

BAB 4. ENTITY RELATIONSHIP DIAGRAM

4.1. Tujuan Instruksional

A. Tujuan Instruksional Umum

Mahasiswa memahami konsep ERD

B. Tujuan Instruksional Khusus

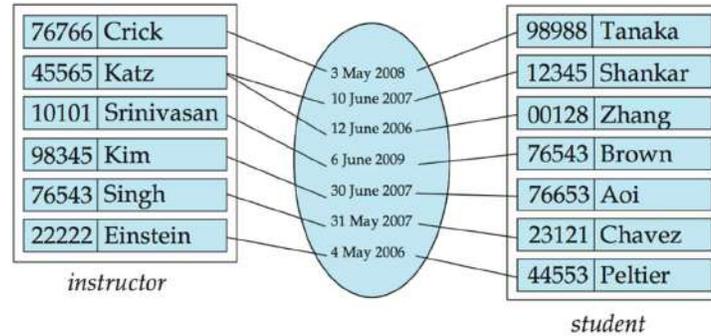
Mahasiswa dapat menggambarkan ERD

4.2. Model ER - Pemodelan Database

- a) Model ER dikembangkan untuk memfasilitasi desain database dengan memungkinkan spesifikasi skema perusahaan yang mewakili struktur logis keseluruhan dari database.
- b) Model ER sangat berguna dalam memetakan makna dan interaksi perusahaan dunia nyata ke skema konseptual. Karena kegunaan ini, banyak alat perancangan basis data menarik konsep dari model ER.
- c) Model data ER menggunakan tiga konsep dasar:
 1. set entitas,
 2. hubungan set,
 3. atribut.
- d) Model ER juga memiliki representasi diagram yang terkait, diagram ER, yang dapat mengekspresikan struktur logis keseluruhan dari sebuah database secara grafis.

4.2.1. Set Entitas

- a. Entitas adalah objek yang ada dan dapat dibedakan dari objek lain.
Contoh: orang tertentu, perusahaan, acara, pabrik
- b. Kumpulan entitas adalah sekumpulan entitas dengan jenis yang sama yang berbagi properti yang sama.
Contoh: kumpulan semua orang, perusahaan, pohon, hari libur
- c. Suatu entitas diwakili oleh seperangkat atribut; yaitu, properti deskriptif yang dimiliki oleh semua anggota kumpulan entitas.
Contoh: instructor = (ID, name, street, city, salary)
 1. course= (course_id, title, credits)
- d. Bagian dari atribut membentuk kunci utama dari kumpulan entitas; yaitu, mengidentifikasi setiap anggota kumpulan secara unik.



4.2.3. Degree of a Relationship Set

a) Hubungan biner

- melibatkan dua set entitas (atau gelar dua).
- sebagian besar himpunan relasi dalam sistem basis data adalah biner.

b) Hubungan antara lebih dari dua set entitas jarang terjadi. Sebagian besar hubungan bersifat biner. (Lebih lanjut tentang ini nanti.)

Contoh: siswa bekerja pada proyek-proyek penelitian di bawah bimbingan seorang instruktur.

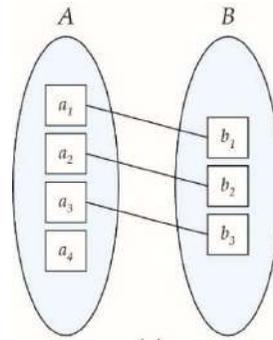
hubungan `proj_guide` adalah hubungan terner antara instruktur, siswa, dan proyek.

4.2.4. Mapping Cardinality Constraints

- a) Ekspresikan jumlah entitas tempat entitas lain dapat dikaitkan melalui himpunan relasi.
- b) Paling berguna dalam menggambarkan himpunan relasi biner.
- c) Untuk hubungan biner, pengaturan kardinalitas pemetaan harus salah satu dari jenis berikut:

a. One to one (Satu ke satu)

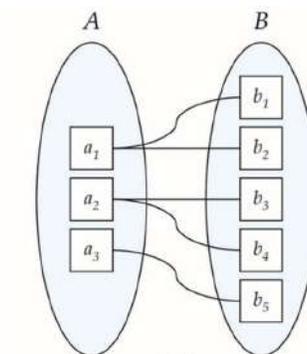
Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan begitu sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada entitas A.



One to one

b. One to many (Satu ke banyak)

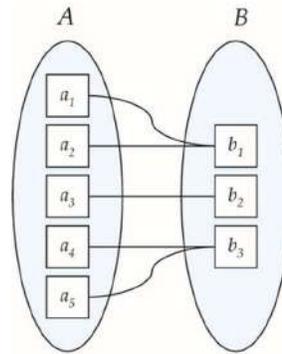
Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan tidak sebaliknya dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan entitas A.



One to many

c. Many to one (Banyak ke Satu)

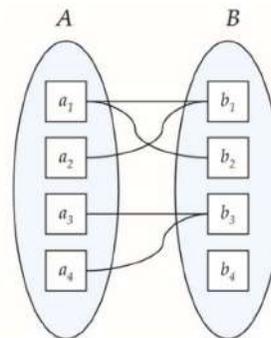
Setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan tidak sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Many to one

d. Many to many(Banyak ke Banyak)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A.



Many to many

4.2.5. Atribut Kompleks

- Simple and composite attributes.
- Single-valued and multivalued attributes

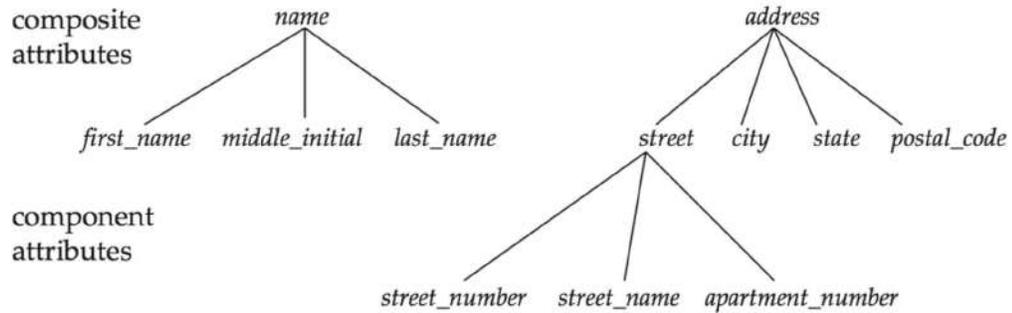
Example: multivalued attribute: *phone_numbers*

- Derived **attributes**

Dapat dihitung dari atribut lain

Example: age, given date_of_birth

- Domain – **himpunan nilai yang diizinkan untuk setiap atribut**



Gambar : Composite Attributes

4.2.6. Atribut Redundant

- a) Misalkan kita memiliki set entitas:
 - instructor, with attributes: ID, name, dept_name, salary
 - department, with attributes: dept_name, building, budget
- b) mencontohkan fakta bahwa setiap instruktur memiliki departemen terkait menggunakan hubungan menetapkan inst_dept.
- c) Atribut dept_name muncul di kedua set entitas. Karena ini adalah kunci utama untuk departemen set entitas, itu mereplikasi informasi yang ada dalam hubungan dan karena itu berlebihan dalam mengatur instruktur entitas dan perlu dihapus.
- d) TAPI: ketika mengkonversi kembali ke tabel, dalam beberapa kasus atribut akan diperkenalkan kembali, seperti yang akan kita lihat nanti.

4.2.7. Entitas Lemah

- a) Pertimbangkan entitas bagian, yang diidentifikasi secara unik oleh course_id, semester, year, dan sec_id.
- b) Jelas, entitas bagian terkait dengan entitas saja. Misalkan kita membuat hubungan mengatur sec_course antara bagian set entitas dan tentu saja.
- c) Perhatikan bahwa informasi di sec_course redundan, karena bagian sudah memiliki atribut course_id, yang mengidentifikasi kursus yang terkait dengan bagian tersebut.
- d) Salah satu pilihan untuk menangani redundansi ini adalah menyingkirkan hubungan sec_course; namun, dengan melakukannya hubungan antara bagian dan jalur menjadi tersirat dalam suatu atribut, yang tidak diinginkan.
- e) Cara alternatif untuk menangani redundansi ini adalah dengan tidak menyimpan atribut course_id di entitas bagian dan hanya menyimpan atribut yang tersisa

section_id, year, dan semester. Namun, bagian kumpulan entitas tidak memiliki atribut yang cukup untuk mengidentifikasi entitas bagian tertentu secara unik; Meskipun masing-masing bagian entitas berbeda, bagian untuk kursus yang berbeda dapat berbagi section_id, tahun, dan semester yang sama.

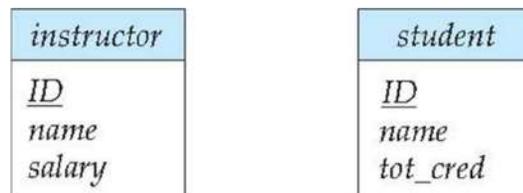
- f) Untuk mengatasi masalah ini, kami memperlakukan hubungan sec_course sebagai hubungan khusus yang memberikan informasi tambahan, dalam hal ini, course_id, yang diperlukan untuk mengidentifikasi bagian entitas secara unik.
- g) Gagasan dari entitas yang lemah mengatur meresmikan intuisi di atas. Satu set entitas lemah adalah entitas yang keberadaannya bergantung pada entitas lain, yang disebut entitas identifikasinya; alih-alih mengasosiasikan kunci utama dengan entitas yang lemah, kami menggunakan entitas pengidentifikasi, bersama dengan atribut tambahan yang disebut diskriminator untuk mengidentifikasi entitas lemah secara unik. Satu set entitas yang bukan kumpulan entitas yang lemah disebut kumpulan entitas yang kuat.
- h) Setiap entitas lemah harus dikaitkan dengan entitas pengidentifikasi; yaitu, kumpulan entitas yang lemah dikatakan keberadaan tergantung pada set entitas yang mengidentifikasi. Set entitas yang mengidentifikasi dikatakan memiliki entitas lemah yang ditetapkan yang diidentifikasinya. Hubungan yang menghubungkan entitas lemah yang ditetapkan dengan himpunan entitas pengidentifikasi disebut hubungan identifikasinya.
- i) Perhatikan bahwa skema relasional yang akhirnya kita buat dari bagian kumpulan entitas memang memiliki atribut course_id, untuk alasan yang akan menjadi jelas nanti, meskipun kita telah menjatuhkan course_id atribut dari bagian set entitas.

4.3. E-R Diagrams

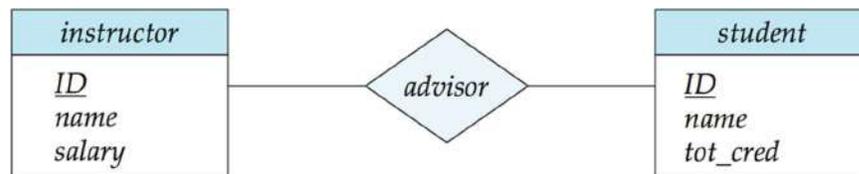
- a) ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan
- b) ERD digunakan untuk memodelkan struktur data dan hubungan antar data
- c) Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. Dengan ERD kita mencoba menjawab pertanyaan seperti:
 - o Data apa yang diperlukan ?
 - o Bagaimana data yang satu berhubungan dengan yang lain ?

4.3.1. Entity

- a) Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat.
- b) Individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain.
- c) Entitas dapat direpresentasikan secara grafis sebagai berikut:
 - o Rectangles mewakili set entitas.
 - o Atribut yang tercantum di dalam persegi panjang entitas
 - o Garis bawah menunjukkan atribut kunci primer

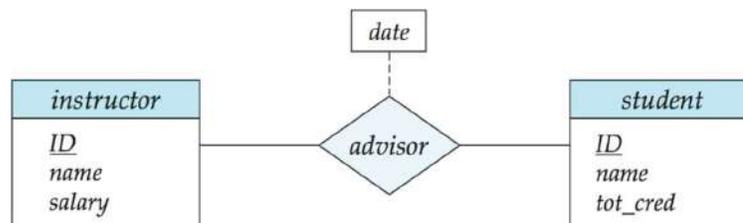


Gambar : contoh entitas

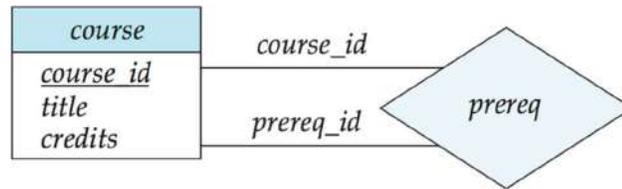


Gambar : hubungan entitas

Keterangan : bentuk Diamond mewakili set hubungan



Gambar : Hubungan entitas dengan atribut



Gambar : roles

Keterangan :

- Kumpulan entitas dari suatu hubungan tidak perlu berbeda
Setiap kemunculan set entitas memainkan "peran" dalam hubungan
- Label "course_id" dan "prereq_id" disebut sebagai peran.



Gambar : Cardinality Constraints

Keterangan :

- Kami mengungkapkan batasan kardinalitas dengan menggambar garis yang diarahkan (\rightarrow), menandakan "satu," atau garis tidak terarah (-), menandakan "banyak," antara himpunan relasi dan kumpulan entitas.
- Hubungan satu-ke-satu antara seorang instruktur dan seorang siswa:
 - Seorang siswa dikaitkan dengan paling banyak satu instruktur melalui penasihat hubungan
 - Seorang siswa dikaitkan dengan paling banyak satu departemen melalui stud_dept



Gambar : Hubungan One-to-Many

Keterangan :

- hubungan satu-ke-banyak antara seorang instruktur dan seorang siswa.

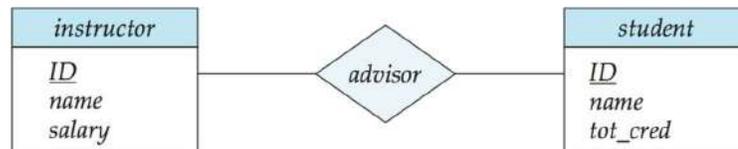
- instruktur dikaitkan dengan beberapa siswa (termasuk 0) melalui penasihat
- seorang siswa dikaitkan dengan paling banyak satu instruktur melalui penasihat



Gambar : Hubungan Many-to-One

Keterangan :

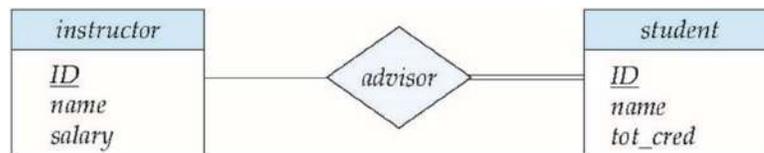
- Dalam hubungan banyak-ke-satu antara seorang instruktur dan seorang siswa,
 - instruktur dikaitkan dengan paling banyak satu siswa melalui penasihat,
 - dan seorang siswa dikaitkan dengan beberapa (termasuk 0) instruktur melalui penasihat



Gambar : Hubungan Many-to-Many

Keterangan :

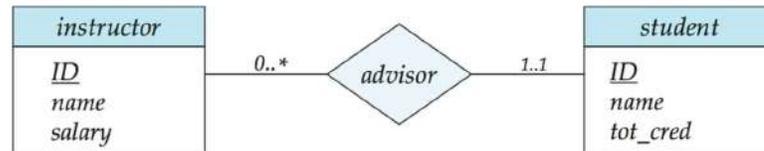
- Seorang instruktur dikaitkan dengan beberapa siswa (mungkin 0) melalui penasihat
- Seorang siswa dikaitkan dengan beberapa instruktur (mungkin 0) melalui penasihat



Gambar : Total and Partial Participation

Keterangan :

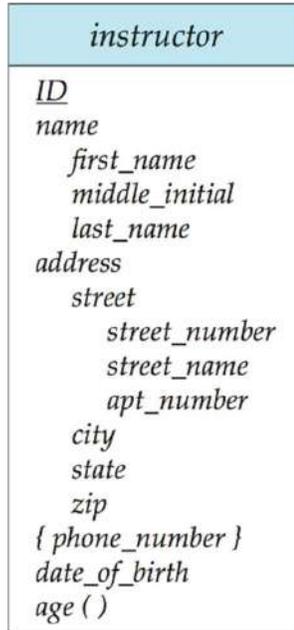
- a. Partisipasi total (ditunjukkan oleh garis ganda): setiap entitas dalam kumpulan entitas berpartisipasi dalam setidaknya satu hubungan dalam himpunan relasi.
- b. Partisipasi siswa dalam hubungan penasihat adalah total
 - setiap siswa harus memiliki instruktur yang terkait
- c. Partial participation: beberapa entitas tidak dapat berpartisipasi dalam hubungan apa pun dalam himpunan relasi
 - Contoh: partisipasi instruktur dalam penasihat bersifat parsial



Gambar : Notation for Expressing More Complex Constraints

Keterangan :

- a. Garis mungkin memiliki kardinalitas minimum dan maksimum terkait, ditunjukkan dalam bentuk l..h, di mana l adalah minimum dan h kardinalitas maksimum.
 - Nilai minimum 1 menunjukkan total partisipasi.
 - Nilai maksimum 1 menunjukkan bahwa entitas berpartisipasi dalam paling banyak satu hubungan
 - Nilai maksimum * menunjukkan tidak ada batasan.
- b. Instruktur dapat memberi saran kepada 0 atau lebih siswa. Seorang siswa harus memiliki 1 penasihat; tidak dapat memiliki banyak penasihat.



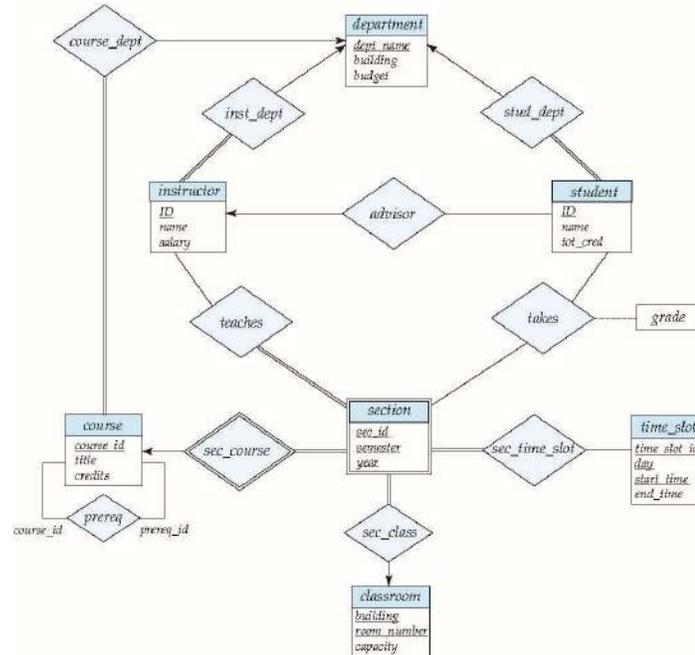
Gambar : Notation to Express Entity with Complex Attributes

4.3.2. Mengekspresikan Entitas Lemah

- Dalam diagram E-R, himpunan entitas yang lemah digambarkan melalui persegi panjang ganda.
- Kami menggarisbawahi diskriminator dari entitas lemah yang ditetapkan dengan garis putus-putus.
- Hubungan yang mengatur menghubungkan entitas lemah yang ditetapkan ke kumpulan entitas kuat yang mengidentifikasi dilukiskan oleh berlian ganda.
- Kunci utama untuk bagian - (*course_id*, *sec_id*, *semester*, *year*).



Gambar : skema entitas lemah



Gambar : E-R Diagram for a University Enterprise

4.4.Reduction to Relation Schemas(Pengurangan Skema Relasi)

- Set entitas dan himpunan relasi dapat dinyatakan secara seragam sebagai skema relasi yang mewakili isi dari basis data.
- Database yang sesuai dengan diagram E-R dapat diwakili oleh kumpulan skema.
- Untuk setiap himpunan entitas dan himpunan hubungan ada skema unik yang diberi nama himpunan entitas terkait atau himpunan relasi.
- Setiap skema memiliki sejumlah kolom (umumnya sesuai dengan atribut), yang memiliki nama unik.

4.4.1. Menggambarkan Entitas (Representing Entity Sets)

- Set entitas yang kuat mengurangi ke skema dengan atribut yang sama
student(ID, name, tot_cred)
- Set entitas yang lemah menjadi tabel yang mencakup kolom untuk kunci primer dari kumpulan entitas kuat yang mengidentifikasi bagian (course_id, sec_id, sem, year)

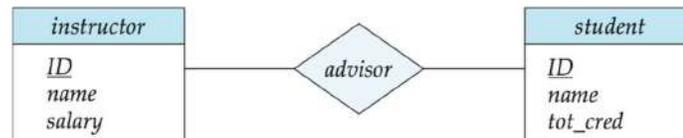


4.4.2. Menggambarkan Hubungan (Representing Relationship Sets)

- a) Kumpulan hubungan banyak-ke-banyak direpresentasikan sebagai skema dengan atribut untuk kunci utama dari dua set entitas yang berpartisipasi, dan atribut deskriptif dari himpunan relasi.

Contoh: skema untuk advisor yang ditetapkan hubungan

$advisor = (s_id, i_id)$



4.4.3. Representasi Entitas dengan Atribut Komposit

- a) Atribut komposit diratakan dengan membuat atribut terpisah untuk setiap atribut komponen

Contoh: diberikan entitas mengatur instruktur dengan nama atribut komposit dengan atribut komponen `first_name` dan `last_name` skema yang sesuai dengan kumpulan entitas memiliki dua atribut `name_first_name` dan `name_last_name`

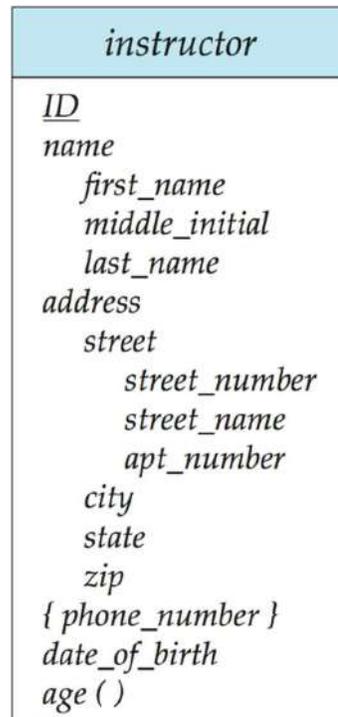
Prefix dihilangkan jika tidak ada ambiguitas

(`name_first_name` bisa menjadi `first_name`)

- b) Mengabaikan atribut multivalai, skema instruktur diperpanjang.

instructor(*ID*,

first_name, *middle_initial*, *last_name*,
street_number, *street_name*,
apt_number, *city*, *state*, *zip_code*,
date_of_birth)



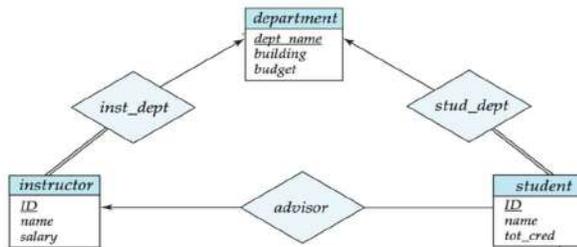
4.4.4. Representasi Entitas dengan Atribut Multinilai

- Atribut multinilai M dari suatu entitas E diwakili oleh skema EM terpisah
- Skema EM memiliki atribut yang bersesuaian dengan kunci primer E dan atribut yang sesuai dengan atribut multinilai M
- Contoh: Atribut nomor ponsel multinilai diwakili oleh skema :? $inst_phone = (ID, phone_number)$
- Setiap nilai dari atribut multinilai memetakan ke tuple terpisah dari relasi pada skema EM
 - Misalnya, entitas instruktur dengan kunci utama 22222 dan nomor telepon 456-7890 dan 123-4567 memetakan ke dua tuple: (22222, 456-7890) dan (22222, 123-4567)

4.4.5. Redundansi Skema (Redundancy of Schemas)

- Set hubungan banyak-ke-satu dan satu-ke-banyak yang bersifat total pada banyak sisi dapat diwakili dengan menambahkan atribut ekstra ke sisi "banyak(many)", yang berisi kunci utama dari sisi "satu(one)"

- b) Contoh: Daripada membuat skema untuk hubungan mengatur `inst_dept`, tambahkan atribut `dept_name` ke skema yang timbul dari instruktur set entitas.

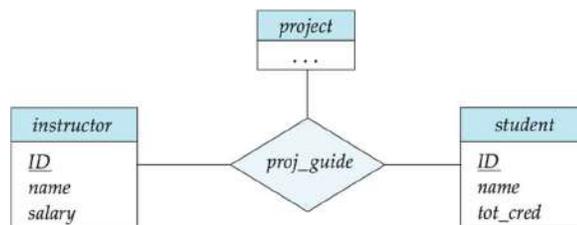


- c) Untuk pengaturan hubungan satu-ke-satu, kedua sisi dapat dipilih untuk bertindak sebagai sisi "banyak(many)".
Yaitu, atribut ekstra dapat ditambahkan ke salah satu tabel yang terkait dengan dua set entitas.
- d) Jika partisipasi bersifat parsial pada sisi "banyak(many)", mengganti skema dengan atribut ekstra dalam skema yang sesuai dengan sisi "banyak(banyak)" dapat mengakibatkan nilai nol.
- e) Skema yang terkait dengan himpunan hubungan yang menghubungkan entitas lemah yang diset ke kumpulan entitas kuat yang mengidentifikasi bersifat berlebihan.
- f) Contoh: Skema bagian sudah berisi atribut yang akan muncul di skema `sec_course`



4.4.6. Kumpulan Hubungan Non-biner (Non-binary Relationship Sets)

- a) Sebagian besar himpunan relasi adalah biner.
- b) Ada saat-saat ketika lebih nyaman untuk merepresentasikan hubungan sebagai non-biner.
- c) Diagram E-R dengan Hubungan Ternary

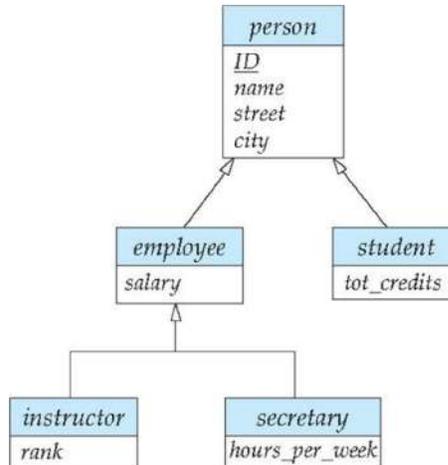


4.4.7. Batasan Kardinalitas pada Ternary Relationship (Cardinality Constraints on Ternary Relationship)

- a) Kami mengizinkan paling banyak satu panah keluar dari hubungan terner (atau lebih tinggi) untuk menunjukkan kendala kardinalitas
- b) Sebagai contoh, panah dari proj_guide ke instruktur menunjukkan setiap siswa memiliki paling banyak satu panduan untuk sebuah proyek
- c) Jika ada lebih dari satu anak panah, ada dua cara untuk mendefinisikan artinya.
 - Sebagai contoh, hubungan terner R antara A, B dan C dengan panah ke B dan C dapat berarti
 - a. Setiap entitas A dikaitkan dengan entitas unik dari B dan C atau
 - b. Setiap pasangan entitas dari (A, B) dikaitkan dengan entitas C yang unik, dan setiap pasangan (A, C) dikaitkan dengan B yang unik
- d) Setiap alternatif telah digunakan dalam berbagai formalisme
- e) Untuk menghindari kebingungan, kami melarang lebih dari satu panah

4.4.8. Spesialisasi (Specialization)

- a) Proses desain top-down; kami menetapkan sub-pengelompokan dalam kumpulan entitas yang berbeda dari entitas lain dalam kumpulan.
- b) Sub-pengelompokan ini menjadi kumpulan entitas tingkat rendah yang memiliki atribut atau berpartisipasi dalam hubungan yang tidak berlaku untuk kumpulan entitas tingkat yang lebih tinggi.
- c) Digambarkan oleh komponen segitiga berlabel ISA (mis., Instruktur "adalah" orang).
- d) Attribute inheritance - set entitas level yang lebih rendah mewarisi semua atribut dan partisipasi hubungan dari entitas level yang lebih tinggi yang disetel ke mana ia terhubung.
- e) Contoh Specialization:
 - Overlapping – employee and student
 - Disjoint – instructor and secretary
 - Total and partial



4.4.8.1. Spesialisasi melalui Skema (Representing Specialization via Schemas)

a) Method 1:

1. Bentuk skema untuk entitas tingkat yang lebih tinggi
2. Membentuk skema untuk setiap set entitas level yang lebih rendah, termasuk kunci primer dari set entitas level yang lebih tinggi dan atribut lokal

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

3. Kelemahan: mendapatkan informasi tentang, karyawan memerlukan akses dua hubungan, yang sesuai dengan skema tingkat rendah dan yang sesuai dengan skema tingkat tinggi

b) Method 2:

1. Bentuk skema untuk setiap entitas yang ditetapkan dengan semua atribut lokal dan yang diwariskan.

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

2. Kelemahan: nama, jalan, dan kota dapat disimpan secara berlebihan untuk orang-orang yang merupakan siswa dan karyawan

4.4.9. Generalisasi (Generalization)

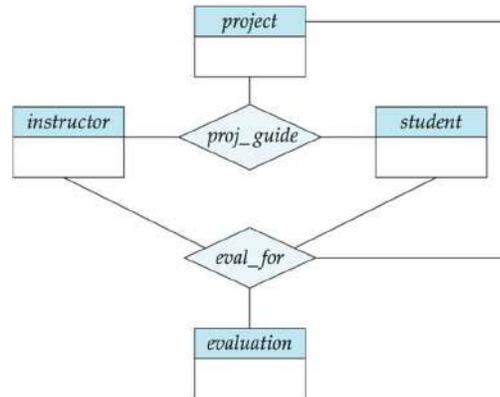
- a) Proses desain bottom-up - menggabungkan sejumlah set entitas yang berbagi fitur yang sama ke dalam set entitas tingkat yang lebih tinggi.
- b) Spesialisasi dan generalisasi adalah inversi sederhana satu sama lain; mereka direpresentasikan dalam diagram E-R dengan cara yang sama.
- c) Istilah spesialisasi dan generalisasi digunakan secara bergantian.

4.4.9.1. Desain pada Spesialisasi / Generalisasi

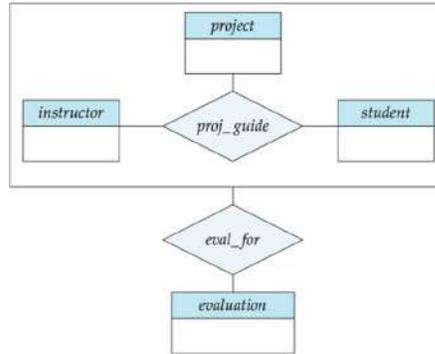
- a) Batasan komplet - menentukan apakah entitas dalam set entitas level yang lebih tinggi harus termasuk paling tidak satu set entitas level bawah dalam suatu generalisasi.
 1. total: entitas harus milik salah satu set entitas tingkat lebih rendah
 2. parsial: entitas tidak perlu menjadi milik salah satu set entitas tingkat lebih rendah
- b) Generalisasi parsial adalah default. Kita dapat menentukan generalisasi total dalam diagram ER dengan menambahkan total kata kunci dalam diagram dan menggambar garis putus-putus dari kata kunci ke kepala panah hampa terkait yang berlaku (untuk total generalisasi), atau ke rangkaian panah hampa -kursi yang berlaku (untuk generalisasi yang tumpang tindih).
- c) Generalisasi siswa adalah total: Semua entitas siswa harus baik lulusan atau sarjana. Karena set entitas level yang lebih tinggi tiba melalui generalisasi umumnya terdiri dari hanya entitas-entitas dalam set entitas tingkat yang lebih rendah, batasan kelengkapan untuk set entitas tingkat lebih tinggi yang umum biasanya total.

4.4.10. Aggregation

- a) Pertimbangkan hubungan `proj_guide`, yang kita lihat sebelumnya.
- b) Misalkan kita ingin merekam evaluasi siswa dengan panduan tentang proyek.



- c) Hubungan menetapkan `eval_for` dan `proj_guide` mewakili informasi yang tumpang tindih.
 1. Setiap hubungan `eval_for` sesuai dengan hubungan `proj_guide`
 2. Namun, beberapa hubungan `proj_guide` mungkin tidak sesuai dengan hubungan `eval_for` apapun
 - Jadi kita tidak dapat membuang hubungan `proj_guide`
- d) Hilangkan redundansi ini melalui agregasi
 1. Perlakukan hubungan sebagai entitas abstrak
 2. Memungkinkan hubungan antar hubungan
 3. Abstraksi hubungan menjadi entitas baru
- e) Hilangkan redundansi ini melalui agregasi tanpa memperkenalkan redundansi, diagram berikut mewakili:
 1. Seorang siswa dipandu oleh instruktur tertentu pada proyek tertentu
 2. Seorang siswa, instruktur, kombinasi proyek dapat memiliki evaluasi yang terkait



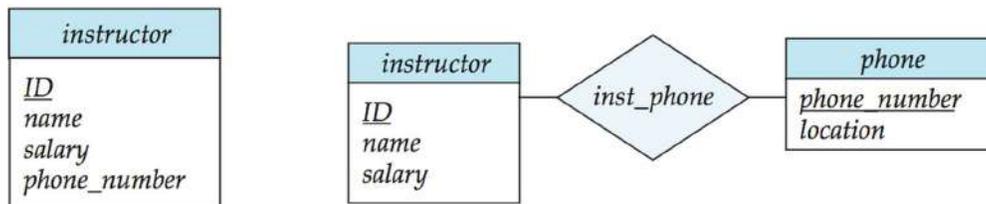
4.4.10.1. Agregasi melalui Skema (Representing Aggregation via Schemas)

- a) Untuk mewakili agregasi, buat skema yang berisi
 1. Kunci utama dari hubungan gabungan,
 2. Kunci utama dari kumpulan entitas terkait
 3. Atribut deskriptif apa pun
- b) Dalam contoh kita:
 1. Skema eval_for adalah:
eval_for (s_ID, project_id, i_ID, evaluation_id)
 2. Skema proj_guide adalah mubazir.

4.5.Design Issues

4.5.1. Entitas vs Atribut

- a) Penggunaan set entitas vs. Atribut

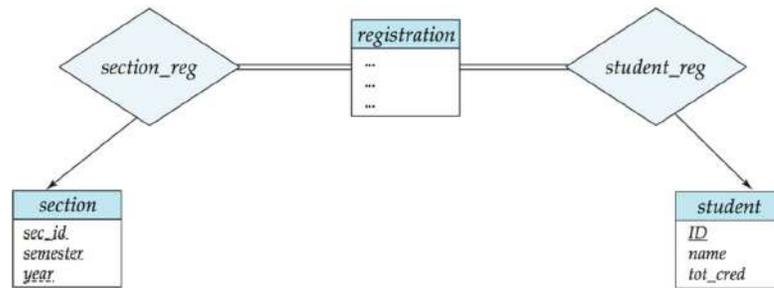


- b) Penggunaan ponsel sebagai entitas memungkinkan informasi tambahan tentang nomor telepon (ditambah beberapa nomor telepon)

4.5.2. Entitas vs Set Hubungan

- a) Penggunaan set entitas vs. set hubungan

Panduan yang memungkinkan adalah untuk menetapkan suatu hubungan yang ditetapkan untuk menggambarkan suatu tindakan yang terjadi antar entitas



- b) Penempatan atribut hubungan
 Misalnya, tanggal atribut sebagai atribut advisor atau sebagai atribut student.

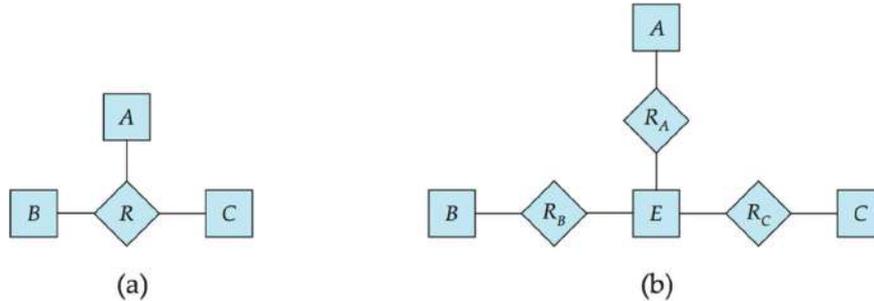
4.5.3. Binary vs hubungan Non-Binary

- a) Meskipun dimungkinkan untuk mengganti hubungan non-biner (n -ary, untuk $n > 2$) yang ditetapkan oleh sejumlah himpunan relasi biner yang berbeda, himpunan relasi n -ary menunjukkan lebih jelas bahwa beberapa entitas berpartisipasi dalam satu hubungan.
- b) Beberapa hubungan yang tampak non-biner mungkin lebih baik diwakili menggunakan hubungan biner
- Sebagai contoh, hubungan orangtua 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 tidak biner
 - Contoh: proj_guide

4.5.4. Konversi Hubungan Non-Biner ke Bentuk Biner

- a) Secara umum, setiap hubungan non-biner dapat diwakili menggunakan hubungan biner dengan membuat kumpulan entitas buatan.
- Ganti R antara entitas set A , B dan C oleh entitas mengatur E , dan tiga set hubungan:
 1. R_A , terkait E dan A
 2. R_B , terkait E dan B
 3. R_C , menghubungkan E dan C
 - Buat atribut identifikasi untuk E dan tambahkan atribut apa pun dari R ke E
 - Untuk setiap hubungan (a_i, b_i, c_i) di R , buat

1. entitas baru e_i dalam entitas mengatur E
2. menambahkan (e_i, a_i) ke R_A
3. tambahkan (e_i, b_i) ke R_B
4. tambahkan (e_i, c_i) ke R_C

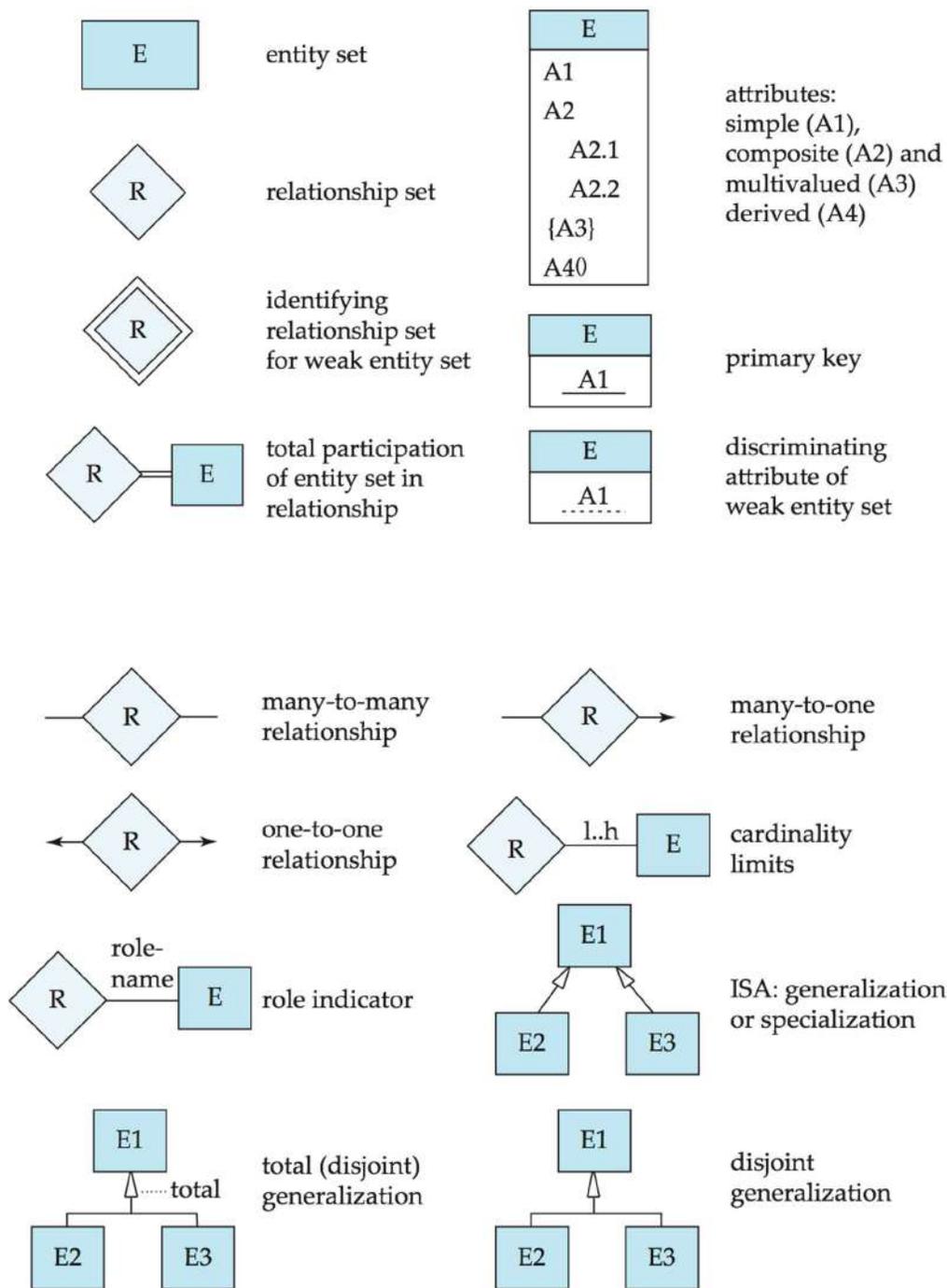


- b) Juga perlu menerjemahkan kendala
- Menerjemahkan semua kendala mungkin tidak dapat dilakukan
 - Mungkin ada contoh dalam skema terjemahan yang "tidak dapat berhubungan dengan contoh R
 - Latihan: tambahkan kendala pada hubungan R_A , R_B dan R_C untuk memastikan bahwa entitas yang baru dibuat sesuai dengan tepat satu entitas di setiap entitas set A, B dan C
 - Kita dapat menghindari menciptakan atribut identifikasi dengan membuat E kumpulan entitas lemah (dijelaskan secara singkat) yang diidentifikasi oleh tiga himpunan relasi.

4.5.5. Keputusan Desain E-R

- a) Penggunaan atribut atau entitas yang ditetapkan untuk mewakili suatu objek.
- b) Apakah konsep dunia nyata paling baik diekspresikan oleh suatu himpunan entitas atau himpunan relasi.
- c) Penggunaan hubungan terner versus sepasang hubungan biner.
- d) Penggunaan set entitas yang kuat atau lemah.
- e) Penggunaan spesialisasi / generalisasi - berkontribusi pada modularitas dalam desain.
- f) Penggunaan agregasi - dapat memperlakukan entitas agregat yang ditetapkan sebagai satu kesatuan tanpa memperhatikan detail strukturnya.

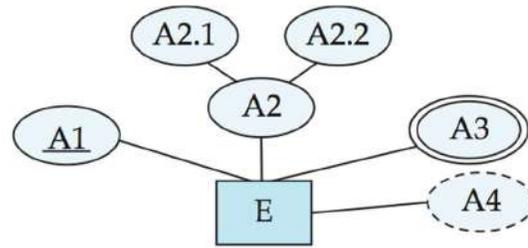
4.5.6. Simbol yang Digunakan dalam Notasi E-R



4.5.7. Notifikasi ER Alternatif

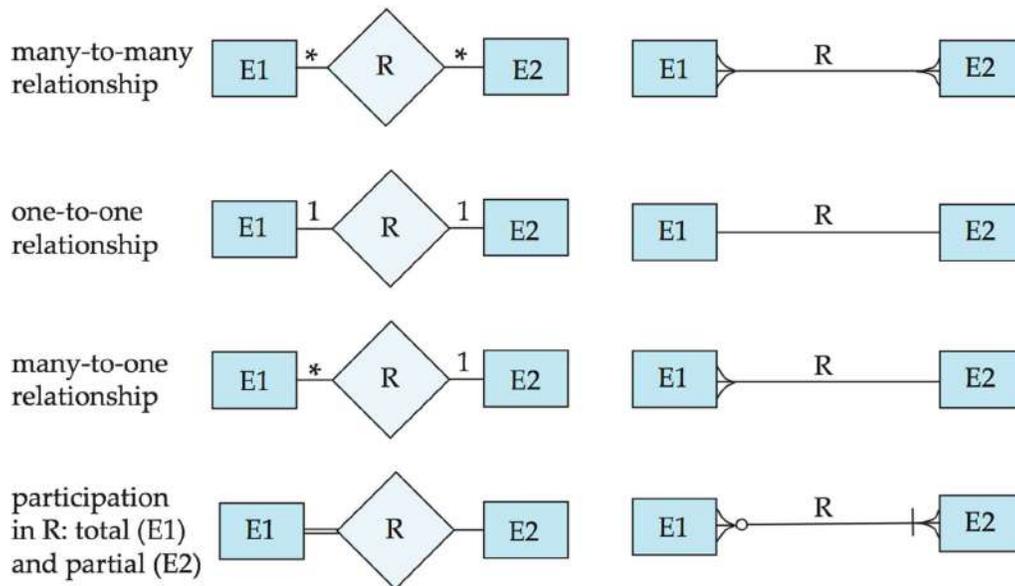
a) Chen, IDE1FX, ...

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



Chen

IDE1FX (Crows feet notation)

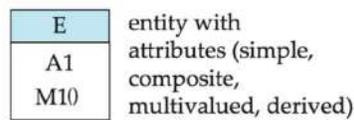


4.6. UML

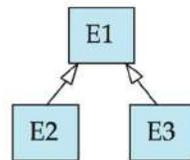
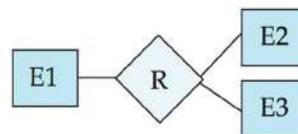
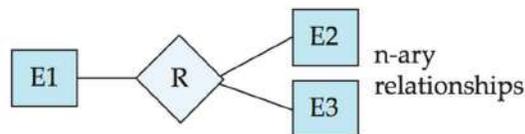
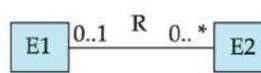
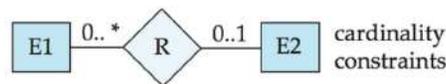
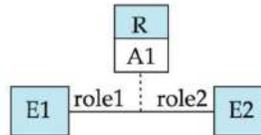
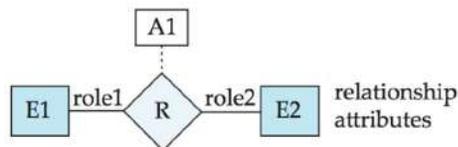
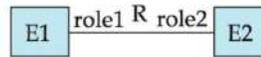
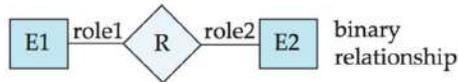
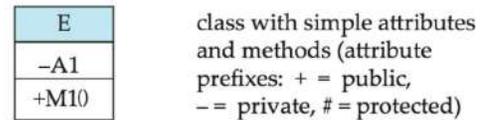
- a) UML : Bahasa Pemodelan Terpadu
- b) UML memiliki banyak komponen untuk memodelkan aspek-aspek berbeda dari keseluruhan sistem perangkat lunak
- c) Diagram Kelas UML sesuai dengan Diagram E-R, tetapi beberapa perbedaan.

4.6.1. ER vs. UML Class Diagrams

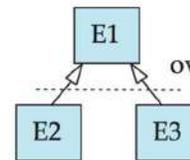
ER Diagram Notation



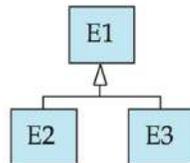
Equivalent in UML



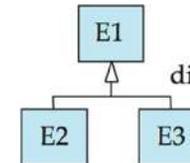
overlapping generalization



overlapping



disjoint generalization



disjoint

- Set hubungan biner direpresentasikan dalam UML hanya dengan menggambar garis yang menghubungkan kumpulan entitas. Hubungan mengatur nama ditulis berdekatan dengan garis.
- Peran yang dimainkan oleh entitas yang ditetapkan dalam himpunan relasi juga dapat ditentukan dengan menulis nama peran pada baris, berdekatan dengan kumpulan entitas.
- Hubungan nama set dapat secara bergantian ditulis dalam kotak, bersama dengan atribut dari himpunan relasi, dan kotak terhubung, menggunakan garis putus-putus, ke garis yang menggambarkan himpunan relasi.