

## **BAB III PEMODELAN DATA MENGGUNAKAN MODEL ENTITY-RELATIONSHIP (MODEL E-R)**

### **A. Capaian Pembelajaran Mata Kuliah**

Mahasiswa mampu menjelaskan konsep basis data, dan merancang basis data mulai dari tahap pemodelan data dengan model E-R sampai model relasional, dimana hasil rancangan memenuhi kaedah normalisasi.

### **B. Sub-Capaian Pembelajaran Mata Kuliah**

Mahasiswa mampu menjelaskan konsep pemodelan data menggunakan Model *Entity-Relationship* (Model E-R)

### **C. Indikator – Indikator Pembelajaran**

Mahasiswa mampu:

1. Menjelaskan komponen dasar dalam model *entity-relationship*
2. Memberikan contoh yang termasuk entitas, relasi dan atribut
3. Membedakan atribut berdasarkan jenis-jenisnya
4. Menjelaskan tentang derajat relasi
5. Menjelaskan rasio kardinalitas dan partisipasi
6. Menjelaskan jenis relasi *n-ary*

### **D. Materi**

#### **1. Pendahuluan**

Pemodelan konseptual adalah tahap yang sangat penting dalam merancang aplikasi basis data. Secara umum, istilah aplikasi basis data mengacu pada basis data tertentu dan program terkait yang mengimplementasikan kueri dan pembaruan basis data. Misalnya, aplikasi basis data BANK yang melacak rekening nasabah akan menyertakan program yang mengimplementasikan pembaruan basis data terkait dengan setoran dan penarikan oleh nasabah. Program-program ini akan menyediakan antarmuka pengguna grafis (*Graphical User Interface-GUI*) yang mudah digunakan dengan menggunakan formulir dan menu untuk pengguna akhir aplikasi (pelanggan bank atau teller bank). Selain itu, sekarang umum untuk menyediakan antarmuka untuk program-program ini kepada pelanggan BANK melalui perangkat seluler menggunakan aplikasi seluler (contoh: *m-banking*). Oleh karena itu, bagian utama dari aplikasi basis data akan

memerlukan desain, implementasi, dan pengujian program aplikasi. Secara tradisional, desain dan pengujian program aplikasi telah dianggap sebagai bagian dari rekayasa perangkat lunak daripada desain basis data. Dalam banyak alat desain perangkat lunak, metodologi desain basis data dan metodologi rekayasa perangkat lunak saling terkait karena kegiatan ini sangat terkait.

Dalam bab ini, kita mengikuti pendekatan tradisional berkonsentrasi pada struktur basis data dan kendala selama desain basis data konseptual. Kami menyajikan konsep pemodelan dengan model *entity-relationship* (E-R), yang merupakan model data konseptual tingkat tinggi yang populer. Model ini dan variasinya sering digunakan untuk desain konseptual aplikasi basis data, dan banyak alat desain basis data yang menggunakan konsep ini. Pada bab ini, dijelaskan konsep dan batasan struktur data dasar dari model E-R dan mendiskusikan penggunaannya dalam desain skema konseptual untuk aplikasi basis data.

Pada bab ini disusun sebagai berikut:

## **2. Contoh Aplikasi Basis Data**

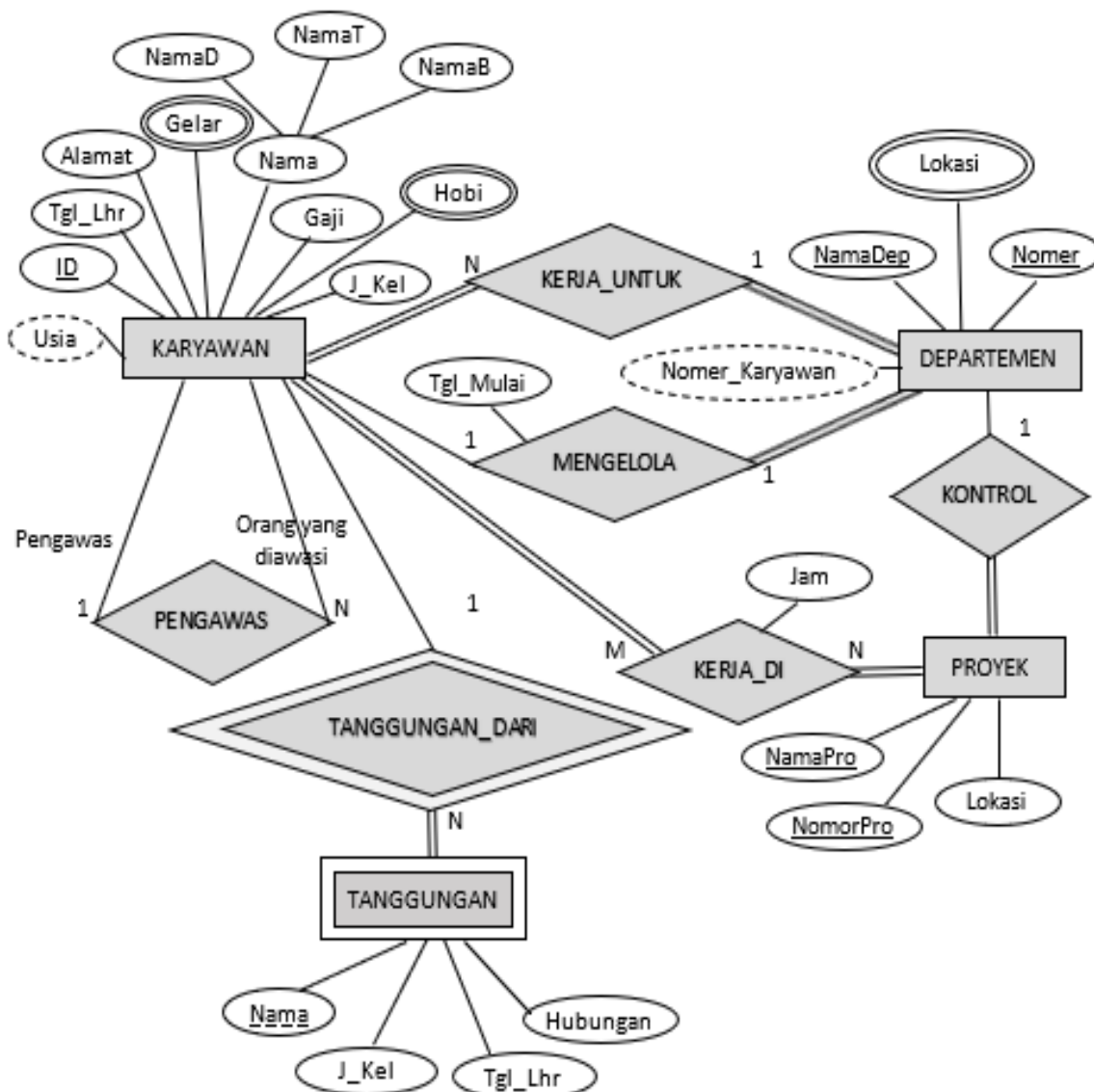
Pada bagian ini dijelaskan contoh aplikasi basis data, yang disebut PERUSAHAAN, yang berfungsi untuk menggambarkan konsep model E-R dasar dan penggunaannya dalam desain skema. Dalam hal ini, akan dicantumkan juga persyaratan data untuk basis data yang digunakan, dan kemudian membuat skema konseptualnya selangkah demi selangkah saat konsep pemodelan dengan menggunakan model E-R diperkenalkan. Basis data PERUSAHAAN melacak karyawan, departemen, dan proyek perusahaan. Misalkan setelah tahap pengumpulan dan analisis kebutuhan, perancang basis data memberikan deskripsi *miniworld* yang merupakan bagian dari perusahaan yang akan direpresentasikan dalam basis data.

- Perusahaan diorganisasikan dalam beberapa departemen. Setiap departemen memiliki nama unik, nomor unik, dan karyawan tertentu yang mengelola departemen. Akan dilakukan pelacakan terhadap tanggal mulai yang menunjukkan seorang karyawan mulai mengelola departemen. Sebuah departemen mungkin memiliki beberapa lokasi.
- Sebuah departemen mengendalikan sejumlah proyek, yang masing-masing memiliki nama unik, nomor unik, dan berada pada satu lokasi.
- Basis data akan menyimpan nama setiap karyawan, nomor unik karyawan, tanggal lahir, gelar, alamat, gaji, jenis kelamin, usia, dan hobi. Seorang karyawan

ditugaskan pada satu departemen, tetapi dapat mengerjakan beberapa proyek, yang tidak harus dikendalikan oleh departemen yang sama. Hal ini diperlukan untuk melacak jumlah jam per minggu seorang karyawan bekerja pada setiap proyek, serta melacak pengawas langsung dari setiap karyawan (yang merupakan karyawan lain).

- Basis data akan melacak tanggungan setiap karyawan untuk tujuan asuransi, termasuk nama depan setiap tanggungan, jenis kelamin, tanggal lahir, dan hubungan dengan karyawan.

Gambar 3.1 menunjukkan bagaimana skema untuk aplikasi basis data ini dapat ditampilkan melalui diagram E-R. Selanjutnya akan dijelaskan secara bertahap saat konsep model E-R disajikan.



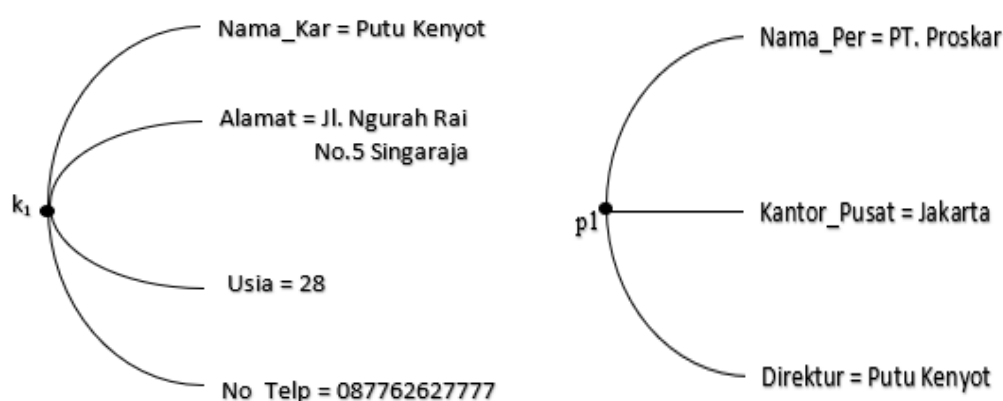
Gambar 3.1 Diagram E-R untuk Basis Data PERUSAHAAN

### 3. Jenis Entitas, Kumpulan Entitas (*Entity Set*), Atribut, dan Kunci (*Keys*)

Model E-R menggambarkan data sebagai entitas, relasi, dan atribut.

#### 3.1 Entitas

**Entitas** merupakan representasi dari konsep dasar model E-R. Entitas adalah objek pada dunia nyata dengan eksistensinya dan dapat dibedakan dengan objek lainnya. Entitas dapat berupa objek dengan keberadaan fisik (misalnya, orang, mobil, rumah, buku, atau karyawan tertentu) atau mungkin objek dengan keberadaan konseptual (misalnya, perusahaan, pekerjaan, atau perkuliahan pada sebuah universitas). Setiap entitas memiliki atribut (properti tertentu yang mendeskripsikannya). Misalnya, entitas KARYAWAN dapat dijelaskan dengan nama karyawan, usia, alamat, gaji, dan pekerjaan karyawan tersebut. Entitas tertentu akan memiliki nilai untuk setiap atributnya. Nilai-nilai atribut yang menggambarkan setiap entitas menjadi bagian utama dari data yang disimpan dalam basis data.



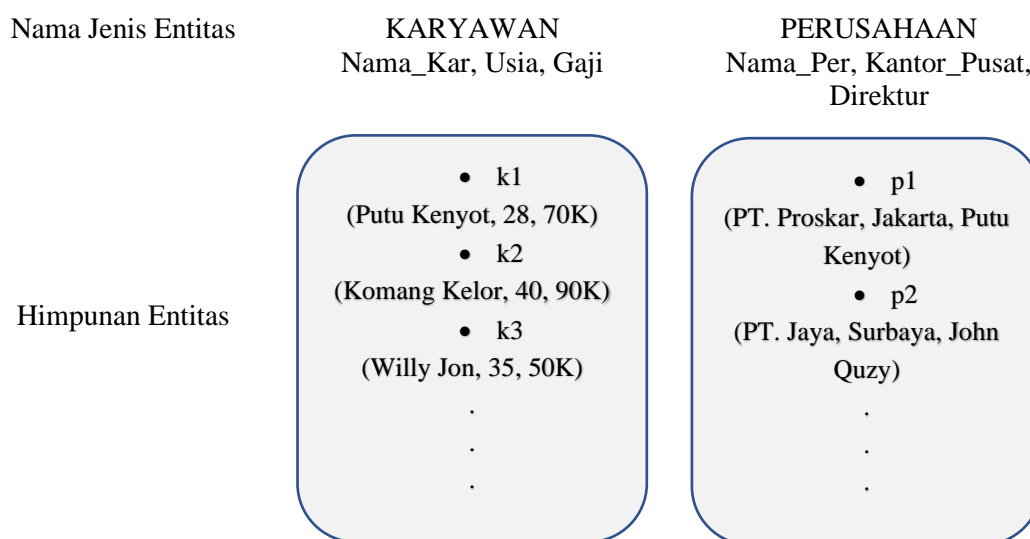
Gambar 3.2 Contoh dua entitas beserta atributnya, yaitu  $k_1$  adalah KARYAWAN, dan  $p_1$  adalah PERUSAHAAN

Gambar 3.2 menunjukkan dua contoh entitas dan nilai atributnya. Entitas KARYAWAN  $k_1$  memiliki empat atribut, yaitu Nama\_Kar, Alamat, Usia, dan No\_Telp; nilai dari masing-masing atributnya adalah 'Putu Kenyot', 'Jl. Ngurah Rai No.5 Singaraja', '28', dan '087762627777'. Entitas PERUSAHAAN  $p_1$  memiliki tiga atribut, yaitu Nama\_Per, Kantor\_Pusat, dan Direktur; nilai masing-masing atributnya adalah 'PT. Proskar', 'Jakarta', dan 'Putu Kenyot'.

#### 3.2 Jenis Entitas dan Himpunan Entitas (*Entity Set*)

**Jenis Entitas dan Himpunan Entitas.** Sebuah basis data biasanya berisi kelompok entitas yang serupa. Misalnya, sebuah perusahaan yang mempekerjakan ratusan karyawan mungkin ingin menyimpan informasi serupa mengenai masing-

masing karyawan. Entitas karyawan ini memiliki atribut yang sama, tetapi setiap entitas memiliki nilainya sendiri untuk setiap atribut. **Jenis entitas** didefinisikan sebagai himpunan (atau *set*) entitas yang memiliki atribut yang sama. Setiap jenis entitas dalam basis data dijelaskan dengan nama dan atributnya. Gambar 3.3 menunjukkan dua jenis entitas, yaitu KARYAWAN dan PERUSAHAAN, dan daftar beberapa atribut untuk masing-masing entitas. Beberapa entitas individu dari setiap jenis juga diilustrasikan, bersama dengan nilai atributnya. Semua entitas dari jenis entitas tertentu dalam basis data disebut **kumpulan entitas** atau **himpunan entitas**. Himpunan entitas biasanya menggunakan nama yang sama dengan jenis entitas, meskipun keduanya merupakan dua konsep yang terpisah. Misalnya, KARYAWAN mengacu pada jenis entitas serta himpunan semua entitas karyawan saat ini dalam basis data.



Gambar 3.3 Dua jenis entitas, yaitu KARYAWAN dan PERUSAHAAN, serta contoh beberapa anggota dari masing-masing entitas

Jenis entitas direpresentasikan dalam diagram E-R (lihat Gambar 3.1) sebagai kotak persegi panjang dan dituliskan nama jenis entitas didalamnya. Atribut direpresentasikan sebagai bentuk oval yang dilengkapi dengan tulisan untuk jenis atribut dan dihubungkan pada jenis entitasnya dengan garis lurus.

### 3.3 Jenis Entitas Lemah (*Weak Entity Type*)

Jenis entitas yang tidak memiliki atribut kunci disebut **jenis entitas lemah**. Untuk atribut kunci akan dibahas pada bagian 3.5. Sebaliknya, jenis entitas yang memiliki atribut kunci disebut sebagai **jenis entitas kuat** (*Strong Entity Type*). Jenis-jenis entitas kuat telah kita bahas sebelumnya, seperti contoh entitas PERUSAHAAN, dan entitas

KARYAWAN merupakan contoh jenis entitas kuat. Entitas yang termasuk dalam jenis entitas lemah diidentifikasi dan dikaitkan dengan entitas tertentu dari jenis entitas lain dalam kombinasi dengan salah satu nilai atributnya. Jenis entitas lain ini disebut sebagai **jenis entitas pengidentifikasi atau pemilik**, dan jenis relasi yang menghubungkan jenis entitas lemah dengan pemiliknya disebut sebagai **relasi pengidentifikasian** jenis entitas lemah. Jenis entitas lemah selalu memiliki batasan partisipasi total (**ketergantungan eksistensi**) sehubungan dengan relasi pengidentifikasiannya karena entitas lemah tidak dapat diidentifikasi tanpa entitas pemilik. Untuk materi ketergantungan eksistensi akan dibahas pada bagian 3.4. Namun, tidak setiap ketergantungan eksistensi menghasilkan jenis entitas lemah. Misalnya, entitas IJIN\_MENGEMUDI tidak bisa ada kecuali jika dihubungkan dengan entitas ORANG, dan entitas tersebut memiliki kuncinya sendiri (nomor\_ijin) sehingga bukan termasuk entitas lemah.

Pertimbangkan jenis entitas TANGGUNGAN, terkait dengan KARYAWAN, yang digunakan untuk melacak tanggungan setiap karyawan melalui relasi 1:N (*one-to-many*).

Contoh:

Atribut pada entitas TANGGUNGAN adalah Nama (nama depan yang ditanggung), Tgl\_Lhr (tanggal lahir), J\_Kel (jenis kelamin), dan Hubungan (jenis hubungannya dengan karyawan). Dua tanggungan dari dua karyawan yang berbeda mungkin, secara kebetulan, memiliki nilai yang sama untuk Nama, Tgl\_Lhr, J\_Kel, dan Hubungan, tetapi mereka masih merupakan entitas yang berbeda. Mereka diidentifikasi sebagai entitas yang berbeda hanya setelah menentukan entitas karyawan tertentu yang terkait dengan setiap tanggungan. Setiap entitas karyawan dikatakan memiliki entitas tanggungan yang terkait dengannya.

Jenis entitas lemah biasanya memiliki **kunci parsial**, yang merupakan atribut yang secara unik dapat mengidentifikasi entitas lemah yang terkait dengan entitas pemilik yang sama.

Contoh:

jika diasumsikan bahwa tidak ada dua tanggungan dari karyawan yang sama yang memiliki nama depan yang sama, atribut Nama dari TANGGUNGAN adalah kunci parsial. Dalam kasus terburuk, atribut komposit dari semua atribut entitas lemah akan menjadi kunci parsial.

Dalam diagram E-R, baik jenis entitas lemah dan relasi pengidentifikasiannya dibedakan dengan mengelilingi kotak dan belah ketupat dengan garis ganda (lihat Gambar 3.1). Atribut kunci parsial digarisbawahi dengan garis putus-putus.

### 3.4 Atribut

Setiap entitas dijelaskan oleh sekumpulan atribut (misalnya, Karyawan = (Nama, Alamat, Usia, Gaji). Setiap atribut memiliki nama, dan dikaitkan dengan entitas dan domain nilai yang legal. Namun, informasi tentang domain atribut tidak disajikan pada diagram E-R. Dalam diagram E-R yang ditunjukkan pada Gambar 3.1, setiap atribut diwakili oleh gambar oval dengan nama di dalamnya.

Dalam model E-R, dikenal ada beberapa jenis atribut, yaitu atribut sederhana versus atribut komposit, atribut bernilai tunggal versus atribut bernilai banyak, dan atribut tersimpan versus atribut turunan.

- 1) Atribut Sederhana (*Simple Attribute*) versus Atribut komposit (*Composite Attribute*)

**Atribut sederhana** adalah atribut yang diambil dari domain nilai atomik. Jenis atribut ini juga disebut dengan atribut bernilai tunggal, atau atribut yang tidak dapat dibagi-bagi lagi menjadi sub-sub atribut yang lebih sederhana.

Contoh:

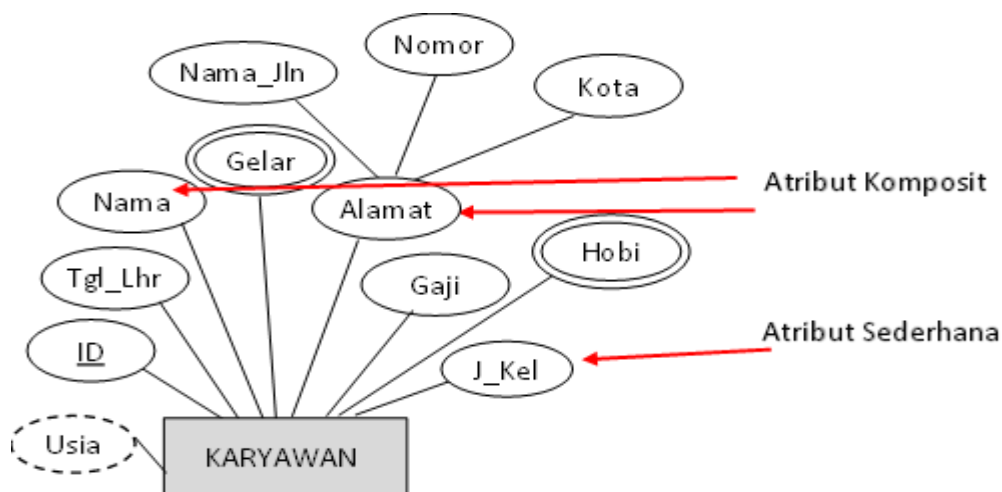
Berdasarkan Gambar 3.4, yang termasuk atribut sederhana dalam basis data PERUSAHAAN pada entitas KARYAWAN, adalah Gaji = {2.000.000}, J\_Kel = {Laki-laki}, dan Usia = {29}.

**Atribut komposit** adalah atribut yang terdiri dari hierarki atribut atau atribut yang masih dapat diuraikan lagi menjadi sub-sub atribut yang lebih sederhana yang masing-masing memiliki makna. Atribut komposit juga disebut dengan atribut bernilai majemuk.

Contoh:

Pada Gambar 3.4, yang termasuk atribut komposit adalah Alamat dan Nama. Alamat dapat terdiri dari Nama\_Jln, Nomor, dan Kota. Sedangkan atribut Nama dapat terdiri dari NamaD (nama depan), NamaT (nama tengah), dan NamaB (nama belakang). Berdasarkan Gambar 3.2, nilai dari atribut alamat, adalah Nama\_Jln = {Jl. Ngurah Rai}, Nomor = {No. 5}, dan Kota = {Singaraja}. Sedangkan nilai dari atribut nama, adalah NamaD = {Putu}, dan NamaB =

{Kenyon}. Dalam kondisi ini, terkadang ada beberapa nama seseorang yang tidak berisi nama tengah.



Gambar 3.4 Contoh atribut sederhana dan atribut komposit pada entitas KARYAWAN

## 2) Atribut Bernilai Tunggal (*Single-Valued Attribute*) versus Atribut Bernilai Banyak (*Multivalued Attribute*)

Sebagian besar atribut memiliki nilai tunggal untuk entitas tertentu; atribut seperti itu disebut **atribut bernilai tunggal**. Misalnya, Usia adalah atribut bernilai tunggal dari seseorang. Atribut lain yang bisa dikatakan memiliki nilai tunggal adalah Gaji, dan J\_Kel (jenis kelamin). Dapat dikatakan bahwa atribut bernilai tunggal hampir sama dengan atribut bernilai sederhana. Dalam beberapa kasus, sebuah atribut dapat memiliki sekumpulan nilai untuk entitas yang sama. Misalnya, atribut Warna untuk mobil, dan atribut Gelar atau atribut Hobi untuk seseorang. Mobil dengan satu warna memiliki nilai tunggal, sedangkan mobil dengan dua warna memiliki dua nilai warna. Demikian pula untuk gelar, satu orang mungkin tidak memiliki gelar sarjana, orang lain mungkin memilikinya, dan orang ketiga mungkin memiliki dua gelar atau lebih. Oleh karena itu, orang yang berbeda dapat memiliki jumlah nilai yang berbeda untuk atribut gelar. Begitu juga dengan hobi, dimana seseorang pasti memiliki hobi. Akan tetapi terkadang ada beberapa orang yang memiliki hobi lebih dari satu. Atribut seperti itu disebut **atribut bernilai banyak**. Atribut bernilai banyak mungkin memiliki batas bawah dan atas untuk membatasi jumlah nilai yang diizinkan untuk setiap entitas individu. Misalnya, atribut Warna dari sebuah mobil mungkin dibatasi untuk

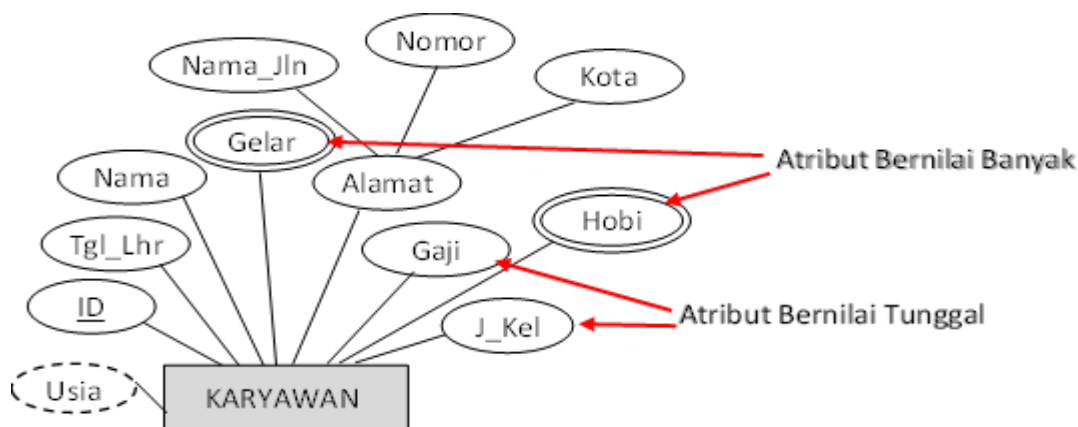


memiliki antara satu dan dua nilai, jika kita berasumsi bahwa sebuah mobil dapat memiliki paling banyak dua warna.

Contoh:

Berdasarkan Gambar 3.5, yang termasuk sebagai atribut bernilai tunggal, adalah atribut J\_Kel yang merupakan atribut jenis kelamin, atribut Usia, dan atribut Gaji. Sebagai catatan bahwa, setiap orang di dunia ini hanya memiliki satu jenis kelamin. Nilai dari atribut ini, yaitu  $J\_Kel = \{\text{Laki-laki}\}$  atau kemungkinan  $J\_Kel = \{\text{Perempuan}\}$ . Sedangkan untuk nilai atribut Usia, misalnya  $Usia = \{28\}$ . Dan untuk atribut Gaji, misalnya  $Gaji = \{2.000.000\}$

Untuk atribut bernilai banyak (*multivalued attribute*), adalah gelar dan hobi. Seperti dijelaskan sebelumnya, bahwa seseorang mungkin memiliki satu gelar, atau lebih dari satu gelar, begitu juga berlaku untuk atribut hobi. Nilai untuk atribut gelar, adalah  $Gelar = \{\text{S. Kom., M. Kom., Ph. D}\}$ , dan nilai untuk atribut hobi, yaitu  $Hobi = \{\text{memasak, olah raga}\}$ . Untuk atribut bernilai banyak disimbolkan dengan gambar berbentuk oval dengan garis ganda (*double*).

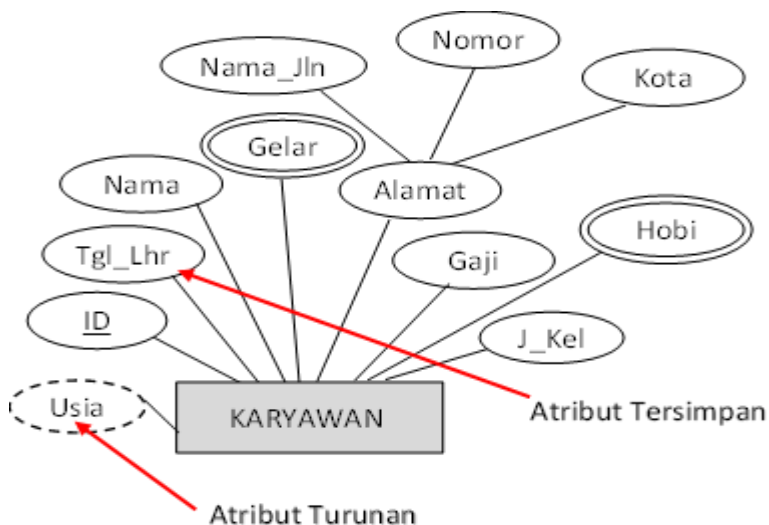


Gambar 3.5 Contoh atribut bernilai tunggal dan atribut bernilai banyak pada entitas KARYAWAN

### 3) Atribut Tersimpan (*Stored Attribute*) dan Atribut Turunan (*Derivied Attribute*)

Dalam beberapa kasus, dua (atau lebih) nilai atribut saling berkaitan. Misalnya, atribut Usia dan Tgl\_Lhr (tanggal lahir) seseorang. Untuk entitas orang tertentu, nilai atribut Usia dapat ditentukan dari tanggal saat ini (hari ini) dan nilai dari atribut Tgl\_Lhr orang tersebut. Oleh karena itu, atribut Usia disebut **atribut turunan** karena nilainya diturunkan dari atribut Tgl\_Lhr, yang disebut **atribut**

**tersimpan.** Untuk lebih jelasnya dapat dilihat pada Gambar 3.6. Untuk atribut turunan disimbolkan dengan gambar berbentuk oval dengan garis putus-putus.



Gambar 3.6 Contoh atribut tersimpan dan atribut turunan pada entitas KARYAWAN

#### 4) Nilai *NULL* (Non Mandatoty Attribute)

**Nilai NULL.** Dalam beberapa kasus, entitas tertentu mungkin tidak memiliki nilai yang berlaku untuk suatu atribut. Misalnya, atribut NamaT (nama tengah) dari nama seseorang karyawan. Mungkin saja, nama seseorang hanya terdiri dari satu suku kata atau dua suku kata. Demikian pula untuk atribut Gelar, hanya berlaku untuk orang yang memiliki gelar apakah itu gelar sarjana, gelar master atau gelar doktor. Untuk situasi seperti itu, nilai khusus yang disebut NULL dibuat. Atribut NamaT akan memiliki nilai NULL, dan seseorang yang tidak memiliki gelar akan memiliki nilai NULL pada atribut Gelar. Nilai NULL juga dapat digunakan jika kita tidak mengetahui nilai atribut untuk entitas tertentu, misalnya, jika nomor telepon dari 'Putu Kenyot' pada Gambar 3.2 tidak diketahui.

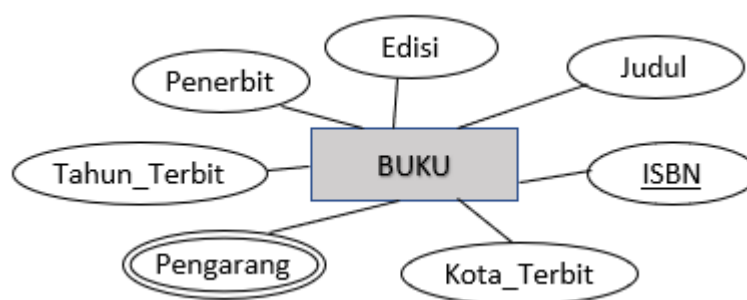
### 3.5 Kunci (Key)

**Atribut Kunci dari Jenis Entitas.** Batasan penting pada entitas dari sebuah jenis entitas adalah kunci atau batasan keunikan pada atribut. Jenis entitas biasanya memiliki satu atau lebih atribut yang nilainya berbeda/unik untuk setiap entitas individu dalam himpunan entitas. Atribut seperti itu disebut **atribut kunci**, dan nilainya dapat digunakan untuk mengidentifikasi setiap entitas secara unik. Misalnya, atribut Nama\_Per adalah kunci pada jenis entitas PERUSAHAAN pada Gambar 3.3 karena tidak ada dua perusahaan yang boleh memiliki nama yang sama. Untuk jenis entitas

KARYAWAN, atribut kunci khusus adalah ID. Terkadang beberapa atribut bersama-sama membentuk kunci, artinya kombinasi nilai atribut harus berbeda untuk setiap entitas.

Menentukan bahwa atribut adalah kunci dari jenis entitas berarti bahwa properti keunikan sebelumnya harus dimiliki untuk setiap kumpulan entitas dari jenis entitas. Oleh karena itu, ini adalah batasan yang melarang dua entitas memiliki nilai yang sama untuk atribut kunci pada saat yang sama. Ini bukan milik dari kumpulan entitas tertentu. Ini adalah batasan pada kumpulan entitas apa pun dari jenis entitas pada titik waktu kapan pun. Batasan kunci ini diturunkan dari batasan *miniworld* yang diwakili oleh basis data.

Sebagai contoh yang ditunjukkan pada Gambar 3.7, dimana entitas BUKU memiliki beberapa atribut. Entitas BUKU ini memiliki atribut yang unik yaitu ISBN. Semua entitas buku memiliki atribut ISBN, atribut ini unik untuk setiap buku yang ada. Dalam notasi diagram E-R (Gambar 3.7 (a)), ada sebuah atribut yang digarisbawahi yaitu ISBN yang merupakan kuncinya.



(a)

(Pengarang), Tahun\_Terbit, Judul, Edisi, ISBN, Kota\_Terit, Penerbit



Gambar 3.7 Jenis entitas BUKU dengan atribut kunci yaitu ISBN. (a) Notasi Diagram E-R, dan (b) Himpunan Entitas dengan tiga entitas individu

### 3.6 Domain Atribut

**Kumpulan Nilai (Domain) dari Atribut.** Setiap atribut sederhana dari jenis entitas dikaitkan dengan kumpulan nilai (atau domain nilai), yang menentukan kumpulan nilai yang dapat ditetapkan ke atribut tersebut untuk setiap entitas individu. Pada Gambar 3.3, jika rentang usia yang diizinkan untuk karyawan adalah antara 20 dan 50, dapat ditentukan kumpulan nilai atribut Usia dari KARYAWAN menjadi himpunan bilangan bulat antara 20 dan 50. Demikian pula, kita dapat menentukan kumpulan nilai untuk atribut Nama\_Kar menjadi kumpulan string yang berisi karakter alfabet yang dipisahkan oleh karakter kosong, dan seterusnya. Kumpulan nilai biasanya tidak ditampilkan dalam diagram E-R dasar dan mirip dengan tipe data dasar yang tersedia di sebagian besar bahasa pemrograman, seperti *Integer*, *String*, *Boolean*, *Float*, tipe *Auto\_Number*, dan sebagainya. Tipe data tambahan untuk mewakili tipe data seperti *Date*, *Time*, dan konsep lainnya juga digunakan.

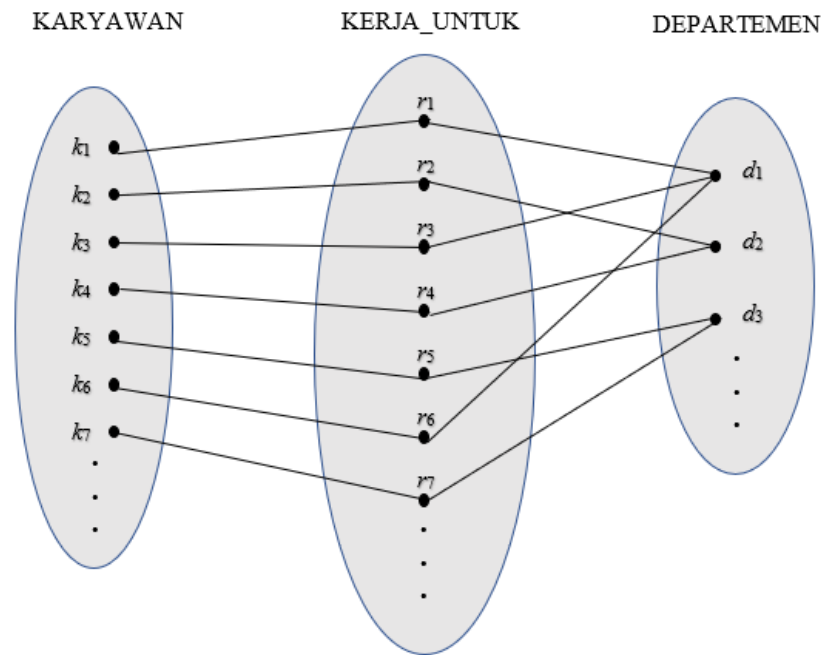
### 4. Jenis Relasi (*Relationship Type*), Himpunan Relasi (*Relationship Set*), Peran (*Role*), dan Batasan Struktural (*Structural Constraint*)

Pada Gambar 3.8 – 3.11, ada beberapa hubungan implisit di antara berbagai jenis entitas. Faktanya, setiap kali atribut dari satu jenis entitas merujuk ke jenis entitas lain, beberapa relasi ada. Misalnya, atribut Manajer pada entitas DEPARTEMEN mengacu pada karyawan yang mengelola departemen; atribut Departemen\_Pengontrol pada entitas PROYEK mengacu pada departemen yang mengontrol proyek; atribut Pengawas pada entitas KARYAWAN mengacu pada karyawan lain (orang yang mengawasi karyawan ini); atribut Departemen pada entitas KARYAWAN mengacu pada departemen tempat karyawan bekerja; dan seterusnya. Dalam model E-R, referensi ini tidak boleh direpresentasikan sebagai atribut tetapi sebagai relasi. Skema basis data PERUSAHAAN pertama pada Gambar 3.8-3.11 akan disempurnakan untuk mewakili relasi secara eksplisit. Dalam desain awal jenis entitas, relasi biasanya ditangkap dalam bentuk atribut. Saat desain disempurnakan, atribut ini diubah menjadi relasi antara jenis entitas.

#### 4.1 Jenis Relasi, Himpunan Relasi, dan *Instance*

Sebuah relasi  $R$  di antara  $n$  buah jenis entitas  $K_1, K_2, \dots, K_n$  mendefinisikan sebuah himpunan asosiasi atau himpunan relasi antar entitas dari jenis entitas ini. Serupa dengan kasus jenis entitas dan himpunan entitas, jenis relasi dan himpunan relasi yang sesuai biasanya disebut dengan nama yang sama, yaitu  $R$ . Secara matematis, himpunan relasi  $R$  adalah himpunan *instance* relasi  $r_i$ , di mana setiap  $r_i$  menghubungkan  $n$  buah entitas secara individu yaitu  $(k_1, k_2, \dots, k_n)$ , dan setiap entitas  $k_j$  di  $r_i$  adalah anggota himpunan entitas  $K$ ,  $1 \leq j \leq n$ . Oleh karena itu, himpunan relasi adalah relasi matematis pada  $K_1, K_2, \dots, K_n$ ; alternatifnya, dapat didefinisikan sebagai sub himpunan dari produk Cartesian dari himpunan entitas  $K_1 \times K_2 \times \dots \times K_n$ . Masing-masing jenis entitas  $K_1, K_2, \dots, K_n$  dikatakan berpartisipasi pada jenis relasi  $R$ ; sama halnya dengan, masing-masing entitas secara individu yaitu  $k_1, k_2, \dots, k_n$  dikatakan berpartisipasi dalam *instance* relasi  $r_i = (k_1, k, \dots, k_n)$

Secara informal, setiap *instance* relasi  $r_i$  di  $R$  adalah himpunan entitas, di mana himpunan entitas tersebut menunjuk tepat satu entitas dari setiap jenis entitas yang berpartisipasi. Setiap *instance* relasi seperti  $r_i$  mewakili fakta bahwa entitas yang berpartisipasi dalam  $r_i$  terkait dalam beberapa cara dalam situasi *miniworld* yang sesuai. Misalnya, pertimbangkan jenis relasi KERJA\_UNTUK antara dua jenis entitas KARYAWAN dan DEPARTEMEN, yang menghubungkan setiap karyawan dengan departemen tempat karyawan tersebut bekerja. Setiap *instance* relasi dalam himpunan relasi KERJA\_UNTUK menghubungkan satu entitas KARYAWAN dan satu entitas DEPARTEMEN. Gambar 3.12 mengilustrasikan contoh ini, di mana setiap *instance* relasi  $r_i$  ditampilkan terhubung ke entitas KARYAWAN dan DEPARTEMEN yang berpartisipasi dalam  $r_i$ .



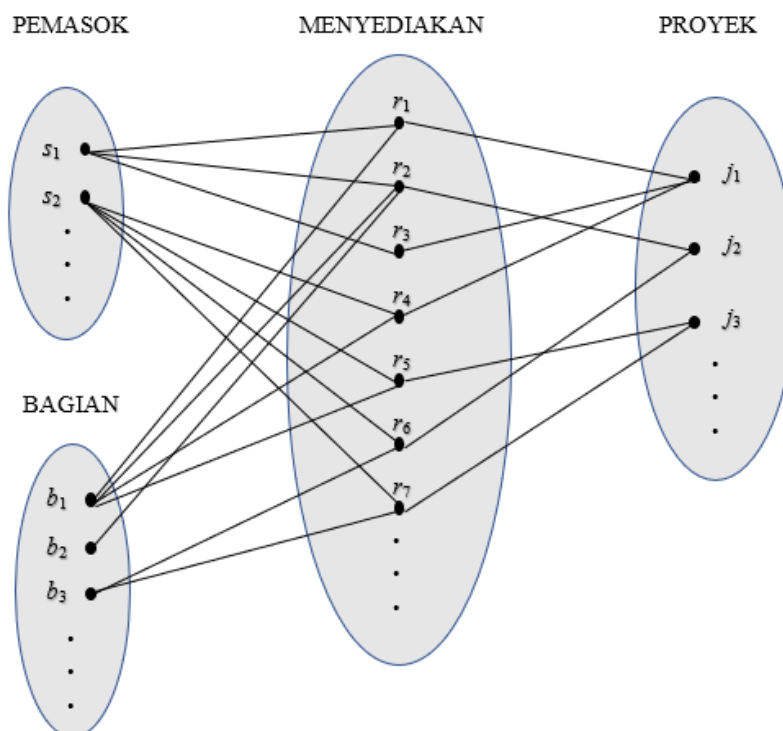
Gambar 3.12 Beberapa *instance* dalam himpunan relasi KERJA\_UNTUK, yang menunjukkan jenis relasi KERJA\_UNTUK antara KARYAWAN dan DEPARTEMEN

Dalam *miniworld* yang diwakili oleh Gambar 3.12, karyawan  $k_1$ ,  $k_3$ , dan  $k_6$  bekerja untuk departemen  $d_1$ ; karyawan  $k_2$  dan  $k_4$  bekerja untuk departemen  $d_2$ ; dan karyawan  $k_5$  dan  $k_7$  bekerja untuk departemen  $d_3$ .

Dalam diagram E-R, jenis relasi ditampilkan sebagai belah ketupat, yang dihubungkan oleh garis lurus ke kotak persegi panjang yang mewakili jenis entitas yang berpartisipasi. Nama relasi ditampilkan dalam belah ketupat tersebut (lihat Gambar 3.1).

#### 4.2 Derajat Relasi, Nama Peran, dan Relasi Rekursif

**Derajat relasi** adalah jumlah entitas yang berpartisipasi. Oleh karena itu, relasi KERJA\_UNTUK adalah derajat dua. Jenis relasi yang memiliki derajat dua disebut relasi **biner**, dan jenis relasi yang memiliki derajat tiga disebut relasi **ternary**. Contoh relasi ternary adalah MENYEDIAKAN, ditunjukkan pada Gambar 3.13, di mana setiap *instance* relasi  $r_i$  menghubungkan tiga entitas, yaitu pemasok  $s$ , bagian  $b$ , dan proyek  $j$ , kapanpun  $s$  memasok dari bagian  $b$  ke proyek  $j$ . Derajat relasi yang umum digunakan adalah relasi biner.



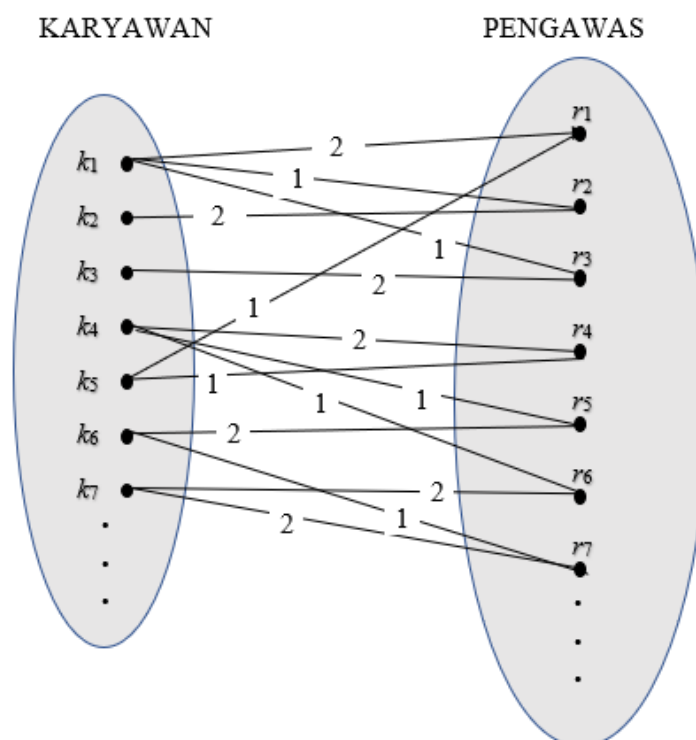
Gambar 3.13 Beberapa *instance* relasi pada himpunan relasi *ternary* MENYEDIAKAN

**Relasi sebagai Atribut.** Kadang-kadang nyaman untuk memikirkan jenis relasi biner dalam hal atribut. Perhatikan jenis relasi KERJA\_UNTUK pada Gambar 3.12. Seseorang dapat memikirkan atribut yang disebut Departemen pada jenis entitas KARYAWAN, di mana nilai Departemen untuk setiap entitas KARYAWAN adalah (menunjuk ke) entitas DEPARTEMEN tempat karyawan tersebut bekerja. Oleh karena itu, nilai yang ditetapkan untuk atribut Departemen ini adalah himpunan semua entitas DEPARTEMEN, yang merupakan himpunan entitas DEPARTEMEN. Inilah yang dilakukan pada Gambar 3.8-3.11 ketika membuat desain awal dari jenis entitas KARYAWAN untuk basis data PERUSAHAAN.

**Nama Peran dan Relasi Rekursif.** Setiap jenis entitas yang berpartisipasi dalam jenis relasi memainkan peran tertentu dalam relasi. Nama peran menandakan peran yang dimainkan oleh entitas yang berpartisipasi dalam setiap *instance* relasi, dan ini membantu menjelaskan apa arti dari relasi tersebut. Misalnya, dalam jenis relasi KERJA\_UNTUK, KARYAWAN berperan sebagai karyawan atau pekerja dan DEPARTEMEN berperan sebagai departemen atau pemberi kerja.

Nama peran secara teknis tidak diperlukan dalam jenis relasi di mana semua jenis entitas yang berpartisipasi berbeda, karena setiap nama dari jenis entitas yang berpartisipasi dapat digunakan sebagai nama peran. Namun, dalam beberapa kasus, jenis

entitas yang sama berpartisipasi lebih dari satu kali dalam jenis relasi dengan peran yang berbeda. Dalam kasus seperti itu, nama peran menjadi penting untuk membedakan makna peran yang dimainkan oleh setiap entitas yang berpartisipasi. Jenis relasi seperti itu disebut **relasi rekursif** atau relasi yang menunjuk diri sendiri. Gambar 3.14 menunjukkan sebuah contoh. Jenis relasi PENGAWASAN menghubungkan karyawan dengan pengawas, di mana entitas karyawan dan pengawas adalah anggota dari himpunan entitas KARYAWAN yang sama. Oleh karena itu, jenis entitas KARYAWAN berpartisipasi dua kali dalam PENGAWASAN: sekali dalam peran pengawas (atau pimpinan), dan sekali dalam peran sebagai orang yang diawasi (atau bawahan). Setiap *instance* relasi  $r_i$  dalam PENGAWASAN menghubungkan dua entitas karyawan yang berbeda  $k_j$  dan  $k_l$ , salah satunya berperan sebagai pengawas dan yang lainnya berperan sebagai orang yang diawasi. Pada Gambar 3.14, garis bertanda '1' mewakili peran pengawas, dan garis bertanda '2' mewakili peran orang yang diawasi; maka,  $k_1$  mengawasi  $k_2$  dan  $k_3$ ,  $k_4$  mengawasi  $k_6$  dan  $k_7$ , dan  $k_5$  mengawasi  $k_1$  dan  $k_4$ .



Gambar 3.14 Relasi rekursif dari PENGAWASAN antara KARYAWAN dalam perannya (1) sebagai pengawas dan KARYAWAN dalam perannya (2) sebagai orang yang diawasi. Dalam contoh ini, setiap contoh relasi harus terhubung dengan dua garis, satu ditandai dengan '1' (pengawas) dan yang lainnya dengan '2' (orang yang diawasi).



### 4.3 Batasan (*Constraint*) pada Jenis Relasi Biner (*Binary Relationship Type*)

Jenis relasi biasanya memiliki batasan tertentu yang membatasi kemungkinan kombinasi entitas yang dapat berpartisipasi dalam himpunan relasi yang sesuai. Batasan ini ditentukan dari situasi *miniworld* yang diwakili oleh relasi. Misalnya, pada Gambar 3.12, jika perusahaan memiliki aturan bahwa setiap karyawan harus bekerja tepat untuk satu departemen, maka batasan ini akan digambarkan dalam bentuk skema. Dalam hal ini, batasan relasi biner akan dibedakan menjadi dua jenis, yaitu: **rasio kardinalitas dan partisipasi**.

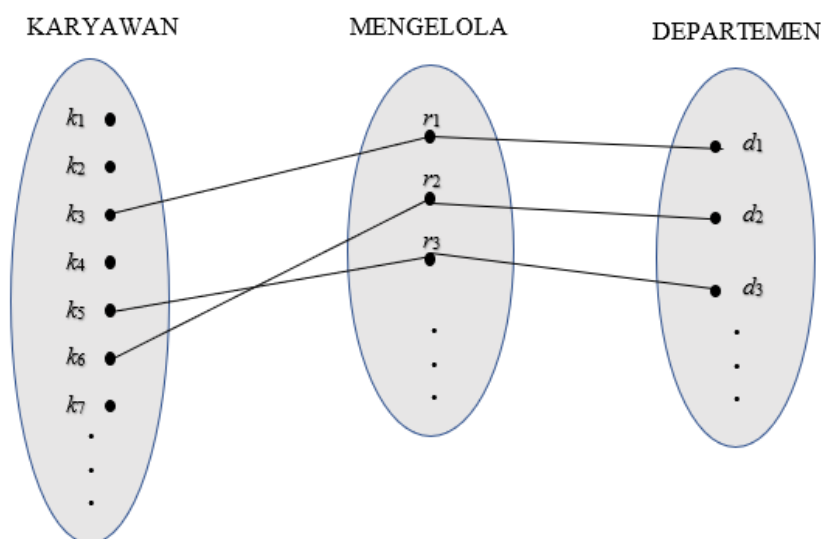
**Rasio Kardinalitas untuk Relasi Biner.** Rasio kardinalitas untuk relasi biner menentukan jumlah maksimum *instance* relasi yang dapat diikuti oleh entitas. Misalnya, dalam jenis relasi biner antara relasi KERJA\_UNTUK, dan relasi DEPARTEMEN: KARYAWAN memiliki **rasio kardinalitas 1:N** (*one-to-many*), yang artinya setiap departemen (1) dapat dikaitkan/dihubungkan dengan (yaitu, mempekerjakan) sejumlah/banyak karyawan (N), tetapi seorang karyawan dapat dikaitkan dengan (bekerja untuk) paling banyak satu departemen (1). Ini berarti bahwa untuk jenis relasi KERJA\_UNTUK, entitas departemen tertentu dapat dikaitkan dengan sejumlah karyawan (N menunjukkan tidak ada jumlah maksimum). Di sisi lain, seorang karyawan dapat dikaitkan dengan maksimal satu departemen. **Rasio kardinalitas yang mungkin untuk jenis relasi biner** adalah 1:1 (*one-to-one*), 1:N (*one-to-many*), N:1 (*many-to-one*), dan M:N (*many-to-many*).

Contoh

- 1) **Relasi biner 1:1** (*one-to-one*). Relasi biner 1:1 dapat terjadi pada jenis relasi MENGELOLA (Gambar 3.15), yang menghubungkan entitas departemen dengan entitas karyawan yang mengelola departemen tersebut. Ini mewakili batasan *miniworld*, dimana seorang karyawan dapat mengelola paling banyak satu departemen dan satu departemen dapat memiliki paling banyak satu manajer.
- 2) **Relasi biner M:N**. Jenis relasi KERJA\_DI (Gambar 3.16) adalah rasio kardinalitas M:N, dalam aturan *miniworld* bahwa seorang karyawan dapat mengerjakan beberapa proyek dan sebuah proyek dapat dikerjakan oleh beberapa karyawan.

Rasio kardinalitas untuk relasi biner direpresentasikan pada diagram E-R dengan menampilkan 1, M, dan N pada simbol bentuk belah ketupat seperti yang ditunjukkan pada Gambar 3.1. Perhatikan dalam notasi ini, bahwa maksimum (N) tidak dapat ditentukan atau maksimum satu (1) pada partisipasi.

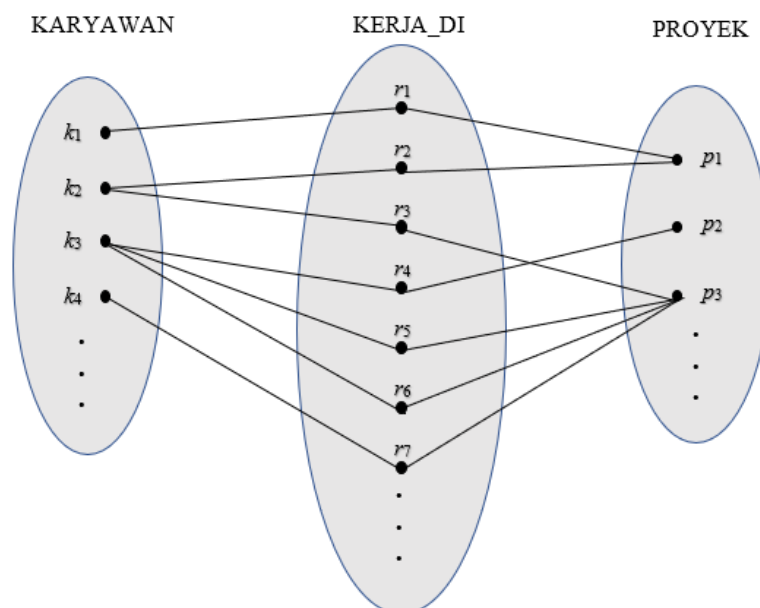
**Batasan Partisipasi dan Ketergantungan Eksistensi.** Batasan partisipasi menentukan apakah keberadaan/eksistensi suatu entitas bergantung pada hubungannya dengan entitas lain melalui jenis relasi. Batasan ini menentukan jumlah minimum *instance* relasi yang dapat diikuti oleh setiap entitas dan kadang-kadang disebut **batasan kardinalitas minimum**. Ada dua jenis batasan partisipasi, yaitu **total dan parsial**. Jika kebijakan perusahaan menyatakan bahwa setiap karyawan harus bekerja untuk suatu departemen, maka entitas karyawan hanya dapat ada jika berpartisipasi dalam setidaknya satu *instance* relasi KERJA\_UNTUK (Gambar 3.12). Dengan demikian, partisipasi KARYAWAN pada relasi KERJA\_UNTUK disebut **partisipasi total**, artinya setiap entitas dalam himpunan entitas karyawan harus terhubung dengan entitas departemen melalui KERJA\_UNTUK. **Partisipasi total** juga disebut **ketergantungan eksistensi**. Pada Gambar 3.15 kita tidak mengharuskan setiap karyawan untuk mengelola suatu departemen, sehingga partisipasi KARYAWAN dalam jenis relasi MENGELOLA bersifat parsial, artinya beberapa atau sebagian himpunan entitas karyawan terhubung dengan beberapa entitas departemen melalui MENGELOLA, tetapi tidak harus selalu semua. Jadi ada karyawan yang tidak mengelola (menjadi manajer) sebuah departemen. Kita akan mengacu pada rasio kardinalitas dan batasan partisipasi, secara bersama-sama, sebagai batasan struktural dari jenis relasi.



Gambar 3.15 Relasi 1:1 (*one-to-one*) yang terjadi antara KARYAWAN dan DEPARTEMEN pada jenis relasi MENGELOLA

Dalam diagram E-R, partisipasi total (atau ketergantungan eksistensi) ditampilkan sebagai garis ganda yang menghubungkan jenis entitas yang berpartisipasi ke relasi, sedangkan partisipasi parsial diwakili oleh satu baris (lihat Gambar 3.1). Perhatikan

bahwa dalam notasi ini, kita dapat menentukan tidak ada minimum (partisipasi parsial) atau minimal satu (partisipasi total).



Gambar 3.16 Relasi M:N, KERJA\_DI

#### 4.4 Atribut pada Jenis Relasi

Jenis relasi juga dapat memiliki atribut, mirip dengan jenis entitas.

Contoh:

1. Misalnya, untuk mencatat jumlah jam per minggu bahwa seorang karyawan tertentu bekerja pada proyek tertentu, kita dapat menyertakan atribut Jam untuk jenis relasi KERJA\_DI pada Gambar 3.16.
2. Untuk memasukkan tanggal manajer mulai mengelola sebuah departemen melalui atribut Tgl\_Mulai untuk jenis relasi MENGELOLA pada Gambar 3.15.

Perhatikan bahwa atribut jenis relasi 1:1 atau 1:N dapat dimigrasikan ke salah satu jenis entitas yang berpartisipasi. Misalnya, atribut Tgl\_Mulai untuk relasi MENGELOLA dapat berupa atribut KARYAWAN (manajer) atau DEPARTEMEN, meskipun secara konseptual itu milik MENGELOLA. Hal ini karena MENGELOLA adalah relasi 1:1, jadi setiap departemen atau entitas karyawan berpartisipasi pada paling banyak satu contoh relasi. Oleh karena itu, nilai atribut Tgl\_Mulai dapat ditentukan secara terpisah, baik oleh entitas departemen yang berpartisipasi atau oleh entitas karyawan (manajer) yang berpartisipasi.

Untuk jenis relasi 1:N, atribut relasi hanya dapat dimigrasikan ke jenis entitas di sisi-N relasi. Misalnya, pada Gambar 3.12, jika relasi KERJA\_UNTUK juga memiliki atribut Tgl\_Mulai yang menunjukkan kapan seorang karyawan mulai bekerja pada sebuah departemen, atribut ini dapat dimasukkan sebagai atribut KARYAWAN. Oleh karena setiap karyawan bekerja paling banyak untuk satu departemen, maka dari itu karyawan berpartisipasi paling banyak pada satu *instance* relasi KERJA\_UNTUK, tetapi sebuah departemen dapat memiliki banyak karyawan, masing-masing dengan tanggal mulai yang berbeda. Dalam jenis relasi 1:1 dan 1:N, keputusan untuk menempatkan atribut relasi, sebagai atribut jenis relasi atau sebagai atribut dari jenis entitas yang berpartisipasi, dapat ditentukan secara subjektif oleh perancang skema.

Untuk jenis relasi M:N (*many-to-many*), beberapa atribut dapat ditentukan oleh kombinasi entitas yang berpartisipasi dalam *instance* relasi, bukan oleh entitas tunggal. Atribut tersebut harus ditentukan sebagai atribut relasi. Contohnya adalah atribut Jam dari relasi M:N pada KERJA\_DI (Gambar 3.16); jumlah jam per minggu seorang karyawan yang bekerja pada sebuah proyek ditentukan oleh kombinasi antara karyawan-proyek dan tidak secara terpisah oleh salah satu entitas.

## 5. Jenis Relasi dengan Derajat yang Lebih Tinggi dari Dua

Di bagian 4.3 telah didefinisikan derajat dari jenis relasi sebagai jumlah jenis entitas yang berpartisipasi dan jenis relasi dengan derajat dua disebut **biner**/*binary*, sedangkan jenis relasi yang berderajat tiga disebut **ternary**. Di bagian ini, akan diuraikan 1) perbedaan antara relasi biner dan jenis relasi dengan derajat yang lebih tinggi, 2) kapan harus memilih relasi dengan derajat yang lebih tinggi versus biner, dan 3) bagaimana menentukan batasan pada relasi dengan derajat yang lebih tinggi.

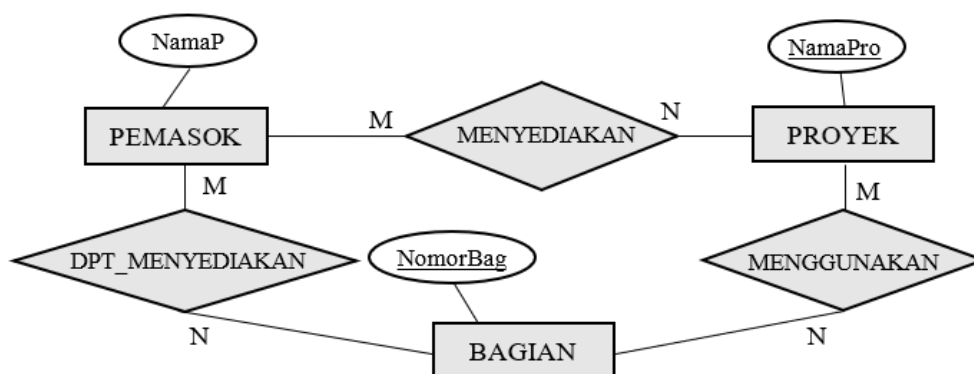
### 5.1 Memilih antara Relasi Biner dan Relasi Ternary (atau relasi dengan derajat yang lebih tinggi)

Notasi diagram E-R untuk jenis relasi *ternary* ditunjukkan pada Gambar 3.17 (a), yang menampilkan skema untuk jenis relasi MENYEDIAKAN, yang ditampilkan pada tingkat *instance* pada Gambar 3.13. Ingat bahwa himpunan relasi MENYEDIAKAN adalah himpunan *instance* relasi (s, j, b), di mana artinya adalah bahwa s adalah PEMASOK yang saat ini menyediakan BAGIAN p ke PROYEK j. Secara umum, jenis relasi R dengan derajat n akan memiliki n sisi dalam diagram E-R, satu sisi menghubungkan R ke setiap jenis entitas yang berpartisipasi.

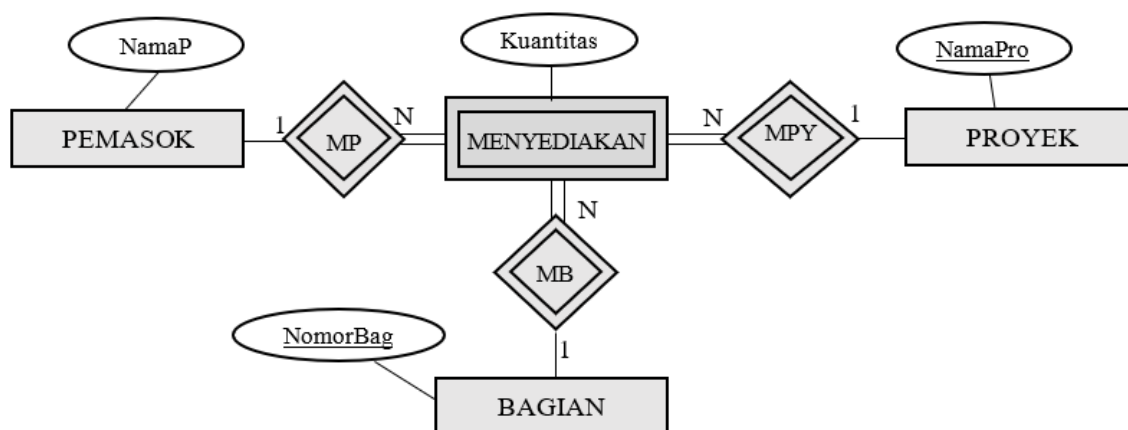
Gambar 3.17 (b) menunjukkan diagram E-R untuk jenis relasi *three binary* DPT\_MENYEDIAKAN, MENGGUNAKAN, dan MENYEDIAKAN. Secara umum, jenis relasi *ternary* mewakili informasi yang berbeda dari jenis relasi *three binary*. Pertimbangkan jenis relasi *three binary* DPT\_MENYEDIAKAN, MENGGUNAKAN, dan MENYEDIAKAN. Misalkan DPT\_MENYEDIAKAN, antara PEMASOK dan BAGIAN, menyertakan *instance* (s, b) kapan pun pemasok s dapat menyediakan bagian b (untuk proyek apa pun); MENGGUNAKAN, antara PROYEK dan BAGIAN, termasuk sebuah *instance* (j, b) setiap kali proyek j menggunakan bagian b; dan MENYEDIAKAN, antara PEMASOK dan PROYEK, termasuk *instance* (s, j) setiap kali pemasok menyediakan beberapa bagian ke proyek j. Keberadaan tiga *instance* relasi (s, b), (j, b), dan (s, j) pada relasi DPT\_MENYEDIAKAN, MENGGUNAKAN, dan MENYEDIAKAN, masing-masing, tidak selalu menyiratkan bahwa sebuah *instance* (s, j, b) ada pada relasi *ternary* MENYEDIAKAN, karena artinya berbeda. Seringkali sulit untuk memutuskan apakah relasi tertentu harus direpresentasikan sebagai jenis relasi derajat  $n$  atau harus dipecah menjadi beberapa jenis relasi dengan  $n$  derajat yang lebih kecil. Perancang harus membuat keputusan ini pada semantik atau makna dari situasi tertentu yang diwakili. Solusi khusus adalah memasukkan relasi *ternary* ditambah satu atau lebih relasi biner, jika mereka mewakili arti yang berbeda dan jika semua dibutuhkan oleh aplikasi.



(a)

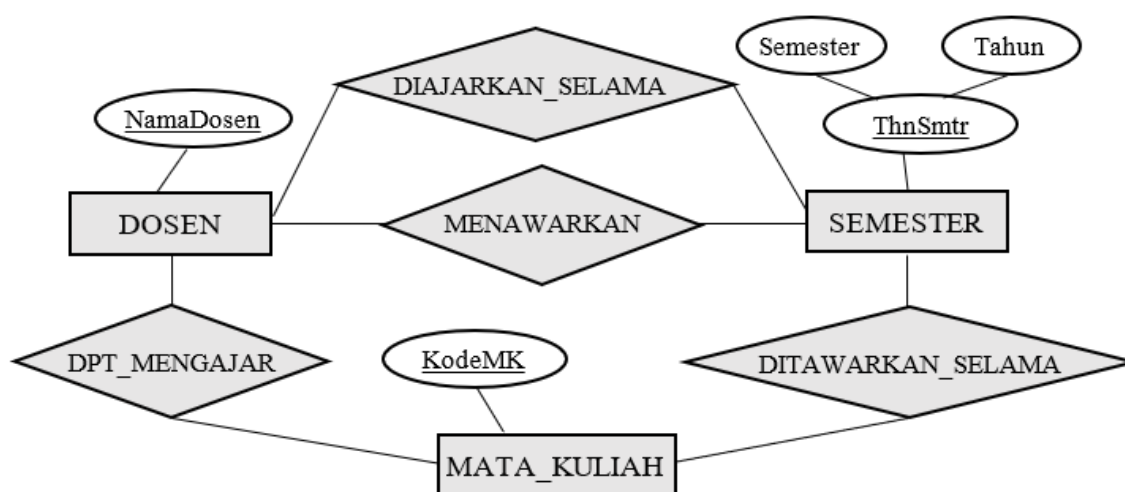


(b)



(c)

Gambar 3.17 Jenis relasi *Ternary*, (a) relasi MENYEDIAKAN, (b) relasi *three binary* yang tidak ekuivalen terhadap relasi MENYEDIAKAN, dan (c) relasi MENYEDIAKAN yang direpresentasikan sebagai jenis entitas lemah



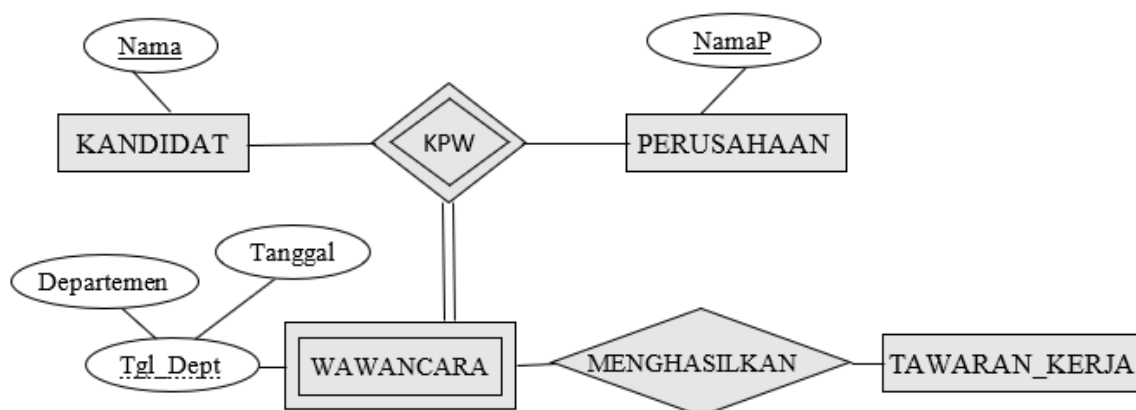
Gambar 3.18 Contoh lain jenis relasi *ternary* vs relasi *binary*

Beberapa alat desain basis data didasarkan pada variasi model E-R yang hanya mengijinkan relasi biner. Dalam hal ini, relasi *ternary* seperti MENYEDIAKAN harus direpresentasikan sebagai jenis entitas lemah, tanpa kunci parsial dan dengan tiga relasi pengidentifikasi. Tiga jenis entitas yang berpartisipasi PEMASOK, BAGIAN, dan PROYEK bersama-sama merupakan jenis entitas pemilik (lihat Gambar 3.17 (c)). Oleh karena itu, entitas dalam jenis entitas lemah MENYEDIAKAN pada Gambar 3.17 (c) diidentifikasi dengan kombinasi tiga entitas pemiliknya yaitu PEMASOK, BAGIAN, dan PROYEK. Hal ini juga memungkinkan untuk mewakili relasi *ternary* sebagai jenis entitas reguler dengan memperkenalkan kunci buatan atau pengganti. Dalam contoh ini, atribut kunci ID\_Sedia dapat digunakan untuk jenis entitas menyediakan, mengubahnya menjadi jenis entitas reguler. Tiga relasi biner N:1 menghubungkan MENYEDIAKAN ke masing-masing dari tiga jenis entitas yang berpartisipasi.

Contoh lain ditunjukkan pada Gambar 3.18. Jenis relasi *ternary* PENAWARAN mewakili informasi tentang dosen yang menawarkan mata kuliah selama semester tertentu; maka dari itu termasuk *instance* relasi  $(i, s, m)$  setiap kali DOSEN  $i$  menawarkan MATA KULIAH  $m$  selama SEMESTER  $s$ . Tiga jenis relasi biner yang ditunjukkan pada Gambar 3.18 memiliki arti sebagai berikut: DPT\_MENGAJAR menghubungkan mata kuliah dengan dosen yang dapat mengajar mata kuliah tersebut, DIAJARKAN\_SELAMA menghubungkan semester dengan dosen yang mengajar beberapa mata kuliah selama semester tersebut, dan DITAWARKAN\_SELAMA menghubungkan semester dengan mata kuliah yang ditawarkan selama semester itu oleh dosen manapun. Relasi *ternary* dan biner ini mewakili informasi yang berbeda, tetapi batasan tertentu harus ada di antara relasi tersebut. Misalnya, *instance* relasi  $(i, s, m)$  tidak boleh ada di MENAWARKAN kecuali ada *instance*  $(i, s)$  pada DIAJARKAN\_SELAMA, *instance*  $(s, m)$  ada pada DITAWARKAN\_SELAMA, dan *instance*  $(i, m)$  ada pada DPT\_MENGAJAR. Namun, kebalikannya tidak selalu benar; kita mungkin memiliki *instance*  $(i, s)$ ,  $(s, m)$ , dan  $(i, m)$  dalam tiga jenis relasi biner tanpa *instance* yang sesuai  $(i, s, m)$  di MENAWARKAN. Perhatikan bahwa dalam contoh ini, berdasarkan arti dari relasi, kita dapat menyimpulkan bahwa *instance* DIAJARKAN\_SELAMA dan DITAWARKAN\_SELAMA dari *instance* pada MENAWARKAN, tetapi kita tidak dapat menyimpulkan *instance* DPT\_MENGAJAR; oleh karena itu, DIAJARKAN\_SELAMA dan DITAWARKAN\_SELAMA berlebihan dan dapat diabaikan.

Meskipun secara umum tiga relasi biner tidak dapat menggantikan relasi *ternary*, mereka dapat melakukannya di bawah batasan tambahan tertentu. Dalam contoh ini, jika relasi DPT\_MENGAJAR adalah 1:1 (seorang dosen hanya dapat mengajar satu mata kuliah, dan satu mata kuliah hanya dapat diajarkan oleh satu dosen), maka relasi *ternary* MENAWARKAN dapat diabaikan karena dapat disimpulkan dari ketiganya, yaitu relasi biner DPT\_MENGAJAR, DIAJARKAN\_SELAMA, dan DITAWARKAN\_SELAMA. Perancang skema harus menganalisis arti dari setiap situasi spesifik untuk memutuskan apakah yang dibutuhkan jenis relasi biner atau *ternary*.

Perhatikan bahwa dimungkinkan untuk memiliki jenis entitas lemah dengan jenis relasi pengidentifikasi *ternary* (atau *n-ary*). Dalam hal ini, jenis entitas lemah dapat memiliki beberapa jenis entitas pemilik. Contohnya ditunjukkan pada Gambar 3.19. Contoh ini menunjukkan bagian dari basis data yang melacak kandidat/calon yang diwawancarai untuk pekerjaan di berbagai perusahaan, yang merupakan bagian dari basis data agen tenaga kerja. Dalam persyaratan, seorang kandidat dapat mengikuti beberapa kali wawancara dengan perusahaan yang sama (misalnya, dengan departemen perusahaan yang berbeda atau pada tanggal yang berbeda), tetapi tawaran pekerjaan dibuat berdasarkan salah satu wawancara. Di sini, WAWANCARA direpresentasikan sebagai entitas lemah dengan dua entitas pemilik, yaitu KANDIDAT dan PERUSAHAAN dengan kunci parsial Tgl\_Dept. Entitas WAWANCARA diidentifikasi secara unik oleh kandidat, perusahaan, dan kombinasi tanggal wawancara dan departemen yang melakukan wawancara.



Gambar 3.21 Jenis Entitas Lemah WAWANCARA dengan sebuah Jenis Relasi yang Teridentifikasi *Ternary*



## 5.2 Batasan pada Relasi *Ternary* (atau relasi dengan derajat yang lebih tinggi)

Ada dua notasi untuk menentukan batasan struktural pada relasi  $n$ -ary, dan mereka menentukan batasan yang berbeda. Dengan demikian keduanya harus digunakan jika penting sepenuhnya untuk menentukan batasan struktural pada relasi *ternary* atau tingkat yang lebih tinggi. Notasi pertama didasarkan pada notasi rasio kardinalitas relasi biner yang ditampilkan pada Gambar 3.1. Di sini, 1,  $M$ , atau  $N$  ditentukan pada setiap garis partisipasi (simbol  $M$  dan  $N$  mewakili banyak atau sembarang angka). Batasan ini akan diilustrasikan menggunakan relasi MENYEDIAKAN pada Gambar 3.17.

Ingat bahwa himpunan relasi MENYEDIAKAN adalah himpunan *instance* relasi  $(s, j, b)$ , di mana  $s$  adalah PEMASOK,  $j$  adalah PROYEK, dan  $b$  adalah BAGIAN. Misalkan ada batasan bahwa untuk kombinasi bagian proyek tertentu, hanya satu pemasok yang akan digunakan (hanya satu pemasok yang menyediakan untuk bagian tertentu ke proyek tertentu). Dalam hal ini, kami menempatkan 1 pada partisipasi PEMASOK, dan  $M, N$  pada partisipasi PROYEK, BAGIAN pada Gambar 3.17. Ini menentukan batasan bahwa kombinasi  $(j, b)$  tertentu dapat muncul paling banyak satu kali dalam himpunan relasi karena setiap kombinasi (PROYEK, BAGIAN) tersebut secara unik menentukan satu pemasok. Oleh karena itu, setiap *instance* relasi  $(s, j, b)$  diidentifikasi secara unik dalam himpunan relasi dengan kombinasi  $(j, b)$ , yang menjadikan  $(j, b)$  sebagai kunci untuk himpunan relasi. Dalam notasi ini, partisipasi yang memiliki 1 dimiliki oleh mereka yang tidak diperlukan untuk menjadi bagian dari kunci pengidentifikasi untuk himpunan relasi. Jika ketiga kardinalitas adalah  $M$  atau  $N$ , maka kuncinya adalah kombinasi dari ketiga partisipan.

Notasi kedua didasarkan pada notasi  $(min, max)$  untuk relasi biner. Notasi  $(min, max)$  pada partisipasi di sini menetapkan bahwa setiap entitas terkait dengan setidaknya  $min$  dan paling banyak  $max$  *instance* relasi dalam himpunan relasi. Batasan ini tidak ada hubungannya dengan penentuan kunci dari relasi  $n$ -ary, di mana  $n > 2$ , tetapi menentukan jenis batasan yang berbeda yang menempatkan batasan pada berapa banyak *instance* relasi yang dapat diikuti oleh setiap entitas.

## E. Rangkuman

Dalam bab ini, telah disajikan konsep pemodelan model data konseptual tingkat tinggi, dan model entitas-relationship (E-R). Kita mulai dengan membahas peran yang dimainkan

model data tingkat tinggi dalam proses desain basis data, dan selanjutnya akan disajikan kumpulan contoh persyaratan basis data untuk basis data PERUSAHAAN, yang merupakan salah satu contoh yang digunakan ada bab ini. Di bagian awal akan didefinisikan konsep model E-R yang meliputi entitas dan atributnya. Kemudian dilanjutkan dengan membahas nilai NULL dan menyajikan berbagai jenis atribut, yaitu:

- Atribut Sederhana (*Simple Attribute*) versus Atribut komposit (*Composite Attribute*)
- Atribut Bernilai Tunggal (*Single-Valued Attribute*) versus Atribut Bernilai Banyak (*Multivalued Attribute*)
- Atribut Tersimpan (*Stored Attribute*) dan Atribut Turunan (*Derivied Attribute*)

Selanjutnya dibahas pula konsep model E-R di tingkat skema:

- Jenis entitas dan himpunan entitas
- Atribut kunci dari jenis entitas
- Himpunan nilai (domain) atribut
- Jenis hubungan dan himpunan relasi
- Peran partisipasi jenis entitas dalam jenis relasi

Disajikan pula dua metode untuk menentukan batasan struktural pada jenis relasi. Metode pertama membedakan dua jenis batasan struktural, yaitu:

- Rasio kardinalitas (1:1 (*one-to-one*), 1:N (*one-to-many*), M:N (*many-to-many*) untuk relasi biner)
- Batasan partisipasi (partisipasi total, dan partisipasi parsial)

Pad bab ini juga dibahas metode lain sebagai alternatif untuk menentukan batasan struktural, yaitu dengan menentukan angka minimum dan maksimum (min, maks) pada partisipasi setiap jenis entitas dalam jenis relasi. Kami membahas jenis entitas lemah dan konsep terkait jenis entitas pemilik, mengidentifikasi jenis relasi dan atribut kunci parsial.

## F. Latihan Soal

1. Buatlah daftar berbagai kasus di mana penggunaan nilai NULL dapat digunakan!
2. Definiskanlah istilah-istilah dari: entitas, atribut, nilai atribut, *instance* relasi, atribut komposit, atribut bernilai banyak, atribut turunan, atribut kunci, dan himpunan nilai (domain)!

3. Apakah yang dimaksud dengan jenis entitas dan himpunan entitas? Jelaskan perbedaan antara entitas, jenis entitas, dan himpunan entitas!
4. Jelaskan perbedaan antara atribut dan himpunan nilai!
5. Apakah yang dimaksud dengan jenis relasi? Jelaskan perbedaan antara *instance* relasi, jenis relasi, dan himpunan relasi!
6. Apakah itu peran partisipasi? Kapan perlu menggunakan nama peran dalam deskripsi jenis relasi?
7. Jelaskan dua alternatif untuk menentukan batasan struktural pada jenis relasi! Apa kelebihan dan kekurangan masing-masing batasan struktural?
8. Dalam kondisi apa atribut dari jenis relasi biner dapat dimigrasikan menjadi atribut dari salah satu jenis entitas yang berpartisipasi?
9. Ketika kita memikirkan relasi sebagai atribut, apa himpunan nilai dari atribut ini?
10. Apakah yang dimaksud dengan jenis relasi rekursif? Berikan beberapa contoh dari jenis relasi rekursif ini!
11. Kapan konsep entitas lemah digunakan dalam pemodelan data? Definisikan istilah jenis entitas pemilik, tipe entitas lemah, identifikasi jenis relasi, dan kunci parsial!
12. Dapatkah relasi pengidentifikasi dari jenis entitas lemah memiliki derajat yang lebih besar dari dua? Berikan contoh untuk menggambarkan jawaban Anda!

## G. Referensi

- Bagui, Sikha Saha & Richard Walsh Earp. 2023. Database Design Using Entity-Relationship Diagrams. Third Edition. New York: CRC Press
- Connolly, Thomas M., & Carolyn E. Begg. 2015. Database Systems: A Practical Approach to Design, Implementation, and Management. England: Pearson.
- Dantes, Gede Rasben, Komang Setemen, Ni Wayan Marti, I Ketut Resika Arthana, Kadek Surya Mahedy, dan Putu Hendra Suputra. 2017. Pengantar Basis Data. Depok: Rajawali Press
- Date, C. J. 2019. Database Design and Relational Theory: Normal Forms and All That Jazz. Second Edition. USA: Apress
- Elmasri, Ramez & Shamkant B. Navathe. 2016. Fundamental of Database Systems. Seventh Edition. United State: Pearson.
- Fathansyah. 2018. Basis Data. Revisi Ketiga. Bandung: Informatika.
- Silberschatz, A., Hendry F. Korth, & S. Sudarshan. 2020. Database System Concepts. Seventh Edition. New York: McGraw-Hill Education.

Watt, Adrienne & Nelson Eng. 2014. Database Design. Second Edition. Victoria: BCCampus.