



VERSI 1.1
13 FEBRUARY, 2023

BASIS DATA

MODUL 1 MATERI : KONSEP DASAR BASIS DATA

TIM PENYUSUN :

BRIANSYAH SETIO WIYONO, S. KOM

CLARISSA SANINDITA REIKISYIFA

CHINTYA TRIA DIANA OKTAVIANI

PRESENTED BY: LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

BASIS DATA

PERSIAPAN MATERI

Praktikan mempelajari konsep dasar basis data, cara instalasi, dan penggunaan tools oracle data modeller.

TUJUAN

1. Praktikan memahami konsep dasar dari basis data.
 2. Praktikan mampu memodelkan informasi ke dalam bentuk Logical dan Relational.
 3. Praktikan dapat mengetahui tools oracle data modeler dan mampu membuat dalam bentuk Logical dan Relational.
-

TARGET MODUL

Praktikan dapat memahami atribut, entitas, data, relasi, kardinalitas, opsionalitas, precision, scale, supertype, subtype, arc dan barred.

PERSIAPAN SOFTWARE/APLIKASI

Oracle SQL Developer Data Modeler :

<https://www.oracle.com/database/sqldeveloper/technologies/sql-data-modeler/download/>

MATERI POKOK

A. KONSEP DASAR BASIS DATA

Basis Data merupakan kumpulan data yang terpusat dan terstruktur yang disimpan pada sistem komputer. Basis data menyediakan fasilitas untuk mengambil, menambah, memodifikasi, dan menghapus data bila diperlukan. Selain itu basis data juga menyediakan fasilitas untuk mengubah data yang diambil menjadi informasi yang berguna.

B. PENGANTAR PEMODELAN DATA RELASIONAL

Database relasional menyimpan informasi dalam tabel dengan baris dan kolom (sebuah tabel). Tabel adalah kumpulan record dimana terdiri dari sebuah baris disebut atau instance dan kolom disebut sebagai bidang (atau atribut). Berikut merupakan komponen penting dalam pemodelan data relasional :

1. ENTITAS

Entitas adalah Objek yang memiliki sesuatu yang nyata (eksistensinya) dan dapat dibedakan dengan objek yang lainnya. Objek tersebut dapat berupa orang, benda,

ataupun hal lainnya. Biasanya entitas digambarkan dalam bentuk persegi panjang. Contoh : dalam aktifitas perkuliahan terdapat entitas mahasiswa, dosen, mata kuliah.

2. ATRIBUT

Atribut adalah karakteristik atau ciri yang mendefinisikan entitas sehingga dapat membedakan antara entitas satu dengan entitas lainnya. Dalam sebuah entitas pasti memiliki satu atau lebih atribut untuk mendefinisikan karakteristik dari entitas tersebut. Biasanya atribut digambarkan dalam bentuk oval. Contoh atribut dari Entitas Mahasiswa yaitu nama, NIM, tanggal lahir. Salah satu macam atribut yang nantinya akan dipelajari adalah Atribut Identifier yaitu digunakan untuk menentukan suatu entitas sebagai kunci atau secara unik (Primary key). **Primary Key (Atribut Khusus)** adalah kandidat key yang dipilih untuk mengidentifikasi baris data secara unik. Contoh atribut identifier dalam entitas mahasiswa adalah NIM

3. DATA / INSTANCE

Data / instance adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata.

4. RELASI

Relasi adalah hubungan antara beberapa entitas atau antar entitas satu dengan entitas yang lainnya. Relasi biasa digambarkan dengan belah ketupat.

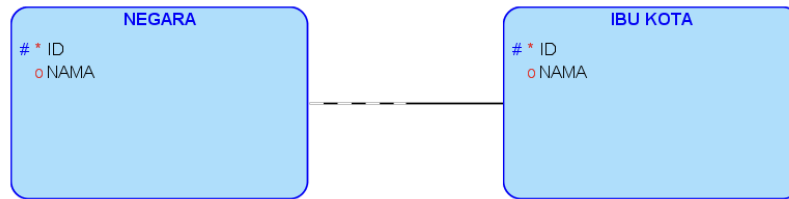
5. KARDINALITAS

Kardinalitas relasi merujuk kepada relasi maksimum yang terjadi dari himpunan entitas yang satu ke himpunan entitas yang lain dan begitu juga sebaliknya. Derajat kardinalitas dibagi menjadi 3 jenis :

a) One-to-One

Relasi ini menggunakan notasi 1:1. Kardinalitas one to one terjadi jika satu entitas X hanya berelasi dengan satu entitas Y ataupun sebaliknya.

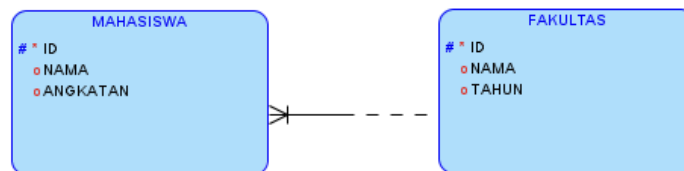
Contohnya: relasi antara tabel negara dan tabel ibu kota dimana satu negara hanya memiliki satu ibukota dan satu ibu kota hanya dimiliki satu kota.



b) One-to-Many atau Many-to-One

Relasi ini menggunakan notasi 1:N. Kardinalitas ini terjadi jika satu entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya. Dalam data modeller disimbolkan dengan one / single toe di satu ujung dan many / crow's feet di ujung lainnya.

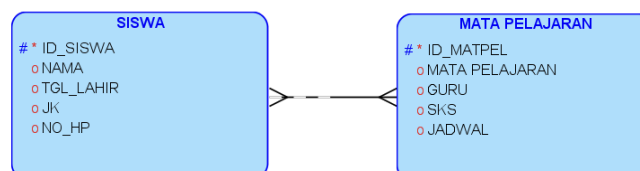
Contohnya: relasi entitas antara tabel mahasiswa dan tabel fakultas. Dimana satu fakultas terdiri dari banyak mahasiswa dan satu mahasiswa hanya berada di satu fakultas.



c) Many-to-Many

Relasi ini menggunakan notasi M:N. Kardinalitas Many to Many terjadi jika banyak entitas X berelasi dengan banyak entitas Y, ataupun sebaliknya. Dalam data modeller disimbolkan dengan many / crow's feet di kedua ujung.

Contohnya: Relasi antar tabel siswa dan tabel mata kuliah. Dimana satu baris siswa bisa berhubungan dengan banyak baris mata pelajaran begitu juga sebaliknya.



6. OPSIONALITAS

Opsionalitas Relasi menunjukkan haruskah atau tidak harus adanya suatu relasi antara satu entitas dengan entitas lainnya.

a) **Optional**

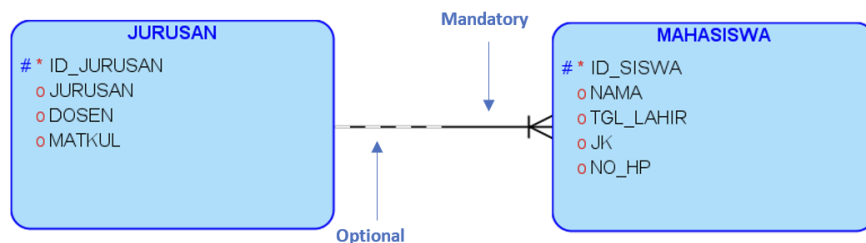
Optional dapat berupa nol (tidak ada) catatan yang sesuai. Dalam SQL Data Modeller disimbolkan dengan garis putus-putus.

Contoh : Setiap jurusan tidak harus memiliki Mahasiswa.

b) **Mandatory**

Mandatory merupakan sejumlah atribut yang ada pada suatu table yang harus berisi data dan tidak boleh kosong (wajib), setidaknya satu data yang cocok di setiap entitas. Dalam SQL Data Modeller disimbolkan dengan garis solid.

Contoh : Setiap Mahasiswa harus memilih satu jurusan.



7. PRECISION & SCALE

a) **Precision**

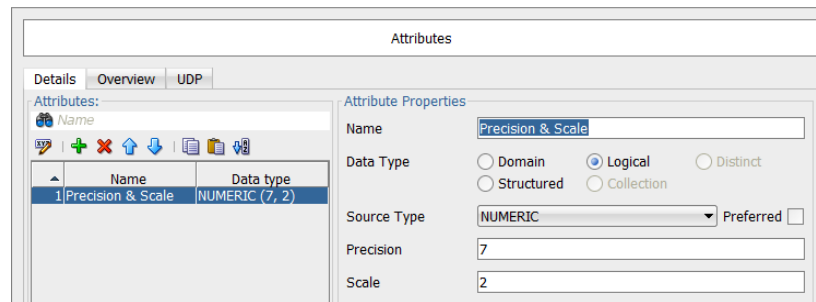
Precision adalah jumlah digit yang bisa dimiliki oleh angka desimal. Precision terdapat pada tipe data numerik dengan jumlah maksimum digit desimal yang signifikan (didefinisikan).

b) **Scale**

Scale adalah jumlah digit di sebelah kanan setelah titik desimal atau setelah koma. Scale terdapat pada tipe data numerik dengan skala digit desimal yang didefinisikan.

Contoh :Jika ada angka 21,253, Maka angka ini memiliki precision 5 dan scale 3. Untuk penulisannya bisa menggunakan decimal (5,3) yang berarti memiliki precision 5 dan scale 3.

Berikut ini cara memasukkan precision dan scale dan harus menggunakan tipe data numerik:



8. SUPERTYPE & SUBTYPE

a) Supertype

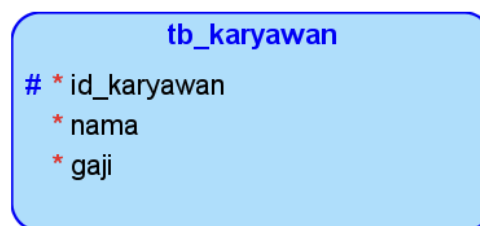
Entitas yang memiliki atribut dan method yang sama untuk satu atau lebih entitas subtype yang mana juga dapat disebut sebagai parent class.

b) Subtype

Entitas hasil pewarisan (*inheritance*) atribut dan method dari entitas supertype serta memungkinkan untuk memiliki atribut dan method khusus lainnya.

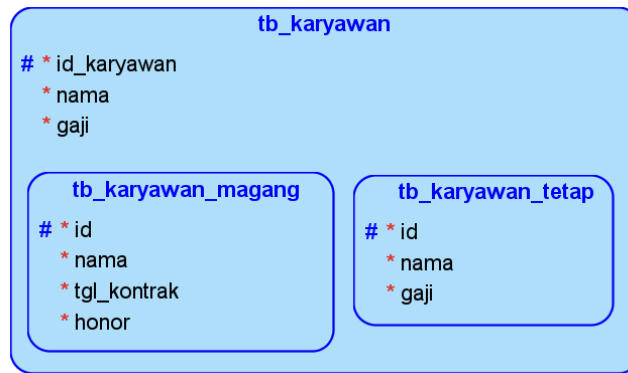
Contoh:

Supertypes : tb_karyawan



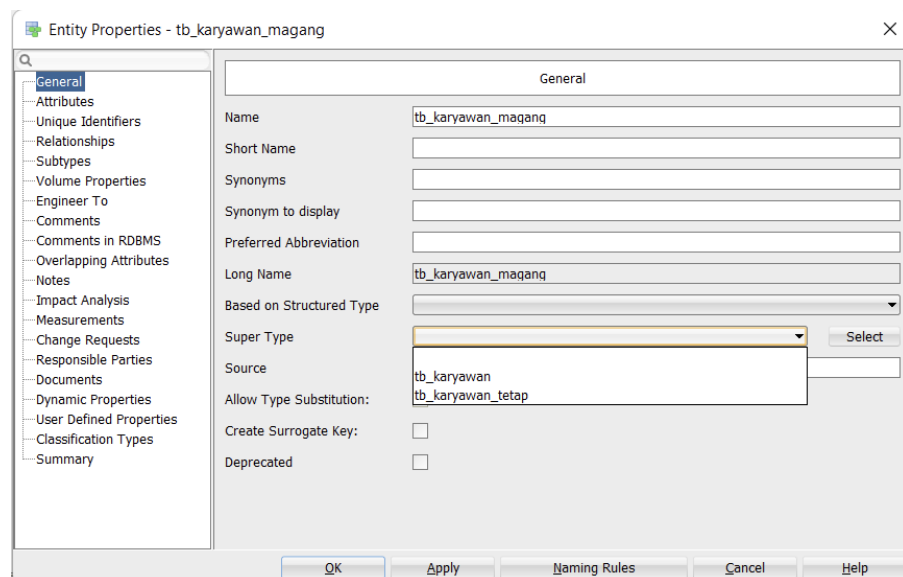
Subtypes : tb_karyawan_tetap dan tb_karyawan_magang

tb_karyawan_tetap	tb_karyawan_magang
- id	- id
- nama	- nama
- gaji	- tanggal_kontrak
	- honor



Cara Pembuatan Logical Model seperti diatas adalah sebagai berikut:

- 1) Setelah menentukan entitas supertype dan subtype, pilih entitas subtype hingga muncul dialog seperti ini:



- 2) Pilih entitas yang akan menjadi supertype, klik Ok.
- 3) Entitas subtype akan otomatis masuk ke dalam entitas supertype.
- 4) Atur ukuran kotak entitasnya sedemikian rupa hingga atribut dapat terlihat dengan jelas.

9. ARC

Arc dalam pemodelan data membantu perancang memperjelas relasi XOR (Exclusive OR) dalam pada database. Hubungan OR Eksklusif adalah hubungan antara satu entitas

dan dua (atau lebih) entitas lain dimana hanya satu dari hubungan bisa eksis dalam satu waktu.

- Contoh XOR :

*Seorang pemenang undian, hanya akan menerima sepeda motor **ATAU** uang tunai.*

Dari kalimat diatas dapat ditarik kesimpulan bahwa pemenang undian tersebut hanya dapat menerima sepeda motor, atau hanya dapat menerima uang tunai, **HANYA SALAH SATU TIDAK KEDUANYA.**

- Contoh lain :

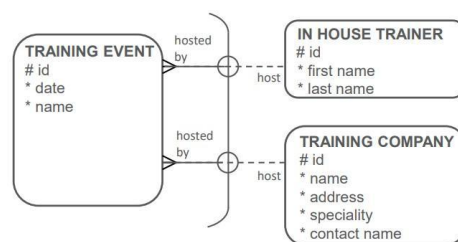
Entitas training event dapat memiliki relasi arc ke in house trainer dan training company.

Dari contoh di atas dapat dijelaskan ketika suatu training event sudah berelasi dengan salah in house trainer, maka tidak bisa berelasi dengan training company, begitu juga sebaliknya.

Kesimpulannya ARC menunjukkan bahwa setiap instance dari entitas tersebut hanya dapat memiliki satu relasi yang valid dari relasi di dalam busur pada satu waktu.

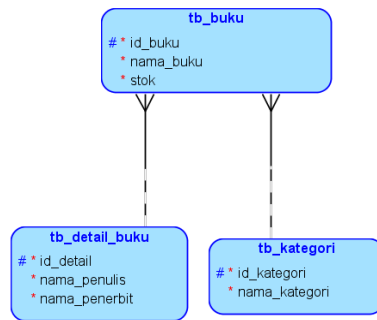
Ciri-ciri Arc:

- Selalu terdiri dari 1 entitas.
- Dapat terdiri dari 2 atau lebih relasi.
- Tidak semua relasi dari suatu entitas dapat dimasukkan dalam arc.
- Harus memiliki semua relasi mandatory atau semua relasi opsional.

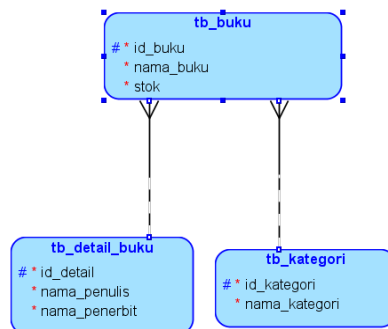


Cara Pembuatan:

- Buat 3 entitas dan 2 kardinalitas

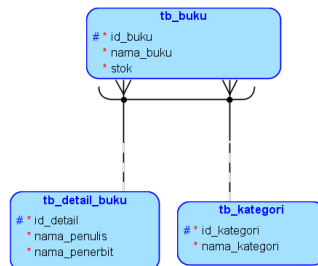


b) Klik ctrl + pilih Entitas Utama dan ke - dua Kardinalitas



c) Kemudian akan muncul ikon , klik ikon New Arc tersebut.

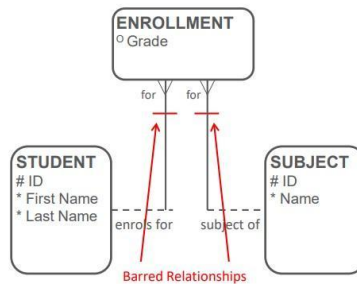
d) Hasilnya akan seperti gambar dibawah ini



10. BARRED

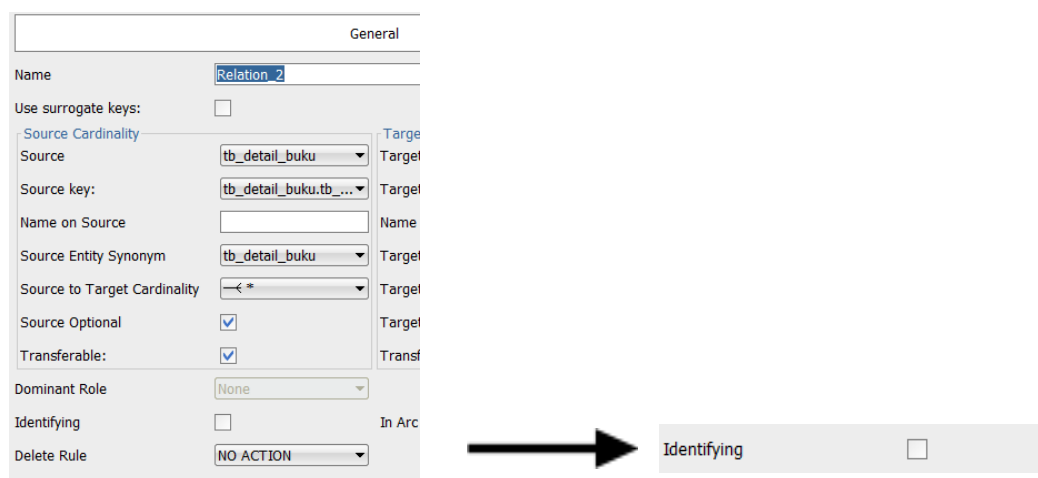
Relasi yang berasal dari entitas persimpangan dengan UID yang sama. Pada studi kasus dibawah:

- Setiap mahasiswa bisa saja melakukan enrollment pada satu atau lebih subject.
- Setiap subject bisa saja di-enroll oleh satu atau lebih mahasiswa.



Cara Pembuatan:

- Pilih One to Many Relations
- Pada **Relations Properties Dialog** → **General**, klik **Identifying**.



11. Transferability dan Non-Transferable Relationship

a) Transferability

Transferability adalah sebuah relasi antar entitas yang nilainya dapat berpindah (Transferable). Dalam suatu desain database yang dibuat ada sebuah relasi antar entitas. Setiap entitas yang berelasi dengan entitas lainnya saling berpengaruh.

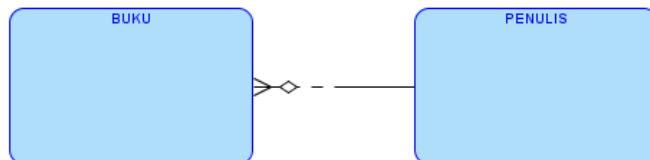
Contoh : Dalam pembuatan desain database Kepegawaian terdapat hubungan antara EMPLOYEE dan DEPARTMENT.

Setiap EMPLOYEE ditugaskan ke sebuah DEPARTMENT. Dalam hal ini seorang EMPLOYEE bisa dipindah tugaskan untuk ke sebuah DEPARTMENT lain, hal ini kita sebut dengan relasi Transferability.

b) Non-Transferable

Non-Transferability adalah sebuah relasi antar entitas yang nilainya tidak dapat berpindah/dipindahkan (Transferable). Ini merupakan kebalikan dari relasi Transferability. Untuk non-transferable relationship ditandai dengan lambang diamond.

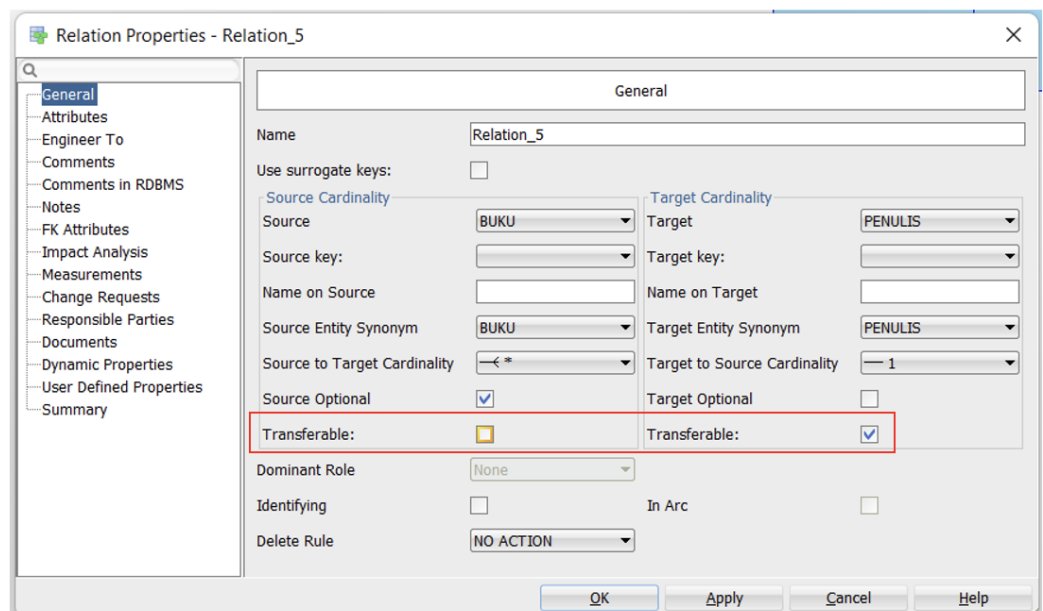
Contoh : Dalam pembuatan desain database Perpustakaan terdapat hubungan antara BUKU dan PENULIS.



Satu BUKU tidak bisa untuk berpindah nilai menjadi PENULIS yang lain, karena nilai BUKU sudah ditentukan nilainya di awal dan tidak dapat digantikan. Dalam hal ini hubungan relasi antara BUKU dan PENULIS disebut Non-Transferable.

Cara Pembuatan :

- Pilih Relations yang kalian butuhkan untuk menghubungkan tabel
- Pada **Relations Properties Dialog** → **Transferable**, klik **V** untuk menjadi **Transferability** dan hapus **V** untuk **Non-Transferable**

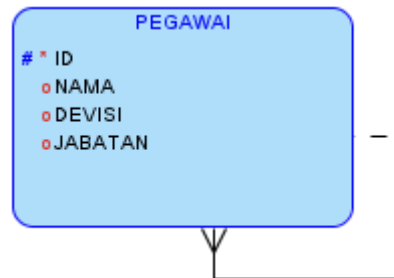


12. Recursive Relationship

Rekursif adalah pemodelan data yang memiliki relasi pada entitas itu sendiri.

Contoh :

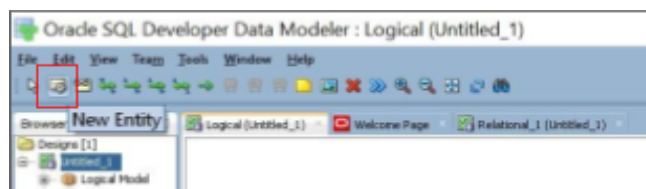
Pada sebuah perusahaan terjadi dimana seorang pegawai dapat menjadi kepala atau atasan pegawai lainnya.



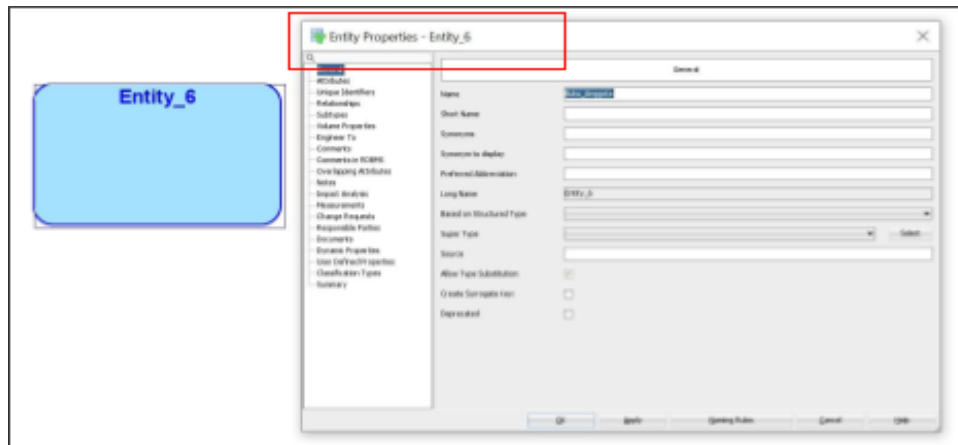
KEGIATAN PERCOBAAN

A. Tutorial Pembuatan Conceptual Data Model (CDM)

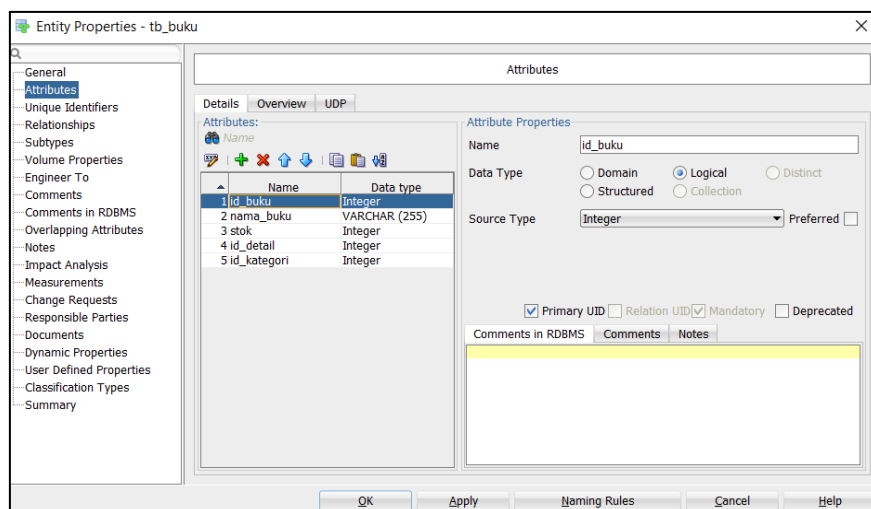
1. Unduh dan install SQL Developer Data Modeler pada link <https://www.oracle.com/tools/downloads/sql-data-modeler-downloads.html> serta sesuaikan dengan jenis OS yang Anda gunakan saat ini.
2. Setelah melakukan proses instalasi dengan benar, jalankan aplikasi dengan nama file **"datamodeler"** pada folder hasil unggahan dalam format zip yang telah di-extract.
3. Pastikan Anda berada pada lembar kerja **"Logical (Untitled_1)"**.
4. Buatlah entitas-entitas yang dibutuhkan dengan mengklik ikon berikut:



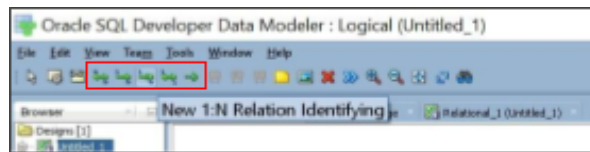
5. Arahkan kursor pada lembar kerja dan bentuklah entitas (cara yang dilakukan untuk membentuk persegi pada entitas dapat dilakukan seperti membuat *shape* pada Microsoft Word). Setelah entitas dibuat, maka akan muncul jendela **"Entity Properties"** seperti pada gambar.



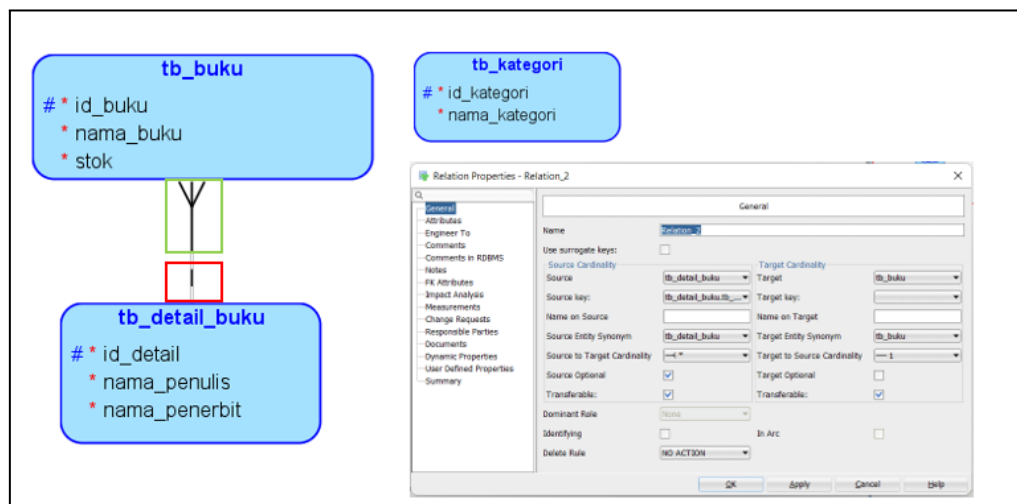
6. Pilih menu **“General”** dan atur nama entitas yang sesuai, kemudian tekan tombol **“Apply”** dan **“OK”**.
7. Untuk membuat atribut-atribut yang dibutuhkan, Anda dapat beralih pada menu **“Attributes”**. Tekan ikon tambah yang berwarna hijau untuk menambahkan atribut. Atur kembali atribut pada kolom **“Attributes Properties”**.
8. Isilah nama yang sesuai, tipe data dengan pilihan **“Logical”**, serta **“Source Type”** atau jenis dari atribut yang dapat berupa data *numeric*, *varchar*, *varbinary*, dll. Apabila atribut tersebut merupakan **primary key**, Anda dapat mencentang pilihan **“Primary UID”** dan apabila atribut bersifat wajib diisi (*mandatory*), Anda dapat mencentang pilihan **“Mandatory”** pada kolom tersebut. Namun, jika atribut tersebut tidak bersifat wajib (*optional*), Anda tidak perlu mencentang apapun. Kemudian, tekan tombol **“Apply”** dan **“OK”**.



9. Susunlah relasi yang dibutuhkan dengan memanfaatkan ikon-ikon yang ada pada gambar. Setiap ikon memiliki kriteria masing-masing. Anda dapat mengecek setiap kriterianya dengan mengarahkan kursor pada ikon relasi.

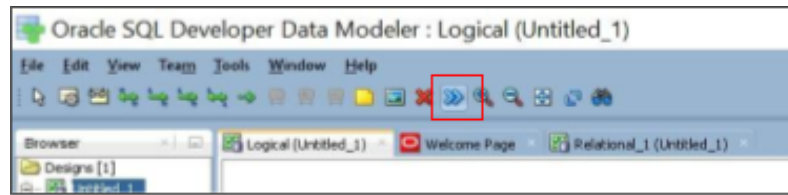


10. Tariklah garis relasi dari entitas satu dengan entitas lainnya, kemudian akan muncul jendela “**Relationship Properties**”. Pilih menu “**General**” dan atur kembali jenis relasi yang dipilih. Beberapa pengaturan yang harus Anda isi adalah nama relasi serta kardinalitas entitas sumber dan tujuan.
11. Pada contoh berikut telah dipilih relasi **One-to-Many (1:N)** antara entitas “**tb_buku**” dan “**tb_detail_buku**”.
12. Perhatikan kotak berwarna merah dan hijau pada gambar. **Kotak merah** merupakan garis kardinalitas dari entitas “**tb_buku**” dan **kotak hijau** merupakan simbol dari opsionalitas dari entitas yang sama. Apabila kardinalitas dari entitas tersebut bersifat opsional, maka Anda dapat mencentang pilihan “**Source Optional**” dan garis yang terbentuk akan menjadi putus-putus. Sedangkan, jika pilihan tersebut tidak dicentang, garis yang terbentuk adalah garis *solid* (tidak bersifat opsional). Hal demikian juga terjadi pada entitas tujuan.

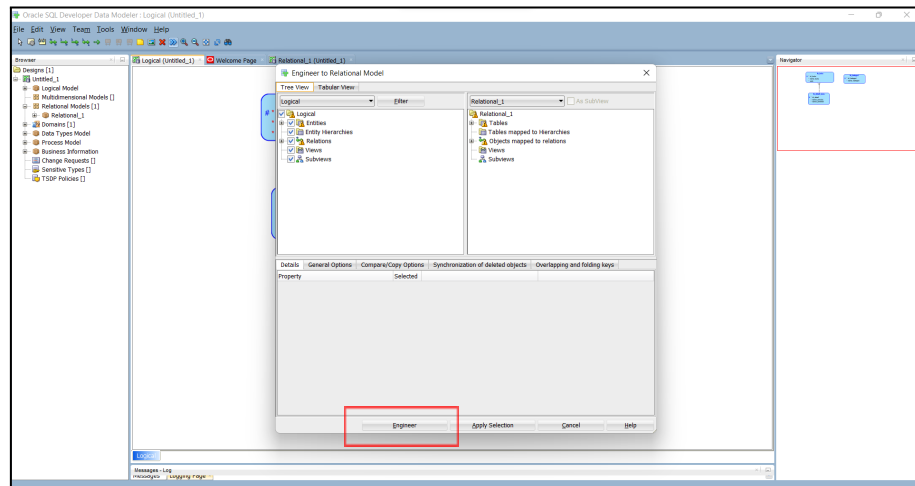


B. Tutorial Konversi CDM Ke PDM

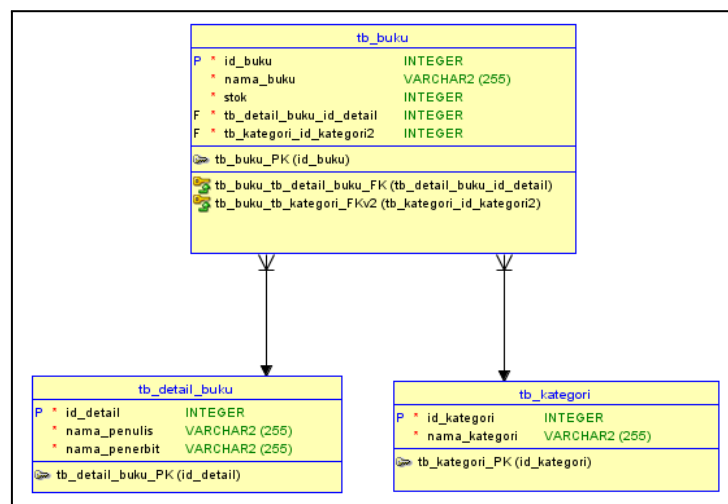
1. Setelah semua entitas dan relasi dibentuk pada CDM, pilihlah ikon pada kotak berwarna merah berikut untuk dikonversi.



2. Tekan tombol “**Engineer**” pada jendela yang muncul.



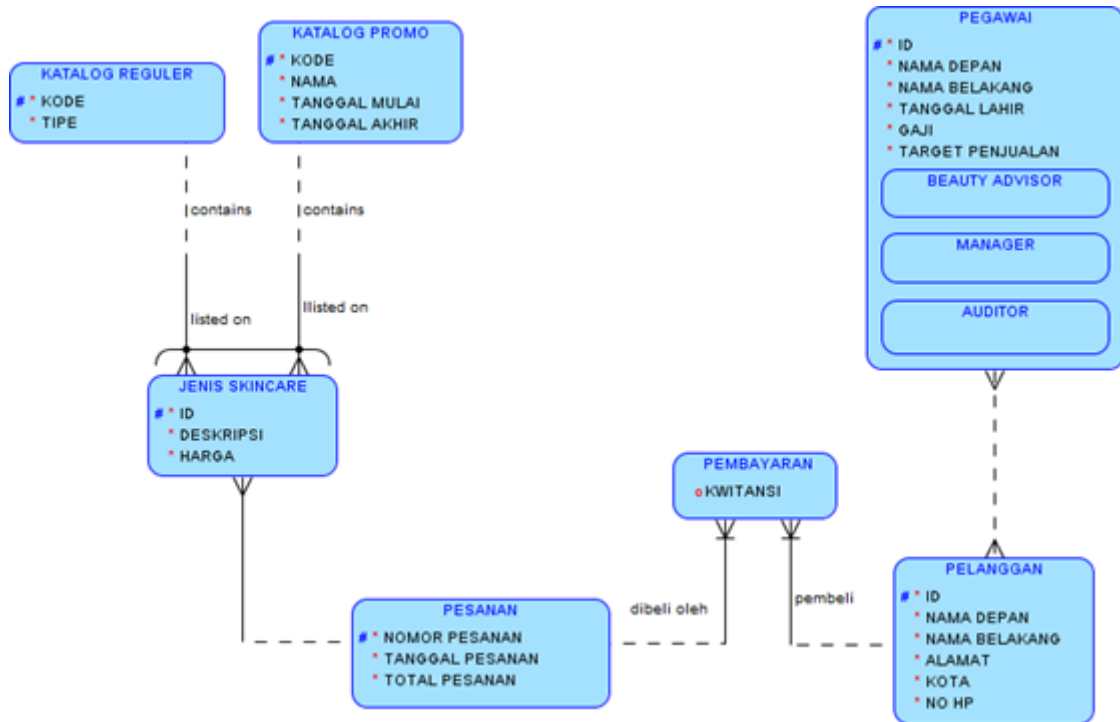
3. Setelah menekan tombol tersebut, Anda akan dialihkan pada lembar kerja “**Relational_1 (Untitled_1)**” dengan hasil yang tampak pada gambar.



4. Simpan seluruh desain *database* yang telah Anda buat pada lokasi yang sesuai.

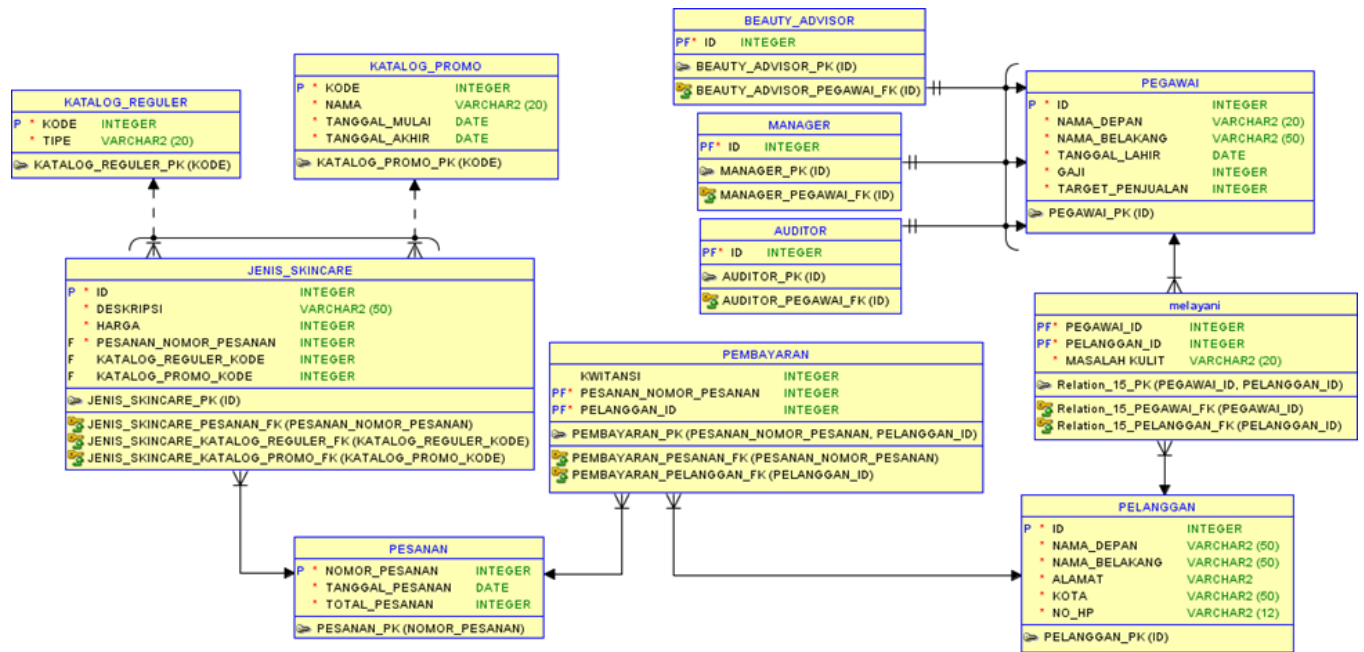
LATIHAN

1. Pada suatu toko Skincare terdapat beberapa barang yang termasuk ke dalam katalog reguler atau katalog promo. Toko tersebut memiliki pegawai yang terbilang banyak dan dibagi berdasarkan jabatan seperti manager, auditor, beauty advisor, dsb. Buatlah CDM sesuai dengan studi kasus di atas!



Bisa seperti ini atau bisa kalian tambahkan agar lebih detail

2. Buatlah konversi ke PDM dari hasil CDM yang telah dibuat pada soal nomor 1 seperti gambar di bawah



3. Pelajari alur pembuatan serta detail dari setiap bagian entitas yang terbentuk pada *Physical Data Model* yang telah Anda buat