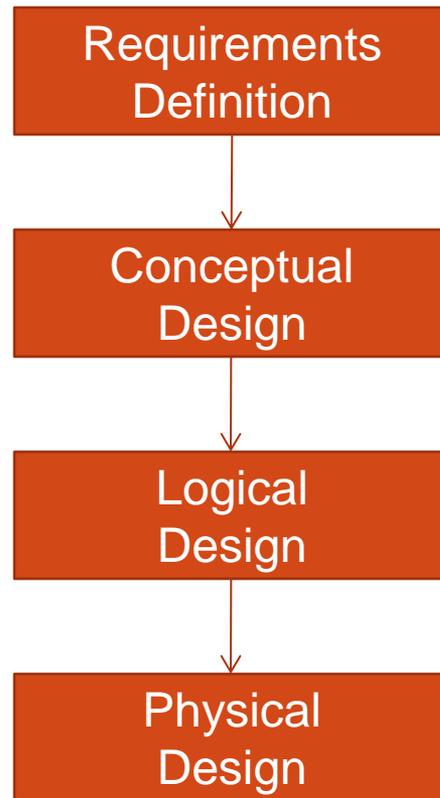


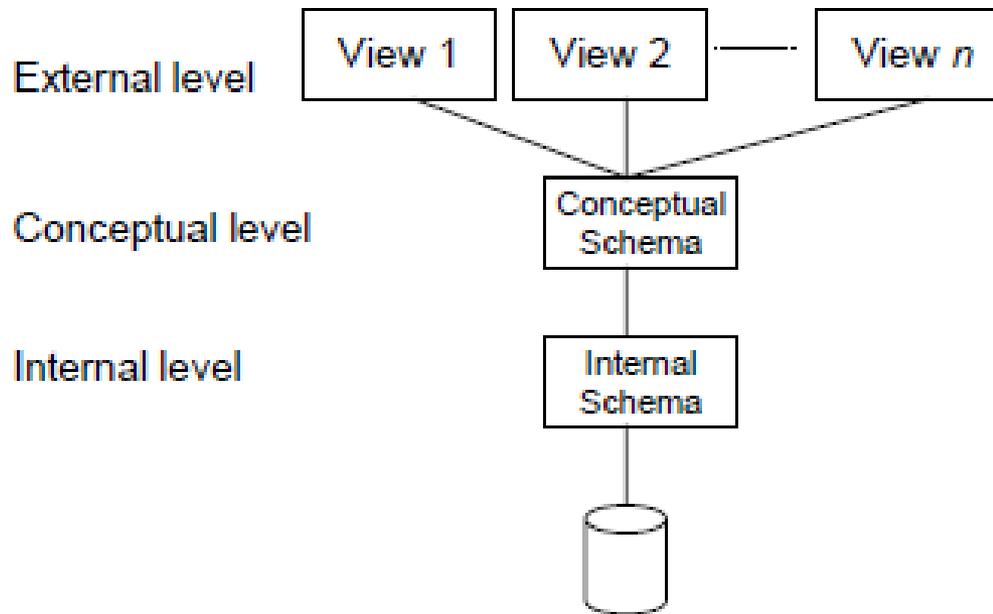
Database Design I

TPI4210
Sistem dan Teknologi Informasi

Database Design Life Cycle



Recap: ANSI/SPARC architecture



Requirements Definition

- Identifikasi dan analisis tampilan-tampilan user
- “user” mungkin seorang manusia, sebuah laporan yang harus dibuat atau sebuah tipe transaksi khusus yang harus diakomodasi oleh DBMS
- Tahap ini masuk dalam *external level* dalam ANSI/SPARC Architecture
- Output dalam tahap ini adalah sebuah pernyataan spesifikasi yang menggambarkan kebutuhan dan keinginan semua user serta kebutuhan dan karakteristik data

Conceptual Design

- Pengembangan data model untuk keperluan seluruh perusahaan/ organisasi
- Conceptual level dalam ANSI/SPARC Architecture
- Tidak terikat oleh model/metode implementasi dalam *physical level*
- Beberapa metode desain dapat digunakan, termasuk dengan pendekatan ER (*Entity Relationship*)

Logical Design

- Pengembangan data model untuk model basis data yang telah ditetapkan (*relational, hierarchical, network, object oriented*)
- Tidak terikat oleh implementasi dengan DBMS tertentu
- Conceptual level dalam ANSI/SPARC Architecture

Physical Design

- Pengembangan strategi implementasi fisik dari *logical data model*
- Penentuan organisasi file, struktur penyimpanan dalam media penyimpan dan metode akses yang tepat sehingga dapat memenuhi kebutuhan user dengan efisien.
- Tahap ini terikat oleh jenis DBMS yang digunakan
- Internal level dalam ANSI/SPARC Architecture

ER Modeling

- ER model dikembangkan oleh Peter Chen pada 1976 untuk desain basis data
- ER model bisa digunakan untuk logical/conceptual design
- ER diagram memberi visualisasi dari sistem yang didesain
- Komponen dasar:
 - Entity
 - Attribute
 - Relationship



ER Modeling: Entities

- Entity (Chen, 1976):
Sesuatu yang bisa diidentifikasi dengan unik dan pasti
- Entitas lemah – entitas yang keberadaannya tergantung pada keberadaan entitas lain
- Entitas kuat/Entitas – tidak tergantung pada entitas lain
- Entitas disimbolkan dengan segiempat didalam ER Diagram

ER Modeling: Entities



Staff Member

Entitas kuat

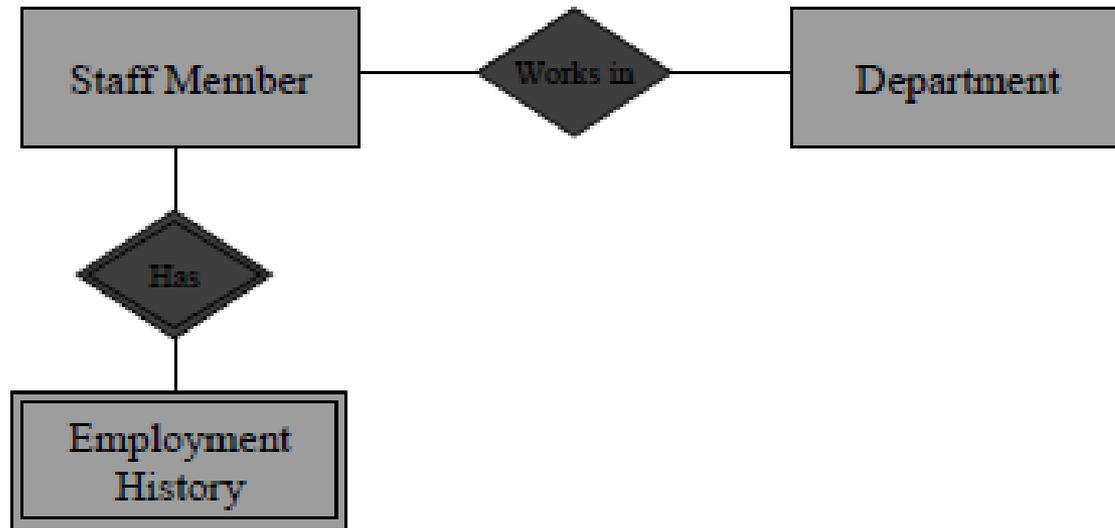
Employment
History

Entitas lemah, keberadaannya
tergantung entitas lain

ER Modelling: Relationships

- *Relationship* menyatakan hubungan antar entitas
- Setiap *relationship* dalam ER diagram diberi nama yang menunjukkan keterkaitan antara entitas yang terhubung
- Secara umum, *binary relationship* (hubungan antara 2 entitas) menjadi fokus dalam ER diagram

ER Modeling: Entities

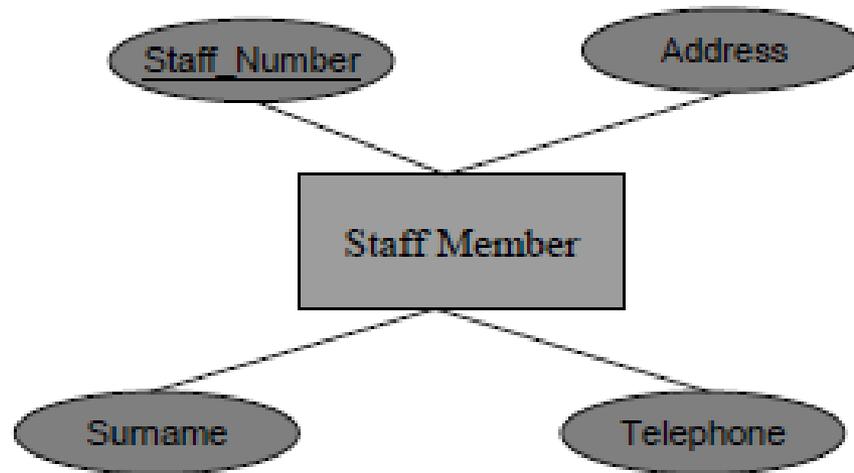


ER Modeling: Attributes

- *Attribute* adalah data mengenai sebuah entitas atau hubungan
- *Attributes* dari entitas STAFF MEMBER (contoh):
 - Staff_number
 - Surname
 - Address
 - Telephone_number
 - Date_of_birth
 - Position
 - Tax_file_number

ER Modeling: Attributes

Attribute bisa disertakan dalam ER diagram



ER Modeling: Attributes

- Domain dari sebuah *attribute* adalah set nilai yang bisa diberikan pada *attribute* tersebut
- Contoh: domain dari *attribute* `staff_number` bisa di set untuk bilangan *integer* antara 1 - 100

ER Modeling: Keys

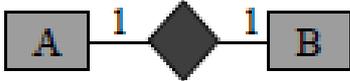
- *Candidate key* adalah sebuah *attribute* atau set *attribute* yang bisa mengidentifikasi dengan pasti entri dari entitas
- Sebuah *candidate key* dipilih sebagai alat identifikasi entri dalam entitas.
- *Candidate key* yang dipilih tersebut disebut *primary key*, sisanya adalah *alternate key*
- *Composite key* adalah *candidate key* yang terdiri dari lebih satu *attribute*

ER Modelling: Keys

- *Candidate key* untuk entitas STAFF MEMBER:
 - *staff_number*
 - *tax_file_number*
- Kedua *attribute* ini dapat mengidentifikasi seorang staff
- Hanya satu dari kedua *attribute* tersebut bisa digunakan sebagai *primary key*

Cardinality Constraints

Dalam sebuah binary relationship, tiga kemungkinan *cardinality ratio*:

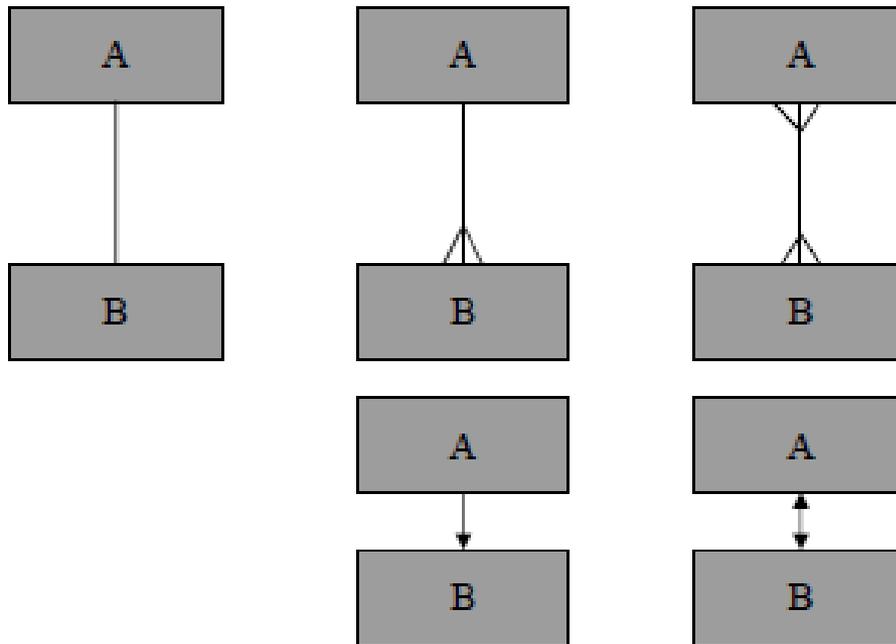
- satu ke satu (1:1) 

Setiap 1 anggota A memiliki hubungan dengan 1 anggota B
Setiap 1 anggota B memiliki hubungan dengan 1 anggota A
- satu ke banyak (1:M) 

Setiap 1 anggota A memiliki hubungan dengan banyak anggota B
Setiap 1 anggota B memiliki hubungan dengan 1 anggota A
- banyak ke banyak (M:N) 

Setiap 1 anggota A memiliki hubungan dengan banyak anggota B
Setiap 1 anggota B memiliki hubungan dengan banyak anggota A

Alternative Notation



Local vs Global Data Models

- Setiap tampilan untuk user mungkin didefinisikan dari *data model* lokal
- *Data model* global adalah *data model* dalam seluruh organisasi/perusahaan yang didapat dari penggabungan seluruh model tampilan masing-masing user

Logical Design (Relational)

Memetakan desain konseptual dalam *relational schema*

1. Menetapkan *relation* dari *conceptual data model* lokal
2. Normalisasi *relation*
3. Menggabungkan *logical data model* lokal dalam sebuah data model global

Deriving Relations

- Membuat *relation* untuk setiap entitas dan *relationship* dalam sebuah ER diagram
- Mendefinisikan *integrity constraints* standard yang relevan dalam fase ini.
- DBDL (*Database Design Language*) dapat digunakan untuk mendefinisikan relation dan *integrity constraints*

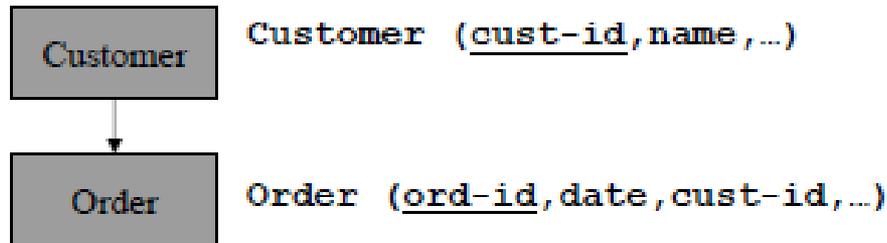
Integrity Constraints

- *Integrity constraint* adalah syarat kondisi yang harus dipenuhi dalam database
- Syarat-syarat ini diterapkan untuk menjamin konsistensi data
- Tidak semua RDMBS mendukung penerapan *integrity constraints*

Common Integrity Constraints

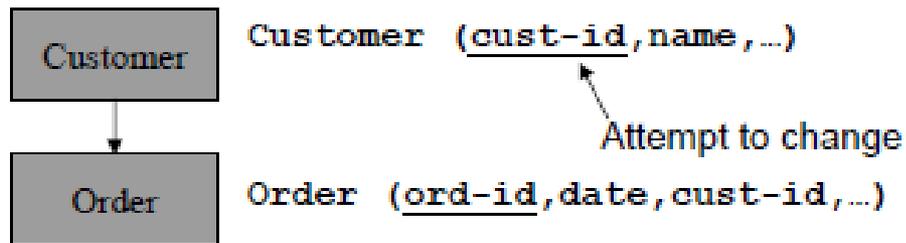
- *Entity integrity:*
“*primary key* tidak boleh *null*”
- *Referential integrity:*
“sebuah *foreign key* hanya boleh bernilai *null* atau sama dengan *primary key* dalam *relation* yang lain”

Referential Integrity



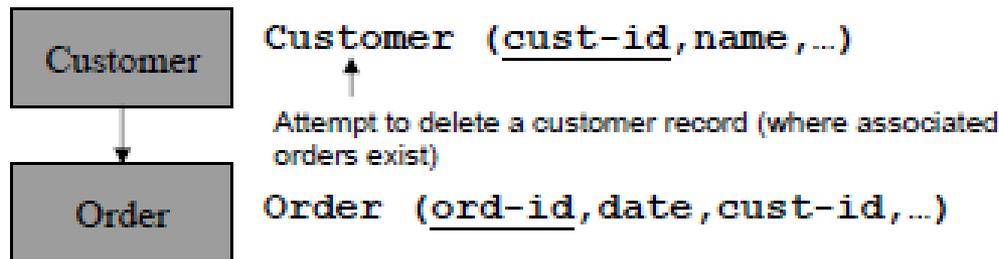
- Setiap order hanya bisa dilakukan oleh customer yang telah terdaftar dalam *relation* customer (atau bernilai *null*).
- Bagaimana jika ingin merubah sebuah customer_id padahal dia memiliki order atau menghapusnya?
- Sebuah aturan harus diterapkan untuk mengakomodasi hal-hal tersebut sehingga *referential integrity* tetap terjaga

Ref. Integrity Update Rules



- Update restricted: tidak mengizinkan adanya perubahan
- Update cascaded: update cust_id untuk seorang customer beserta seluruh ordernya
- Update nullified: update cust_id untuk seorang customer dan set *null* pada cust_id untuk seluruh order yang telah ia buat

Ref. Integrity Deletion Rules



- Delete restricted: tidak bisa dihapus
- Delete cascaded: hapus cust_id untuk seorang customer dan hapus seluruh ordernya
- Delete nullified: hapus seorang customer dan set *null* pada cust_id untuk seluruh order yang telah ia buat

DBDL

- DBDL dapat digunakan untuk mendefinisikan *relation*, *key* dan *integrity constraint*
- *Relation*, *attribute* dan *primary key* dinyatakan dengan notasi standard
- *Attribute* yang boleh *null* ditandai dengan *
- *Alternate key* dinyatakan dengan AK
- *Foreign key* dinyatakan dengan FK beserta aturan untuk *update* dan *delete*
- *Default*: *update cascaded*, *delete restricted*

DBDL - Example

```
Customer (cust-id, surname, initial,  
         address*, phone*, tax_file_number,  
         dept-id)
```

```
AK tax_file_number
```

```
SK surname
```

```
FK dept-id --> Department DLT CSCD
```

```
Department (dept-id, name, location)
```

```
SK name
```

```
CUSTOMER(cus-id, surname, initials,  
         {order-no, order-date})
```

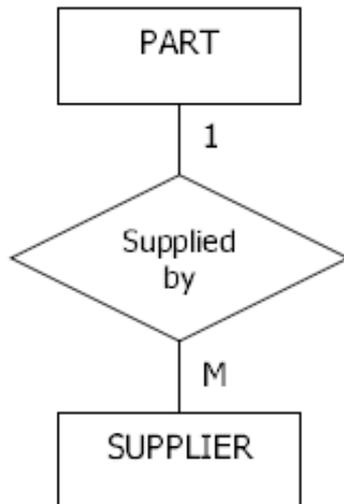
Latihan

- Untuk setiap ER diagram berikut, nyatakan relation dalam DBDL. Attribute dari entitas-entitas adalah sbb:

PART: part-id (primary key), description, unit price

SUPPLIER: supp-id (primary key), name, address, telephone

(a)



(b)

