukkan peran yang dimainkan oleh kelas yang melekat pada akhir di dekat tempat nama tersebut muncul. Penggunaan nama peran bersifat opsional. Anda dapat menentukan nama peran sebagai pengganti atau sebagai tambahan pada nama asosiasi. Anda dapat menunjukkan arah asosiasi secara eksplisit dengan menggunakan segitiga padat di samping nama asosiasi.

Asosiasi
Hubungan bernama antara atau di antara kelas objek.

Peran asosiasi
Akhir dari suatu asosiasi yang menghubungkannya dengan suatu kelas.

GAMBAR 8-27
Contoh hubungan asosiasi dengan derajat yang berbeda



Gambar 8-27 menunjukkan dua hubungan unary, Is-married-to dan Manages. Di salah satu ujung hubungan Manages, kami telah menamai peran sebagai "manager," yang menyiratkan bahwa seorang karyawan dapat memainkan peran sebagai manajer. Kami belum menamai peran lainnya, tetapi kami telah menamai asosiasinya. Ketika nama peran tidak muncul, Anda mungkin menganggap nama peran sebagai nama kelas yang melekat pada ujung tersebut (Fowler, 2000). Misalnya, Anda dapat menyebut peran untuk ujung kanan hubungan Is-assigned pada Gambar 8-27 "Parking Place."

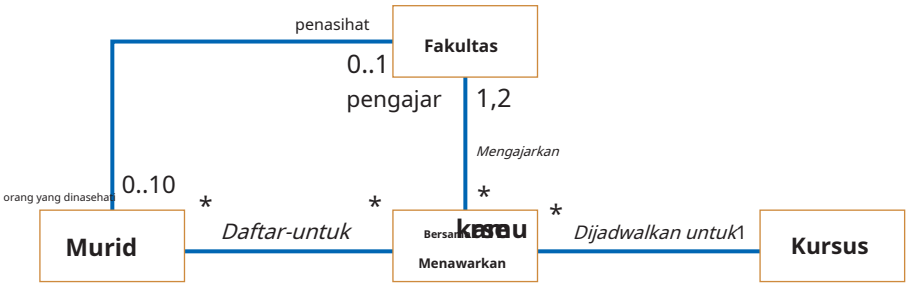
Keserbaragaman
Spesifikasi yang menunjukkan berapa banyak objek yang berpartisipasi dalam hubungan tertentu.

Setiap peran memiliki**keserbaragaman**, yang menunjukkan berapa banyak objek yang berpartisipasi dalam hubungan asosiasi tertentu. Misalnya, multiplisitas 2..5 menunjukkan bahwa minimal dua dan maksimal lima objek dapat berpartisipasi dalam hubungan tertentu. Oleh karena itu, multiplisitas tidak lain hanyalah kendala kardinalitas, yang Anda lihat dalam diagram ER. Selain nilai integer, batas atas multiplisitas dapat berupa * (tanda bintang), yang menunjukkan batas atas tak terhingga. Jika satu nilai integer ditentukan, artinya rentang hanya mencakup nilai tersebut.

Multiplisitas untuk kedua peran dalam hubungan Is-married-to pada Gambar 8-27 adalah 0..1, yang menunjukkan bahwa seseorang mungkin lajang atau menikah dengan satu orang. Multiplisitas untuk peran manajer dalam hubungan Manages adalah 0..1 dan untuk peran lainnya adalah 0..*, yang menyiratkan bahwa seorang karyawan dapat dikelola hanya oleh satu manajer, tetapi seorang manajer dapat mengelola nol hingga banyak karyawan.

Pada Gambar 8-27, kami menunjukkan hubungan terner yang disebut Persediaan antara Vendor, Suku Cadang, dan Gudang. Seperti yang lebih disukai dalam diagram ER, kami merepresentasikan hubungan terner menggunakan kelas dan menempatkan nama hubungan di sana.

Diagram kelas pada Gambar 8-28 menunjukkan beberapa asosiasi biner. Diagram menunjukkan bahwa seorang mahasiswa mungkin memiliki seorang penasihat, sementara seorang anggota fakultas dapat memberikan nasihat hingga



GAMBAR 8-28
Contoh asosiasi biner

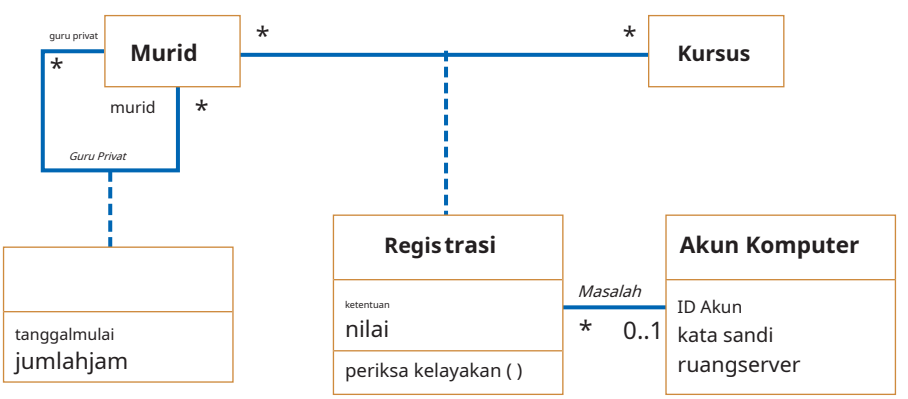
maksimum sepuluh siswa. Selain itu, meskipun suatu kursus mungkin memiliki beberapa penawaran, penawaran kursus tertentu dijadwalkan untuk tepat satu kursus. UML memungkinkan Anda untuk menentukan secara numerik setiap kelipatan. Misalnya, diagram menunjukkan bahwa penawaran kursus dapat diajarkan oleh satu atau dua instruktur (1,2). Anda dapat menentukan satu angka (misalnya, 2 untuk anggota tim bridge), rentang (misalnya, 11–14 untuk pemain tim sepak bola yang berpartisipasi dalam permainan tertentu), atau serangkaian angka dan rentang yang terpisah (misalnya, 3, 5, 7 untuk jumlah anggota komite dan 20–32, 35–40 untuk beban kerja dalam jam per minggu karyawan perusahaan.

Gambar 8-28 juga menunjukkan bahwa anggota fakultas berperan sebagai instruktur, serta penasihat. Sementara peran penasihat mengidentifikasi objek Fakultas yang terkait dengan objek Mahasiswa, peran penerima mengidentifikasi kumpulan objek Mahasiswa yang terkait dengan objek Fakultas. Kita dapat menamai asosiasi tersebut, misalnya, Advises, tetapi, dalam kasus ini, nama peran cukup bermakna untuk menyampaikan semantik hubungan tersebut.

MEWAKILI KELAS ASOSIATIF

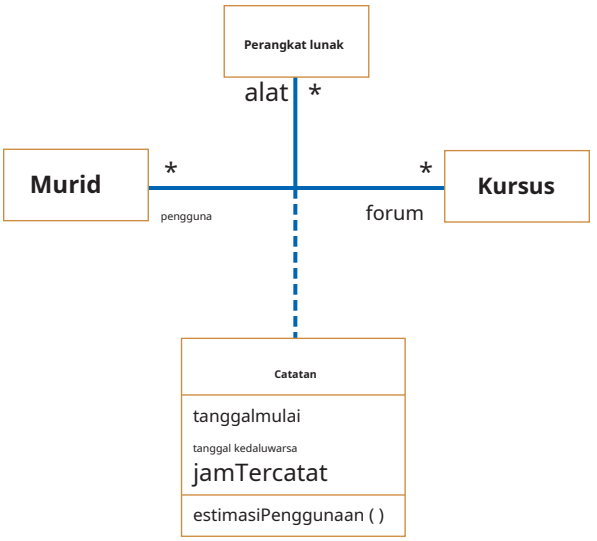
Ketika sebuah asosiasi itu sendiri memiliki atribut atau operasinya sendiri atau ketika ia berpartisipasi dalam hubungan dengan kelas lain, maka akan berguna untuk memodelkan asosiasi tersebut sebagai **kelas asosiatif**(sama seperti kita menggunakan entitas asosiatif dalam diagram ER). Misalnya, dalam Gambar 8-29, atribut term dan grade benar-benar termasuk dalam asosiasi many-to-many antara Student dan Course. Nilai mahasiswa untuk suatu mata kuliah tidak dapat ditentukan kecuali mahasiswa dan mata kuliah tersebut diketahui. Demikian pula, untuk menemukan term(term) di mana mahasiswa mengambil mata kuliah tersebut, baik student maupun mata kuliah tersebut harus diketahui. Operasi check Eligibility, yang menentukan apakah seorang mahasiswa memenuhi syarat untuk mendaftar pada mata kuliah tertentu, juga termasuk dalam asosiasi, bukan salah satu dari dua kelas yang berpartisipasi. Kami juga telah menangkap fakta bahwa, untuk beberapa registrasi mata kuliah, akun komputer diberikan kepada mahasiswa. Atas alasan ini, kami memodelkan Registration sebagai kelas asosiasi, dengan serangkaian fiturnya sendiri dan asosiasi dengan kelas lain (Akun Komputer). Demikian pula, untuk asosiasi Tutor unary,

Kelas asosiatif
Suatu asosiasi yang memiliki atribut atau operasinya sendiri atau yang berpartisipasi dalam hubungan dengan kelas lain.



GAMBAR 8-29
Diagram kelas yang menunjukkan kelas asosiatif

GAMBAR 8-30
Hubungan terner dengan kelas asosiasi



mulai Tanggal dan jumlah Jam (jumlah jam bimbingan) benar-benar milik asosiasi, dan karenanya muncul dalam kelas asosiasi terpisah.

Anda memiliki pilihan untuk menampilkan nama kelas asosiasi pada jalur asosiasi, pada simbol kelas, atau keduanya. Ketika asosiasi hanya memiliki atribut tetapi tidak memiliki operasi atau tidak berpartisipasi dalam asosiasi lain, pilihan yang disarankan adalah menampilkan nama pada jalur asosiasi, tetapi menghilangkannya dari simbol kelas asosiasi, untuk menekankan "sifat asosiasinya" (*Panduan Notasi UML,* 1997). Itulah cara kami menampilkan asosiasi Tutor. Di sisi lain, kami telah menampilkan nama asosiasi Registrasi—yang memiliki dua atribut dan satu operasinya sendiri, serta asosiasi yang disebut Masalah dengan Akun Komputer—di dalam persegi panjang kelas untuk menekankan "sifat kelasnya."

Gambar 8-30 menunjukkan hubungan terner antara kelas Siswa, Perangkat Lunak, dan Kursus. Hubungan ini menangkap fakta bahwa siswa menggunakan berbagai perangkat lunak untuk kursus yang berbeda. Misalnya, kita dapat menyimpan informasi bahwa Mary Jones menggunakan Microsoft Access dan Oracle untuk kursus Manajemen Basis Data, Rational Rose dan Visual C++ untuk kursus Pemodelan Berorientasi Objek, dan Level5 Object untuk Kursus Sistem Pakar. Sekarang anggaplah kita ingin memperkirakan jumlah jam per minggu yang akan dihabiskan Mary menggunakan Oracle untuk kursus Manajemen Basis Data. Proses ini benar-benar milik asosiasi terner, dan bukan milik salah satu kelas individual. Oleh karena itu, kita telah membuat kelas asosiatif yang disebut Log, yang di dalamnya kita telah mendeklarasikan operasi yang disebut estimasi Penggunaan. Selain operasi ini, kita telah menetapkan tiga atribut yang termasuk dalam asosiasi: beginDate, expiryDate, dan hoursLogged. Atau, kelas asosiatif Log dapat ditempatkan di persimpangan garis asosiasi, seperti yang ditunjukkan pada Gambar 8-16; dalam kasus ini, multiplisitas diperlukan pada semua garis di samping kelas Log.

MEWAKILI STEREOTIP UNTUK ATRIBUT

Dalam diagram ER, kami menetapkan atribut sebagai kunci utama, dan kami menetapkannya sebagai multinilai, turunan, dan jenis lainnya. Ini juga dapat dilakukan dalam diagram kelas dengan menempatkan stereotip di sebelah atribut. Stereotip hanya memperluas kosakata UML umum. Misalnya, dalam Gambar 8-31, usia adalah atribut turunan dari Siswa karena dapat dihitung dari tanggal lahir dan tanggal saat ini. Karena perhitungan tersebut merupakan kendala pada kelas objek, perhitungan ditampilkan pada diagram ini dalam kurung kurawal di dekat kelas objek Siswa. Selain itu, crseCode adalah kunci utama untuk kelas Kursus. Properti atribut lainnya dapat ditampilkan dengan cara yang sama.

GAMBAR 8-31
Stereotip

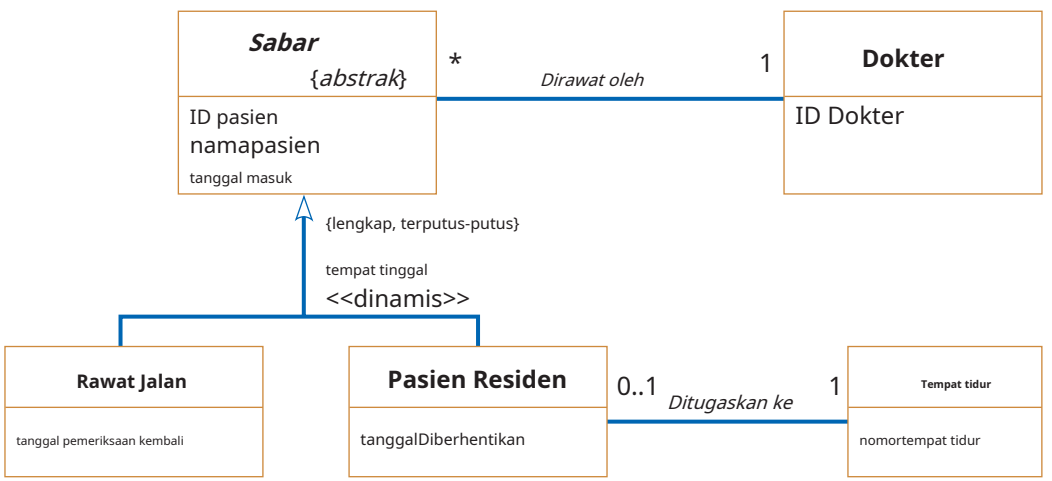
MEWAKILI GENERALISASI

Dalam pendekatan berorientasi objek, Anda dapat mengabstraksikan fitur-fitur umum (atribut dan operasi) di antara beberapa kelas, serta hubungan yang mereka ikuti, ke dalam kelas yang lebih umum, seperti yang kita lihat dengan hubungan supertipe/subtipe dalam diagram ER. Kelas-kelas yang digeneralisasi disebut subkelas, dan kelas yang digeneralisasi disebut superkelas.

Pertimbangkan contoh yang ditunjukkan pada Gambar 8-32, yang merupakan diagram kelas yang setara dengan diagram ER pada Gambar 8-18. Ada dua jenis pasien: pasien rawat jalan dan pasien residen. Fitur yang dimiliki oleh semua pasien—patientId, patientName, dan admitDate—disimpan dalam superkelas Pasien, sedangkan fitur yang khusus untuk jenis pasien tertentu disimpan dalam subkelas yang sesuai (misalnya, checkbackDate dari Rawat Jalan). Jalur generalisasi ditunjukkan sebagai garis padat dari subkelas ke superkelas, dengan panah di ujungnya, dan menunjuk ke arah superkelas. Kami juga menetapkan bahwa generalisasi ini bersifat dinamis, yang berarti bahwa suatu objek dapat mengubah subtipe. Generalisasi ini juga lengkap (tidak ada subkelas lain) dan terpisah (subkelas tidak tumpang tindih). Meskipun tidak menggunakan terminologi yang sama persis, aturan bisnis generalisasi yang sama yang kita lihat dengan diagram ER dapat direpresentasikan.

Anda dapat menunjukkan dasar generalisasi dengan menentukan diskriminator di samping jalur. Diskriminator menunjukkan properti kelas objek mana yang sedang diabstraksikan oleh hubungan generalisasi tertentu. Anda dapat melakukan diskriminasi hanya pada satu properti pada satu waktu. Misalnya, pada Gambar 8-32, kami melakukan diskriminasi kelas Pasien berdasarkan tempat tinggal.

Sebuah contoh dari subkelas juga merupakan contoh dari superkelasnya. Misalnya, pada Gambar 8-32, contoh Rawat Jalan juga merupakan contoh Pasien. Oleh karena itu, generalisasi juga disebut sebagai hubungan Is-a. Selain itu, sebuah subkelas mewarisi semua fitur dari superkelasnya. Misalnya, pada Gambar 8-32, selain fitur-fiturnya sendiri, subkelas juga mewarisi fitur-fitur yang dimilikinya.



GAMBAR 8-32
Contoh generalisasi, pewarisan, dan kendala

fitur khusus—checkbackDate—subkelas Rawat Jalan mewarisi patientId, patientName, admitDate, dan operasi apa pun (tidak ditampilkan) dari Pasien.

Kelas abstrak
Suatu kelas yang tidak mempunyai instansi langsung, tetapi keturunannya mungkin mempunyai instansi langsung.

Kelas beton
Suatu kelas yang dapat memiliki instansi langsung.

Perhatikan bahwa pada Gambar 8-32, kelas Pasien ditulis miring, yang berarti bahwa kelas tersebut adalah kelas abstrak.**kelas abstrak**adalah kelas yang tidak memiliki contoh langsung tetapi keturunannya mungkin memiliki contoh langsung (Booch, 1994; Rumbaugh et al., 1991).*Catatan:* Anda juga bisa menuliskan kata*abstrak*dalam kurung kurawal tepat di bawah nama kelas. Ini khususnya berguna ketika Anda membuat diagram kelas secara manual.) Kelas yang dapat memiliki contoh langsung (misalnya, Pasien Rawat Jalan atau Pasien Residen) disebut**kelas beton**. Oleh karena itu, dalam contoh ini, Pasien Rawat Jalan dan Pasien Residen dapat memiliki instansi langsung, tetapi Pasien tidak dapat memiliki instansi langsungnya sendiri.

Kelas abstrak Pasien berpartisipasi dalam hubungan yang disebut Diobati oleh dengan Dokter, yang menyiratkan bahwa semua pasien—baik pasien rawat jalan maupun pasien rawat inap—dirawat oleh dokter. Selain hubungan yang diwariskan ini, kelas Pasien Residen memiliki hubungan khusus sendiri yang disebut Ditugaskan ke dengan Tempat Tidur, yang menyiratkan bahwa hanya pasien rawat inap yang dapat ditugaskan ke tempat tidur. Jadi, selain menyempurnakan atribut dan operasi kelas, subkelas juga dapat mengkhususkan hubungan tempat ia berpartisipasi.

Pada Gambar 8-32, kata-kata*menyelesaikan*Dan*menguraikan*telah ditempatkan dalam kurung kurawal di sebelah generalisasi. Mereka menunjukkan kendala semantik di antara subkelas (*menyelesaikan*sesuai dengan spesialisasi total dalam notasi Hubungan Entitas yang Diperluas [EER] [lihat Hoffer et al., 2010], sedangkan*tidak lengkap*sesuai dengan spesialisasi parsial). Salah satu kata kunci UML berikut dapat digunakan:

- *Tumpang tindih.*Keturunan dapat diturunkan dari lebih dari satu subkelas (sama dengan aturan tumpang tindih dalam diagram EER).
- *Menguraikan.*Suatu keturunan tidak boleh diturunkan dari lebih dari satu subkelas (sama dengan aturan disjoint dalam diagram EER).
- *Menyelesaikan.*Semua subkelas telah ditentukan (baik ditampilkan maupun tidak). Tidak ada subkelas tambahan yang diharapkan (sama dengan aturan spesialisasi total dalam diagram EER).
- *Tidak lengkap.*Beberapa subkelas telah ditetapkan, tetapi daftarnya diketahui tidak lengkap. Ada beberapa subkelas tambahan yang belum ada dalam model (sama dengan aturan spesialisasi parsial dalam diagram EER).

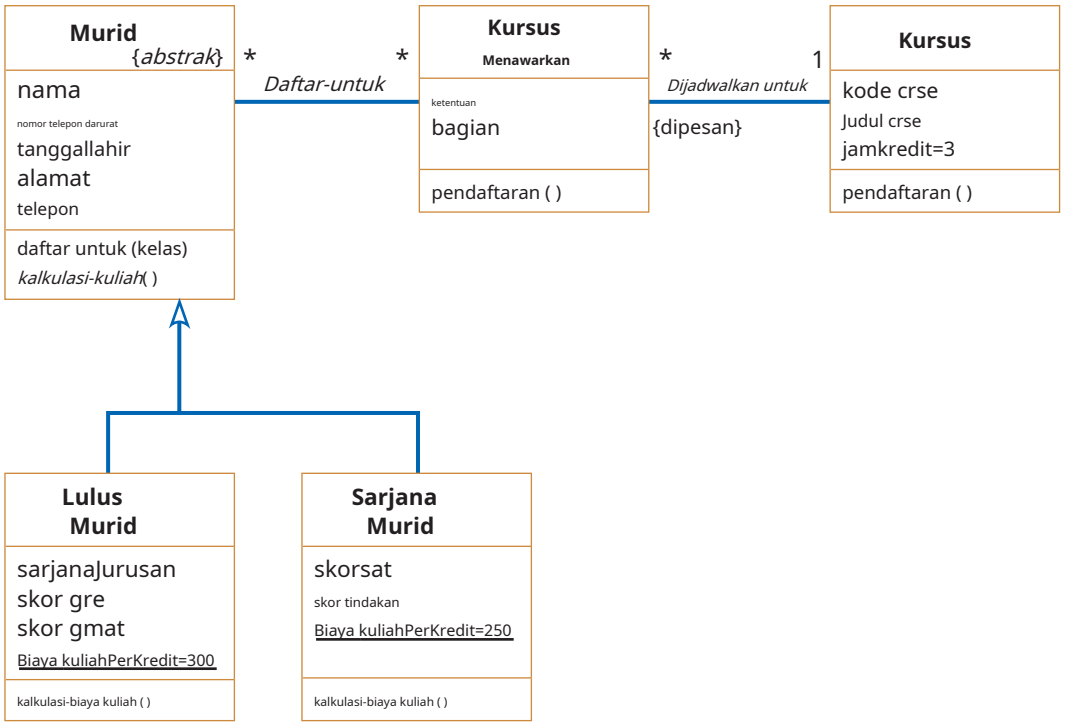
Pada Gambar 8-33, kami merepresentasikan mahasiswa pascasarjana dan sarjana dalam model yang dikembangkan untuk penagihan mahasiswa. Operasi calc-tuition menghitung biaya kuliah yang harus dibayarkan mahasiswa; jumlah ini bergantung pada biaya kuliah per jam kredit (tuitionPerCred), mata kuliah yang diambil, dan jumlah jam kredit (creditHrs) untuk setiap mata kuliah tersebut. Biaya kuliah per jam kredit, pada gilirannya, bergantung pada apakah mahasiswa tersebut adalah mahasiswa pascasarjana atau sarjana. Dalam contoh ini, jumlah tersebut adalah $300 untuk semua mahasiswa pascasarjana dan $250 untuk semua mahasiswa sarjana. Untuk menunjukkan hal ini, kami telah menggarisbawahi atribut tuitionPerCred di masing-masing dari dua subkelas, beserta nilainya. Atribut seperti itu disebut**atribut ruang lingkup kelas**, yang menentukan nilai umum untuk seluruh kelas, dan bukan nilai spesifik untuk satu contoh (Rumbaugh et al., 1991).

Atribut lingkup kelas
Atribut suatu kelas yang menentukan nilai umum untuk seluruh kelas, dan bukan nilai spesifik untuk suatu contoh.

Anda dapat menentukan nilai default awal dari suatu atribut menggunakan tanda sama dengan (=) setelah nama atribut (lihat creditHrs untuk kelas Course pada Gambar 8-33). Perbedaan antara spesifikasi nilai awal dan atribut lingkup kelas adalah bahwa yang pertama memungkinkan kemungkinan nilai atribut yang berbeda untuk instans suatu kelas, dan yang kedua memaksa semua instans untuk berbagi nilai yang sama.

Selain menentukan banyaknya peran asosiasi, Anda juga dapat menentukan properti lain, misalnya, apakah objek yang memainkan peran tersebut diurutkan atau tidak. Pada gambar, kami menempatkan batasan kata kunci “{ordered}” di samping akhir Penawaran Kursus dari hubungan Scheduled-for untuk menunjukkan fakta bahwa penawaran untuk kursus tertentu diurutkan ke dalam daftar—misalnya, menurut istilah dan bagian. Batasan default pada peran adalah “{unordered}.”

Subkelas Mahasiswa Pascasarjana mengkhususkan kelas Mahasiswa abstrak dengan menambahkan empat atribut—undergradMajor, greScore, gmatScore, dan tuitionPerCred—dan dengan



GAMBAR 8-33
Polimorfisme, abstrak operasi, atribut ruang lingkup kelas, dan pemesanan

menyempurnakan operasi kalkulus yang diwariskan. Perhatikan bahwa operasi tersebut ditampilkan dalam huruf miring di dalam kelas Student, yang menunjukkan bahwa itu adalah operasi abstrak.**operasi abstrak**mendefinisikan bentuk atau protokol operasi, tetapi bukan implementasinya. Dalam contoh ini, kelas Student mendefinisikan protokol operasi kalkulus, tanpa menyediakan protokol yang sesuai.**metode**(implementasi aktual dari operasi tersebut). Protokol tersebut mencakup jumlah dan jenis argumen, jenis hasil, dan semantik yang dimaksudkan dari operasi tersebut. Dua subkelas konkret, Mahasiswa Pascasarjana dan Mahasiswa S1, menyediakan implementasinya sendiri dari operasi kalkulasi. Perhatikan bahwa karena kelas-kelas ini konkret, mereka tidak dapat menyimpan operasi abstrak.

Operasi abstrak
Menentukan bentuk atau protokol operasi, tetapi bukan implementasinya.

Metode
Pelaksanaan suatu operasi.

Penting untuk dicatat bahwa, meskipun kelas Mahasiswa Pascasarjana dan Mahasiswa Sarjana memiliki operasi kalkulasi biaya kuliah yang sama, mereka mungkin menerapkan operasi tersebut dengan cara yang sangat berbeda. Misalnya, metode yang menerapkan operasi untuk mahasiswa pascasarjana mungkin menambahkan biaya pascasarjana khusus untuk setiap mata kuliah yang diambil mahasiswa. Fakta bahwa operasi yang sama dapat diterapkan pada dua atau lebih kelas dengan cara yang berbeda dikenal sebagai**polimorfisme**, sebuah konsep kunci dalam sistem berorientasi objek (Booch, 1994; Rumbaugh et al., 1991). Operasi pendaftaran dalam Gambar 8-33 mengilustrasikan contoh lain dari polimorfisme. Sementara operasi pendaftaran dalam Penawaran Kursus menghitung pendaftaran untuk penawaran atau bagian kursus tertentu, operasi dengan nama yang sama dalam Kursus menghitung pendaftaran gabungan untuk semua bagian dari kursus tertentu.

Polimorfisme
Operasi yang sama dapat diterapkan pada dua atau lebih kelas dengan cara yang berbeda.

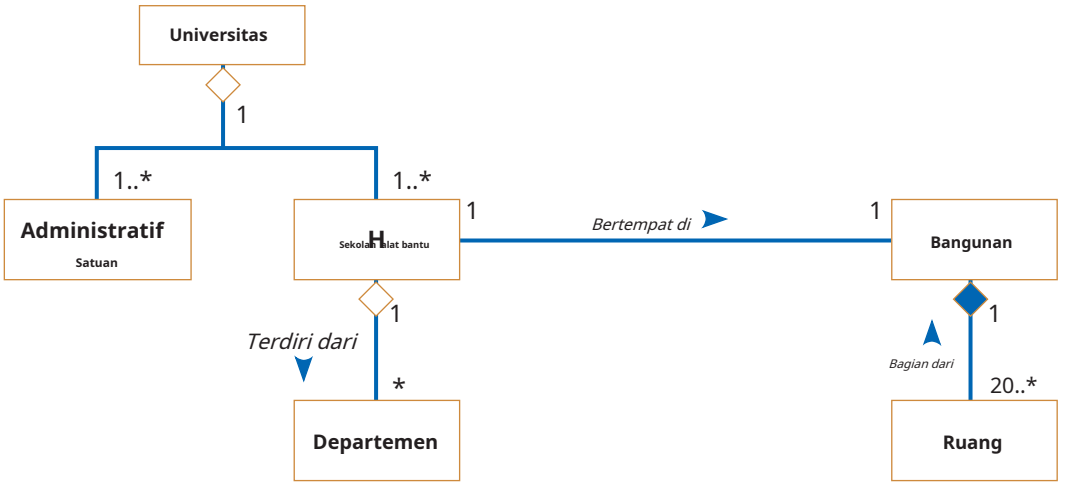
MEWAKILI AGREGASI

Sebuah**pengumpulan**menyatakan hubungan bagian-dari antara objek komponen dan objek agregat. Ini adalah bentuk hubungan asosiasi yang lebih kuat (dengan semantik "bagian-dari" yang ditambahkan) dan direpresentasikan dengan berlian berongga di ujung agregat.

Pengumpulan
Hubungan bagian-bagian antara suatu objek komponen dan suatu objek agregat.

Gambar 8-34 menunjukkan struktur agregasi sebuah universitas. Perhatikan bahwa berlian di salah satu ujung hubungan antara Bangunan dan Ruang berbentuk padat, bukan berongga. Berlian padat mewakili bentuk agregasi yang lebih kuat yang dikenal sebagai

GAMBAR 8-34
Agregasi dan komposisi



Komposisi

Hubungan bagian-bagian di mana bagian-bagian hanya menjadi bagian dari satu objek utuh, dan bagian-bagian tersebut hidup dan mati bersama keseluruhan objek.

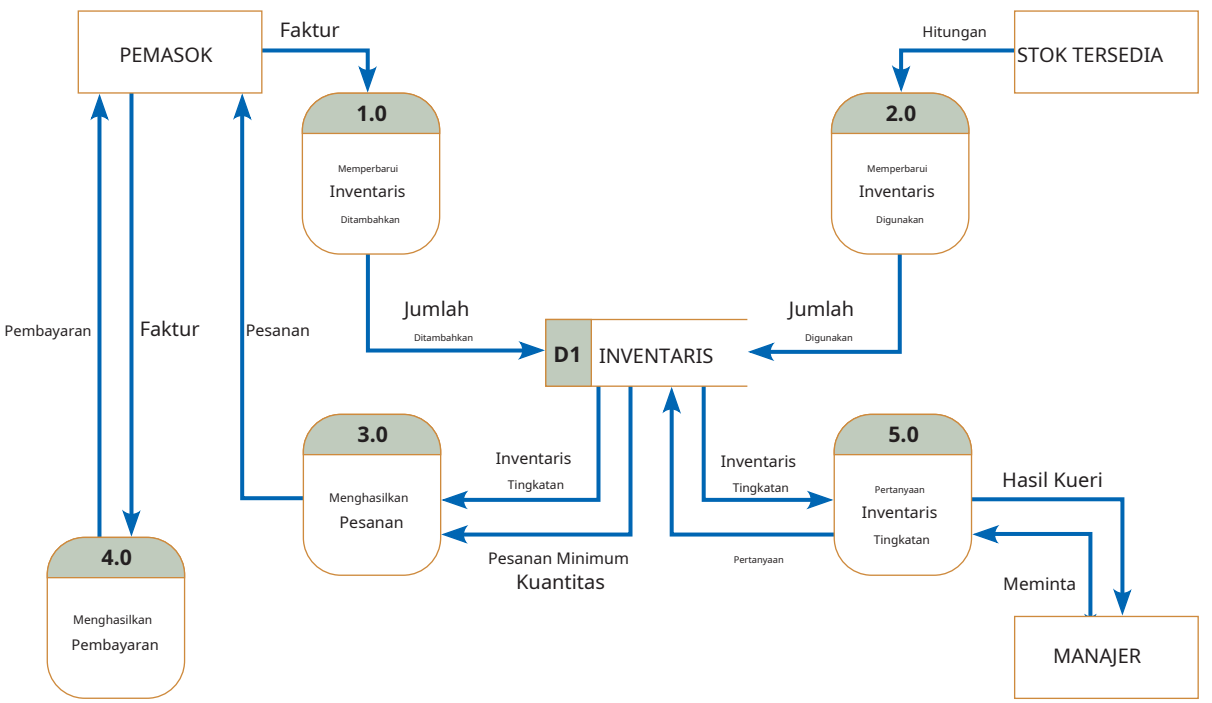
komposisi. Dalam komposisi, bagian objek hanya milik satu objek utuh; misalnya, sebuah ruangan hanya merupakan bagian dari satu bangunan. Oleh karena itu, multiplisitas pada ujung agregat tidak boleh melebihi satu. Bagian dapat dibuat setelah pembuatan keseluruhan objek; misalnya, ruangan dapat ditambahkan ke bangunan yang sudah ada. Namun, setelah bagian dari komposisi dibuat, bagian tersebut hidup dan mati bersama keseluruhan; penghapusan objek agregat akan berlanjut ke komponen-komponennya. Jika sebuah bangunan dihancurkan, misalnya, maka semua ruangannya juga akan dihancurkan. Namun, adalah mungkin untuk menghapus bagian sebelum agregatnya hancur, sama seperti mungkin untuk menghancurkan sebuah ruangan tanpa merobohkan sebuah bangunan.

Lebih suka Hoosier BURGER

CONTOH PEMODELAN DATA KONSEPTUAL DI HOOSIER BURGER

GAMBAR 8-35
Diagram aliran data Level-0 untuk sistem kontrol inventaris logis baru Hoosier Burger

Bab 7 menyusun proses dan persyaratan logika untuk sistem kontrol inventaris baru untuk Hoosier Burger. DFD dan tabel keputusan (diulang di sini sebagai Gambar 8-35 dan 8-36) menjelaskan persyaratan untuk sistem baru ini. Tujuan dari sistem ini adalah untuk



Kondisi/ Langkah Tindakan	Aturan						
	1	2	3	4	5	6	7
Jenis barang	P	P	P	P	P	P	N
Waktu dalam seminggu	D	Kami	D	Kami	D	Kami	–
Musim tahun ini	A	A	S	S	H	H	–
Pesanan harian tetap	X		X		X		
Perintah akhir pekan tetap		X		X		X	
Jumlah pesanan minimum							X
Pengurangan hari libur					X	X	
Pengurangan musim panas			X	X			

GAMBAR 8-36
Tabel keputusan yang diperkecil untuk penataan ulang inventaris Hoosier Burger

memantau dan melaporkan perubahan dalam tingkat persediaan bahan baku dan menerbitkan pesanan dan pembayaran material kepada pemasok. Dengan demikian, entitas data pusat untuk sistem ini akan menjadi ITEM PERSEDIAAN, yang sesuai dengan penyimpanan data D1 pada Gambar 8-22.

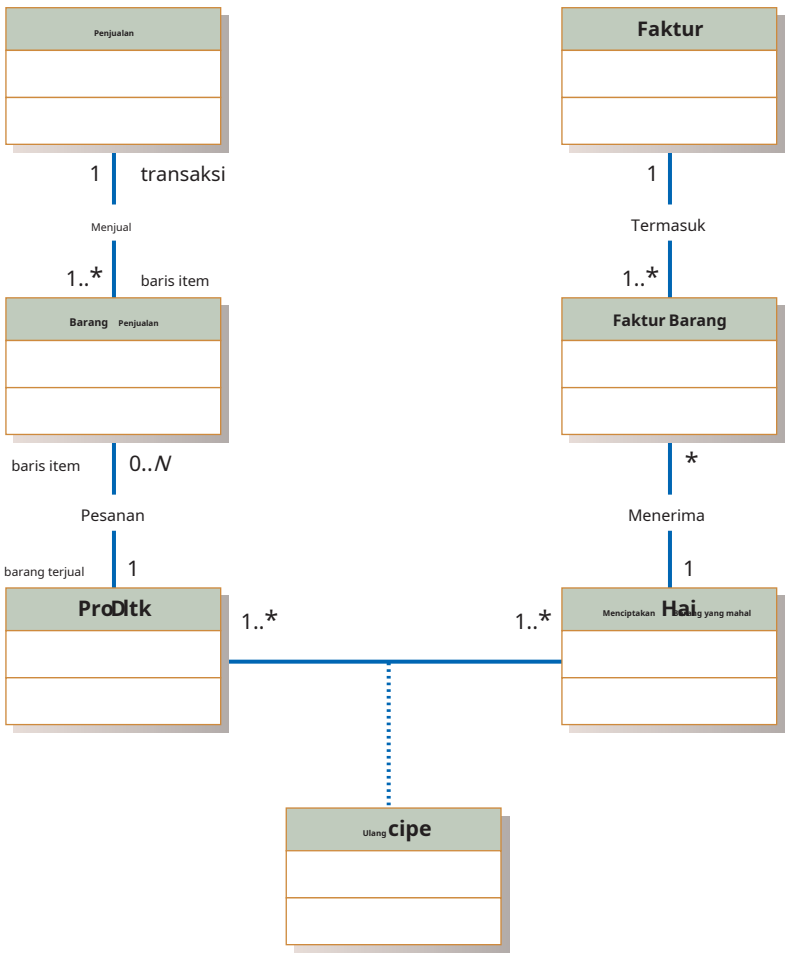
Perubahan dalam tingkat persediaan disebabkan oleh dua jenis transaksi: penerimaan barang baru dari pemasok dan konsumsi barang dari penjualan produk. Persediaan ditambahkan setelah penerimaan bahan baku baru, yang mana Hoosier Burger menerima FAKTUR pemasok (lihat Proses 1.0 pada Gambar 8-35). Setiap FAKTUR menunjukkan bahwa pemasok telah mengirim jumlah tertentu dari satu atau lebih INVOICE_ITEM, yang sesuai dengan ITEM PERSEDIAAN Hoosier. Persediaan digunakan ketika pelanggan memesan dan membayar PRODUK. Artinya, Hoosier melakukan PENJUALAN untuk satu atau lebih PENJUALAN ITEM, yang masing-masing sesuai dengan PRODUK makanan. Karena sistem pemrosesan pesanan pelanggan waktu nyata terpisah dari sistem kontrol persediaan, sumber, STOCK ON HAND pada Gambar 8-35, menggambarkan bagaimana data mengalir dari pemrosesan pesanan ke sistem kontrol persediaan. Terakhir, karena PRODUK makanan terdiri dari berbagai ITEM PERSEDIAAN (dan sebaliknya), Hoosier mempertahankan RESEP untuk menunjukkan berapa banyak dari setiap ITEM PERSEDIAAN yang digunakan untuk membuat satu PRODUK. Dari pembahasan ini, kami telah mengidentifikasi entitas data yang diperlukan dalam model data untuk sistem kontrol inventaris Hoosier Burger yang baru: ITEM PERSEDIAAN, FAKTUR, ITEM FAKTUR, PRODUK, PENJUALAN, PENJUALAN ITEM, dan RESEP. Untuk melengkapi model data, kami harus menentukan hubungan yang diperlukan antara entitas-entitas ini serta atribut untuk setiap entitas.

Kata-kata dalam uraian sebelumnya memberi tahu kita banyak hal yang perlu kita ketahui untuk menentukan hubungan:

- FAKTUR mencakup satu atau lebih ITEM FAKTUR, yang masing-masing sesuai dengan ITEM PERSEDIAAN. Jelas, ITEM FAKTUR tidak dapat ada tanpa FAKTUR terkait, dan seiring waktu akan ada tanda terima nol hingga banyak, atau ITEM FAKTUR, untuk ITEM PERSEDIAAN.
- Setiap PRODUK memiliki RESEP dari ITEM PERSEDIAAN. Dengan demikian, RESEP adalah entitas asosiatif yang mendukung hubungan jenis daftar bahan baku antara PRODUK dan ITEM PERSEDIAAN.
- PENJUALAN menunjukkan bahwa Hoosier menjual satu atau lebih PENJUALAN BARANG, yang masing-masing terkait dengan PRODUK. PENJUALAN BARANG tidak dapat terjadi tanpa PENJUALAN terkait, dan seiring waktu akan ada nol hingga banyak PENJUALAN BARANG untuk PRODUK.

Gambar 8-37 menunjukkan diagram kelas dengan kelas dan hubungan yang dijelaskan di atas. Dalam beberapa kasus, kami menyertakan nama peran (misalnya, Penjualan memainkan peran transaksi)

GAMBAR 8-37
Diagram kelas awal untuk sistem kontrol inventaris Hoosier Burger



dalam asosiasi Sells). RECIPE ditampilkan sebagai kelas asosiasi, bukan sekadar hubungan antara PRODUCT dan INVENTORY ITEM karena kemungkinan memiliki atribut dan perilaku. Sekarang setelah kita memahami kelas data dan hubungannya, kita harus memutuskan elemen data dan perilaku mana yang dikaitkan dengan kelas data dalam diagram ini. Kami telah memilih untuk mengembangkan model data konseptual menggunakan notasi UML, bukan notasi ER, tetapi Anda seharusnya dapat dengan mudah menerjemahkan diagram kelas UML ke dalam diagram ER (yang tersisa sebagai latihan di akhir bagian ini).

Anda mungkin bertanya-tanya pada titik ini mengapa hanya penyimpanan data INVENTORY yang ditampilkan pada Gambar 8-35 ketika ada tujuh kelas data pada diagram kelas. Penyimpanan data INVENTORY sesuai dengan kelas data INVENTORY ITEM pada Gambar 8-37. Kelas data lainnya disembunyikan di dalam proses lain yang diagram tingkat bawahnya belum kami tampilkan. Dalam langkah-langkah penataan persyaratan yang sebenarnya, Anda harus mencocokkan semua kelas data dengan penyimpanan data: setiap penyimpanan data mewakili beberapa subset dari suatu kelas atau diagram ER, dan setiap kelas data atau entitas disertakan dalam satu atau beberapa penyimpanan data. Idealnya, setiap penyimpanan data pada DFD primitif akan menjadi kelas data atau entitas individual.

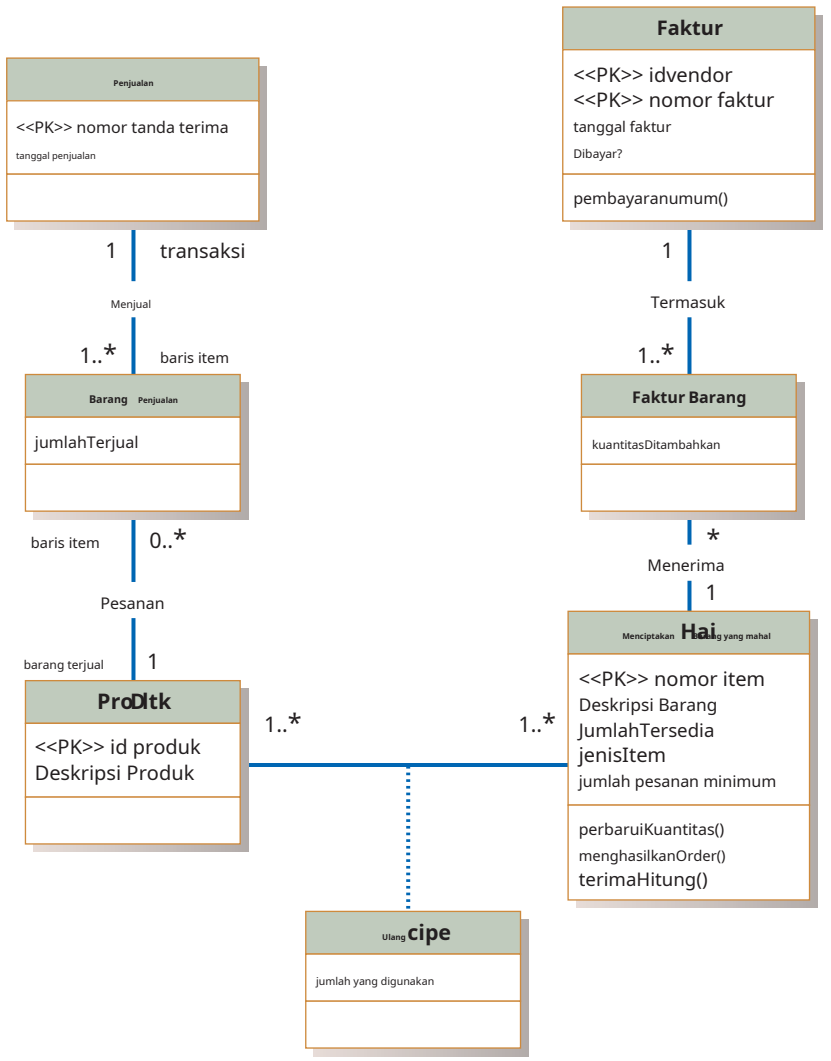
Untuk menentukan elemen data untuk kelas data, kami menyelidiki aliran data masuk dan keluar dari penyimpanan data yang sesuai dengan kelas data dan melengkapinya dengan studi logika keputusan dan logika temporal yang menggunakan atau mengubah data tentang kelas data. Enam aliran data dikaitkan dengan penyimpanan data INVENTORY pada Gambar 8-35. Deskripsi setiap aliran data dalam kamus proyek atau repositori CASE akan mencakup komposisi aliran data, yang kemudian memberi tahu kita data apa yang mengalir masuk atau keluar dari penyimpanan data.

dari penyimpanan data. Misalnya, aliran data Jumlah yang Digunakan yang berasal dari Proses 2.0 menunjukkan seberapa banyak pengurangan atribut Jumlah_dalam_Stok karena penggunaan ITEM PERSEDIAAN untuk memenuhi penjualan pelanggan. Dengan demikian, aliran data Jumlah yang Digunakan menyiratkan bahwa Proses 2.0 akan terlebih dahulu membaca rekaman ITEM PERSEDIAAN yang relevan, kemudian memperbarui atribut Jumlah_dalam_Stok, dan akhirnya menyimpan nilai yang diperbarui dalam rekaman. Bahasa Inggris Terstruktur untuk Proses 2.0 akan menggambarkan logika ini. Setiap aliran data akan dianalisis dengan cara yang sama (ruang tidak memungkinkan kita untuk menunjukkan analisis untuk setiap aliran data).

Analisis aliran data untuk elemen data dilengkapi dengan studi logika keputusan. Misalnya, perhatikan tabel keputusan pada Gambar 8-36. Satu kondisi yang digunakan untuk menentukan proses penataan ulang ITEM INVENTARIS melibatkan Jenis_Item. Dengan demikian, Proses 3.0 pada Gambar 8-35 (yang terkait dengan tabel keputusan ini) perlu mengetahui karakteristik setiap ITEM INVENTARIS ini, sehingga mengidentifikasi atribut lain dari kelas data ini.

Analisis DFD dan tabel keputusan juga menyarankan kemungkinan operasi untuk setiap kelas. Misalnya, kelas Barang Inventaris akan memerlukan operasi untuk memperbarui kuantitas yang tersedia, membuat pesanan pengisian ulang, dan menerima hitungan inventaris.

Setelah mempertimbangkan semua aliran data masuk dan keluar dari penyimpanan data yang terkait dengan kelas data, ditambah semua keputusan dan logika temporal yang terkait dengan pengendalian inventaris, kami memperoleh diagram kelas lengkap, dengan atribut dan operasi, yang ditunjukkan pada Gambar 8-38.



GAMBAR 8-38
Diagram kelas akhir untuk sistem kontrol inventaris Hoosier Burger

RINGKASAN

Kami telah menyajikan proses dan notasi dasar yang digunakan untuk memodelkan persyaratan data dari suatu sistem informasi menggunakan diagram kelas. Model data menunjukkan aturan bisnis yang relatif permanen yang menentukan sifat suatu organisasi. Aturan menentukan karakteristik data seperti karakteristik unik kelas data dan hubungan antara kelas data yang berbeda. Model data menunjukkan kategori utama data, yang disebut kelas untuk notasi UML; asosiasi atau hubungan antara kelas; dan atribut (hanya kelas yang memiliki atribut pada diagram kelas). Jenis khusus

Kelas yang disebut kelas asosiasi sering kali diperlukan untuk mewakili hubungan banyak-ke-banyak antara kelas. Kelas berbeda dari objek. Setiap objek dibedakan dari contoh lain dengan tipe yang sama oleh atribut pengenal (atau atribut). Hubungan adalah perekat yang menyatukan model data. Jumlah minimum dan maksimum objek yang berpartisipasi dalam hubungan mewakili aturan penting tentang sifat organisasi, seperti yang ditetapkan selama penentuan persyaratan.

ISTILAH UTAMA

1. Kelas abstrak
2. Operasi abstrak
3. Agregasi
4. Asosiasi
5. Peran asosiasi
6. Kelas asosiatif
7. Perilaku
8. Diagram kelas
9. Atribut lingkup kelas
10. Operasi lingkup kelas
11. Komposisi
12. Kelas Beton
13. Operasi konstruktor
14. Enkapsulasi
15. Metode
16. Multiplisitas
17. Objek
18. Objek (kelas)
19. Operasi
20. Polimorfisme
21. Operasi kueri
22. Negara
23. Operasi pembaruan

Cocokkan setiap istilah kunci di atas dengan definisi yang paling tepat cocok untuknya.

_____ Suatu bagian objek yang hanya dimiliki oleh satu objek utuh dan hidup serta mati bersama keseluruhannya.

_____ Hubungan bagian-bagian antara suatu objek komponen dan suatu objek agregat.

_____ Operasi yang sama dapat diterapkan pada dua atau lebih kelas dengan cara yang berbeda.

_____ Pelaksanaan suatu operasi.

_____ Menentukan bentuk atau protokol operasi, tetapi bukan implementasinya.

_____ Atribut suatu kelas yang menentukan nilai umum untuk seluruh kelas, dan bukan nilai spesifik untuk suatu contoh.

_____ Suatu kelas yang dapat memiliki instansi langsung.

_____ Suatu kelas yang tidak mempunyai instansi langsung, tetapi keturunannya boleh mempunyai instansi langsung.

_____ Asosiasi yang memiliki atribut atau operasinya sendiri atau yang berpartisipasi dalam hubungan dengan kelas lain.

_____ Menunjukkan berapa banyak objek yang berpartisipasi dalam hubungan tertentu.

_____ Akhir dari suatu asosiasi yang menghubungkannya dengan suatu kelas.

_____ Hubungan antara contoh kelas objek.

_____ Suatu operasi yang diterapkan pada suatu kelas dan bukan pada suatu instansi objek.

_____ Suatu operasi yang mengubah keadaan suatu objek.

_____ Suatu operasi yang mengakses keadaan suatu objek tetapi tidak mengubah keadaan tersebut.

_____ Suatu operasi yang menciptakan contoh baru suatu kelas.

_____ Teknik menyembunyikan detail implementasi internal suatu objek dari pandangan luarnya.

_____ Suatu fungsi atau layanan yang disediakan oleh semua instansi suatu kelas.

_____ Diagram yang menunjukkan struktur statis model berorientasi objek.

_____ Pengelompokan logis objek yang memiliki atribut dan perilaku (metode) yang sama (atau serupa).

_____ Menggambarkan bagaimana suatu objek bertindak dan bereaksi.

_____ Meliputi properti suatu objek (atribut dan hubungan) dan nilai yang dimiliki properti tersebut.

_____ Entitas yang memiliki peran yang terdefinisi dengan baik dalam domain aplikasi, dan memiliki karakteristik status, perilaku, dan identitas.

PERTANYAAN ULASAN

1. Berikan contoh agregasi. Contoh Anda harus mencakup setidaknya satu objek agregat dan tiga objek komponen. Tentukan multiplisitas di setiap ujung semua hubungan agregasi.
2. Bandingkan istilah-istilah berikut:
   a. Kelas objek; objek
   b. Kelas abstrak; kelas konkret

MASALAH DAN LATIHAN

1. Gambarkan diagram kelas yang menunjukkan kelas, atribut, operasi, dan hubungan yang relevan untuk setiap situasi berikut (jika Anda yakin perlu membuat asumsi tambahan, nyatakan dengan jelas untuk setiap situasi):
   a. Suatu perusahaan memiliki sejumlah karyawan. Atribut Karyawan meliputi ID karyawan (kunci utama), nama, alamat, dan tanggal lahir. Perusahaan tersebut juga memiliki beberapa proyek. Atribut Proyek meliputi projectName dan startDate. Setiap karyawan dapat ditugaskan ke satu atau beberapa proyek, atau tidak dapat ditugaskan ke suatu proyek. Suatu proyek harus memiliki setidaknya satu karyawan yang ditugaskan, dan dapat memiliki sejumlah karyawan yang ditugaskan. Tarif penagihan karyawan dapat bervariasi menurut proyek, dan perusahaan ingin mencatat tarif penagihan yang berlaku untuk setiap karyawan saat ditugaskan ke proyek tertentu. Pada akhir setiap bulan, perusahaan mengirimkan cek kepada setiap karyawan yang telah mengerjakan suatu proyek selama bulan tersebut. Jumlah cek didasarkan pada tarif penagihan dan jam yang dicatat untuk setiap proyek yang ditugaskan kepada karyawan tersebut.
   b. Sebuah universitas memiliki sejumlah besar mata kuliah dalam katalognya. Atribut Mata Kuliah meliputi Nomor Mata Kuliah (kunci utama), Nama Mata Kuliah, dan satuan mata kuliah. Setiap mata kuliah dapat memiliki satu atau beberapa mata kuliah yang berbeda sebagai prasyarat, atau suatu mata kuliah mungkin tidak memiliki prasyarat. Demikian pula, suatu mata kuliah tertentu dapat menjadi prasyarat untuk sejumlah mata kuliah, atau mungkin bukan prasyarat untuk mata kuliah lainnya. Universitas menambahkan atau menghapus prasyarat untuk suatu mata kuliah hanya jika direktur mata kuliah tersebut mengajukan permintaan resmi untuk tujuan tersebut.
   c. Laboratorium memiliki beberapa ahli kimia yang bekerja pada satu atau lebih proyek. Ahli kimia juga dapat menggunakan jenis peralatan tertentu pada setiap proyek. Atribut Ahli Kimia meliputi nama dan nomor telepon. Atribut Proyek meliputi Nama proyek dan Tanggal mulai. Atribut Peralatan meliputi No seri dan biaya. Organisasi ingin mencatat assignDate—yaitu, tanggal saat item peralatan tertentu ditugaskan ke ahli kimia tertentu yang bekerja pada proyek tertentu—serta total Jam, yaitu, jumlah total jam ahli kimia telah menggunakan peralatan untuk proyek tersebut. Organisasi juga ingin melacak penggunaan setiap jenis peralatan oleh seorang ahli kimia. Hal itu dilakukan dengan menghitung jumlah rata-rata jam ahli kimia telah menggunakan peralatan itu pada semua proyek yang ditugaskan. Seorang ahli kimia harus ditugaskan untuk setidaknya satu proyek dan satu item peralatan. Item peralatan tertentu tidak perlu ditugaskan, dan proyek tertentu tidak perlu ditugaskan baik seorang ahli kimia atau item peralatan.
   d. Suatu mata kuliah dapat memiliki satu atau beberapa bagian terjadwal, atau mungkin tidak memiliki bagian terjadwal. Atribut Mata Kuliah meliputi courseID, courseName, dan unit. Atribut Section meliputi sectionNumber dan semester. Nilai sectionNumber adalah bilangan bulat (seperti 1 atau 2) yang membedakan satu section dari section lain untuk mata kuliah yang sama, tetapi tidak mengidentifikasi section secara unik. Ada operasi yang disebut findNumSections yang menemukan jumlah section yang ditawarkan untuk mata kuliah tertentu dalam semester tertentu.
   e. Sebuah rumah sakit memiliki sejumlah besar dokter terdaftar. Atribut Dokter meliputi ID Dokter (kunci utama) dan spesialisasi. Pasien dirawat di rumah sakit oleh dokter. Atribut Pasien meliputi ID Pasien (kunci utama) dan Nama Pasien. Setiap pasien yang dirawat harus memiliki tepat satu dokter yang merawat. Seorang dokter dapat secara opsional merawat sejumlah pasien. Setelah dirawat, pasien tertentu harus dirawat oleh setidaknya satu dokter. Seorang dokter tertentu dapat merawat sejumlah pasien, atau dia tidak boleh merawat pasien mana pun. Setiap kali seorang pasien dirawat oleh seorang dokter, rumah sakit ingin mencatat rincian perawatan dengan menyertakan tanggal, waktu, dan hasil perawatan.
2. Seorang mahasiswa, yang atributnya meliputi Nama, Alamat, telepon, dan usia mahasiswa, dapat terlibat dalam beberapa kegiatan berbasis kampus. Universitas mencatat jumlah tahun mahasiswa tersebut telah berpartisipasi dalam kegiatan tertentu dan, pada akhir setiap tahun akademik, mengirimkan laporan kegiatan kepada mahasiswa yang menunjukkan partisipasinya dalam berbagai kegiatan. Gambarkan diagram kelas untuk situasi ini.
3. Bank memiliki tiga jenis rekening: giro, tabungan, dan pinjaman. Berikut ini adalah atribut untuk setiap jenis rekening:

   MEMERIKSA: No_Akun, Tanggal_Dibuka, Saldo, Biaya_Layanan

   TABUNGAN: No_Akun, Tanggal_Pembukaan, Saldo, Suku_Bunga

   PINJAMAN: No_Akun, Tanggal_Pembukaan, Saldo, Suku_Bunga, Pembayaran

   Asumsikan bahwa setiap rekening bank harus menjadi anggota dari salah satu subtipe ini. Pada akhir setiap bulan, bank menghitung saldo di setiap rekening dan mengirimkan laporan kepada nasabah yang memegang rekening tersebut. Perhitungan saldo bergantung pada jenis rekening. Misalnya, saldo rekening giro dapat mencerminkan biaya layanan, sedangkan saldo rekening tabungan dapat mencakup jumlah bunga. Gambarkan diagram kelas untuk menggambarkan situasi tersebut. Diagram Anda harus mencakup kelas abstrak, serta operasi abstrak untuk menghitung saldo.
4. Ubah diagram kelas pada Gambar 8-37 menjadi diagram ER yang setara. Bandingkan kedua diagram tersebut. Jelaskan spesifikasi sistem yang berbeda yang ditunjukkan pada setiap diagram.

REFERENSI

Booch, G. 1994.*Analisis dan Desain Berorientasi Objek dengan Aplikasi,*Edisi ke-2. Redwood City, CA: Benjamin Cummings. Fowler, M. 2000.*UML Distilled: Panduan Singkat untuk Pemodelan Objek Bahasa*, edisi ke-2. Reading, MA: Addison-Wesley. George, J., D. Batra, J. Valacich, dan J. Hoffer. 2007.*Obyek-Analisis dan Desain Sistem Berorientasi*, edisi ke-2. Upper Saddle River, NJ: Prentice Hall.

Rumbaugh, J., M. Blaha, W. Premerlani, F. Eddy, dan W. Lorensen, 1991.*Pemodelan dan Desain Berorientasi Objek.*Upper Saddle River, NJ: Prentice Hall.

*Panduan Notasi UML.*1997. Dokumen diakses dari *www.rational.com/uml*Diakses pada tanggal 19 Februari 2009. Hak cipta dimiliki oleh Rational Software Corporation, Microsoft Corporation, Hewlett-Packard Company, Oracle Corporation, Sterling Software, MCI Systemhouse Corporation, Unisys Corporation, ICON Computing, IntelliCorp, i-Logix, IBM Corporation, ObjecTime Limited, Platinum Technology Incorporated, Ptech Incorporated, Taskon A/S, Reich Technologies, Softeam.