



Bab 6: Desain Basis Data Menggunakan Model E-R

Konsep-konsep Sistem Basis Data, Edisi 7.

©Silberschatz, Korth dan Sudarshan

Lihat www.db-book.com untuk mengetahui ketentuan tentang penggunaan ulang



Fase Desain

- Fase awal - mengkarakterisasi kebutuhan data calon pelanggan secara lengkap pengguna basis data.
- Fase kedua - memilih model data
 - Menerapkan konsep-konsep model data yang dipilih
 - Menerjemahkan persyaratan ini ke dalam skema konseptual dari basis data.
 - Skema konseptual yang dikembangkan sepenuhnya menunjukkan persyaratan fungsional perusahaan.
 - Jelaskan jenis operasi (atau transaksi) yang akan dilakukan pada data.



Fase Desain

- Tahap Akhir - Beranjak dari model data abstrak ke implementasi basis data
 - Desain Logis - Memutuskan skema basis data.
 - Desain basis data mengharuskan kita untuk menemukan koleksi yang "baik" dari skema hubungan.
 - Keputusan bisnis - Atribut apa yang harus kita catat dalam database?
 - Keputusan Ilmu Komputer - Skema relasi apa yang harus kita miliki dan bagaimana atribut harus didistribusikan di antara berbagai skema relasi?
 - Desain Fisik - Memutuskan tata letak fisik database



Alternatif Desain

- Dalam mendesain skema database, kita harus memastikan bahwa kita menghindari dua hal jebakan utama:
 - Redundansi: desain yang buruk dapat mengakibatkan pengulangan informasi.
 - Representasi informasi yang berlebihan dapat menyebabkan ketidakkonsistenan data di antara berbagai salinan informasi
 - Ketidaklengkapan: desain yang buruk dapat membuat aspek-aspek tertentu dari perusahaan menjadi sulit atau tidak mungkin untuk dimodelkan.
- Menghindari desain yang buruk saja tidak cukup. Mungkin ada banyak sekali desain bagus yang harus kita pilih.



Pendekatan Desain

- Model Hubungan Entitas (dibahas dalam bab ini)
 - Memodelkan perusahaan sebagai kumpulan *entitas* dan *hubungan*
 - Entitas: "benda" atau "objek" dalam perusahaan yang dapat dibedakan dari objek lain
 - Dijelaskan oleh satu set *atribut*
 - Hubungan: hubungan di antara beberapa entitas
 - Diwakili secara diagramatis oleh *diagram hubungan entitas*:
- Teori Normalisasi (Bab 7)
 - Memformalkan desain apa yang buruk, dan mengujinya



Garis Besar Model UGD



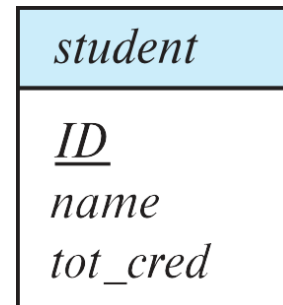
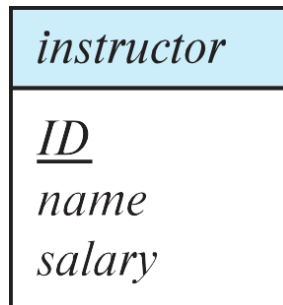
Kumpulan

- **Entitas** adalah objek yang ada dan dapat dibedakan dari objek lainnya.
 - Contoh: orang tertentu, perusahaan, acara, pabrik
- Himpunan **entitas** adalah sekumpulan entitas dengan tipe yang sama yang memiliki kesamaan properti.
 - Contoh: kumpulan semua orang, perusahaan, pohon, hari libur
- Entitas diwakili oleh sekumpulan atribut; yaitu properti deskriptif yang dimiliki oleh semua anggota kumpulan entitas.
 - Contoh:
 $instruktur = (ID, nama, gaji)$
 $course = (course_id, judul, sks)$
- Sebuah subset dari atribut membentuk **kunci utama** dari himpunan entitas; yaitu, secara unik mengidentifikasi setiap anggota himpunan.



Merepresentasikan set Entitas dalam

- Set entitas dapat direpresentasikan secara grafis sebagai berikut:
 - Persegi panjang merepresentasikan set entitas.
 - Atribut yang tercantum di dalam persegi panjang entitas
 - Garis bawah menunjukkan atribut kunci utama





Kumpulan

- **Hubungan** adalah asosiasi di antara beberapa entitas

Contoh:

44553 (Peltier)
entitas *siswa*

penasihat
set hubungan

22222 (Einstein)
entitas *instruktur*

- Himpunan **relasi** adalah relasi matematis di antara $n \geq 2$ entitas, masing-masing diambil dari set entitas

$$\{(e_1, e_2, \dots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \dots, e_n \in E_n\}$$

di mana (e_1, e_2, \dots, e_n) adalah sebuah hubungan

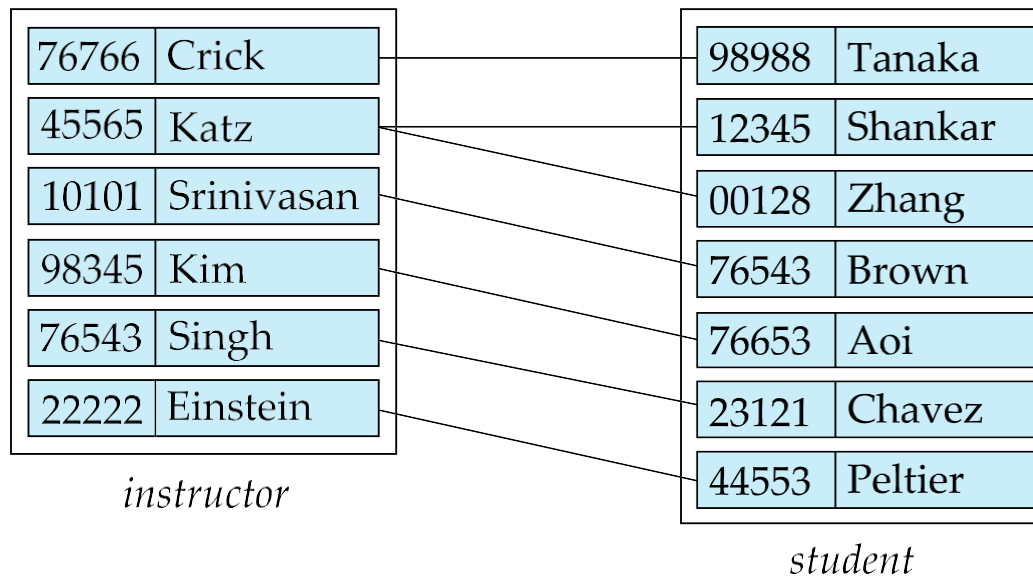
- Contoh:

$$(44553.22222) \in \textit{penasihat}$$



Kumpulan Hubungan

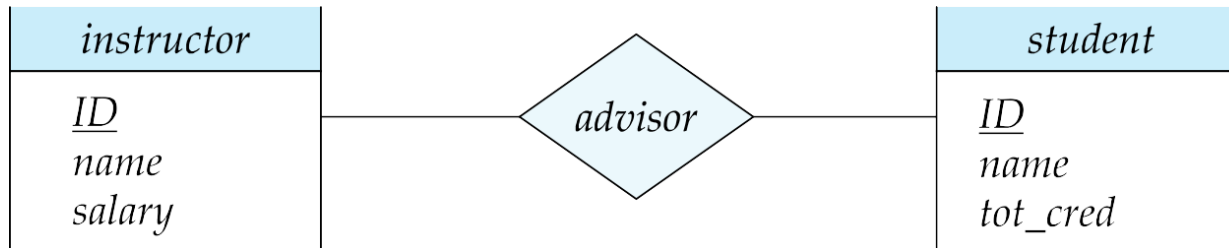
- Contoh: kami mendefinisikan himpunan relasi *penasihat* untuk menunjukkan hubungan antara siswa dan instruktur yang bertindak sebagai penasihat mereka.
- Secara bergambar, kami menarik garis di antara entitas-entitas terkait.





Merepresentasikan Kumpulan Hubungan melalui Diagram ER

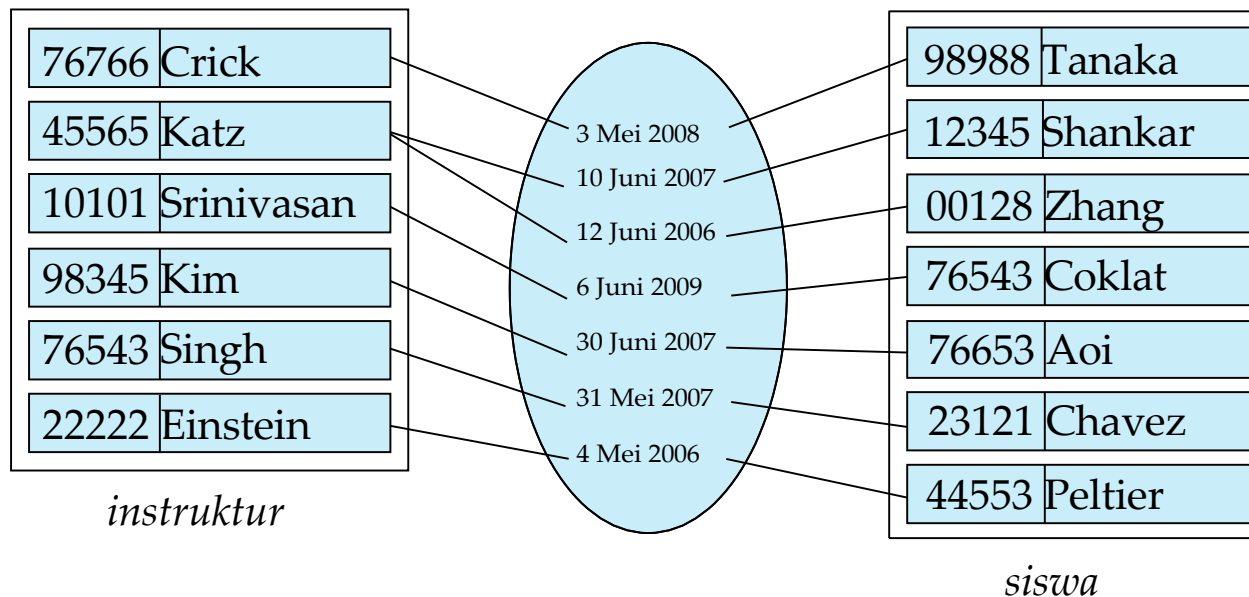
- Berlian merepresentasikan rangkaian hubungan.





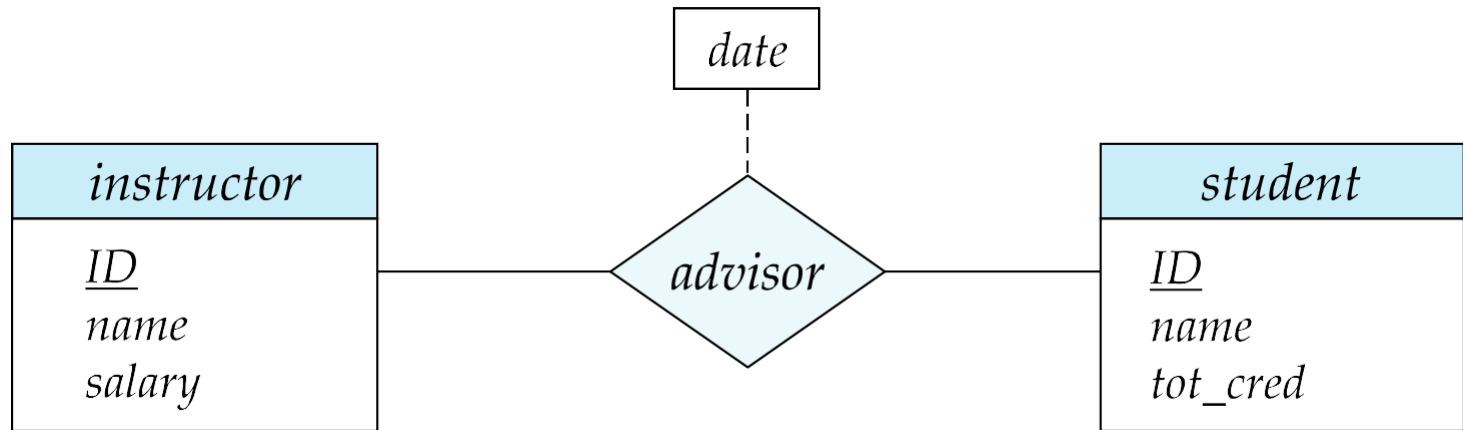
Kumpulan Hubungan

- Sebuah atribut juga dapat dikaitkan dengan himpunan hubungan.
- Misalnya, himpunan hubungan *penasihat* antara himpunan entitas *instruktur* dan *siswa* dapat memiliki atribut *tanggal* yang melacak kapan siswa mulai dikaitkan dengan penasihat





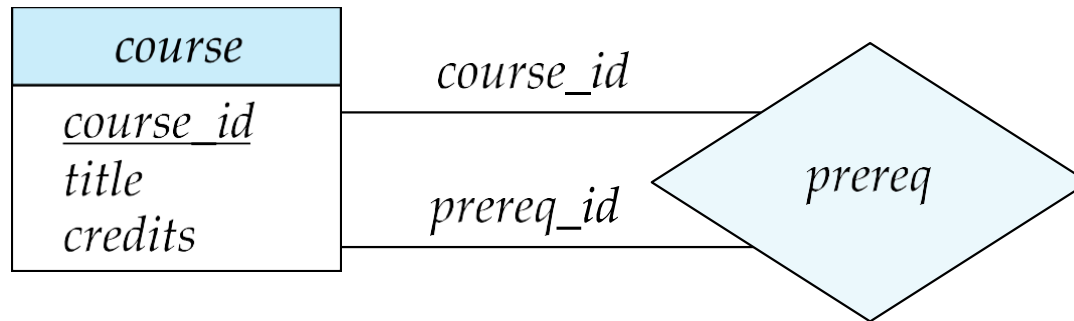
Himpunan Hubungan dengan





Pera

- Kumpulan entitas dari suatu hubungan tidak perlu berbeda
 - Setiap kejadian dari himpunan entitas memainkan "peran" dalam hubungan
- Label "*course_id*" dan "*prereq_id*" disebut **peran**.





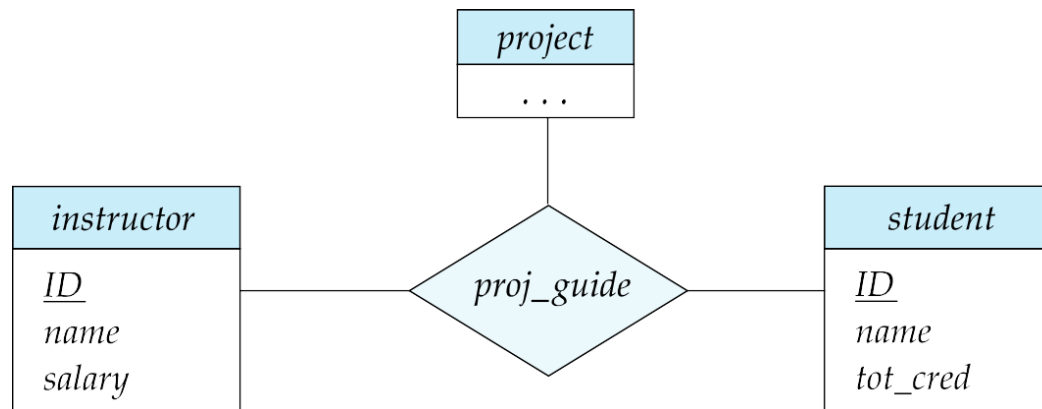
Derajat Himpunan

- Hubungan biner
 - melibatkan dua himpunan entitas (atau derajat dua).
 - sebagian besar himpunan relasi dalam sistem basis data adalah biner.
- Hubungan antara lebih dari dua set entitas jarang terjadi. Kebanyakan hubungan bersifat biner. (Lebih lanjut mengenai hal ini nanti.)
 - Contoh: *siswa* mengerjakan *proyek* penelitian di bawah bimbingan *instruktur*.
 - hubungan *proj_guide* adalah hubungan terner antara *instruktur*, *siswa*, dan *proyek*



Kumpulan Hubungan Non-

- Sebagian besar himpunan hubungan adalah biner
- Ada kalanya akan lebih mudah untuk merepresentasikan hubungan sebagai non-biner.
- Diagram E-R dengan Hubungan Ternary





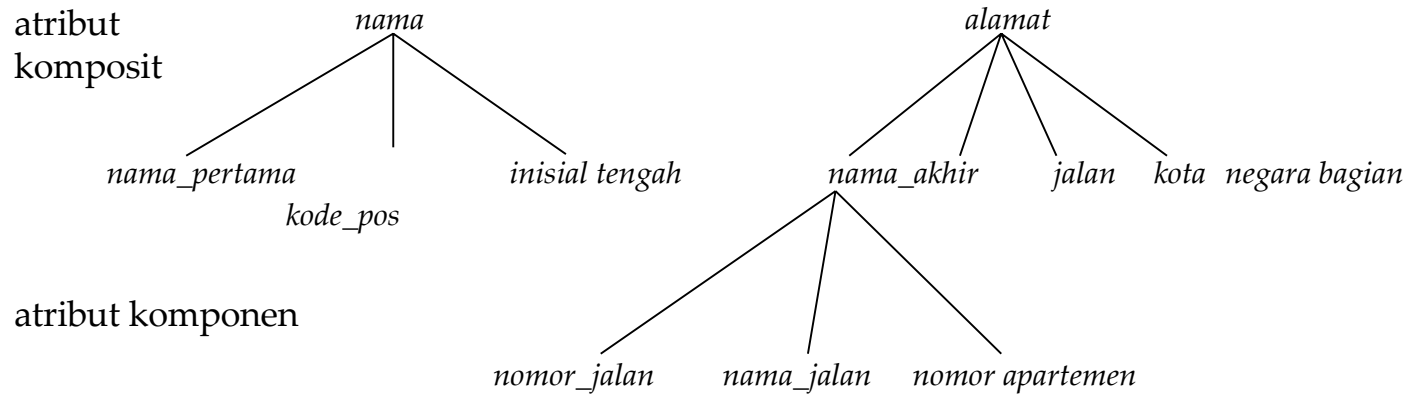
Atribut yang

- Jenis atribut:
 - Atribut **seederhana** dan **komposit**.
 - Atribut **bernilai tunggal** dan **multivariabel**
 - Contoh: atribut multivariabel: *nomor_telepon*
 - Atribut yang **diturunkan**
 - Dapat dihitung dari atribut lain
 - Contoh: usia, tanggal_lahir yang diberikan
- **Domain** - kumpulan nilai yang diizinkan untuk setiap atribut



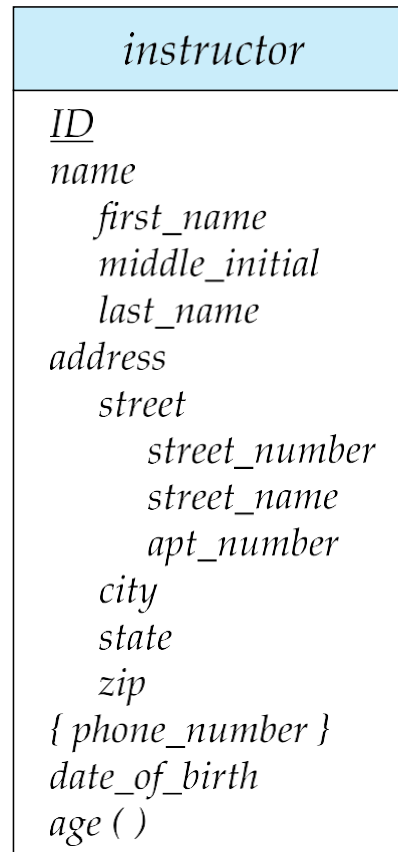
Atribut Komposit

- Atribut komposit memungkinkan kita untuk membagi atribut menjadi beberapa bagian (atribut lain).





Merepresentasikan Atribut Kompleks dalam Diagram ER



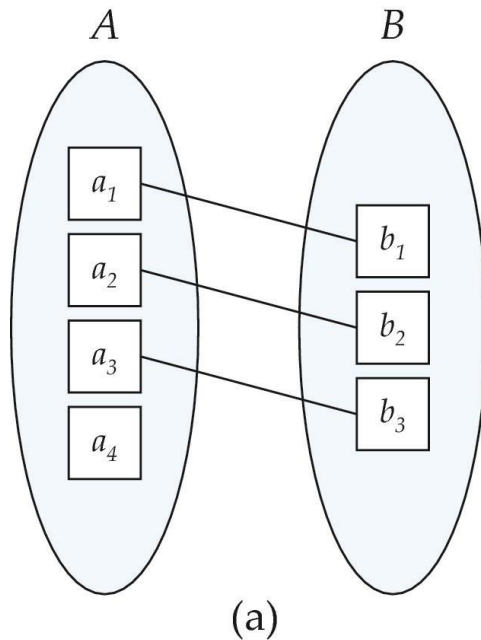


Memetakan Batasan

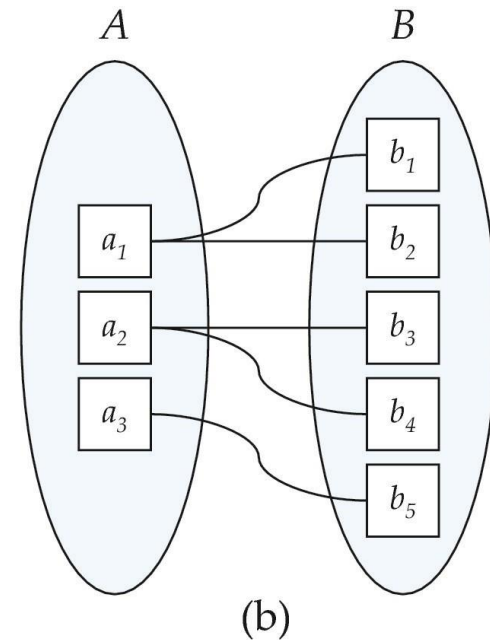
- Menyatakan jumlah entitas yang dapat dikaitkan dengan entitas lain melalui kumpulan hubungan.
- Paling berguna dalam menggambarkan set hubungan biner.
- Untuk set hubungan biner, kardinalitas pemetaan haruslah salah satu dari jenis berikut:
 - Satu lawan satu
 - Satu untuk banyak orang
 - Banyak ke satu
 - Banyak untuk banyak orang



Pemetaan



Satu ke satu

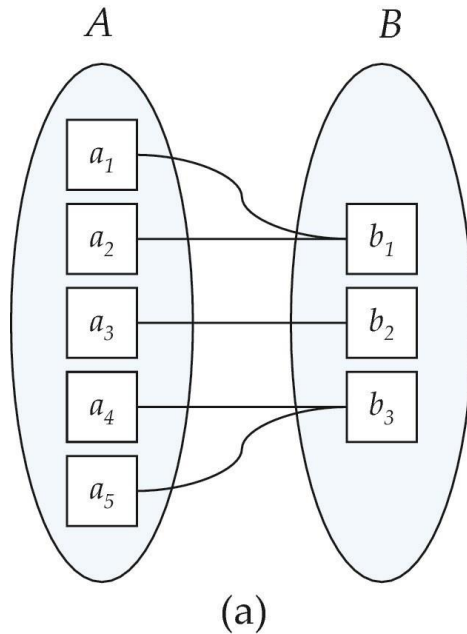


Satu ke banyak

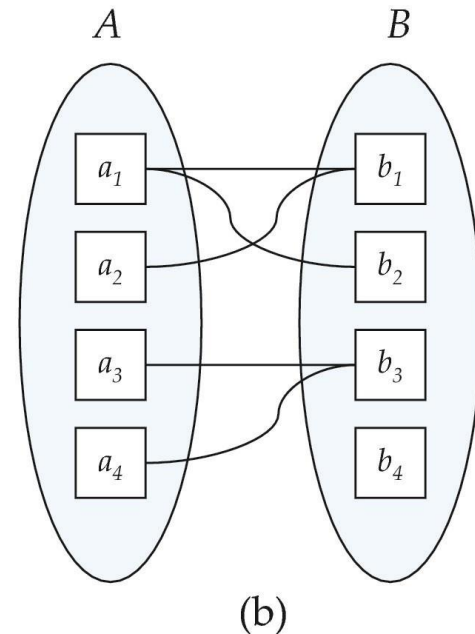
Catatan: Beberapa elemen di A dan B mungkin tidak dapat dipetakan ke elemen apa pun di himpunan lainnya



Pemetaan



Banyak ke satu



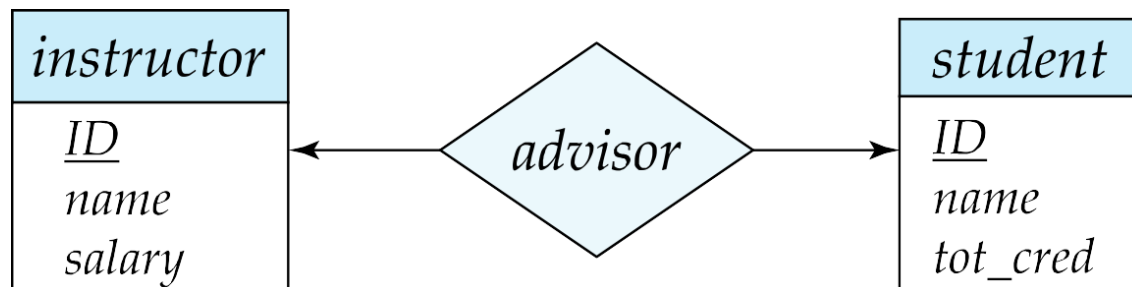
Banyak ke banyak

Catatan: Beberapa elemen di A dan B mungkin tidak dapat dipetakan ke elemen apa pun di himpunan lainnya



Merepresentasikan Batasan Kardinalitas dalam Diagram ER

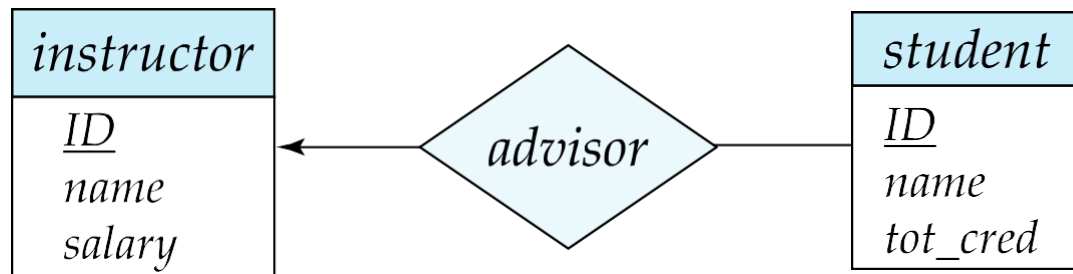
- Kita mengekspresikan batasan kardinalitas dengan menggambar sebuah garis berarah (\rightarrow), yang menandakan "satu", atau sebuah garis tak berarah (-), yang menandakan "banyak", antara himpunan relasi dan himpunan entitas.
- Hubungan satu-ke-satu antara *instruktur* dan *siswa*:
 - Seorang siswa terhubung dengan paling banyak satu *instruktur* melalui hubungan *penasihat*
 - Seorang *siswa* dikaitkan dengan paling banyak satu *departemen* melalui *stud_dept*





Hubungan Satu-ke-

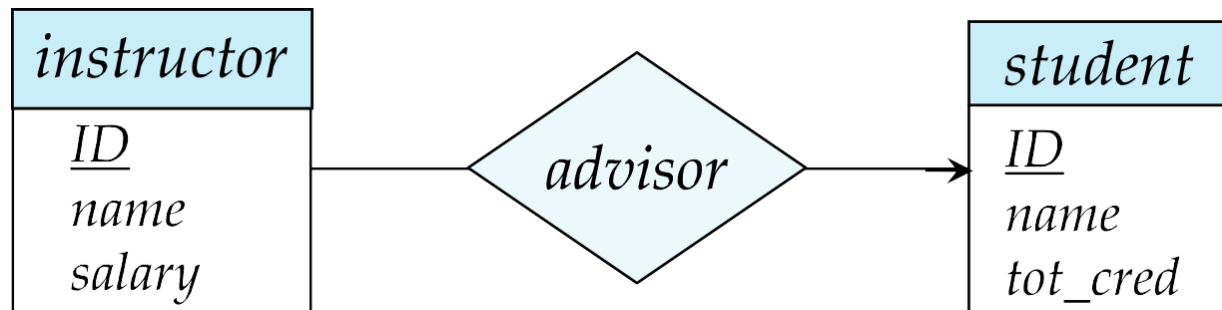
- hubungan satu-ke-banyak antara *instruktur* dan *siswa*
 - seorang instruktur berhubungan dengan beberapa (termasuk 0) siswa melalui *penasihat*
 - seorang siswa berhubungan dengan paling banyak satu instruktur melalui *penasihat*,





Hubungan Banyak-ke-Satu

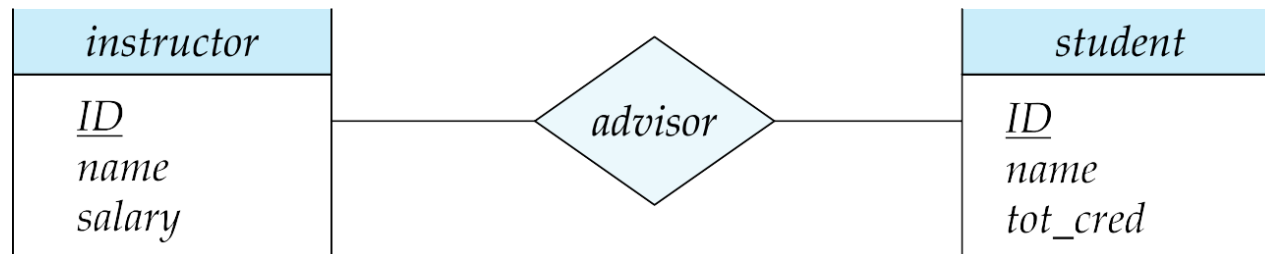
- Dalam hubungan banyak-ke-satu antara *instruktur* dan *siswa*,
 - seorang instruktur berhubungan dengan paling banyak satu siswa melalui *penasihat*,
 - dan seorang siswa dikaitkan dengan beberapa (termasuk 0) instruktur melalui *penasihat*





Hubungan Banyak-ke-

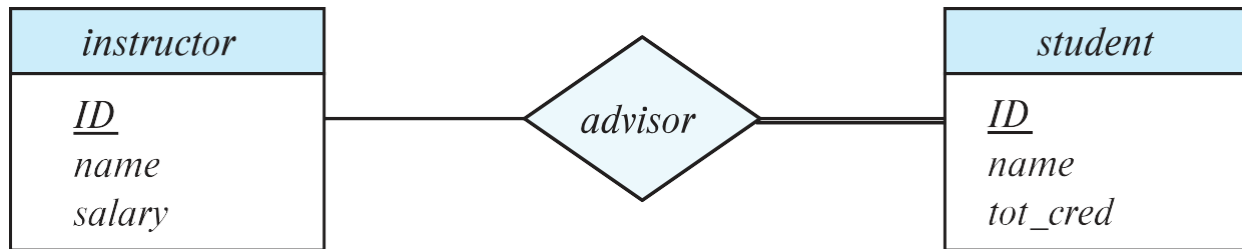
- Seorang instruktur berhubungan dengan beberapa (mungkin 0) siswa melalui *penasihat*
- Seorang siswa dikaitkan dengan beberapa (mungkin 0) instruktur melalui *penasihat*





Partisipasi Total dan Parsial

- **Partisipasi total** (ditunjukkan dengan garis ganda): setiap entitas dalam himpunan entitas berpartisipasi dalam setidaknya satu hubungan dalam himpunan hubungan



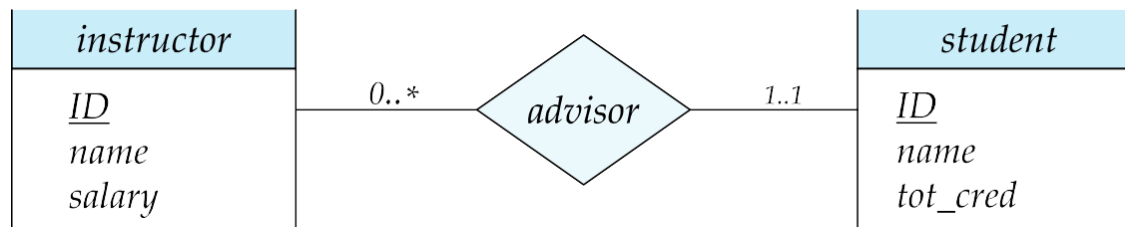
partisipasi *siswa* dalam hubungan dengan *penasihat* adalah total

- setiap *siswa* harus memiliki instruktur terkait
- **Partisipasi parsial**: beberapa entitas mungkin tidak berpartisipasi dalam hubungan apa pun dalam himpunan hubungan
 - Contoh: partisipasi *instruktur* dalam *penasihat* bersifat parsial



Notasi untuk Mengekspresikan Kendala yang Lebih Kompleks

- Sebuah garis dapat memiliki kardinalitas minimum dan maksimum yang terkait, yang ditunjukkan dalam bentuk $l..h$, di mana l adalah minimum dan h adalah kardinalitas maksimum
 - Nilai minimum 1 menunjukkan partisipasi total.
 - Nilai maksimum 1 menunjukkan bahwa entitas berpartisipasi dalam paling banyak satu hubungan
 - Nilai maksimum * menunjukkan tidak ada batas.
- Contoh



- Instruktur dapat membimbing 0 mahasiswa atau lebih. Seorang siswa harus memiliki 1 pembimbing; tidak dapat memiliki lebih dari satu pembimbing



Batasan Kardinalitas pada Hubungan Ternary

- Kami mengizinkan paling banyak satu anak panah keluar dari hubungan terner (atau tingkat yang lebih tinggi) untuk menunjukkan batasan kardinalitas
- Sebagai contoh, tanda panah dari *proj_guide* ke *instruktur* menunjukkan bahwa setiap siswa memiliki paling banyak satu panduan untuk sebuah proyek
- Jika terdapat lebih dari satu tanda panah, ada dua cara untuk menentukan artinya.
 - Sebagai contoh, hubungan terner R antara A , B dan C dengan panah ke B dan C dapat berarti
 1. Setiap entitas A dikaitkan dengan entitas unik dari B dan C atau
 2. Setiap pasangan entitas dari (A, B) diasosiasikan dengan entitas C yang unik, dan setiap pasangan (A, C) diasosiasikan dengan B yang unik
 - Setiap alternatif telah digunakan dalam formalisme yang berbeda
 - Untuk menghindari kebingungan, kami melarang lebih dari satu panah



Kunci

- Primary key menyediakan cara untuk menentukan bagaimana entitas dan relasi dibedakan. Kami akan mempertimbangkannya:
 - Set entitas
 - Kumpulan hubungan.
 - Set entitas yang lemah



Kunci utama untuk Set

- Menurut definisi, entitas individu berbeda.
- Dari perspektif basis data, perbedaan di antara mereka harus dinyatakan dalam bentuk atribut mereka.
- Nilai-nilai dari nilai atribut suatu entitas harus sedemikian rupa sehingga dapat mengidentifikasi entitas secara unik.
 - Tidak ada dua entitas dalam kumpulan entitas yang diperbolehkan memiliki nilai yang sama persis untuk semua atribut.
- Kunci untuk sebuah entitas adalah sekumpulan atribut yang cukup untuk membedakan entitas satu sama lain



Kunci Utama untuk Himpunan

- Untuk membedakan antara berbagai hubungan dari sebuah himpunan hubungan, kita menggunakan kunci utama individu dari entitas dalam himpunan hubungan.
 - Misalkan R adalah sebuah himpunan relasi yang melibatkan himpunan entitas E_1, E_2, \dots, E_n
 - Kunci utama untuk R terdiri dari gabungan kunci utama dari himpunan entitas E_1, E_2, \dots, E_n
 - Jika himpunan relasi R memiliki atribut a_1, a_2, \dots, a_m yang terkait dengannya, maka kunci utama R juga mencakup atribut a_1, a_2, \dots, a_m
- Contoh: set hubungan "penasihat".
 - Kunci utama terdiri dari *instructor.ID* dan *student.ID*
- Pilihan kunci utama untuk himpunan relasi bergantung pada kardinalitas pemetaan himpunan relasi.



Pilihan kunci utama untuk Hubungan Biner

- Hubungan Banyak-ke-Banyak. Gabungan kunci utama sebelumnya adalah superkey minimal dan dipilih sebagai kunci utama.
- Hubungan satu-ke-banyak . Kunci utama dari sisi "Banyak" adalah superkey minimal dan digunakan sebagai kunci utama.
- Hubungan banyak-ke-satu. Kunci utama dari sisi "Banyak" adalah minimal superkey dan digunakan sebagai kunci utama.
- Hubungan satu-ke-satu. Kunci utama dari salah satu set entitas yang berpartisipasi membentuk superkey minimal, dan salah satu dari keduanya dapat dipilih sebagai kunci utama.



Kumpulan

- Pertimbangkan sebuah entitas *bagian*, yang secara unik diidentifikasi oleh *course_id*, *semester*, *tahun*, dan *sec_id*.
- Jelasnya, entitas *section* berhubungan dengan entitas *course*. Misalkan kita membuat himpunan hubungan *sec_course* antara himpunan entitas *section* dan *course*.
- Perhatikan bahwa informasi dalam *sec_course* bersifat redundan, karena *section* sudah memiliki atribut *course_id*, yang mengidentifikasi *course* yang terkait dengan *section* tersebut.
- Salah satu opsi untuk menangani redundansi ini adalah dengan menghilangkan hubungan *sec_course*; namun, dengan melakukan hal tersebut hubungan antara *section* dan *course* menjadi implisit dalam sebuah atribut, yang mana hal ini tidak diinginkan.



Himpunan Entitas

- Cara alternatif untuk menangani redundansi ini adalah dengan tidak menyimpan atribut *course_id* dalam entitas *section* dan hanya menyimpan atribut yang tersisa yaitu *section_id*, *tahun*, dan *semester*.
 - Namun, *bagian* set entitas kemudian tidak memiliki atribut yang cukup untuk mengidentifikasi entitas *bagian* tertentu secara unik
- Untuk menangani masalah ini, kita memperlakukan hubungan *sec_course* sebagai hubungan khusus yang menyediakan informasi tambahan, dalam hal ini, *course_id*, yang diperlukan untuk mengidentifikasi entitas *bagian* secara unik.
- **Himpunan entitas** yang **lemah** adalah **himpunan** yang keberadaannya bergantung pada entitas lain, yang disebut **entitas pengenalnya**
- Alih-alih mengasosiasikan kunci utama dengan entitas yang lemah, kita menggunakan entitas pengenal, bersama dengan atribut tambahan yang disebut **pembeda** untuk mengidentifikasi entitas yang lemah secara unik.



Himpunan Entitas

- **Himpunan** entitas yang bukan merupakan himpunan entitas yang lemah disebut sebagai **himpunan entitas yang kuat**.
- Setiap entitas lemah harus dikaitkan dengan entitas pengenalan; yaitu, himpunan entitas lemah dikatakan **bergantung** pada **keberadaan** himpunan entitas pengenalan.
- Himpunan entitas yang mengidentifikasi dikatakan **memiliki** himpunan entitas lemah yang diidentifikasinya.
- Hubungan yang mengaitkan himpunan entitas lemah dengan himpunan entitas pengenalan disebut **hubungan pengenalan**.
- Perhatikan bahwa skema relasional yang pada akhirnya kita buat dari *bagian* entity set memang memiliki atribut *course_id*, untuk alasan yang akan menjadi jelas nanti, meskipun kita telah menghapus atribut *course_id* dari *bagian* entity set.



Mengekspresikan Himpunan

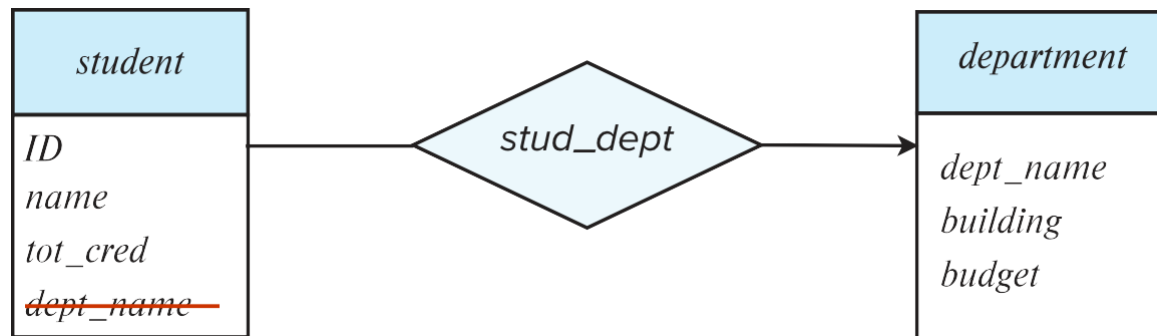
- Dalam diagram E-R, himpunan entitas yang lemah digambarkan melalui persegi panjang ganda.
- Kami menggarisbawahi pembeda dari himpunan entitas yang lemah dengan garis putus-putus.
- Himpunan hubungan yang menghubungkan himpunan entitas lemah dengan himpunan entitas kuat yang mengidentifikasi digambarkan dengan berlian ganda.
- Kunci utama untuk *bagian* - (*id_kuliah*, *id_seksi*, *semester*, *tahun*)





Atribut yang

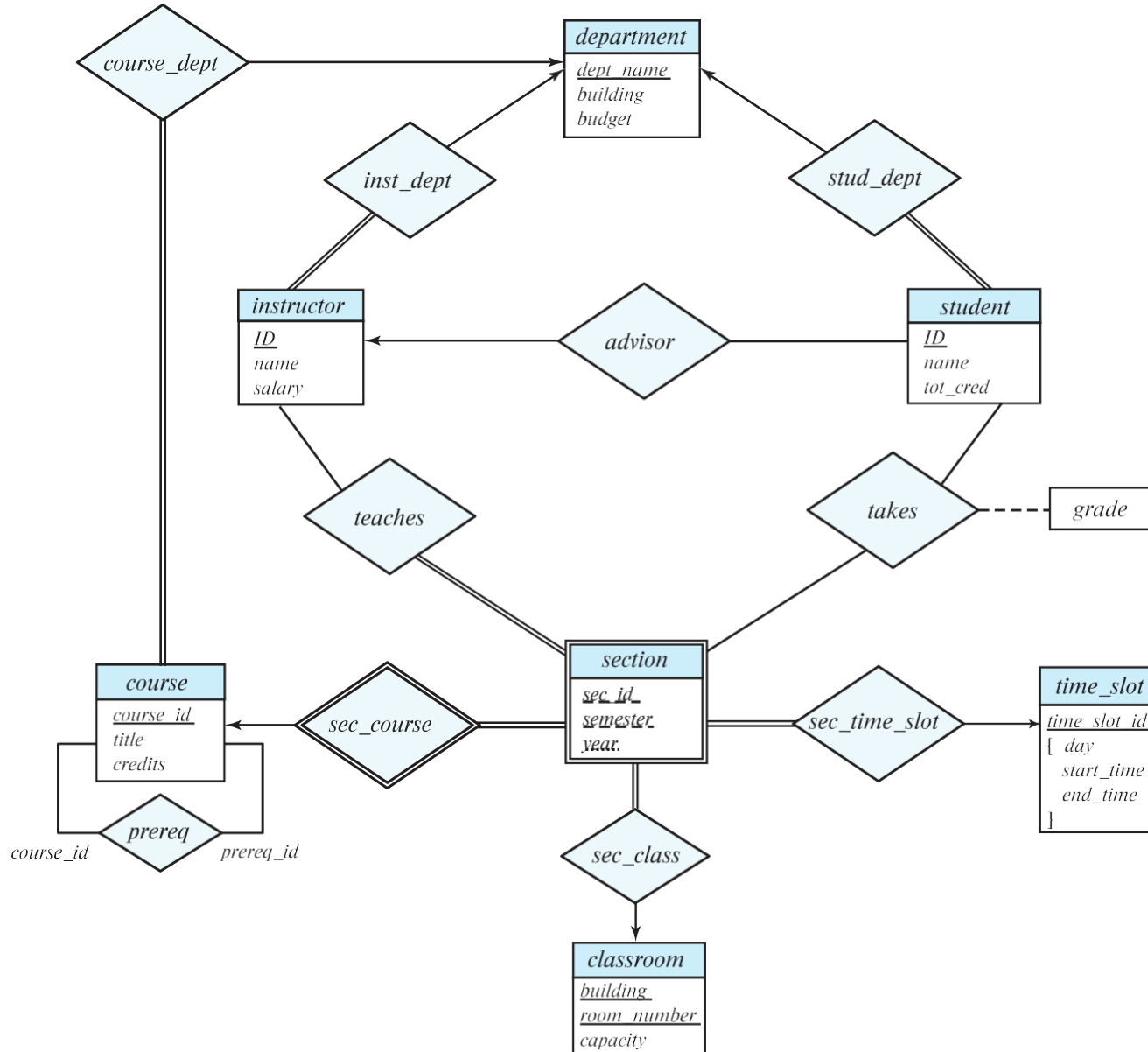
- Misalkan kita memiliki kumpulan entitas:
 - *siswa*, dengan atribut: *ID*, *nama*, *tot_cred*, *nama_departemen*
 - *departemen*, dengan atribut: *nama_departemen*, *gedung*, *anggaran*
- Kami memodelkan fakta bahwa setiap siswa memiliki departemen yang terkait dengan menggunakan himpunan hubungan *stud_dept*
- Atribut *dept_name* pada *siswa* di bawah ini mereplikasi informasi yang ada dalam hubungan dan karena itu mubazir
 - dan perlu dihilangkan.
- TETAPI: ketika mengonversi kembali ke tabel, dalam beberapa kasus, atribut akan diperkenalkan kembali, seperti yang akan kita lihat nanti.



(a) Incorrect use of attribute



Diagram E-R untuk Perusahaan





Reduksi ke Skema Relasi



Reduksi ke Skema Relasi

- Himpunan entitas dan himpunan relasi dapat dinyatakan secara seragam sebagai *skema relasi* yang merepresentasikan isi database.
- Basis data yang sesuai dengan diagram E-R dapat diwakili oleh kumpulan skema.
- Untuk setiap himpunan entitas dan himpunan relasi, terdapat skema unik yaitu menetapkan nama himpunan entitas atau himpunan hubungan yang sesuai.
- Setiap skema memiliki sejumlah kolom (umumnya sesuai dengan atribut), yang memiliki nama unik.



Mewakili Kumpulan

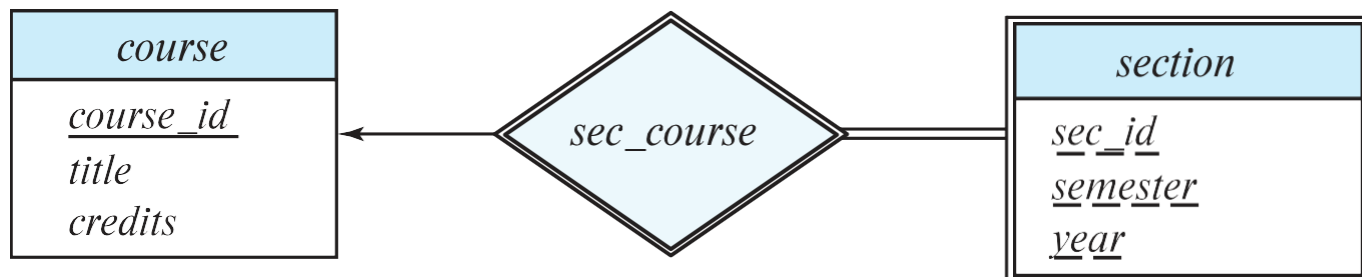
- Kumpulan entitas yang kuat direduksi menjadi skema dengan atribut yang sama

student(ID, nama, tot_cred)

- Himpunan entitas yang lemah menjadi sebuah tabel yang menyertakan kolom untuk kunci utama dari himpunan entitas yang kuat

bagian (id kuliah, id sek, sem, tahun)

- Contoh





Representasi Himpunan Entitas dengan Atribut

<i>instructor</i>
<u>ID</u>
<i>name</i>
<i>first_name</i>
<i>middle_initial</i>
<i>last_name</i>
<i>address</i>
<i>street</i>
<i>street_number</i>
<i>street_name</i>
<i>apt_number</i>
<i>city</i>
<i>state</i>
<i>zip</i>
{ <i>phone_number</i> }
<i>date_of_birth</i>
<i>age</i> ()

- Atribut komposit diratakan dengan membuat atribut terpisah untuk setiap atribut komponen
 - Contoh: diberikan himpunan entitas *instruktur* dengan atribut komposit *nama* dengan atribut komponen *nama_pertama* dan *nama_lengkap* skema yang sesuai dengan himpunan entitas memiliki dua atribut *nama_pertama* dan *nama_lengkap*
 - Awalan dihilangkan jika tidak ada ambiguitas (*nama_depan* bisa menjadi *nama_depan*)
- Mengabaikan atribut multivalued, skema instruktur yang diperluas adalah
 - *instruktur* (*ID*,
 nama_depan, *nama_tengah*, *nama_belakang*,
 nomor_jalan, *nama_jalan*,
 nomor_apartemen, *kota*, *negara*
 bagian, *kode_pos*,
 tanggal_lahir)



Representasi Himpunan Entitas dengan Atribut

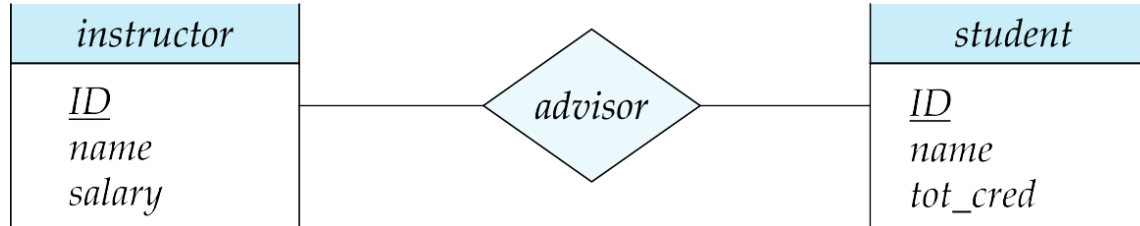
- Atribut multivariabel M dari sebuah entitas E diwakili oleh skema terpisah EM
- Skema EM memiliki atribut yang sesuai dengan kunci utama E dan sebuah atribut yang berhubungan dengan atribut multivariabel M
- Contoh: Atribut multivariabel *nomor_telepon instruktur* diwakili oleh sebuah skema:
 $inst_phone = (\underline{ID}, \underline{nomor_telepon})$
- Setiap nilai dari atribut multivariabel dipetakan ke tuple terpisah dari hubungan pada skema EM
 - Sebagai contoh, entitas *instruktur* dengan kunci utama 22222 dan nomor telepon 456-7890 dan 123-4567 dipetakan ke dua tupel:
(22222, 456-7890) dan (22222, 123-4567)



Merepresentasikan Himpunan

- Himpunan hubungan banyak-ke-banyak direpresentasikan sebagai skema dengan atribut untuk kunci utama dari dua himpunan entitas yang berpartisipasi, dan atribut deskriptif dari himpunan hubungan.
- Contoh: skema untuk *penasihat* rangkaian hubungan

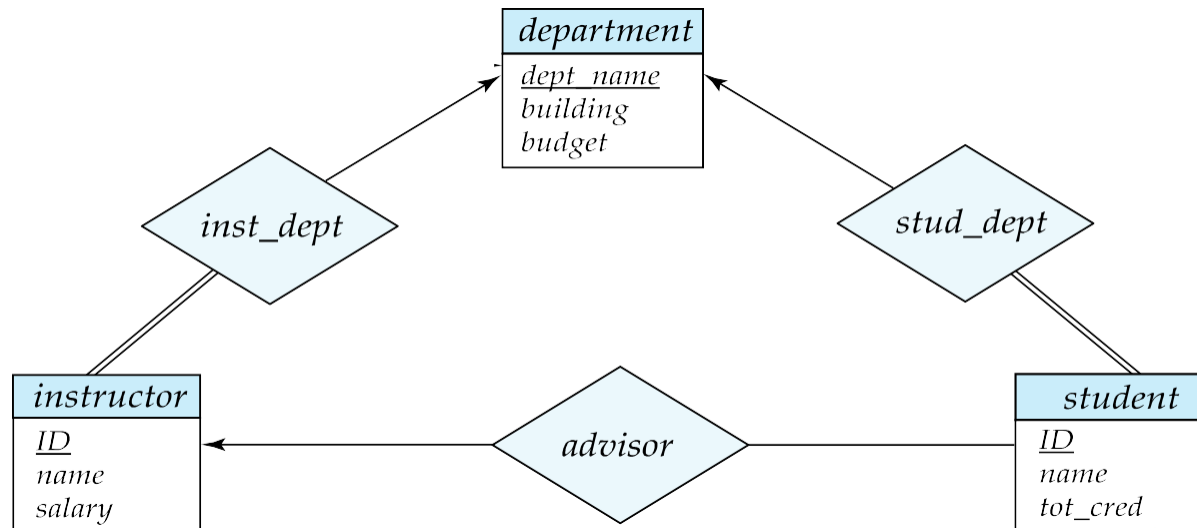
penasihat = (*s_id*, *i_id*)





Redundansi Skema

- Himpunan relasi banyak-ke-satu dan satu-ke-banyak yang total pada sisi banyak dapat direpresentasikan dengan menambahkan atribut tambahan pada sisi "banyak", yang berisi kunci utama pada sisi "satu"
- Contoh: Alih-alih membuat skema untuk himpunan relasi *inst_dept*, tambahkan atribut *dept_name* ke skema yang muncul dari set entitas *instruktur*
- Contoh





Redundansi Skema (Lanjutan)

- Untuk set hubungan satu-ke-satu, salah satu pihak dapat dipilih untuk bertindak sebagai sisi "banyak"
 - Artinya, atribut tambahan dapat ditambahkan ke salah satu tabel yang sesuai dengan dua set entitas
- Jika partisipasi *parsial* pada sisi "banyak", mengganti skema dengan atribut tambahan dalam skema yang sesuai dengan sisi "banyak" dapat menghasilkan nilai nol



Redundansi Skema (Lanjutan)

- Skema yang sesuai dengan himpunan hubungan yang menghubungkan himpunan entitas yang lemah untuk mengidentifikasi himpunan entitas yang kuat adalah berlebihan.
- Contoh: Skema *bagian* sudah berisi atribut yang akan muncul di skema *sec_course*





Fitur E-R yang Diperluas



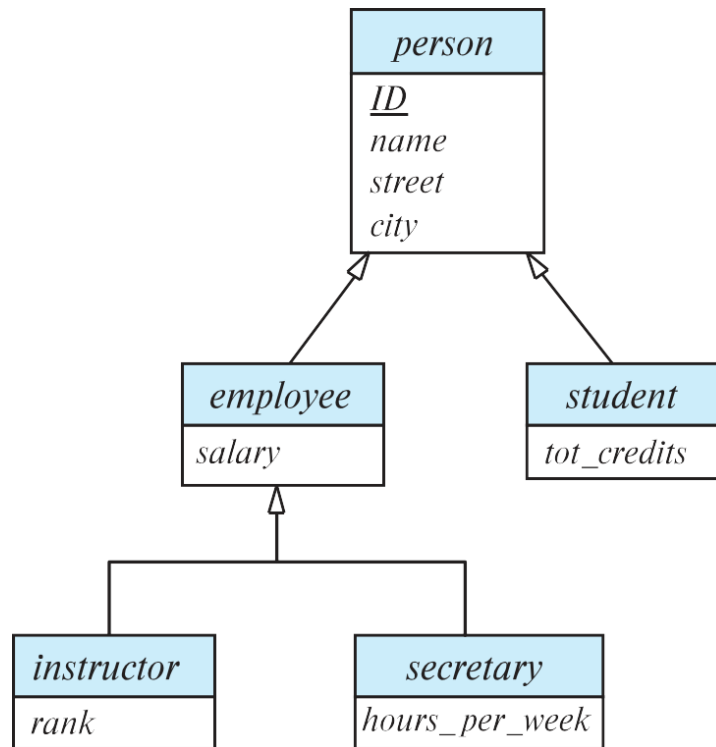
Spesialisasi

- Proses desain dari atas ke bawah; kami menetapkan sub-kelompok dalam himpunan entitas yang berbeda dari entitas lain dalam himpunan tersebut.
- Sub-kelompok ini menjadi himpunan entitas tingkat yang lebih rendah yang memiliki atribut atau berpartisipasi dalam hubungan yang tidak berlaku untuk himpunan entitas tingkat yang lebih tinggi.
- Digambarkan dengan komponen *segitiga* berlabel ISA (misalnya, *instruktur* "adalah" *orang*).
- **Pewarisan atribut** - himpunan entitas tingkat yang lebih rendah mewarisi semua atribut dan partisipasi hubungan dari himpunan entitas tingkat yang lebih tinggi yang ditautkan.



Contoh Spesialisasi

- **Tumpang tindih** - *karyawan* dan *siswa*
- **Terpisah** - *instruktur* dan *sekretaris*
- Total dan parsial





Merepresentasikan Spesialisasi melalui

- Metode 1:
 - Membentuk skema untuk entitas tingkat yang lebih tinggi
 - Membentuk skema untuk setiap set entitas tingkat bawah, termasuk kunci utama set entitas tingkat yang lebih tinggi dan atribut lokal

schema	attributes
person	ID, name, street, city
student	ID, tot_cred
employee	ID, salary

- Kelemahan: untuk mendapatkan informasi tentang, seorang *karyawan* membutuhkan akses ke dua relasi, yang sesuai dengan skema tingkat rendah dan yang sesuai dengan skema tingkat tinggi



Merepresentasikan Spesialisasi sebagai Skema

- Metode 2:
 - Membentuk skema untuk setiap set entitas dengan semua atribut lokal dan atribut yang diwariskan

schema	attributes
person	ID, name, street, city
student	ID, name, street, city, tot_cred
employee	ID, name, street, city, salary

- Kelemahan: *nama*, *jalan*, dan *kota* dapat disimpan secara berlebihan untuk orang-orang yang berstatus pelajar dan karyawan



Generalisasi

- **Proses desain** dari **bawah ke atas** - menggabungkan sejumlah kumpulan entitas yang memiliki fitur yang sama ke dalam kumpulan entitas yang lebih tinggi.
- Spesialisasi dan generalisasi adalah inversi sederhana satu sama lain; keduanya direpresentasikan dalam diagram E-R dengan cara yang sama.
- Istilah spesialisasi dan generalisasi digunakan secara bergantian.



Batasan kelengkapan

- **Batasan kelengkapan** - menentukan apakah suatu entitas dalam himpunan entitas tingkat yang lebih tinggi harus menjadi bagian dari setidaknya satu himpunan entitas tingkat yang lebih rendah dalam suatu generalisasi.
 - **total**: sebuah entitas harus menjadi bagian dari salah satu himpunan entitas tingkat yang lebih rendah
 - **parsial**: sebuah entitas tidak harus menjadi bagian dari salah satu himpunan entitas yang lebih rendah tingkatannya



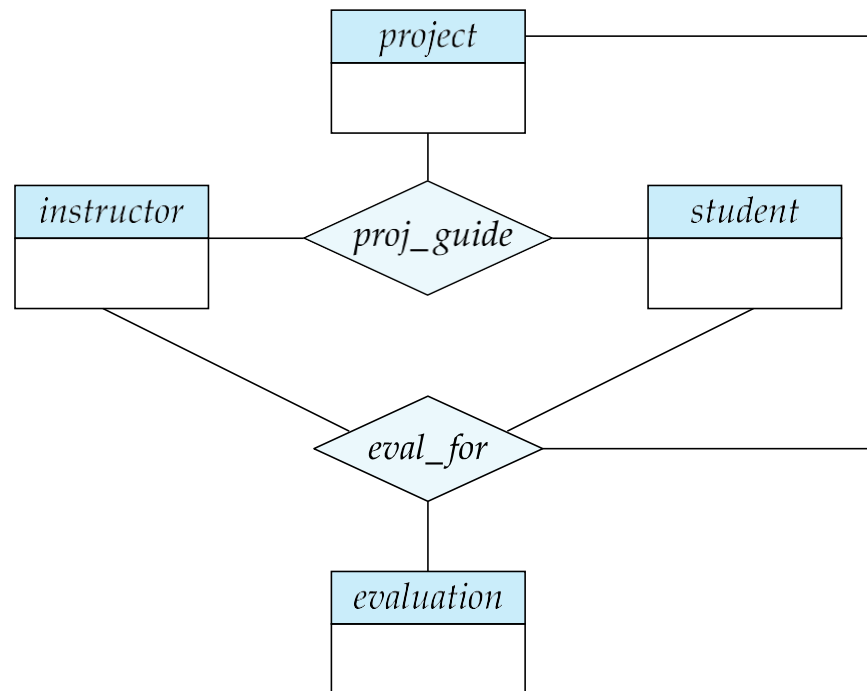
Batasan kelengkapan (Lanjutan)

- Generalisasi parsial adalah default.
- Kita dapat menentukan generalisasi total dalam diagram ER dengan menambahkan kata kunci **total** dalam diagram dan menggambar garis putus-putus dari kata kunci ke kepala panah berongga yang sesuai dengan yang berlaku (untuk generalisasi total), atau ke set kepala panah berongga yang berlaku (untuk generalisasi yang tumpang tindih).
- Generalisasi *mahasiswa* adalah total: Semua entitas mahasiswa haruslah mahasiswa pascasarjana atau sarjana. Karena himpunan entitas tingkat yang lebih tinggi yang diperoleh melalui generalisasi umumnya hanya terdiri dari entitas-entitas di himpunan entitas tingkat yang lebih rendah, batasan kelengkapan untuk himpunan entitas tingkat yang lebih tinggi yang digeneralisasi biasanya adalah total



Agregasi

- Perhatikan hubungan terner *proj_guide*, yang telah kita lihat sebelumnya
- Misalkan kita ingin merekam evaluasi seorang siswa oleh seorang pemandu dalam sebuah proyek





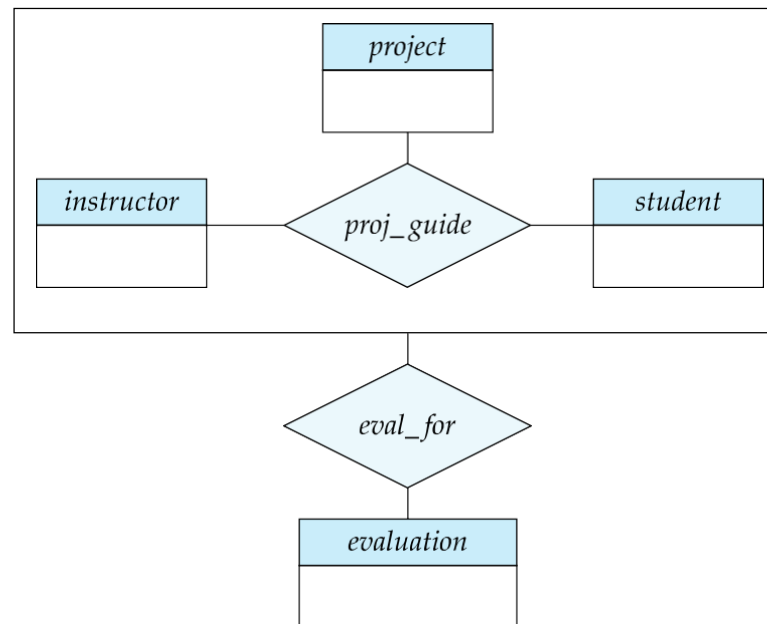
Agregasi (Lanjutan)

- Himpunan relasi *eval_for* dan *proj_guide* merepresentasikan tumpang tindih informasi
 - Setiap relasi *eval_for* berhubungan dengan relasi *proj_guide*
 - Namun, beberapa hubungan *proj_guide* mungkin tidak sesuai dengan *eval_untuk* hubungan
 - Jadi kita tidak bisa membuang hubungan *proj_guide*
- Menghilangkan redundansi ini melalui *agregasi*
 - Perlakukan hubungan sebagai entitas abstrak
 - Memungkinkan hubungan antar hubungan
 - Abstraksi hubungan menjadi entitas baru



Agregasi (Lanjutan)

- Hilangkan redundansi ini melalui *agregasi* tanpa memperkenalkan redundansi, seperti yang ditunjukkan oleh diagram berikut:
 - Seorang siswa dipandu oleh instruktur tertentu pada proyek tertentu
 - Kombinasi siswa, instruktur, dan proyek mungkin memiliki evaluasi terkait





Reduksi ke Skema Relasional

- Untuk merepresentasikan agregasi, buat skema yang berisi
 - Kunci utama dari hubungan gabungan,
 - Kunci utama dari set entitas terkait
 - Setiap atribut deskriptif
- Dalam contoh kita:
 - Skema *eval_for* adalah:
$$eval_for(s_ID, project_id, i_ID, evaluation_id)$$
 - Skema *proj_guide* adalah berlebihan.

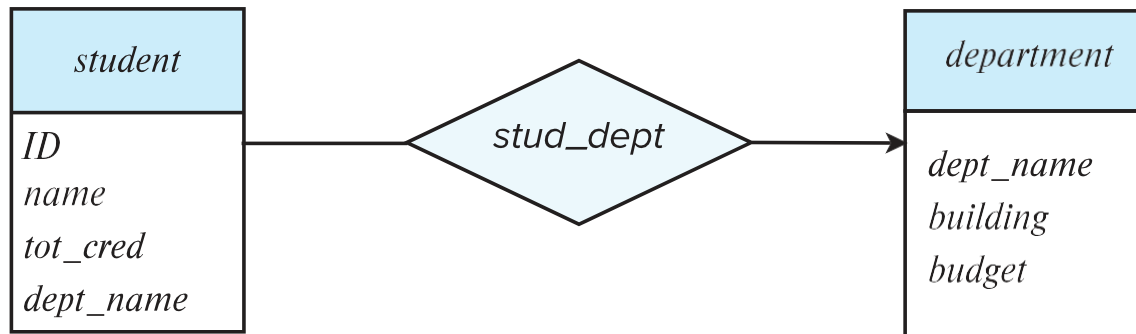


Masalah Desain

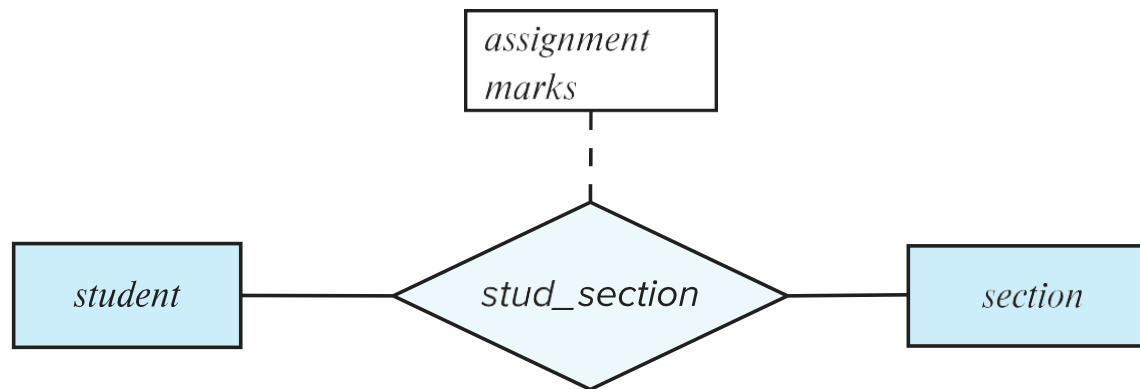


Kesalahan Umum dalam Diagram E-R

- Contoh diagram E-R yang salah



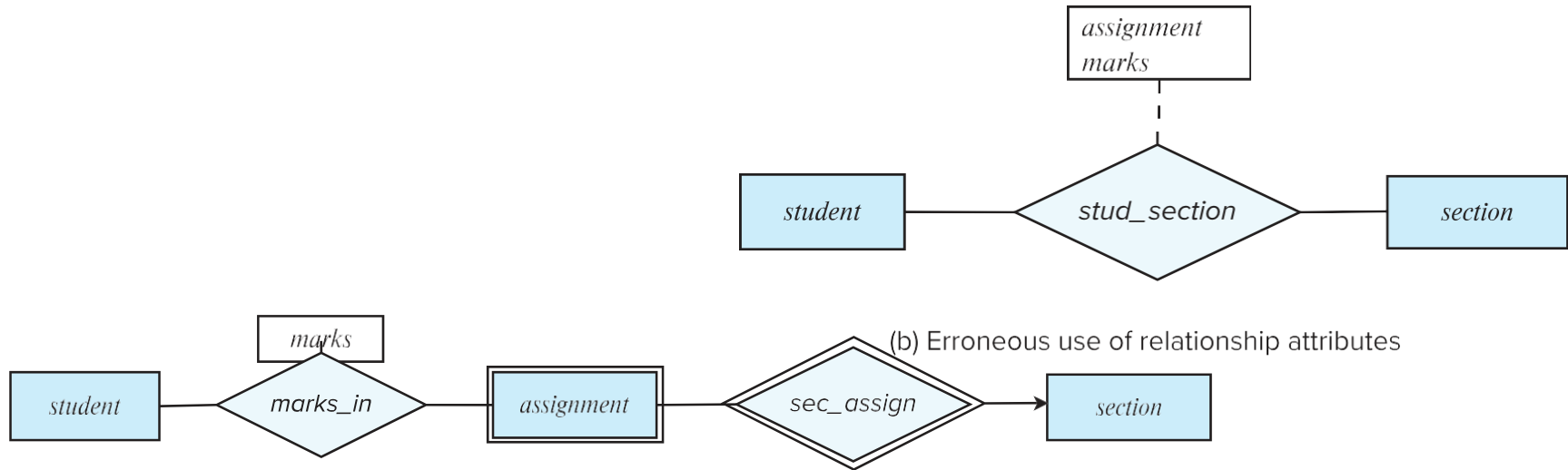
(a) Incorrect use of attribute



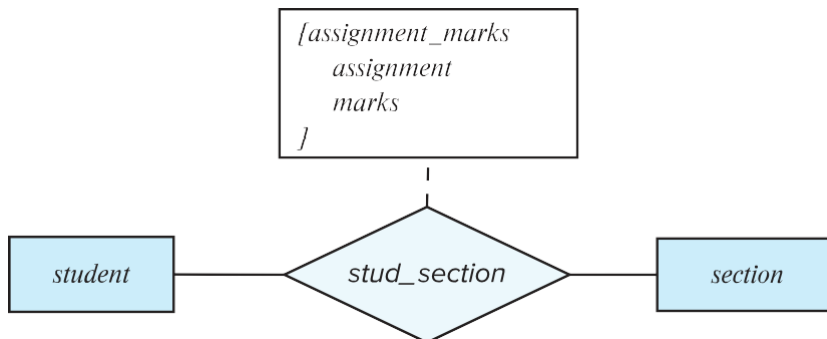
(b) Erroneous use of relationship attributes



Kesalahan Umum dalam Diagram E-R (Lanjutan)



(c) Correct alternative to erroneous E-R diagram (b)

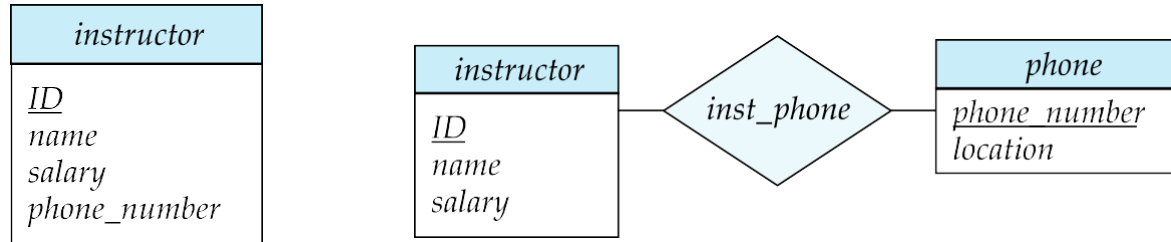


(d) Correct alternative to erroneous E-R diagram (b)



Entitas vs Atribut

- Penggunaan set entitas vs. atribut



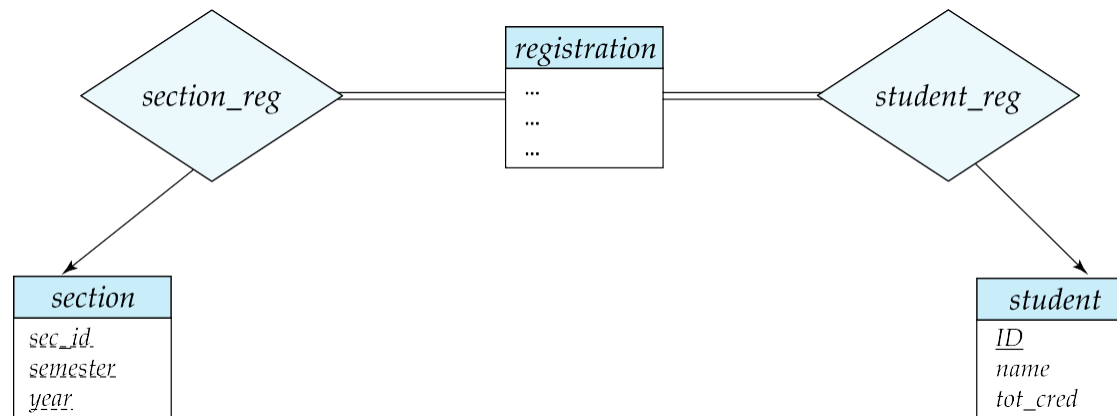
- Penggunaan telepon sebagai entitas memungkinkan informasi tambahan tentang nomor telepon (ditambah beberapa nomor telepon)



Entitas vs. kumpulan

- **Penggunaan kumpulan entitas vs kumpulan relasi**

Pedoman yang mungkin adalah untuk menetapkan set hubungan untuk menggambarkan tindakan yang terjadi di antara entitas



- **Penempatan atribut hubungan**

Misalnya, atribut tanggal sebagai atribut penasihat atau sebagai atribut siswa



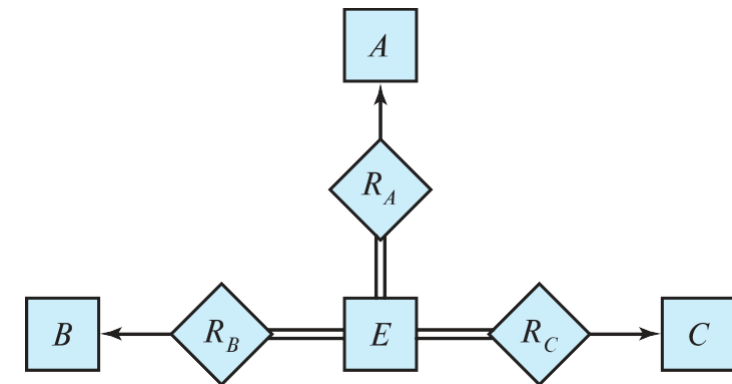
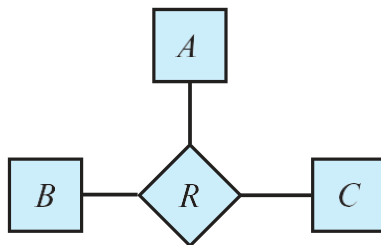
Hubungan Biner Vs. Non-Biner

- Meskipun dimungkinkan untuk mengganti set hubungan non-biner (n -ary, untuk $n > 2$) dengan sejumlah set hubungan biner yang berbeda, set hubungan n -ary menunjukkan dengan lebih jelas bahwa beberapa entitas berpartisipasi dalam satu hubungan.
- Beberapa hubungan yang tampaknya non-biner mungkin lebih baik diwakili dengan menggunakan hubungan biner
 - Sebagai contoh, hubungan *orang tua* terner, yang menghubungkan seorang anak dengan ayah dan ibunya, paling baik digantikan oleh dua hubungan biner, *ayah* dan *ibu*
 - Menggunakan dua hubungan biner memungkinkan informasi parsial (misalnya, hanya ibu yang diketahui)
 - Tetapi ada beberapa hubungan yang secara alami bersifat non-biner
 - Contoh: *proj_guide*



Mengonversi Hubungan Non-Biner ke Bentuk Biner

- Secara umum, setiap hubungan non-biner dapat direpresentasikan menggunakan hubungan biner dengan membuat himpunan entitas buatan.
 - Mengganti R di antara himpunan entitas A , B dan C dengan himpunan entitas E , dan tiga set hubungan:
 - R_A , menghubungkan E dan A
 - R_B , menghubungkan E dan B
 - R_C , menghubungkan E dan C
 - Buat atribut pengenal untuk E dan tambahkan atribut R ke E
 - Untuk setiap relasi (a_i, b_i, c_i) dalam R , buatlah
 - entitas baru e_i dalam himpunan entitas E
 - tambahkan (e_i, a_i) ke R_A
 - tambahkan (e_i, b_i) ke R_B
 - tambahkan (e_i, c_i) ke R_C





Mengonversi Hubungan Non-Biner

- Juga perlu menerjemahkan batasan
 - Menerjemahkan semua kendala mungkin tidak dapat dilakukan
 - Mungkin ada beberapa contoh dalam skema yang diterjemahkan yang tidak dapat berhubungan dengan contoh R
 - Latihan: *tambahkan batasan pada relasi $_{RA}$, $_{RB}$ dan $_{RC}$* untuk memastikan bahwa entitas yang baru dibuat berhubungan dengan tepat satu entitas di masing-masing himpunan entitas A , B dan C
 - Kita dapat menghindari pembuatan atribut pengenalan dengan membuat E sebagai himpunan entitas lemah (dijelaskan segera) yang diidentifikasi oleh tiga himpunan relasi

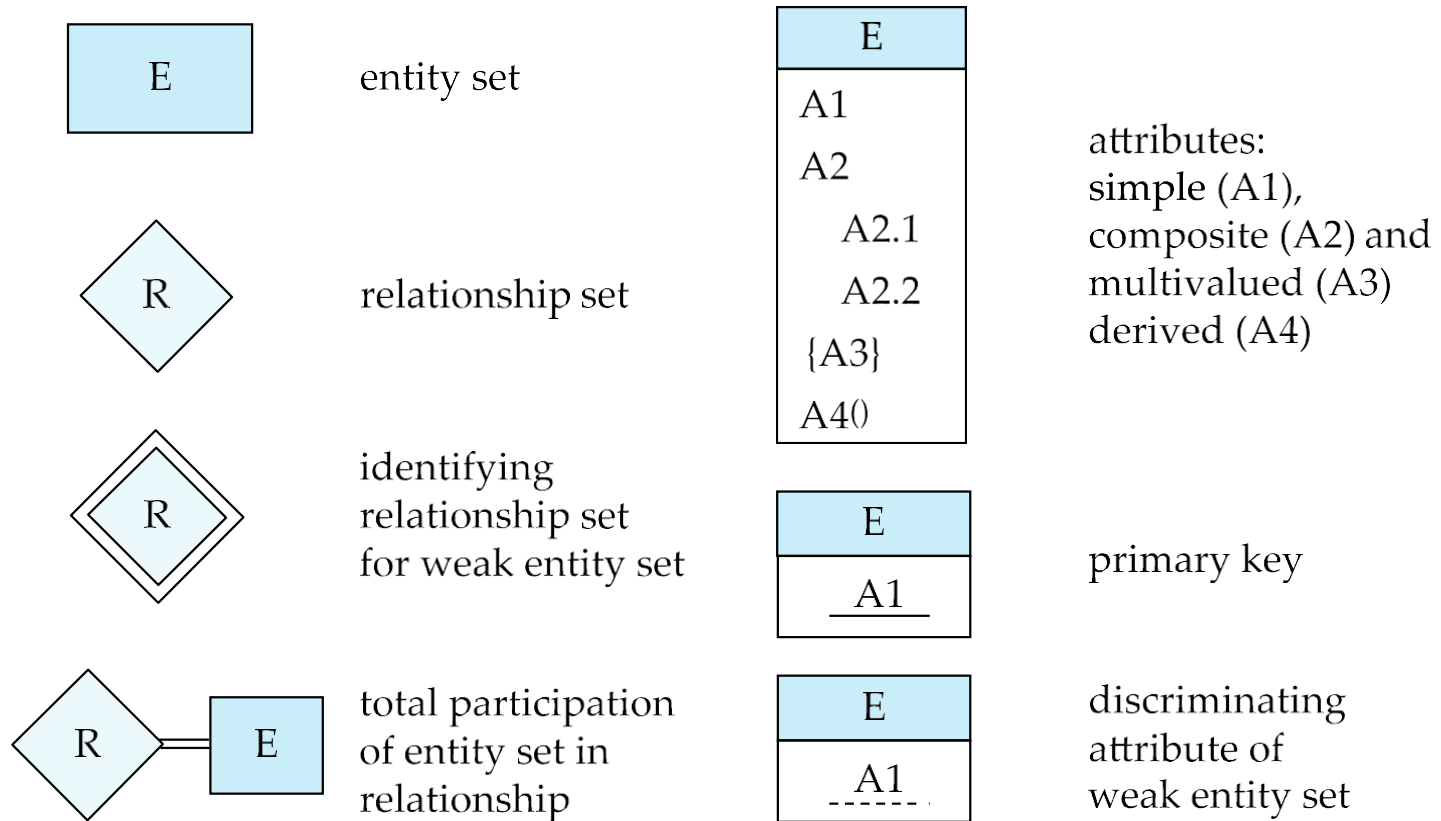


Keputusan Desain E-

- Penggunaan atribut atau entitas yang ditetapkan untuk mewakili suatu objek.
- Apakah sebuah konsep dunia nyata paling baik diekspresikan dengan himpunan entitas atau himpunan relasi.
- Penggunaan hubungan terner versus sepasang hubungan biner.
- Penggunaan set entitas yang kuat atau lemah.
- Penggunaan spesialisasi/ generalisasi - berkontribusi pada modularitas dalam desain.
- Penggunaan agregasi - dapat memperlakukan kumpulan entitas agregat sebagai satu kesatuan tanpa memperhatikan detail struktur internalnya.

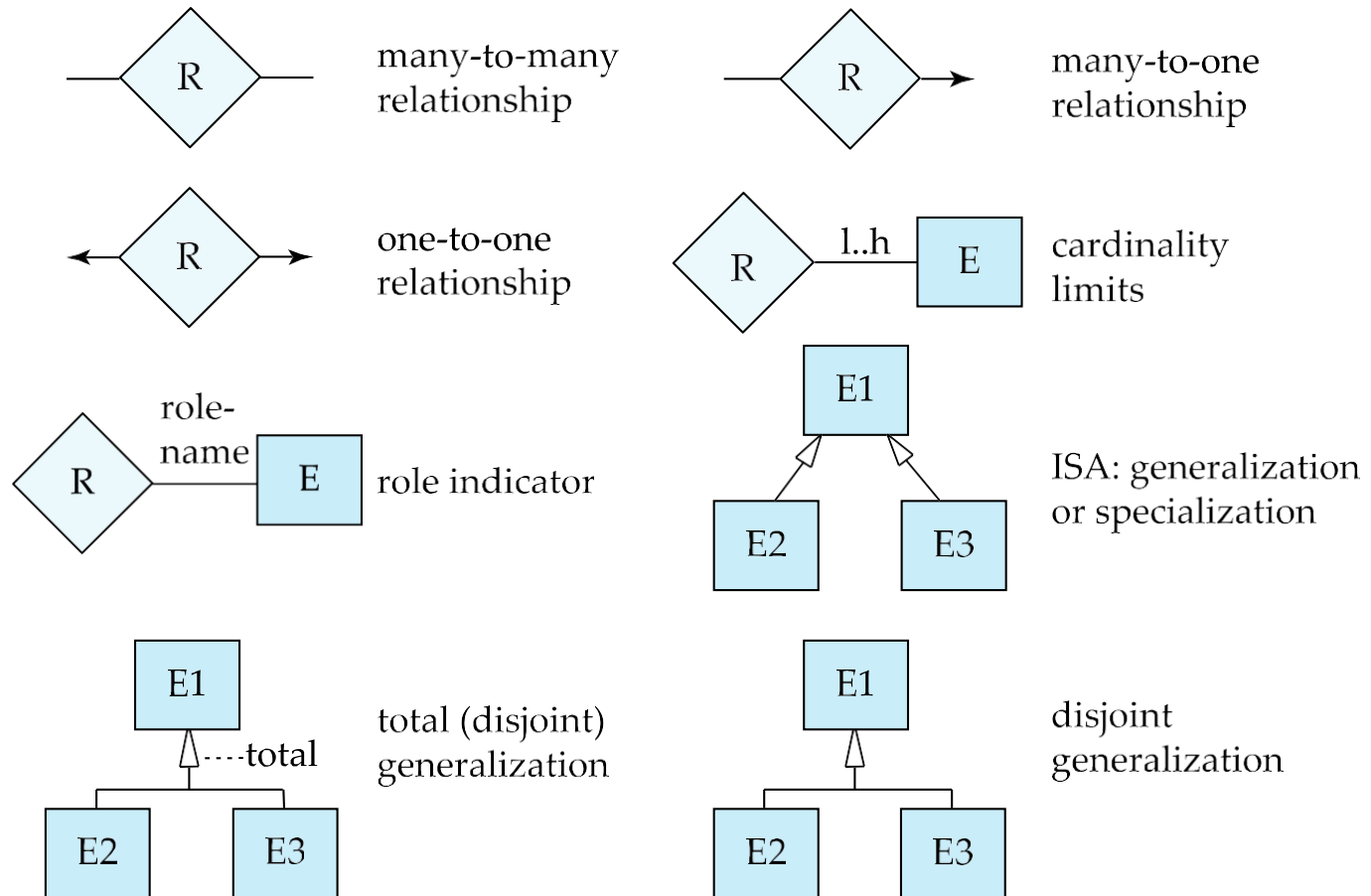


Ringkasan Simbol yang Digunakan dalam





Simbol yang Digunakan dalam Notasi

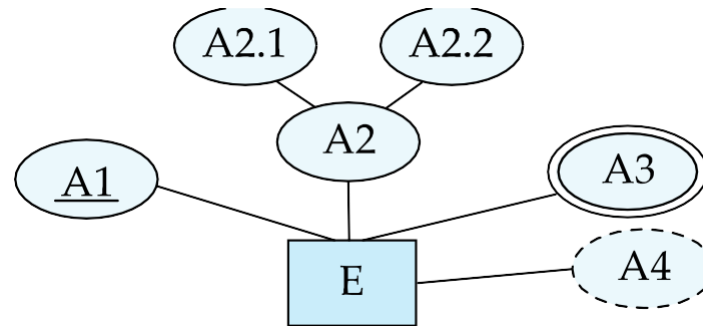




Notasi UGD Alternatif

- Chen, IDE1FX, ...

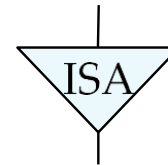
entity set E with
simple attribute A1,
composite attribute A2,
multivalued attribute A3,
derived attribute A4,
and primary key A1



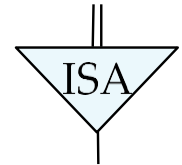
weak entity set



generalization



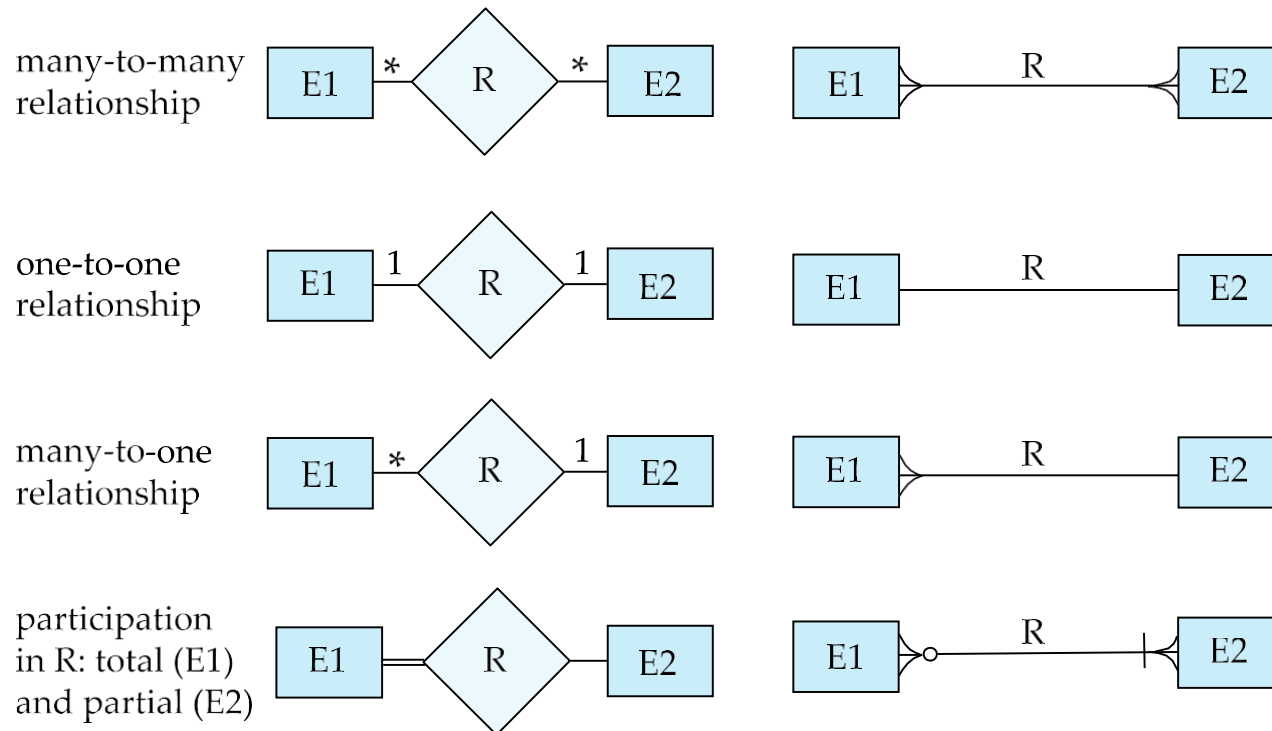
total
generalization





Notasi UGD Alternatif

ChenIDE1FX (Notasi kaki gagak)





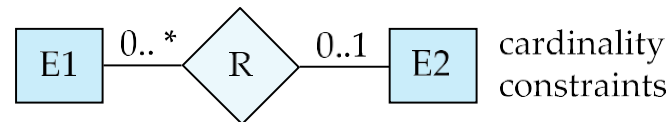
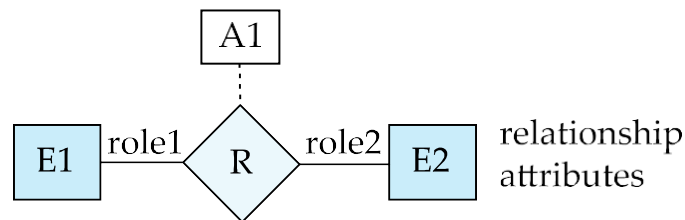
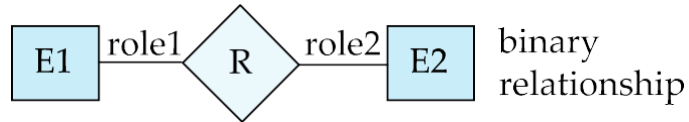
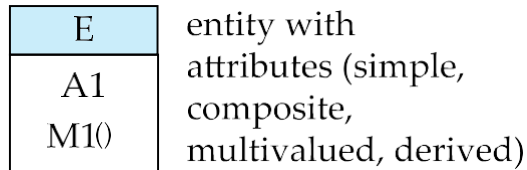
UML

- **UML:** Bahasa Pemodelan Terpadu
- UML memiliki banyak komponen untuk memodelkan secara grafis berbagai aspek dari keseluruhan sistem perangkat lunak
- Diagram Kelas UML sesuai dengan Diagram E-R, tetapi ada beberapa perbedaan.

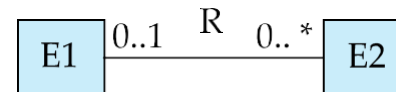
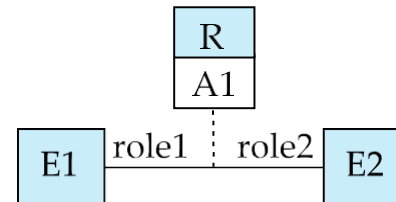
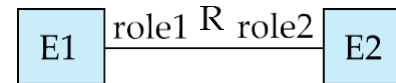
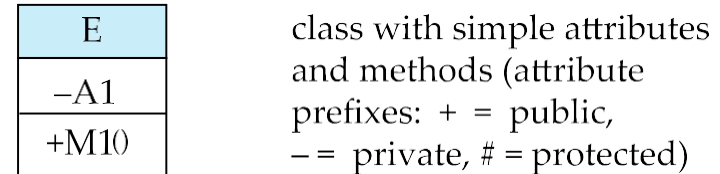


Diagram Kelas ER vs UML

ER Diagram Notation



Equivalent in UML

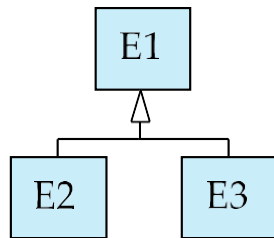
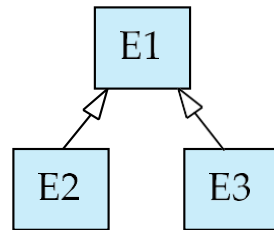
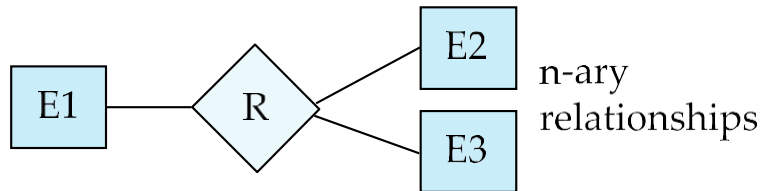


* Perhatikan pembalikan posisi dalam penggambaran batasan kardinalitas



Diagram Kelas ER vs UML

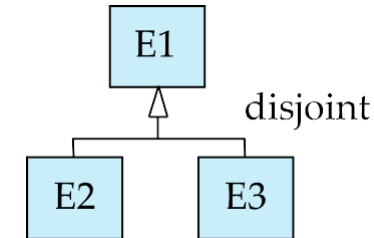
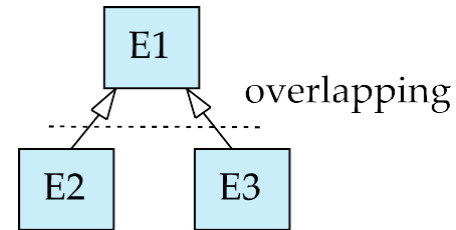
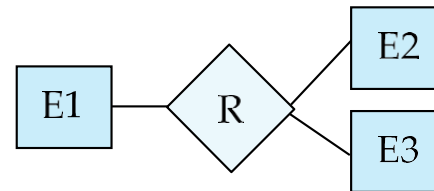
Notasi Diagram ER



overlapping
generalization

disjoint
generalization

Setara dengan UML



overlapping

disjoint

- * Generalisasi dapat menggunakan panah yang digabungkan atau terpisah yang tidak terpisah/tumpang tindih



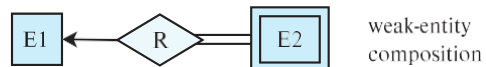
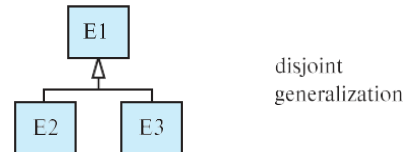
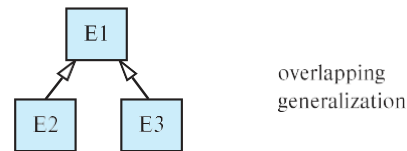
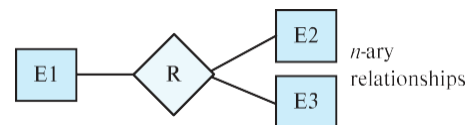
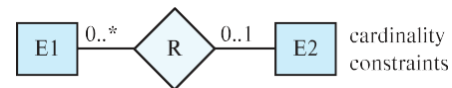
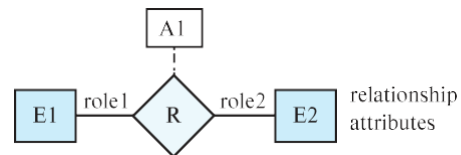
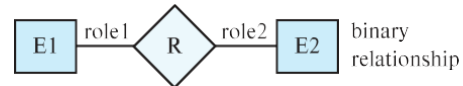
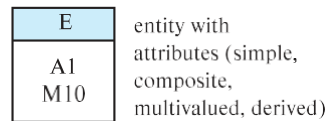
Diagram Kelas UML

- Himpunan hubungan biner direpresentasikan dalam UML hanya dengan menggambar garis yang menghubungkan himpunan entitas. Nama himpunan hubungan ditulis berdekatan dengan garis tersebut.
- Peran yang dimainkan oleh himpunan entitas dalam himpunan relasi juga dapat ditentukan dengan menulis nama peran pada baris, berdekatan dengan himpunan entitas.
- Nama himpunan hubungan dapat dituliskan dalam sebuah kotak, bersama dengan atribut himpunan hubungan, dan kotak tersebut dihubungkan, menggunakan garis putus-putus, ke garis yang menggambarkan himpunan hubungan.

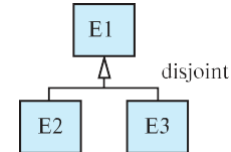
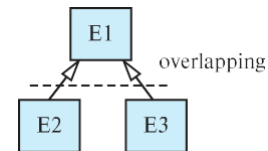
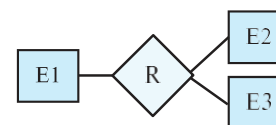
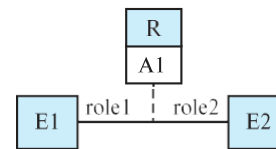
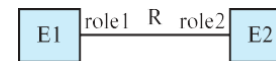
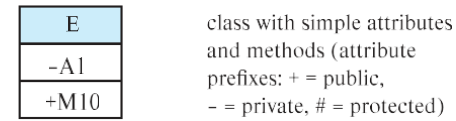


Diagram Kelas ER vs UML

ER Diagram Notation



Equivalent in UML





Aspek Lain dari Desain Basis Data

- Persyaratan Fungsional
- Aliran Data, Alur Kerja
- Evolusi Skema



Akhir Bab 6