



**MODUL**

**DATABASE**

**(CCS120)**

**MODUL SESI 3**

**ENTITY RELATIONSHIP DIAGRAM**

**DISUSUN OLEH**

**NOVIANDI, S.Kom, M.Kom**

**UNIVERSITAS ESA UNGGUL**

**2020**

## BAB III

### (Entity Relationship Diagram)

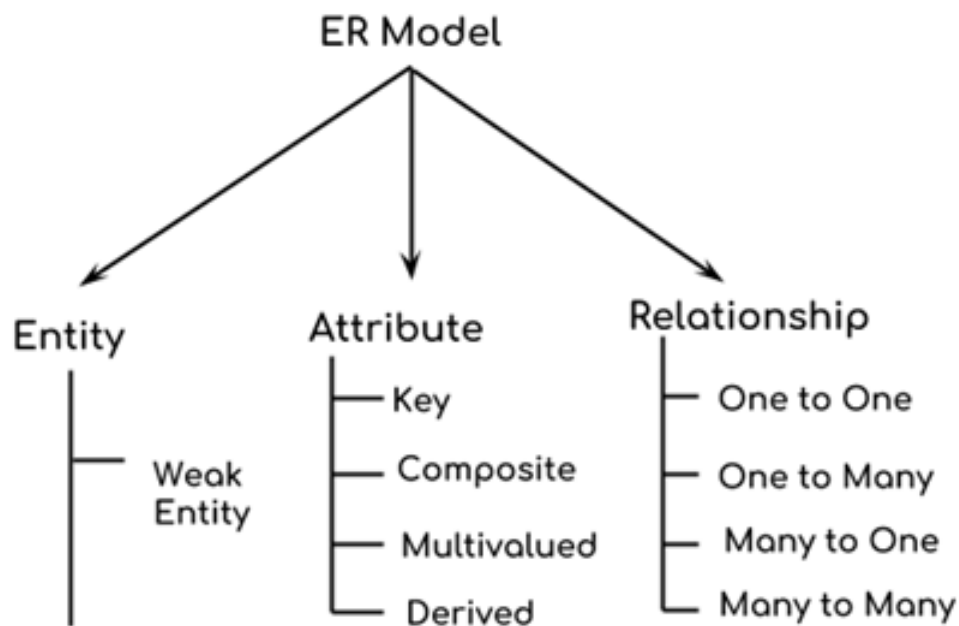
#### Tujuan

1. Bagaimana menggunakan pemodelan Entity – Relationship (ER) dalam desain database.
2. Konsep dasar yang terkait dengan model ER: entitas, relationship, dan atribut.
3. Teknik diagram untuk menampilkan model ER menggunakan Unified Modeling Language (UML).
4. Cara mengidentifikasi dan menyelesaikan masalah dengan model ER yang disebut jebakan koneksi.

#### Teori

##### Entity-Relationship Model

Model data entity-relationship (E-R) menganggap dunia nyata terdiri dari objek-objek dasar, yang disebut **entitas**, dan **relationships** di antara objek-objek.



Gambar 3.1 Komponen ER Diagram

## Konsep Dasar

Model data E-R menggunakan tiga pengertian dasar: entity, relationship, dan atribut.

### 1. Entity

- a. Entitas adalah "**benda**" atau "**objek**" di dunia nyata yang dapat dibedakan dari semua objek lainnya.  
Misalnya, setiap orang dalam suatu perusahaan adalah suatu **entitas**. Entitas memiliki sekumpulan **properti**, dan **nilai** atau beberapa set properti dapat mengidentifikasi entitas secara unik.  
Misalnya, seseorang mungkin memiliki properti **person-id** yang nilainya secara unik mengidentifikasi orang tersebut.
- b. Entity occurrence (Kejadian entitas) adalah objek yang dapat diidentifikasi secara unik dari tipe entitas.
- c. Himpunan entitas adalah sekumpulan entitas dengan tipe yang sama yang berbagi properti, atau atribut yang sama. Himpunan semua orang yang menjadi nasabah di bank tertentu, misalnya, dapat didefinisikan sebagai nasabah himpunan entitas. Entitas individu yang merupakan himpunan dikatakan sebagai perpanjangan dari himpunan entitas. Dengan demikian, semua nasabah bank perorangan merupakan perpanjangan dari nasabah kumpulan entitas.
- d. Setiap jenis entitas ditampilkan sebagai persegi panjang, dilabeli dengan nama entitas, yang biasanya merupakan kata benda tunggal.
- e. Pada UML, huruf pertama dari setiap kata dalam nama entitas adalah huruf besar.  
Misalnya, Staf dan PropertyForRent.

Entitas dapat dibagi menjadi dua jenis, yaitu:

#### 1. Weak Entity (Entitas Lemah)

- Entitas yang tidak dapat diidentifikasi secara unik oleh atributnya sendiri dan bergantung pada hubungan dengan entitas lain disebut entitas lemah.
- Karakteristik entitas yang lemah adalah bahwa setiap kejadian entitas tidak dapat diidentifikasi secara unik hanya dengan menggunakan atribut yang terkait dengan tipe entitas tersebut.

Misalnya, perhatikan bahwa tidak ada kunci utama untuk entitas Preferensi.

- Entitas lemah diwakili oleh persegi panjang ganda.

Misalnya - rekening bank tidak dapat diidentifikasi secara unik tanpa mengetahui bank yang memiliki rekening tersebut, sehingga rekening bank adalah entitas yang lemah.



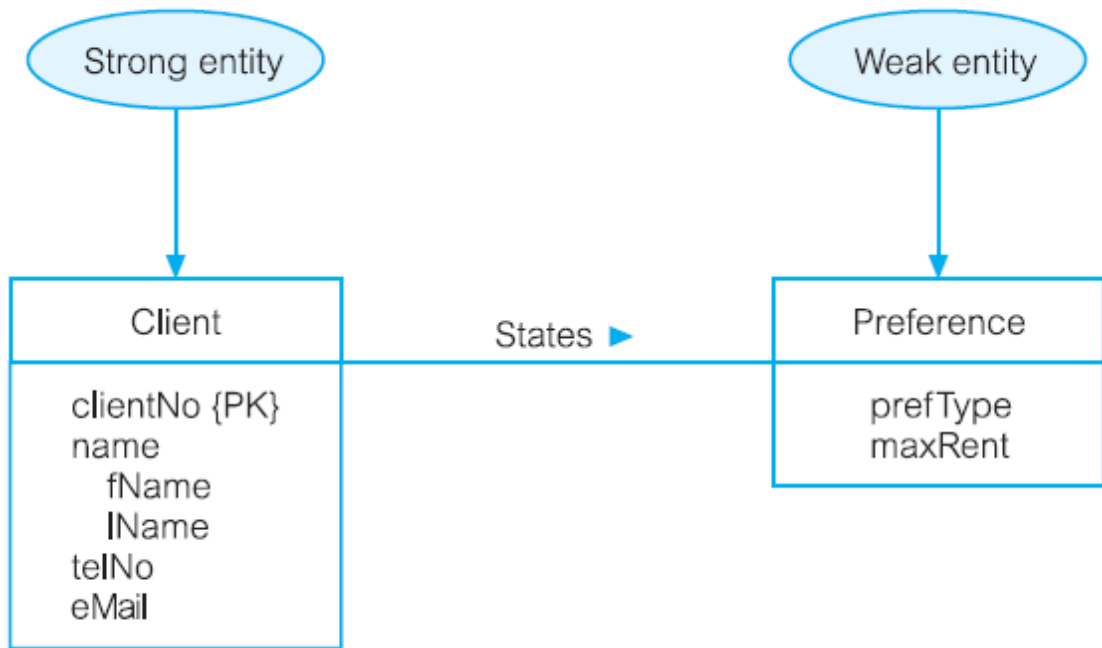
Gambar 3.2 Weak Entity

## 2. Strong Entity

- Jenis entitas yang keberadaannya tidak bergantung pada beberapa jenis entitas lainnya.
- Karakteristik dari tipe entitas yang kuat adalah bahwa setiap kejadian entitas dapat diidentifikasi secara unik menggunakan atribut kunci utama dari tipe entitas tersebut.

Misalnya, dapat mengidentifikasi secara unik setiap anggota staf menggunakan atribut staffNo, yang merupakan kunci utama untuk tipe entitas Staf.

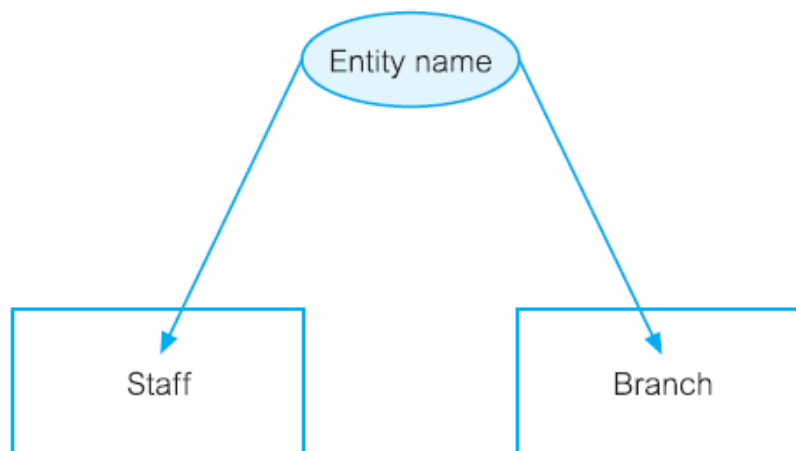
Universitas  
**Esa Unggul**



Gambar 3.3 Tipe entitas yang kuat disebut Klien dan tipe entitas yang lemah yang disebut Preferensi

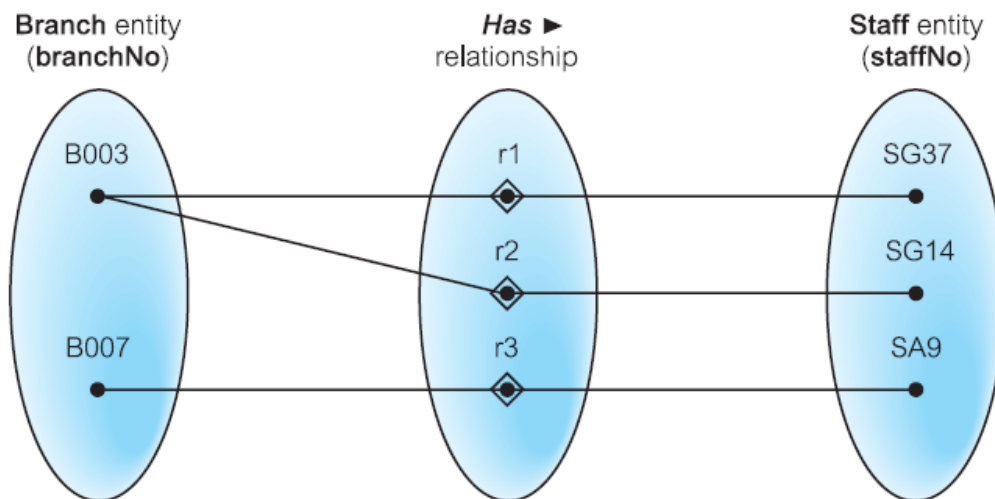
## 2. Relationship

- Suatu hubungan direpresentasikan dengan bentuk wajik pada diagram ER yang menunjukkan hubungan antar entitas.
- Jenis relationship adalah sekumpulan asosiasi antara satu atau beberapa jenis entitas yang berpartisipasi.
- Setiap jenis relationship diberi nama yang menjelaskan fungsinya.

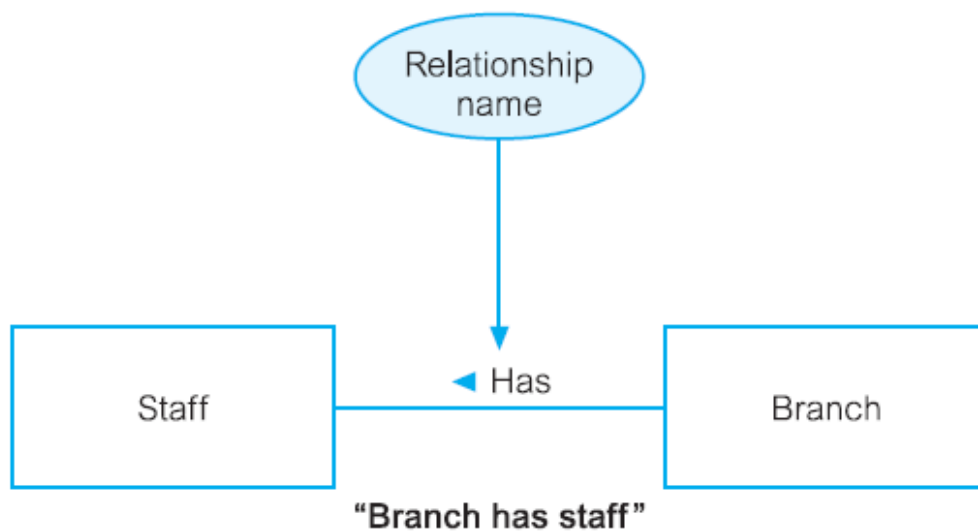


Gambar 3.4 Diagrammatic Representation dari Jenis Entitas Staff dan Branch

- Relationship occurrence adalah Asosiasi unik yang dapat diidentifikasi yang mencakup satu kejadian dari setiap jenis entitas yang berpartisipasi.



Gambar 3.5 Jaringan semantik yang menunjukkan kemunculan individu dari tipe relationship Has.



Gambar 3.6 Representasi diagram dari tipe hubungan Branch *Has* Staff.

### Representasi Diagram dari Jenis Relationship

- Setiap jenis relationship ditampilkan sebagai garis yang menghubungkan tipe entitas terkait dan diberi label dengan nama relationshipnya.

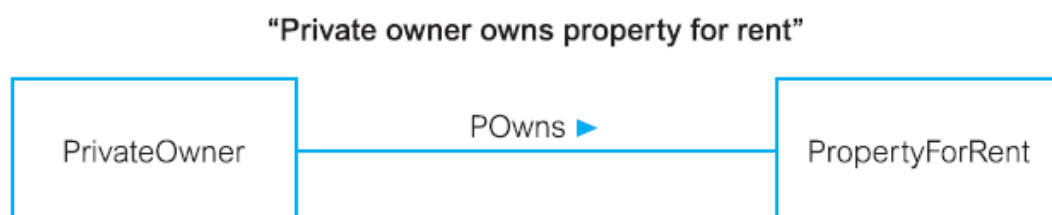
- Biasanya, suatu hubungan dinamai menggunakan kata kerja (misalnya, Supervises atau Manages) atau frasa singkat yang menyertakan kata kerja
- Huruf pertama dari setiap kata dalam nama relationship ditampilkan dalam huruf besar. Jika memungkinkan, nama hubungan harus unik untuk model ER tertentu.
- Relationship hanya diberi label dalam satu arah, yang biasanya berarti bahwa nama relationship hanya masuk akal dalam satu arah.
- setelah nama relationship dipilih, simbol panah ditempatkan di samping nama yang menunjukkan arah yang benar bagi pembaca untuk menafsirkan nama relationship.

### Degree of Relationship Type

- Jumlah tipe entitas yang berpartisipasi dalam suatu hubungan.
- Jumlah peserta dalam suatu jenis relationship disebut derajat relationship.
- Relationship derajat dua disebut **biner**.

Contoh dari hubungan biner adalah hubungan Has yang ditunjukkan pada Gambar 3.6 dengan dua tipe entitas yang berpartisipasi; yaitu Staf dan branch.

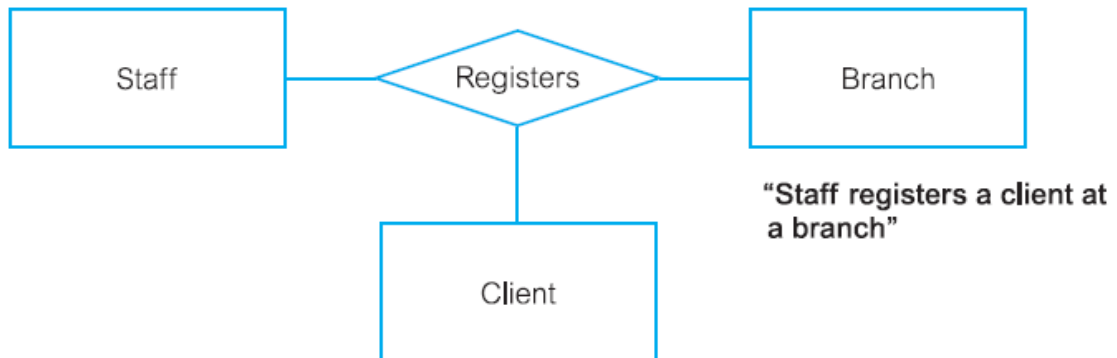
Contoh kedua dari hubungan biner adalah hubungan POwn yang ditunjukkan pada Gambar 3.7 dengan dua tipe entitas yang berpartisipasi; yaitu, PrivateOwner dan PropertyForRent.



Gambar 3.7 Contoh Binary Relationship

- Relationship derajat tiga disebut **ternary**.
- Contoh relasi ternary adalah Register dengan tiga tipe entitas yang berpartisipasi: Staff, Branch, dan Client. Hubungan ini mewakili pendaftaran

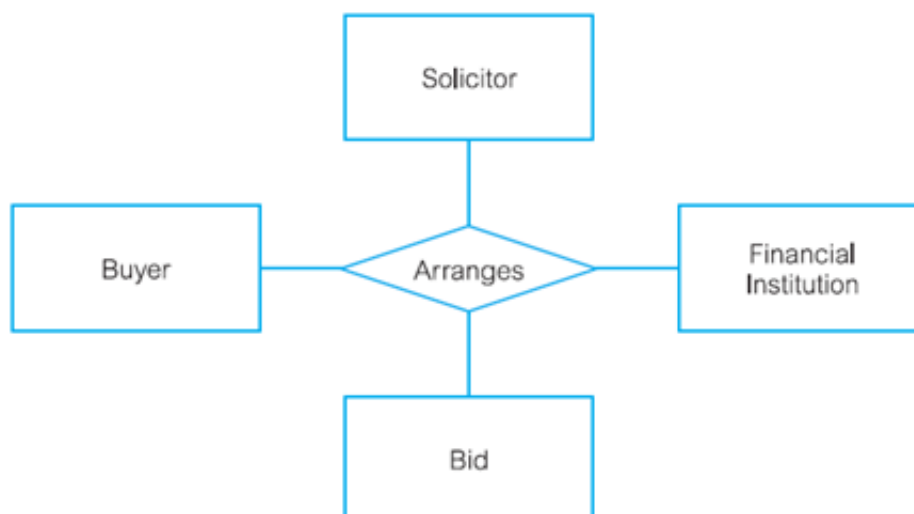
klien oleh anggota staf di branch. Istilah "hubungan kompleks" digunakan untuk menggambarkan hubungan dengan derajat yang lebih tinggi dari biner.



Gambar 3.8 Contoh Ternary Relationship

### Representasi Diagram dari Hubungan yang Kompleks

- Notasi UML menggunakan berlian untuk merepresentasikan hubungan dengan derajat lebih tinggi dari biner.
- Nama relationship ditampilkan di dalam berlian, dan dalam kasus ini, panah arah yang biasanya terkait dengan nama dihilangkan.
- Relationship derajat empat disebut kuartener.
- Relationship kuartener yang disebut Arranges dengan empat jenis entitas yang berpartisipasi, yaitu, Pembeli, Pengacara, Lembaga Keuangan, dan Penawaran (Gambar 3.9)
- Hubungan ini mewakili situasi di mana pembeli, disarankan oleh pengacara dan didukung oleh lembaga keuangan, mengajukan penawaran.



Gambar 3.9 Contoh Quarternary Relationship disebut dengan Arranges



Keterangan:

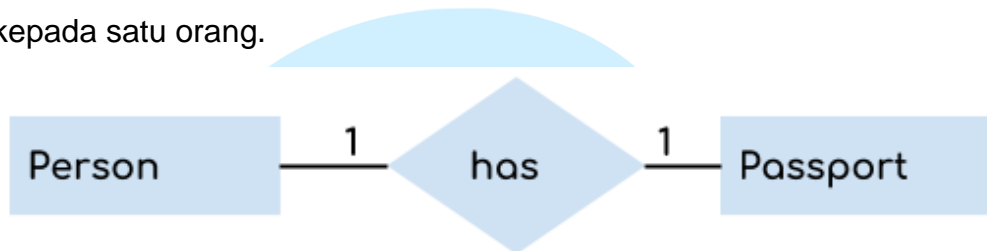
Pengacara mengatur penawaran atas nama pembeli yang didukung oleh lembaga keuangan.

Ada 4 (empat) jenis relationships, yaitu:

a. One to One

- Ketika satu contoh dari suatu entitas dikaitkan dengan satu contoh dari entitas lain, maka itu disebut hubungan One to One.

Misalnya, seseorang hanya memiliki satu paspor dan paspor diberikan kepada satu orang.

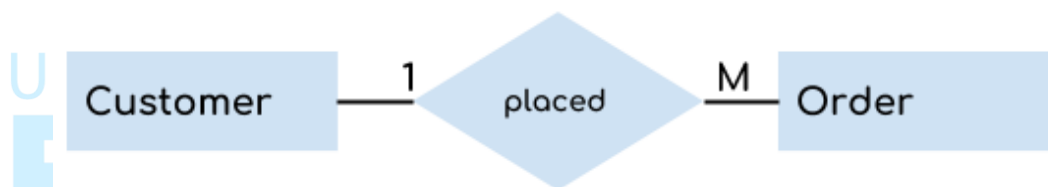


Gambar 3.10 One to One

b. One to Many

- Ketika satu contoh entitas dikaitkan dengan lebih dari satu contoh entitas lain maka itu disebut hubungan One to Many.

Misalnya, pelanggan dapat melakukan banyak pesanan tetapi pesanan tidak dapat dilakukan oleh banyak pelanggan.



Gambar 3.11 One to Many

c. Many to One

- Ketika lebih dari satu contoh entitas dikaitkan dengan satu contoh dari entitas lain maka itu disebut hubungan Many to One.

Misalnya, banyak siswa dapat belajar di satu perguruan tinggi tetapi seorang siswa tidak dapat belajar di banyak perguruan tinggi pada waktu yang sama.



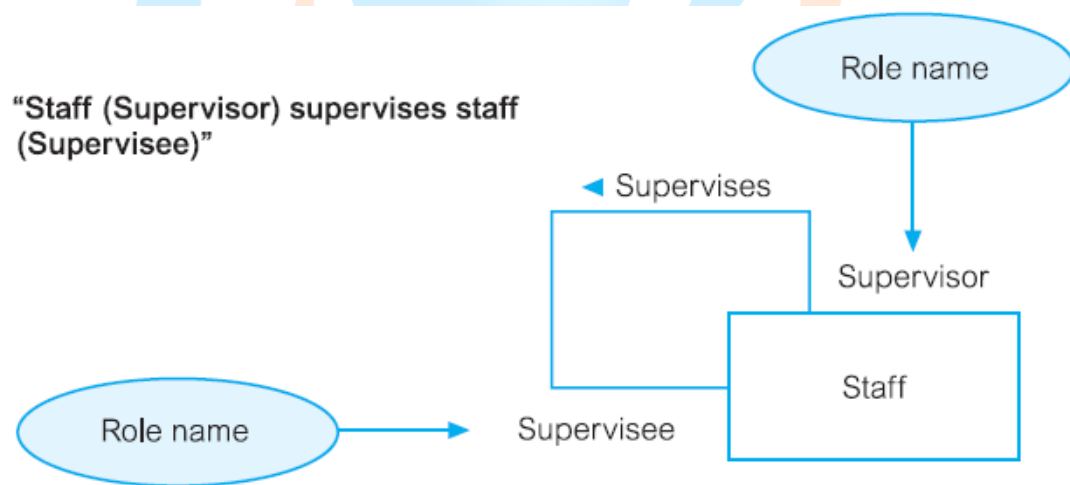
Gambar 3.12 Many to One

d. Many to Many

- Ketika lebih dari satu contoh entitas dikaitkan dengan lebih dari satu contoh entitas lain maka itu disebut hubungan Many toMany. Misalnya, a dapat ditugaskan ke banyak proyek dan proyek dapat ditugaskan ke banyak siswa.

**Recursive Relationship**

- Jenis hubungan di mana jenis entitas yang sama berpartisipasi lebih dari sekali dalam peran yang berbeda



Gambar 3.13 An example of a recursive relationship called *Supervises* with role names Supervisor and Supervisee.

3. Attributes

- Atribut menggambarkan properti dari suatu entitas.
- Atribut direpresentasikan sebagai Oval dalam diagram ER.
- Setiap atribut dikaitkan dengan sekumpulan nilai yang disebut domain. Domain mendefinisikan nilai potensial yang mungkin dimiliki oleh atribut

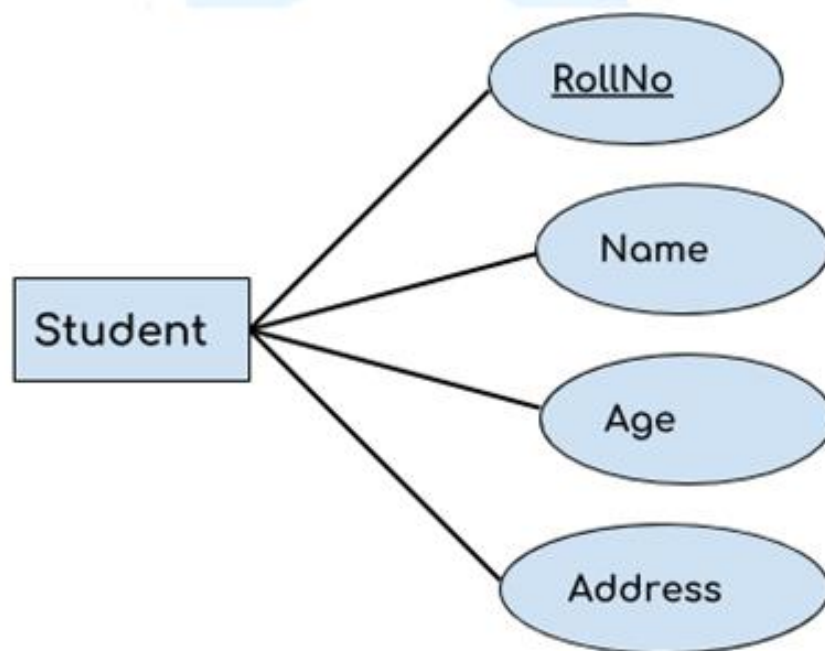
Ada empat jenis atribut:

a. Key attribute (Atribut Kunci)

- Atribut kunci dapat secara unik mengidentifikasi entitas dari kumpulan entitas.

Misalnya, nomor daftar siswa dapat secara unik mengidentifikasi siswa dari sekumpulan siswa.

- Atribut key diwakili oleh oval sama seperti atribut lainnya akan tetapi atribut key tersebut ***digarisbawahi***.

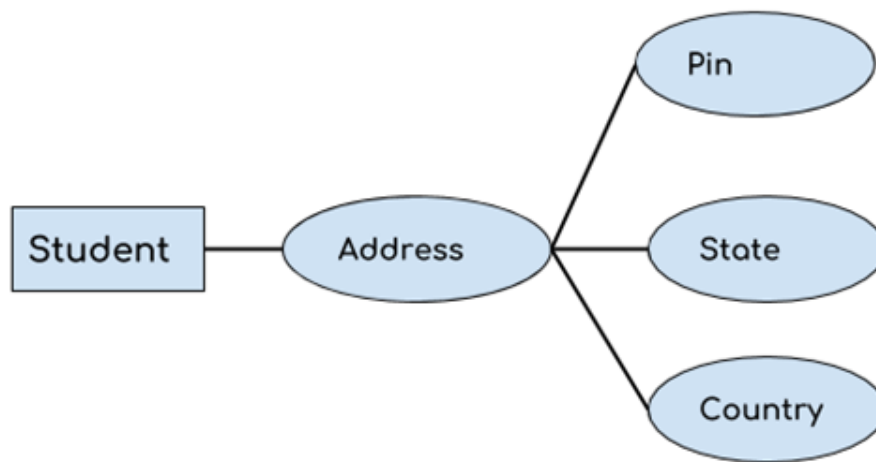


Gambar 3.14 Key Atribut

b. Composite attribute

- Atribut yang merupakan gabungan dari atribut lain dikenal sebagai atribut komposit.

Misalnya, dalam **entitas siswa**, alamat siswa adalah atribut gabungan karena alamat terdiri dari atribut lain seperti kode pin, negara bagian, negara.



Gambar 3.15 Composite attribute

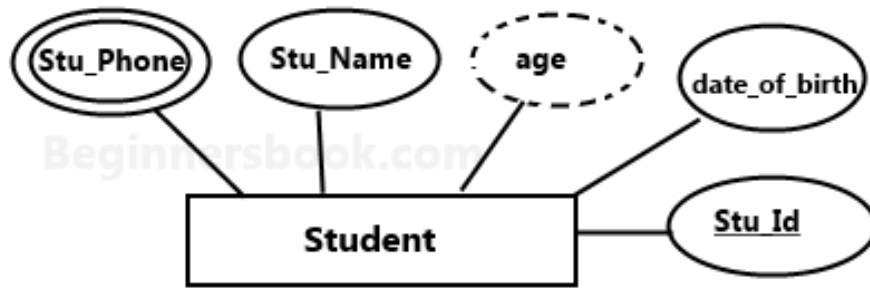
Address adalah composite attribute

c. Multivalued attribute

- Atribut yang dapat menampung banyak nilai dikenal sebagai atribut multinilai.
- Atribut ini diwakili dengan oval ganda dalam Diagram ER. Misalnya; Seseorang dapat memiliki lebih dari satu nomor telepon sehingga atribut nomor telepon multinilai.

d. Derived attribute (Atribut Turunan)

- Atribut yang mewakili nilai yang dapat diturunkan dari nilai atribut terkait atau kumpulan atribut, tidak harus dalam tipe entitas yang sama.
- Atribut turunan adalah atribut yang nilainya dinamis dan berasal dari atribut lain.
- Atribut ini diwakili oleh oval putus-putus dalam Diagram ER. Misalnya; Usia orang adalah atribut turunan karena berubah seiring waktu dan dapat diturunkan dari atribut lain (Tanggal lahir).



Gambar 3.16 Derived Attribute

### Simple and Composite Attributes

- Simple atribut adalah atribut yang terdiri dari satu komponen dengan keberadaan independen.
- Composite atribut adalah atribut yang terdiri dari banyak komponen, masing-masing dengan eksistensi independen.

### Single-valued and Multi-valued Attributes

- Single valued attribute adalah atribut yang menyimpan satu nilai untuk setiap kemunculan tipe entitas.
- Multi valued attribute adalah atribut yang menyimpan beberapa nilai untuk setiap kemunculan dari tipe entitas.

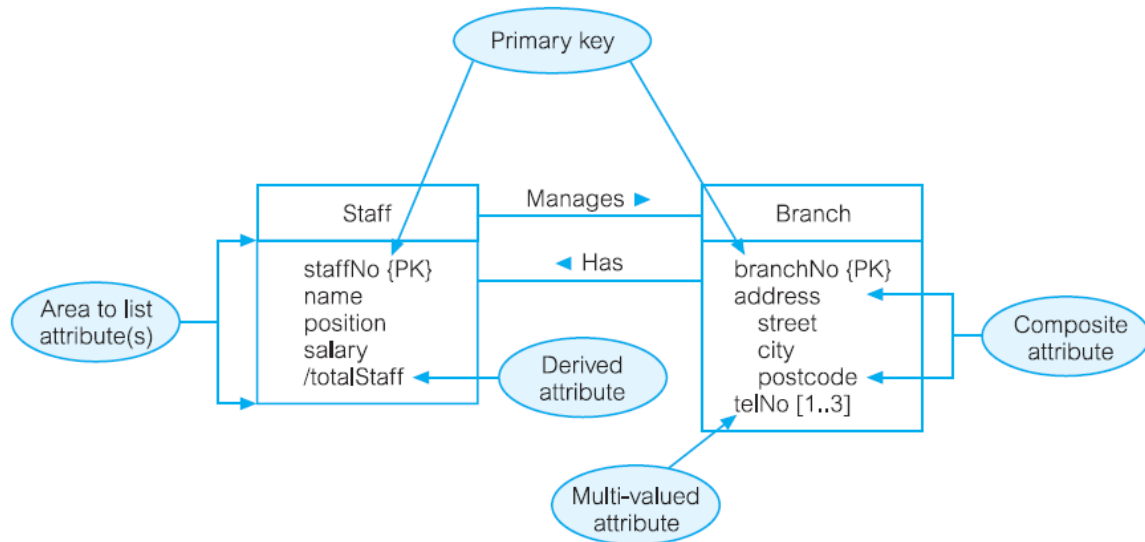
### Key

Ada 3 (tiga) jenis key dalam sebuah atribut, yaitu:

1. Candidate Key  
Kumpulan atribut minimal yang secara unik **mengidentifikasi setiap kemunculan** tipe entitas.
2. Primary Key  
Kunci kandidat yang dipilih untuk **mengidentifikasi secara unik** setiap kemunculan dari tipe entitas.
3. Composite Key  
Candidat key yang terdiri dari dua atau lebih atribut.

## Representasi Diagram dari Atribut

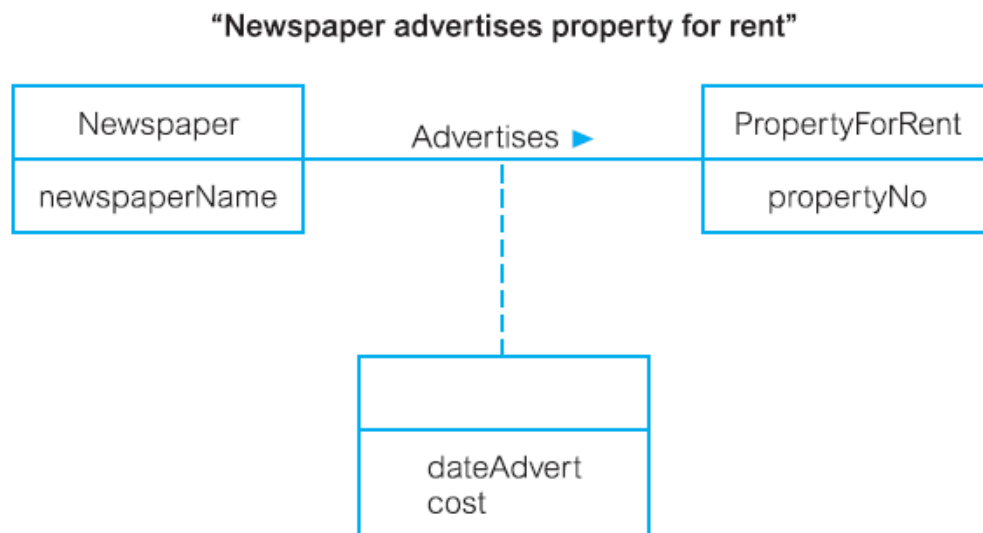
Jika tipe entitas ingin ditampilkan dengan atributnya, maka dibagi dalam bentuk persegi panjang yang mewakili entitas menjadi dua. Bagian atas persegi panjang menampilkan nama entitas dan bagian bawah mencantumkan nama atribut.



Gambar 3.17 Representasi Diagram dari Entitas Staff dan Branch Beserta Atribut

## Atribut pada Relationship

Representasi diagram dari atribut pada relationship



Gambar 3.18 Contoh Relationship Advertises dengan Atribut dateAdvert dan Cost

## Problems dengan ER Models

Permasalahan yang mungkin terjadi saat membangun model ER adalah kemungkinan salah dalam penafsiran dari makna relationship tersebut (Connection Traps).

Dua jenis connection trap (Jebakan koneksi)

### 1. Fan trap

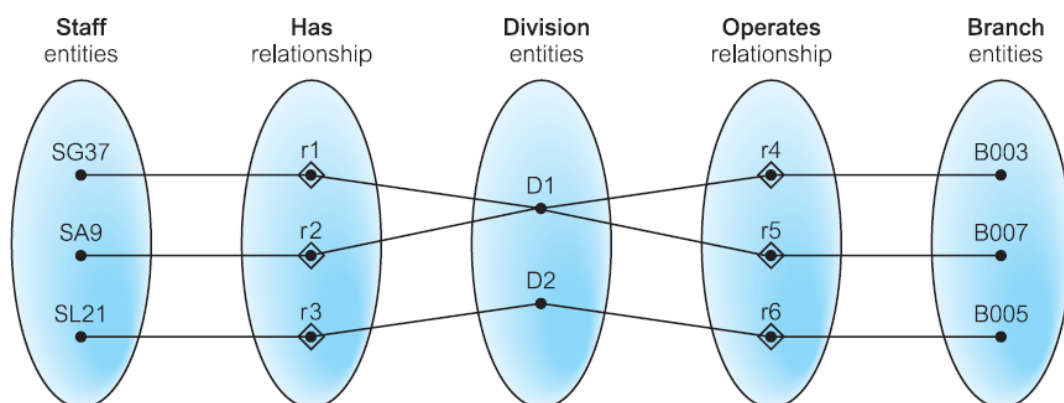
Dimana model merepresentasikan hubungan antara tipe entitas, tetapi jalur antara kejadian entitas tertentu ambigu.



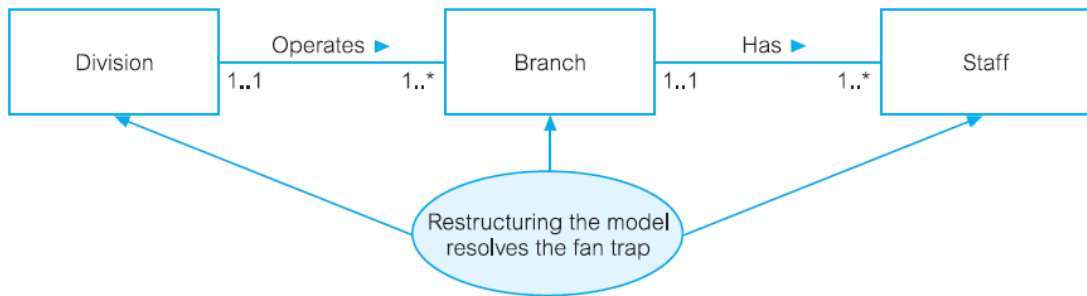
Gambar 3.19 Contoh Fan Trap

Keterangan:

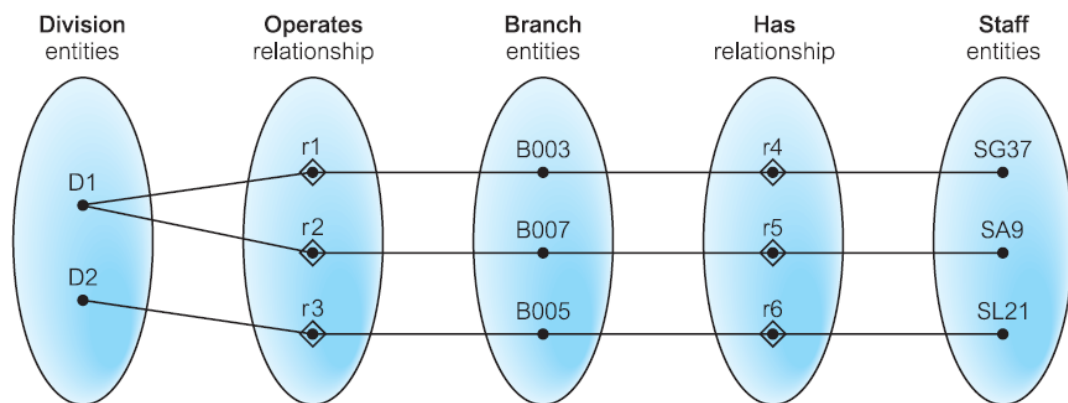
- Model ini merepresentasikan fakta bahwa satu divisi mengoperasikan satu atau lebih branch dan memiliki satu atau lebih staf.
- Namun, masalah muncul ketika kita ingin mengetahui anggota staf mana yang bekerja di branch tertentu.
- Untuk memahami masalah ini, kami memeriksa beberapa kemunculan hubungan *Has* dan *Operates* menggunakan nilai untuk atribut primary key dari tipe entitas Staf, Divisi, dan Branch



Gambar 3.20 Jaringan Semantik dari ER Model dari Gambar 3.19



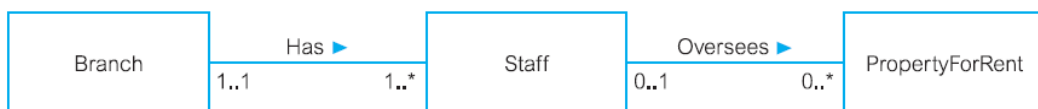
Gambar 3.21 Model ER yang ditunjukkan pada Gambar 3.19 direstrukturisasi untuk melepaskan fan trap



Gambar 3.22 Jaringan semantic dari ER model yang ditunjukkan pada Gambar 3.21

## 2. Chasm trap

Dimana model menyarankan adanya hubungan antara tipe entitas, tetapi jalur tidak ada antara kejadian entitas tertentu.



Gambar 3.23 Contoh Chasm Trap

Keterangan:

- Model ini mewakili fakta bahwa satu cabang memiliki satu atau lebih staf yang mengawasi tidak ada atau lebih properti yang disewakan.
- Kami juga mencatat bahwa tidak semua staf mengawasi properti, dan tidak semua properti diawasi oleh anggota staf.



- Masalah muncul ketika kita ingin mengetahui properti mana yang tersedia di setiap cabang.
- Untuk memahami masalah ini, kami memeriksa beberapa kemunculan hubungan Has dan Oversees menggunakan nilai untuk atribut kunci utama dari tipe entitas Branch, Staff, dan PropertyForRent

## Latihan

1. Buat model ER untuk setiap deskripsi berikut:
  - a. Setiap perusahaan menjalankan empat departemen, dan setiap departemen dimiliki oleh satu perusahaan.
  - b. Setiap departemen dalam bagian (a) mempekerjakan satu atau lebih karyawan, dan setiap karyawan bekerja untuk satu departemen.
  - c. Setiap karyawan di bagian (b) mungkin atau mungkin tidak memiliki satu atau lebih tanggungan, dan setiap tanggungan dimiliki oleh satu karyawan.
  - d. Setiap karyawan di bagian (c) mungkin atau mungkin tidak memiliki riwayat pekerjaan.
  - e. Mewakili semua model ER yang dijelaskan dalam (a), (b), (c), dan (d) sebagai model ER tunggal.

Universitas  
**Esa Unggul**