

PERTEMUAN 9

DESAIN BASIS DATA DENGAN MODEL E-R

A. TUJUAN PEMBELAJARAN

1. Mahasiswa dapat memahami penggunaan Entitas dan Relasi dalam memodelkan data, yang dikenal dengan istilah model-ER.
2. Mahasiswa dapat menganalisa dan menentukan apa saja entitas-entitas yang ada beserta atributnya dari kasus yang diberikan.

B. URAIAN MATERI

1. Relasi

a. Pengertian Relasi

Pengertian relasi *Relationship* yaitu Satu tabel dapat dihubungkan dengan yang lain dalam apa yang dikenal sebagai relasi. Hubungan dapat dibangun ke dalam struktur *database* untuk memfasilitasi pengoperasian gabungan relasional saat runtime. Relasi menunjukkan adanya hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda

- 1) Hubungan: hubungan antara dua atau lebih entitas.
- 2) Himpunan Hubungan: Himpunan hubungan serupa.

Contoh entitas mahasiswa dengan nim = '107586' dan name = 'Darwin' (dari himpunan entitas mahasiswa) memiliki hubungan dengan entitas mata kuliah dengan kode = 'T05' dan Course_name = 'database', yang artinya bahwa Darwin -Siswa mengambil kursus dasar.

Konsep kekuatan hubungan didasarkan pada bagaimana kunci utama dari suatu entitas terkait didefinisikan. Untuk mengimplementasikan hubungan, kunci utama dari satu entitas (entitas induk, biasanya di sisi "satu" dari hubungan satu-ke-banyak) muncul sebagai kunci asing dalam entitas terkait (entitas anak, sebagian besar entitas di sisi "banyak" dari satu banyak hubungan). Terkadang kunci asing juga merupakan komponen kunci utama di entitas terkait.

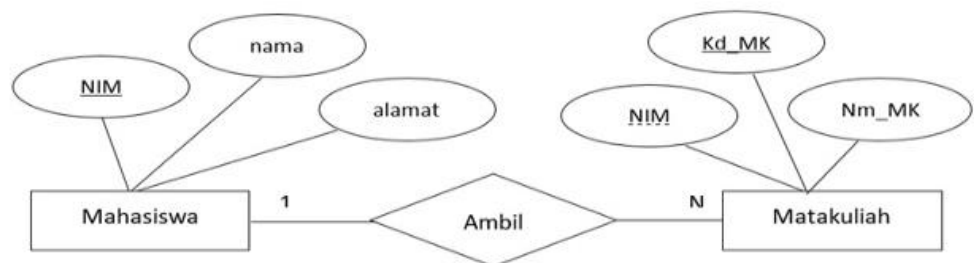
b. Variasi relasi

Relasi atau hubungan yang terjadi pada umumnya dilakukan antar dua entitas yang berbeda yang dikenal dengan nama hubungan biner (*binary relationship*). Namun pada pelaksanaannya dapat pula terjadi hubungan dengan hanya satu entitas atau lebih dari dua entitas. Oleh karena itu hubungan (*relationship*) dalam ERD dikelompokkan menjadi tiga hubungan yaitu:

1) Hubungan Biner

Adapun yang dimaksud dengan hubungan biner adalah hubungan antar dua buah entitas yang berbeda atau hubungan kelompok peran satu entitas dengan kelompok peran entitas lainnya. Penulisan dengan biner ini sangat dianjurkan sekali dalam penulisan ER-D. karena dapat memudahkan membedakan derajat hubungan yang terjadi didalam ER-D. pada gambar diatas ditampilkan dalam dua notasi yang mempunyai fungsi yang sama. Notasi pertama yang umum digunakan dan sering digunakan, sedangkan notasi yang kedua adalah notasi yang baru dan kini sudah mulai banyak digunakan.

Contoh hubungan biner:



Gambar 0.1 Contoh Hubungan Biner

2) Hubungan tunggal (unary *relationship*) atau Recursive *Relationship*

Hubungan tunggal (unary *relationship*) adalah hubungan yang terjadi pada himpunan yang sama. penggunaan hubungan tunggal bisa saja terjadi pada kasus-kasus yang ditemui sehari-hari Hubungan tunggal pada ER-D dapat terjadi pada suatu perusahaan atau lembaga Pendidikan. Hal yang paling mudah sekali bila sekumpulan pegawai atau karyawan yang lebih dari satu akan memimpin pegawai lain. Oleh karena itu hubungan tunggal sebaiknya digunakan bila dalam keadaan terdesak dan tidak dapat dilakukan lagi.

Contoh hubungan unary:

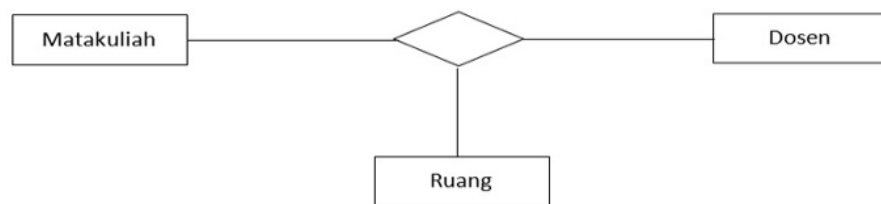


Gambar 0.2 Contoh hubungan unary

3) Hubungan Multi Entitas (N-ary *Relationship*)

Hubungan multi entitas adalah hubungan dari tiga himpunan entitas atau lebih. Sebaiknya bentuk hubungan ini sedapat mungkin dihindari, karena pada hubungan ini dapat mengaburkan derajat hubungan yang ada pada hubungan tersebut. Penulisan ER-D dengan multi entitas dapat terjadi pada perusahaan maupun lembaga Pendidikan namun sekali lagi sebaiknya hindari pemakaian multi entitas ini.

Contoh gambar hubungan N-ary



Gambar 0.3 Contoh hubungan N-ary

c. Karakteristik tabel relasi

Berikut karakteristik relasi tabel

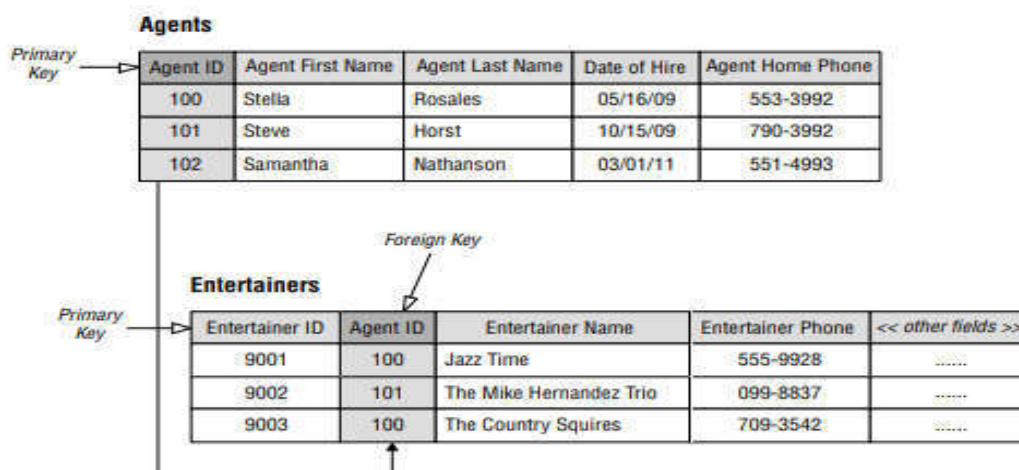
- Tabel dianggap sebagai struktur dua dimensi yang terdiri dari baris dan kolom.
- Setiap baris tabel (tupel) mewakili kejadian entitas tunggal dalam kumpulan entitas.
- Setiap kolom tabel mewakili atribut, dan setiap kolom memiliki nama yang berbeda.
- Setiap perpotongan baris dan kolom mewakili satu nilai data.
- Semua nilai dalam kolom harus sesuai dengan format data yang sama.

- f) Setiap kolom memiliki rentang nilai tertentu yang dikenal sebagai domain atribut.
- g) Urutan baris dan kolom tidak penting bagi DBMS.
- h) Setiap tabel harus memiliki atribut atau kombinasi atribut yang secara unik mengidentifikasi setiap baris.

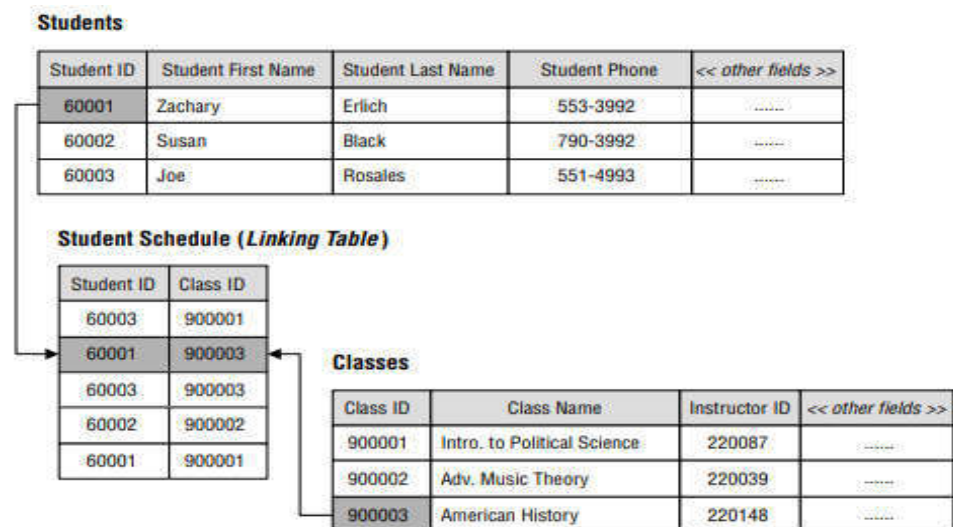
2. Kardinalitas

- a. Pengertian kardinalitas Adapun yang dimaksud dengan kardinalitas adalah:

Mendefinisikan batas minimum dan maksimum untuk setiap kejadian dari suatu entitas yang dapat dihubungkan ke suatu peristiwa di entitas lain. Karena semua relasi bersifat langsung, kardinal harus didefinisikan secara langsung di antara dua entitas kapan pun mereka terhubung (Whitten et.al., 2001:264) hubungan antara dua tabel jika Anda bisa dengan cara tertentu mengasosiasikan *record* dari tabel pertama dengan *record* dari tabel kedua. Kamu dapat membangun sebuah relasi melalui sekumpulan kunci utama "*primary key*" dan asing "*foreign key*" atau melalui tabel ketiga yang diketahui sebagai tabel penautan (juga dikenal sebagai tabel asosiatif). Cara masuk yang Anda buat hubungan sangat bergantung pada tipe hubungan yang ada di antara tabel.



Gambar 0.4 Mengilustrasikan Hubungan yang Dibangun oleh *Primary key*



Gambar 0.5 Ilustrasi Hubungan antara Tabel

relasi adalah komponen penting dari *database* relasional.

- Ini memungkinkan Anda untuk membuat tampilan yang beragam.
- Sangat penting untuk integritas data karena membantu mengurangi redundan data dan menghilangkan data duplikat.

Anda dapat mencirikan setiap hubungan dengan tiga cara: menurut jenis hubungan yang ada di antara tabel, cara di mana masing-masing tabel berpartisipasi, dan sejauh mana setiap tabel berpartisipasi. Anda dapat mencirikan setiap hubungan dengan tiga cara: menurut jenis hubungan yang ada di antara tabel, dengan cara masing-masing tabel berpartisipasi, dan sejauh mana setiap tabel berpartisipasi.

b. Jenis Kardinalitas

Ada tiga jenis kardinalitas dalam *database* yang bisa ada diantara sepasang tabel antara lain: *one-to-one*, *one-to-many*, and *many-to-many*.

One to one adalah Sepasang tabel memiliki hubungan satu-ke-satu atau *one-to-one* ketika satu *record* di tabel pertama hanya terkait dengan satu *record* di tabel kedua, dan satu catatan di tabel kedua hanya terkait dengan satu catatan di tabel pertama. Dalam jenis hubungan ini, satu tabel berfungsi sebagai Tabel "induk" dan yang lainnya berfungsi sebagai tabel "anak". Anda membangun hubungan dengan mengambil salinan kunci utama tabel induk dan menggabungkannya ke dalam struktur tabel anak, di mana itu menjadi kunci asing.

Gambar 9.6 menunjukkan contoh hubungan satu-ke-satu yang khas. Dikasuk ini, KARYAWAN adalah tabel induk dan KOMPENSASI adalah meja anak. Hubungan antara tabel-tabel ini sedemikian rupa sehingga menjadi satu *record* dalam tabel EMPLOYEES hanya dapat dikaitkan dengan satu *record* di tabel KOMPENSASI, dan satu catatan di KOMPENSASI tabel dapat dikaitkan hanya dengan satu *record* dalam tabel EMPLOYEES. Catatan bahwa EMPLOYEE ID memang merupakan kunci utama di kedua tabel. Namun, itu juga akan menjalankan peran kunci asing di tabel anak.

Employees

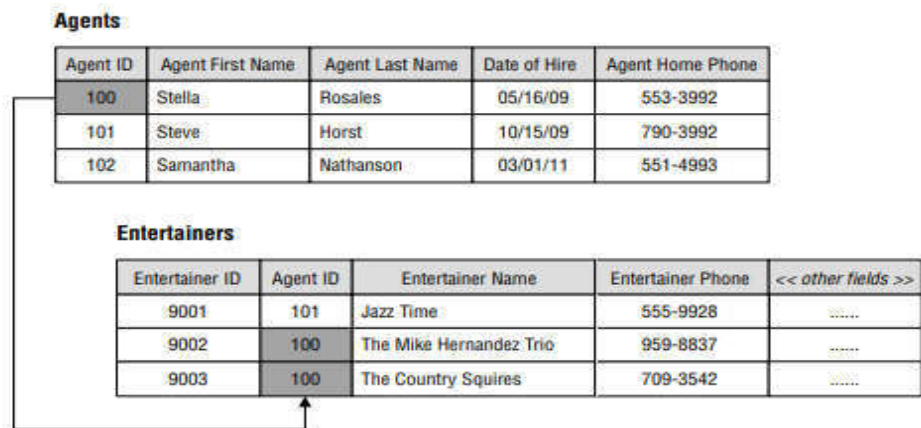
Employee ID	Employee First Name	Employee Last Name	Home Phone	<< other fields >>
100	Zachary	Erlich	553-3992
101	Susan	Black	790-3992
102	Joe	Rosales	551-4993

Compensation

Employee ID	Hourly Rate	Commission Rate	<< other fields >>
100	19.75	3.5%
101	25.00	5.0%
102	22.50	5.0%

Gambar 0.6 Contoh One-to one

one-to many hubungan satu-ke-banyak antara sepasang tabel ketika satu catatan di tabel pertama bisa dikaitkan dengan banyak catatan di tabel kedua, tapi satu catatan di tabel kedua bisa terkait hanya satu catatan di tabel pertama. (Model orang tua / anak yang biasa saya gambarkan hubungan satu-ke-satu juga berfungsi di sini. Dalam hal ini, meja di atas sisi "satu" dari hubungan tersebut adalah tabel induk, dan tabel pada sisi "banyak" adalah tabel anak.) Anda membuat hubungan satu-ke-banyak dengan mengambil salinan kunci utama tabel induk dan menggabungkannya ke dalam struktur tabel anak, di mana itu menjadi kunci asing.



Gambar 0.7 Contoh one-to many

Many-to-Many Sepasang tabel memiliki hubungan banyak-ke-banyak ketika satu catatan di tabel pertama dapat dikaitkan dengan banyak catatan di tabel kedua tabel dan satu catatan di tabel kedua dapat dikaitkan dengan banyak catatan di tabel pertama. Anda membangun hubungan ini dengan sebuah tautan meja. (Anda telah mempelajari sedikit tentang jenis tabel ini di awal bagian ini.) Tabel penautan memudahkan Anda untuk mengaitkan catatan dari satu tabel dengan yang lain dan akan membantu untuk memastikan Anda tidak memiliki masalah menambahkan, menghapus, atau memodifikasi data terkait. Kamu tentukan tabel penghubung dengan mengambil salinan kunci utama dari setiap tabel dalam hubungan dan menggunakannya untuk membentuk struktur baru meja. Bidang-bidang ini sebenarnya memiliki dua peran berbeda: Bersama-sama, mereka terbentuk kunci utama komposit dari tabel penghubung; secara terpisah, mereka masing-masing berfungsi sebagai kunci asing.

Hubungan banyak-ke-banyak yang tidak terjalin dengan baik adalah "belum terselesaikan". Gambar 3.15 menunjukkan contoh klasik dan jelas dari yang belum terselesaikan hubungan banyak-ke-banyak. Dalam hal ini, satu *record* di *file*.

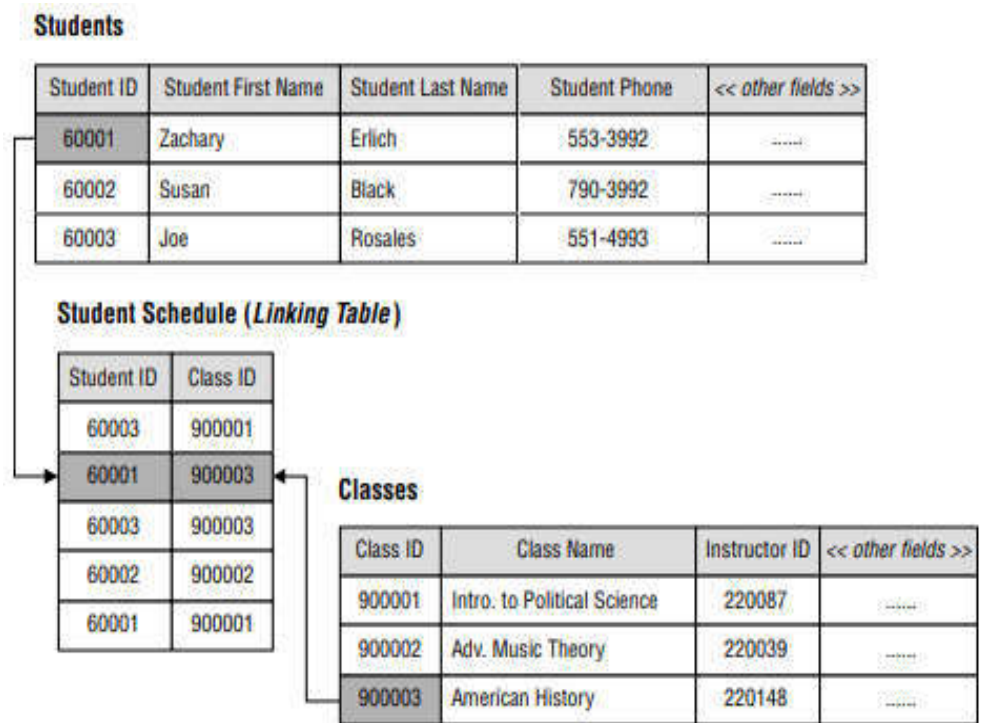
Classes

Class ID	Class Name	Instructor ID	<< other fields >>
900001	Intro. to Political Science	220087
900002	Adv. Music Theory	220039
900003	American History	220148

Gambar 0.8 Table Students

Tabel STUDENTS dapat dikaitkan dengan banyak *record* di CLASSES tabel dan satu *record* dalam tabel CLASSES dapat dikaitkan dengan banyak catatan dalam tabel STUDENTS.

Hubungan ini tidak terselesaikan karena kekhasan yang melekat pada hubungan banyak ke banyak. Masalah utamanya adalah ini: Bagaimana Anda dengan mudah mengaitkan catatan dari tabel pertama dengan catatan di tabel kedua? Untuk membingkai ulang pertanyaan dalam bentuk tabel yang ditunjukkan pada Gambar 3.15, bagaimana Anda mengasosiasikan seorang siswa dengan beberapa kelas atau satu kelas tertentu kelas dengan beberapa siswa? Apakah Anda memasukkan beberapa bidang STUDENT ke dalam *file* Tabel CLASSES? Atau apakah Anda menambahkan beberapa bidang KELAS ke SISWA meja? Salah satu dari pendekatan ini akan menyulitkan Anda untuk bekerja dengan data dalam tabel tersebut dan akan mempengaruhi integritas data secara negatif. Pendekatan terbaik untuk Anda lakukan adalah membuat dan menggunakan tabel penautan, yang akan menyelesaikan hubungan banyak-ke-banyak dengan cara yang paling tepat dan efektif. Gambar 9.9 menunjukkan solusi ini dalam praktiknya.



Gambar 0.9 Relasi Table Students

Penting bagi Anda untuk mengetahui jenis hubungan yang ada antara sepasang tabel karena menentukan bagaimana tabel tersebut terkait, apakah rekaman antara tabel saling bergantung atau tidak, dan jumlah minimum dan maksimum rekaman terkait yang bisa ada di dalam hubungan.

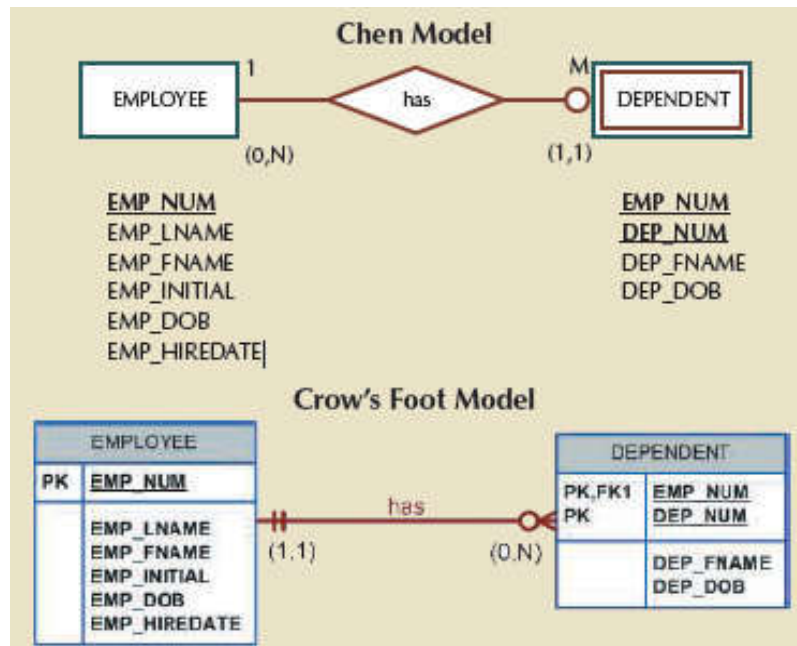
3. Weak Identity (Entitas Lemah)

entitas lemah adalah entitas yang tidak mempunyai key. Salah satu yang memenuhi dua syarat:

- Entitas bergantung pada keberadaan; ia tidak bisa ada tanpa entitas yang dimilikinya sebuah hubungan.
- Entitas memiliki kunci utama yang sebagian atau seluruhnya berasal dari entitas induk dalam hubungan.

Misalnya, polis asuransi perusahaan mengasuransikan karyawan dan tanggungannya. Untuk tujuan menjelaskan polis asuransi, seorang KARYAWAN mungkin atau mungkin tidak memiliki DEPENDENT, tetapi DEPENDENT harus terkait dengan EMPLOYEE. Selain itu, DEPENDENT tidak bisa ada tanpa KARYAWAN; yaitu, seseorang tidak bisa mendapatkan perlindungan asuransi sebagai tanggungan kecuali orang tersebut menjadi

tanggungan seorang pegawai. DEPENDENT adalah entitas lemah dalam hubungan "EMPLOYEE TERGANTUNG." Hubungan ini ditunjukkan pada



Gambar 0.10 Weak entity

Perlu diingat bahwa urutan tabel dibuat dan dimuat sangat penting. Misalnya, dalam hubungan "KURSUS menghasilkan KELAS", tabel KURSUS harus dibuat sebelum tabel CLASS. Lagipula, memiliki CLASS tidak dapat diterima kunci asing tabel merujuk ke tabel KURSUS yang belum ada. Bahkan, Anda harus memuat *file* data dari sisi "1" terlebih dahulu dalam hubungan 1: M untuk menghindari kemungkinan integritas referensial kesalahan, terlepas dari apakah hubungannya lemah atau kuat.

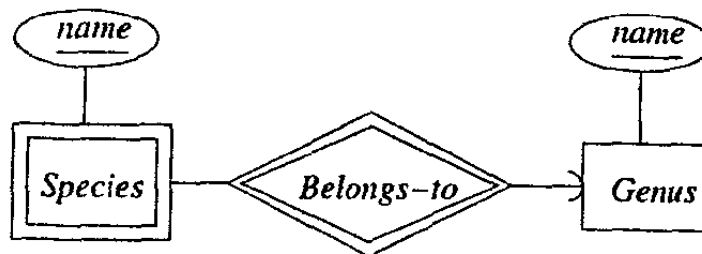
Catatan :

entitas lemah Entitas yang menampilkan adanya ketergantungan dan mewarisi yang utama kunci induknya kesatuan. Misalnya, a DEPENDENT membutuhkan keberadaan *file* KARYAWAN.

Ada kondisi sesekali ketika kunci kumpulan entitas terdiri dari atribut beberapa atau semua, yang dimiliki oleh kumpulan entitas lain. Kumpulan entitas seperti itu disebut himpunan entitas lemah.

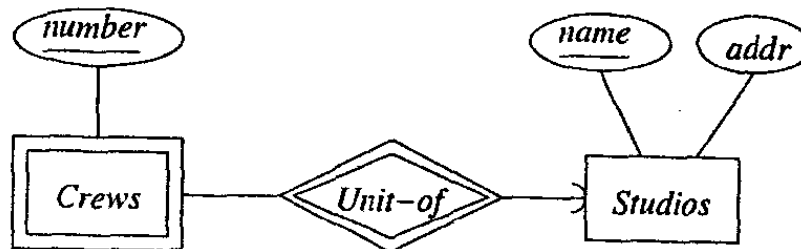
Ada dua sumber utama dari kumpulan entitas yang lemah. Pertama, terkadang kumpulan entitas jatuh ke dalam hierarki berdasarkan klasifikasi yang tidak terkait dengan "hierarki isa" Bagian 2.1.11. Jika entitas dari

himpunan E adalah subunit dari entitas dalam himpunan F, maka itu adalah mungkin nama-nama entitas E tidak unik sampai kita memperhitungkannya nama entitas F di mana entitas E adalah bawahannya. Beberapa contoh nil1 menggambarkan masalahnya.



Gambar 0.11 Model E-R

Contoh 9.12: Sebuah studio film mungkin memiliki beberapa kru film. Para kru mungkin ditunjuk oleh studio tertentu sebagai kru 1, kru 2, dan seterusnya. Namun, studio lain mungkin menggunakan sebutan yang sama untuk kru, jadi nomor atributnya bukan kunci untuk kru. Sebaliknya, untuk memberi nama kru secara unik, kita perlu memberikan keduanya nama studio tempatnya berada dan jumlah kru. Itu situasi ditunjukkan oleh Gambar 9.11. Kunci untuk kumpulan entitas yang lemah adalah Crews sendiri, atribut kayu dan atribut nama dari studio unik tempat kru terkait dengan banyak-satu pinggul Hubungan Unit.



Gambar 0.12 Atribut

Kami tidak dapat memperoleh atribut kunci untuk kumpulan entitas yang lemah jika E adalah himpunan entitas lemah maka kuncinya terdiri dari:

- Nol atau lebih dari atributnya sendiri, dan
- Atribut kunci dari kumpulan entitas yang dijangkau oleh banyak-satu tertentu hubungan dari E ke himpunan entitas lain. Hubungan banyak-satu ini disebut hubungan pendukung untuk E

Agar R, hubungan banyak-satu dari E ke beberapa himpunan entitas F, menjadi hubungan pendukung untuk E, kondisi berikut harus dipatuhi:

- a. R harus berupa hubungan biner, banyak satu dari E ke F.
- b. R harus memiliki integritas referensial dari E ke F. Artinya, untuk setiap entitas E, entitas-F yang terkait dengannya oleh R harus benar-benar ada dalam *database*. Taruh Dengan cara lain, panah bulat dari R ke F harus dijustifikasi.
- c. Atribut yang disuplai F untuk kunci E harus berupa atribut kunci F Namun, jika F itu sendiri lemah, maka beberapa atau semua atribut kunci dari F dipasok ke E akan menjadi atribut kunci dari satu atau lebih himpunan entitas G ke mana F dihubungkan dengan hubungan pendukung. Secara rekursif, jika G lemah, beberapa atribut kunci G akan dipasok dari tempat lain, dan seterusnya.

Jika ada beberapa hubungan pendukung yang berbeda dari E ke F, maka setiap hubungan digunakan untuk memberikan salinan atribut kunci F untuk membantu membentuk kunci E. Perhatikan bahwa entitas e dari E mungkin terkait entitas yang berbeda di F melalui hubungan pendukung yang berbeda dari E. Dengan demikian, kunci dari beberapa entitas berbeda dari F dapat muncul di kunci tersebut nilai yang mengidentifikasi entitas tertentu e dari E.

C. SOAL LATIHAN/TUGAS

1. Jelaskan pengertian dari relasi !
2. Jelaskan pengertian dari relasi biner !
3. Jelaskan pengertian dari relasi unary !
4. Jelaskan pengertian dari relasi N-ary !
5. Jelaskan pengertian dari kardinalitas!
6. Jelaskan kardinalitas one to one dan gambarkan contohnya!
7. Jelaskan kardinalitas *one to many* dan gambarkan contohnya!
8. Jelaskan kardinalitas *many to many* dan gambarkan contohnya!
9. Jelaskan apa itu weak *entity*!

D. REFERENSI

Sikha Bagui and Richard Earp. 2003. *Database Design Using Entity-Relationship Diagrams*. Florida AUERBACH PUBLICATIONS

Michael J. Hernandez. 2013. *Database Design for Mere Mortals A Hands-on Guide to Relational Database Design Third Edition*. New Jersey AADISON-WESLEY

Carlos Coronel & Steven Morris. 2017, 2015. *DATABASE SYSTEMS Design, Implementation, and Management*. Boston Cengage Learning

Hector Garcia-Molina, Jeffrey D. Ullman & Jennifer Widom. 1998. *Database Systems: The Complete Book*. New Jersey Prentice Hall

Adyanata Lubis, S.Kom., M.Kom (2016), *Basis Data Dasar Untuk Mahasiswa Ilmu Komputer*, Pasir pengairan DEEPPUBLISH

GLOSARIUM

N-ary relationship adalah hubungan dari tiga himpunan entitas atau lebih.

Unary relationship adalah hubungan yang terjadi pada dari himpunan yang sama.

Weak entity adalah entitas yang tidak mempunyai key