

# **LAPORAN PRAKTIKUM CODELAB BASDAT 4C**

## **MODUL 2**



**Nama:** Wira Yudha Aji Pratama

**NIM:** 202310370311010

**Kelas:** Basis Data C

**CODELAB:****CODELAB**

1. Sebuah dealer mobil ingin mengembangkan sistem manajemen dealer mobil untuk mengelola mobil-mobil yang tersedia di dealer mereka. Mereka ingin sistem ini mencatat informasi tentang setiap mobil, termasuk merk mobil, seri mobil, jenis mobil, jumlah yang tersedia, dan lokasi dealer. Setiap mobil hanya dapat berada di satu lokasi dealer, tetapi satu lokasi dealer dapat memuat beberapa mobil yang berbeda.
2. Buatlah Logical Data Model (LDM) untuk sistem manajemen dealer mobil ini. Identifikasi serta hubungan antar entitas. Berdasarkan LDM yang telah dibuat, konversikan model tersebut menjadi Physical Data Model (PDM) dengan menambahkan rincian implementasi seperti tipe data kolom dan indeks. **Jangan lupa menerapkan konsep yang dipelajari diatas (UID, Relasi, Kardinalitas, dst)**

Contoh :

Entitas Utama : Dealer, Mobil, dst

Atribut : id\_dealer, merk\_mobil, dst

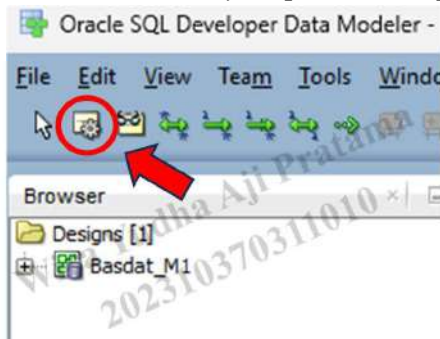
Hubungan : Setiap mobil memiliki satu dealer (Relasi :M)

NB: Pengerjaan codelab dikerjakan dalam bentuk laporan, jelaskan secara detail pengerjaan codelab tersebut berdasarkan materi pada modul 1 dan modul 2 yang mencakup (Relasi, Opsionalitas, Precision & Scale, dll).

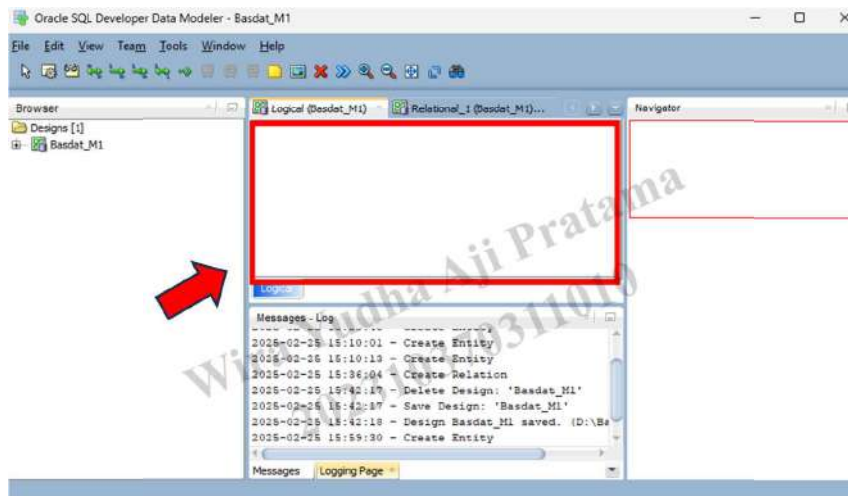
**Catatan: Jika ditemukan laporan yang sama dengan praktikan yang lain, maka asisten berhak tidak memberikan nilai!**

**LANGKAH-LANGKAH:****1. Membuat Entitas Baru**

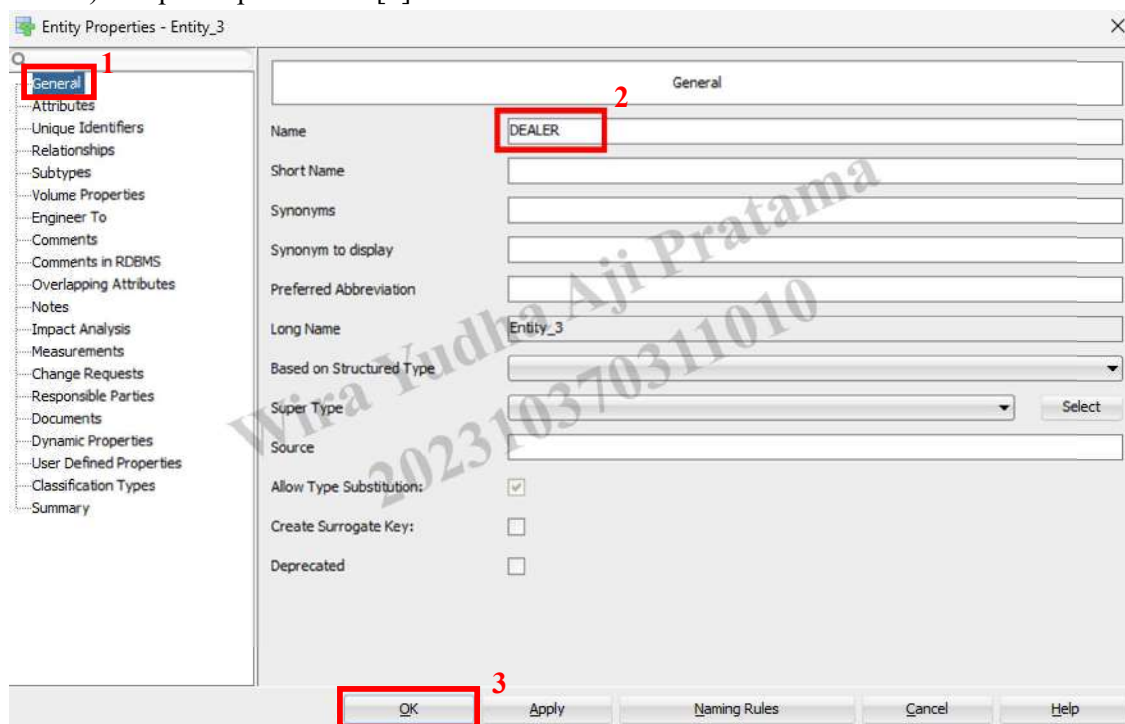
- Langkah pertama yang kita lakukan adalah membuat Entitas Baru. Caranya adalah dengan tekan tombol 'New Entity' seperti contoh pada gambar berikut ini:



- Setelah itu silahkan klik kiri di mana saja pada area yang ditandai pada gambar untuk membuat Entitas Baru:



- Selanjutnya akan muncul tab seperti pada gambar berikut ini. Silahkan ganti nama pada textfield 'Name' [2] di tab 'General' [1] yang diberi tanda pada gambar berikut ini (ganti dengan nama yang sesuai.) Selepas tu pencet 'ok' [3]:



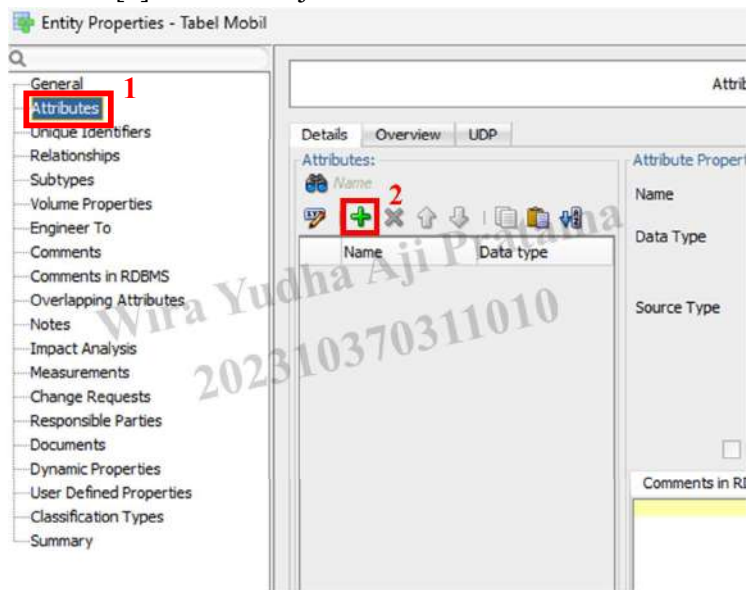
- Jika sudah maka tab CDM akan terlihat seperti gambar dibawah. Saya buat 2 entitas sesuai instruksi di modul, yaitu;
  - 1) DEALER
  - 2) MOBIL



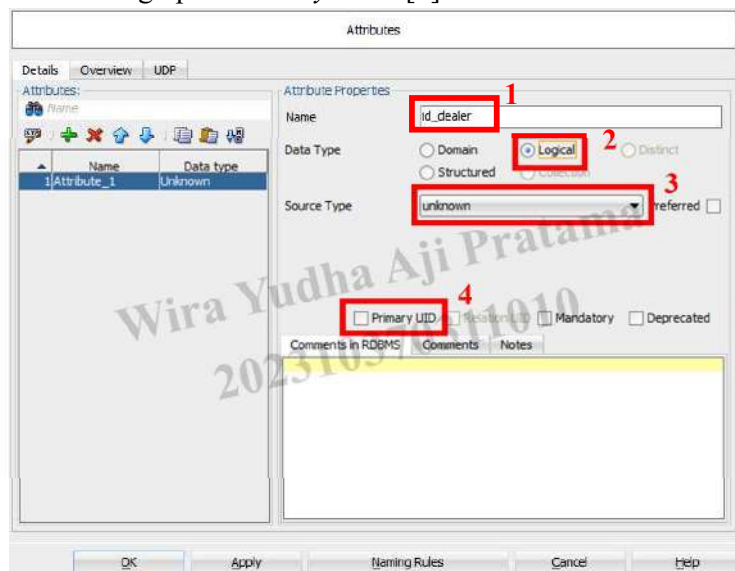
Entitas ini yang nantinya akan kita isi dengan beberapa Atribut.

## 2. Membuat Attribute di Setiap Entitas

- Double klik pada entitas yang telah dibuat, lalu pergi ke tab '*Attributes*' [1]. Disitu silahkan klik ikon '+' [2] berwarna hijau untuk menambahkan Attributes.



- Setelah itu, kita dapat memberi nama pada textfield [1] '*name*'. Berikan nama yang sesuai. Lalu berikan tipe data dengan cara check tipe data '*logical*' [2], kemudian tekan dropdown source type yang awalnya '*unknown*' [3]. Kemudian ubah menjadi tipe data yang sesuai. Kita juga bisa mencentang opsi '*Primary UID*' [4] untuk membuat attribute ini menjadi Primary Key (PK.)



- Dapat dilihat pada gambar dibawah, di nomor [4] itu adalah attribute yang sudah kita buat. Kita juga bisa memberi notes [2], serta memberikan penandaan atribut (Mandatory) [1] yang berarti tidak boleh bernilai NULL. Jika sudah, silahkan tekan '*Apply*' [3]. Kemudian di bagian '*Size*' [5] kita bisa menentukan size dari variabel yang kita pilih (precision & scale untuk variabel seperti float)

Attributes

Details Overview UDP

Attributes:

Name	Data type
1 id_dealer	Integer
2 nama_dealer	VARCHAR (255)
3 lokasi	VARCHAR

Attribute Properties

Name: lokasi

Data Type: ☐ Domain ☒ Logical ☐ Distinct

☐ Structured ☐ Collection

Source Type: VARCHAR Preferred ☐

Size: 255

Units:

☐ Primary UID ☒ Mandatory ☐ Deprecated

Comments in RBMS: Comments Notes

Buttons: OK Apply Cancel Help

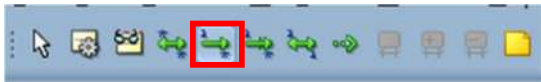
- Jika sudah, maka CDM dapat terlihat sebagai berikut:



Dengan demikian, dapat dikatakan bahwa Entitas 'DEALER' merupakan **Prime Entity** karena dapat berdiri sendiri dan memiliki Primary Key.

### 3. Membuat relasi

- Untuk membuat relasi, pertama tekan icon yang ditandai pada gambar berikut:



- Kemudian klik entitas MOBIL dan hubungkan ke DEALER, maka akan muncul tab sebagai berikut:

General

Name: Relation\_1

Use surrogate keys: ☐

Source Cardinality

Source: DEALER

Source key: DEALER.DEALER PK

Name on Source: dapat memiliki banyak mobil

Source Entity Synonym: DEALER

Source to Target Cardinality: 1-\*

Source Optional: ☐

Transferable: ☒

Dominant Role: None

Identifying: ☐

Delete Rule: NO ACTION

Target Cardinality

Target: MOBIL

Target key:

Name on Target: pat dimiliki oleh satu dealer

Target Entity Synonym: MOBIL

Target to Source Cardinality: 1

Target Optional: ☒

Transferable: ☒

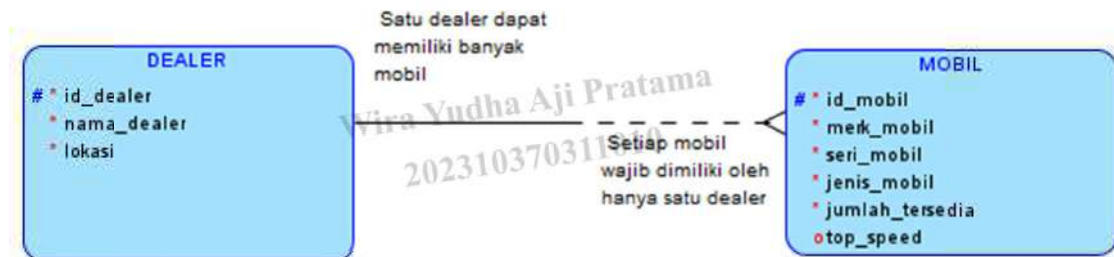
In Arc: ☐

Buttons: OK Apply Cancel Help

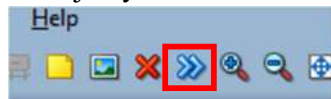


- [1] Ini adalah nama relasi.
- [2] Ini adalah nama relasi dari source (atau target.)
- [3] Ini adalah jenis relasi (many atau one.)
- [4] Ini untuk menentukan hubungannya (optional / mandatory)
- [5] Jika sudah silahkan tekan 'ok.'

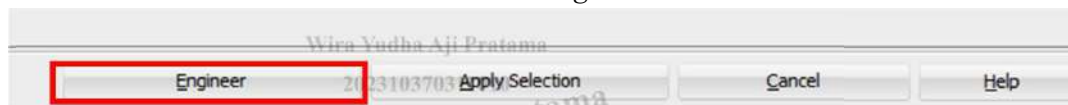
- Jika sudah, maka CDM akan terlihat seperti gambar berikut:



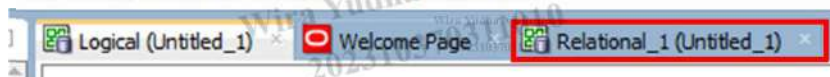
- Selanjutnya silahkan tekan icon pada gambar berikut:



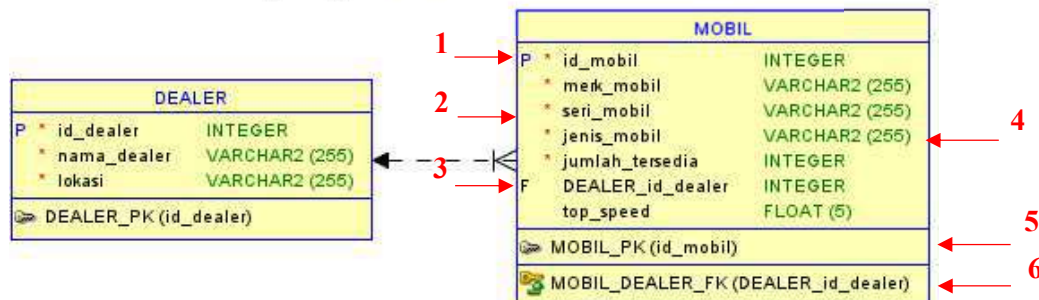
- Lalu akan muncul tab baru. Silahkan tekan 'Engineer'



- Jika sudah silahkan beralih ke tab relational:



- Maka akan terlihat seperti gambar berikut:



- [1] P disini menandakan bahwa attribute tersebut merupakan Primary Key (PK)
- [2] tanda asterisk (\*) menandakan bahwa attribute tersebut bersifat mandatory (tidak boleh NULL)
- [3] F disini berarti attribute tersebut merupakan Foreign Key (FK)
- [4] Tulisan berwarna hijau disebelah kanan merupakan tipe data dan juga size nya.
- [5] Menyediakan detail tambahan, seperti nama constraint (MOBIL\_PK untuk Primary Key).
- [6] Mencakup informasi tentang hubungan Foreign Key (seperti ke tabel lain, yaitu Dealer).

## PENJELASAN

### 1. Penjelasan Jenis-Jenis Entitas:

Jenis Entitas	Penjelasan
Prime Entity	Prime Entity adalah entitas mandiri (independent entity) yang tidak bergantung pada entitas lain untuk eksistensinya. Data dalam entitas ini memiliki makna penuh tanpa membutuhkan referensi dari entitas lain.

Dependent Entity	Dependent Entity adalah entitas yang keberadaannya bergantung pada entitas lain, dalam hal ini Prime Entity. Data di dalam Dependent Entity hanya memiliki makna jika dihubungkan dengan Prime Entity.
------------------	--

## 2. Jenis Entitas yang digunakan:

Nama Entitas	Jenis Entitas	Penjelasan
MOBIL	Prime Entity	Tabel DEALER dapat eksis dan memiliki atribut (id_dealer, nama_dealer, dan lokasi) tanpa memerlukan hubungan dengan entitas lain.
DEALER	Dependent Entity	Tabel MOBIL memiliki kolom id_dealer yang merupakan <i>foreign key</i> dan mengacu pada kolom id_dealer di tabel Dealer. Hal ini menunjukkan bahwa setiap MOBIL harus terkait dengan satu dealer tertentu untuk dapat eksis secara logis dalam database.

## 3. Penjelasan Jenis-Jenis Attribute:





Jenis Atribut	Penjelasan
Tunggal	Tidak bisa dipecah
Gabungan	Bisa dipecah menjadi bagian kecil
Nilai Tunggal	Hanya satu nilai per entitas
Nilai Ganda	Bisa lebih dari satu nilai per entitas
Volatil	Data yang sering berubah seiring waktu atau aktivitas pengguna
Non-Volatil	Data yang tetap atau jarang berubah setelah dibuat

## 4. Jenis Attribute yang digunakan:

DEALER				
Atribut	Tunggal / Gabungan	Nilai Tunggal / Nilai Ganda	Volatil / Non-Volatil	Mandatory / Optional
id_dealer (PK)	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
nama_dealer	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
lokasi	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
MOBIL				
Atribut	Tunggal / Gabungan	Nilai Tunggal / Nilai Ganda	Volatil / Non-Volatil	Mandatory / Optional
id_mobil (PK)	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
merk_mobil	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
seri_mobil	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
jenis_mobil	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
jumlah_tersedia	Tunggal	Nilai Tunggal	Volatil	Mandatory
top_speed	Tunggal	Nilai Tunggal	Non-Volatil	Opsional

id_dealer (FK)	Tunggal	Nilai Tunggal	Non-Volatil	Mandatory
----------------	---------	---------------	-------------	-----------

### 5. Penjelasan Jenis-Jenis Relasi:

Jenis Relasi	Penjelasan
	<b>"Garis Putus-Putus"</b> Digunakan untuk menunjukkan hubungan optional (opsional).
	<b>"Garis Lurus"</b> digunakan untuk menunjukkan hubungan mandatory (wajib).
	<b>"Many"</b> merujuk pada situasi di mana lebih dari satu entitas dari sebuah tabel dapat berhubungan dengan lebih dari satu entitas dari tabel lain.
	<b>"One"</b> mengacu pada situasi di mana satu entitas dari sebuah tabel hanya dapat berhubungan dengan satu entitas dari tabel lain.

### 6. Penjelasan tambahan

- Saya tambahkan attribute tambahan di tabel MOBIL yaitu 'top\_speed' agar dapat menerapkan materi Precision & Scale pada modul 2 ini. Penjelasan Precision dan Scale:
  - Precision: Total digit (baik sebelum maupun sesudah desimal). Dalam attribute yang saya buat, precision adalah 5, artinya kita dapat menyimpan angka hingga 999.99 (tiga digit di sisi kiri desimal dan dua digit di sisi kanan desimal).
  - Scale: Jumlah digit di sisi kanan desimal, di sini 2, memungkinkan penyimpanan nilai desimal hingga dua tempat (misalnya, 180.50).
- Kardinalitas antara Dealer dan Mobil merupakan Kardinalitas One-to-Many (1:N). Yang berarti; Satu Dealer dapat memiliki banyak Mobil (1:N), dan setiap Mobil wajib hanya terhubung ke satu Dealer.