

Model Data

Model data merupakan kumpulan konseptual untuk menggambarkan data, hubungan data, dan batasan data. Model data dibuat dengan tujuan agar kita bisa mengetahui makna dari data dan juga relasi antar datanya. Terdapat banyak cara untuk membuat model data, dan yang paling populer dan banyak digunakan adalah menggunakan ERD (Entity Relationship Diagram).

1. Entity Relationship Model

Model data dalam Basis data kebanyakan dibuat dalam bentuk Entity Relationship Model. Dalam Model ER, data dunia nyata direpresentasikan dalam beberapa komponen diantaranya sebagai berikut.

1. Entity

Entity merupakan individu yang mewakili data (fakta) yang nyata, dan bisa dibedakan dari sesuatu yang lain. Saat menentukan sebuah Entity, kita perlu melihat ruang lingkup yang sedang kita kerjakan. Sebagai contoh, Orang mungkin bisa kita anggap sebagai satu Entity, namun dalam ruang lingkup rumah sakit, bisa jadi tidak dibuat dalam satu Entity, karena orang tergantung jenisnya, apakah orang tersebut Entity pasien atau Entity dokter.

2. Attribute

Setiap Entity pasti memiliki Attribute yang mendeskripsikan karakteristik dari Entity tersebut. Penentuan Attribute dalam Entity biasanya berdasarkan fakta yang ada dan yang dibutuhkan dalam ruang lingkupnya saja. Contoh dalam Entity pasien, kita bisa menambahkan Attribute seperti Nomor pasien, Nama pasien, Alamat pasien, dan lain - lain. Fakta yang tidak termasuk dalam ruang lingkup mungkin tidak perlu dijadikan sebagai Attribute, misalnya Hobby pasien, Harta pasien, dan lain sebagainya. Dalam Attribute, biasanya juga terdapat Attribute kunci yang digunakan sebagai kunci representasi dari Entity tersebut, contohnya ialah Nomor ID pasien.

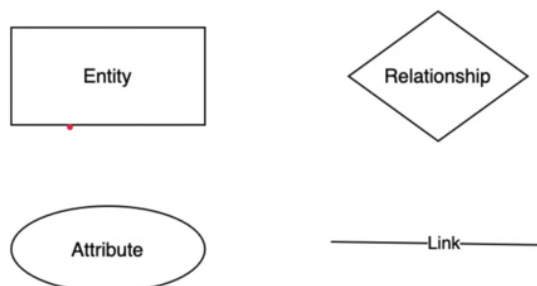
3. Relationship

Relationship merupakan hubungan yang terjadi antar Entity. Misalnya dalam ruang lingkup rumah sakit, hubungan Entity dokter dan pasien adalah merawat.

Entity Relationship Diagram

Setelah menentukan Entity, Attribute, dan Relationship, selanjutnya dalam ER biasanya akan dibuat dalam bentuk diagram. Entity Relationship Diagram (ERD) sangatlah sederhana, kita dapat merepresentasikan tiap Entity, Attribute, Relationship dan hubungannya (Link) dengan gambar sebagai berikut.

Entity Relationship Diagram



Tahapan Pembuatan ERD

Contoh Kasus : Toko Online

1. Menentukan Entity

Hal pertama yang perlu kita tentukan dalam membuat ERD ialah menentukan Entity yang akan terlibat dalam toko online tersebut. Misal dalam toko online, kita akan menentukan tiga Entity yang terlibat, yaitu pembeli, penjual, dan barang (Perlu diingat, bahwasanya dalam toko online yang nyata, sebenarnya banyak sekali Entity yang terlibat).

Kode : Diagram Entity



2. Menentukan Attribute Key

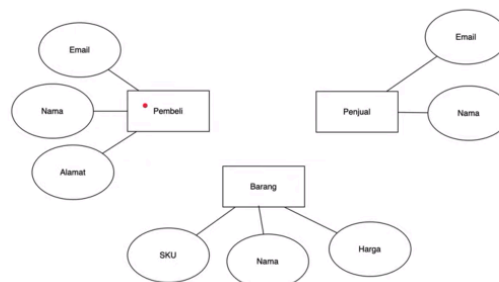
Setiap Entity wajib memiliki Attribute Key, maka selanjutnya kita perlu menentukan Attribute Key dari tiap Entity. Misalnya sebagai berikut.

- Pelanggan, Attribute Keynya adalah Email.
- Pembeli, Attribute Keynya adalah Email.
- Barang, Attribute Keynya adalah SKU (Surat Keterangan Usaha)

3. Menentukan Attribute

Setelah menentukan Attribute key, selanjutnya ialah menentukan Attribute dari tiap Entitynya yang masuk ke ruang lingkup yang akan kita buat.

Diagram : Attribute



4. Menentukan relationship antar Entity

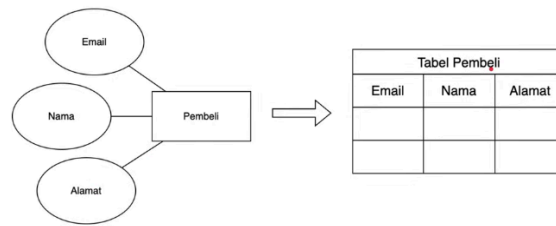
Pada tahap relationship ini, akan dibahas lebih detail pada subbab Relationship.

Implementasi Model Data

Implementasi model data merupakan tahapan untuk membuat basis data secara fisik yang ditempatkan dalam disk dengan bantuan sistem basis data. Setelah selesai membuat ERD, kita bisa mulai membuat implementasinya dalam basis data.

Secara umum, Entity yang kita buat di dalam ERD akan menjadi tabel di dalam basis data. Sedangkan Attribute yang kita buat ERD, akan menjadi kolom di tabel di dalam basis data.

Diagram : Implementasi Entity Pelanggan



2. Model Data Relational

Relationship

Relationship menunjukkan hubungan antar Entity dengan Entity yang lain. Misalnya, hubungan antara Entity penjual dengan Entity barang. Relationship diantara keduanya mengandung arti bahwa penjual sedang berjualan / menjual barang.

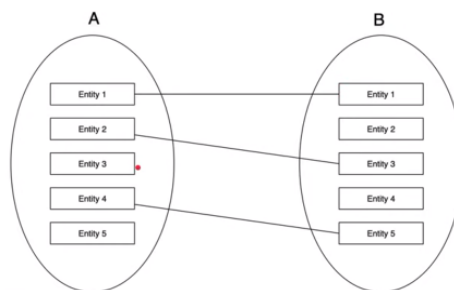
Kardinalitas / Derajat Relasi

Kardinalitas relasi menunjukkan terkait jumlah Entity yang dapat berelasi dengan Entity lainnya. Misalnya, pada kasus toko online, Entity penjual bisa berelasi dengan satu Entity Barang atau lebih ataupun bahkan tidak ada. Berikut merupakan macam - macam dari kardinalitas.

1. One to One (Satu ke Satu)

One to One artinya Entity A hanya bisa berhubungan dengan Entity B dengan maksimal hanya satu Entity saja. Begitupun sebaliknya.

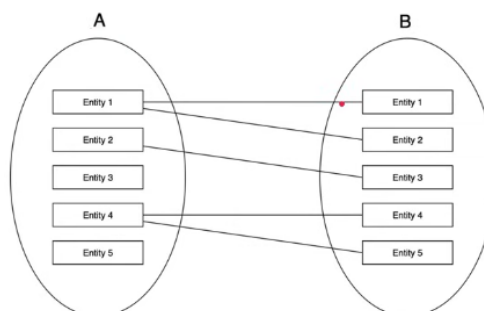
Diagram : One to One



2. One to Many (Satu ke Banyak)

One to Many artinya Entity A dapat berhubungan dengan banyak Entity B. Tetapi tidak sebaliknya, Entity B hanya bisa berelasi dengan satu Entity A.

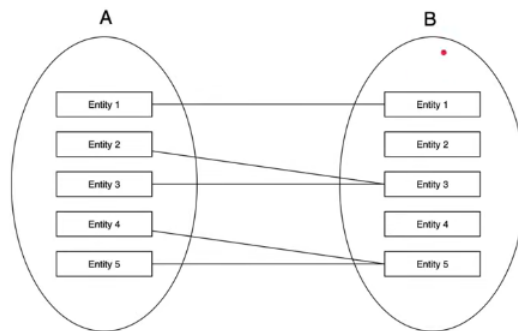
Diagram : One to Many



3. Many to One (Banyak ke Satu)

Many to One artinya Entity A hanya bisa berhubungan dengan satu Entity B. Tetapi tidak sebaliknya, Entity B bisa berelasi ke banyak Entity A.

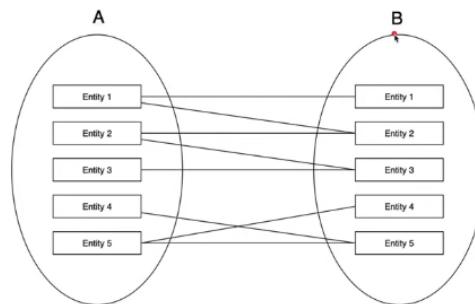
Diagram : Many to One



4. Many to Many (Banyak ke Banyak)

Many to Many artinya Entity A bisa berhubungan dengan banyak Entity B. Begitupun sebaliknya, Entity B bisa berelasi ke banyak Entity A.

Diagram : Many to Many



Tahapan Membuat ERD Lanjutan

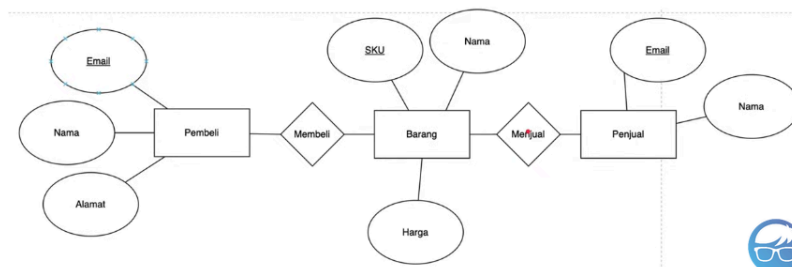
Contoh Kasus : Toko Online (Lanjutan)

Pada materi Entity Relational Model, telah dibuat Entity dan Attribute. Selanjutnya kita akan membuat relationship antar Entitynya.

- Menentukan Relationship

Menentukan relationship juga harus sesuai ruang lingkup yang kita buat. Pada kasus toko online, misal kita buat Relationship dari pembeli ke barang dengan relasi membeli dan Relationship dari penjual ke barang dengan relasi menjual.

Diagram : Relationship

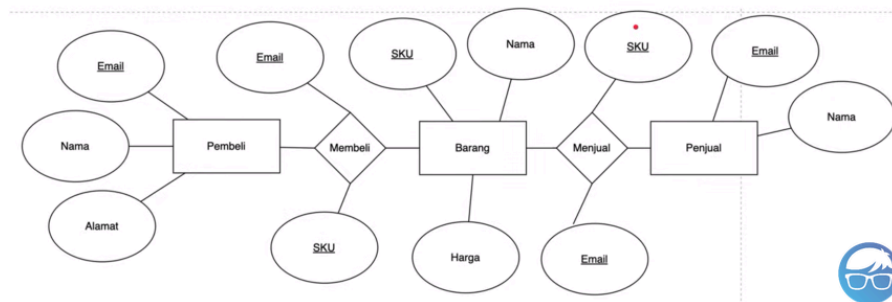


- Menentukan Attribute Foreign Key

Dalam relationship, kita juga bisa menambahkan atribut, dan biasanya kita juga perlu menentukan Attribute Foreign Key, yaitu Attribute Key di Entity yang menjadi digunakan sebagai data untuk lookup (melihat) ke Entity pemilik relasi tersebut.

Pada kasus membeli, misal kita perlu tahu siapa yang membeli (Email Pembeli) dan barang apa yang dibeli (SKU barang). Pada kasus menjual, kita perlu tahu siapa yang menjual (Email penjual) dan barang apa yang dijual (SKU barang).

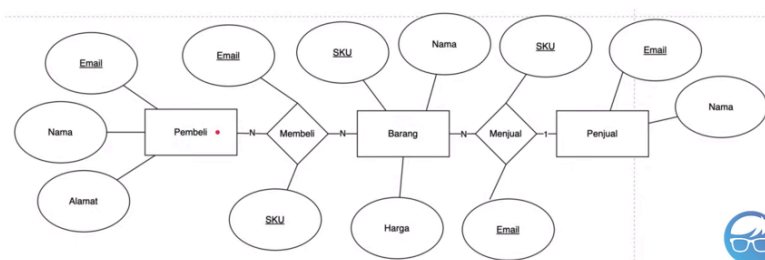
Diagram : Attribute Foreign Key



- Menentukan Derajat Relasi

Pada kasus membeli, kita bisa tentukan sebagai relasi Many to Many, karena satu pembeli bisa membeli banyak barang, satu barang bisa dibeli banyak pembeli. Pada kasus menjual, kita bisa tentukan sebagai relasi One to Many atau Many to One, karena misal satu barang hanya bisa dijual oleh satu penjual, namun satu penjual bisa menjual lebih dari satu barang.

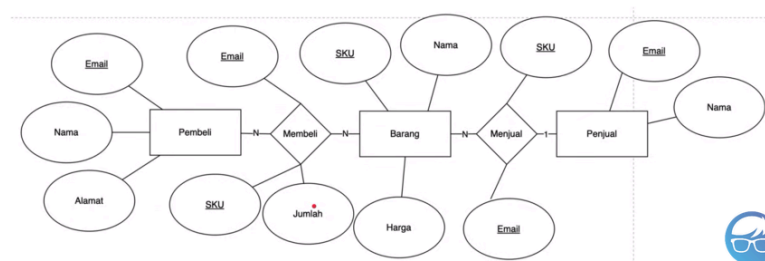
Diagram : Derajat Relasi



- Melengkapi Attribute Tambahan

Relationship pun bisa memiliki Attribute jika memang diperlukan. Pada relationship kita bisa menambahkan Attribute yang bersifat deskriptif. Misal pada relasi Membeli, kita dapat menambahkan atribut jumlah.

Diagram : Attribute

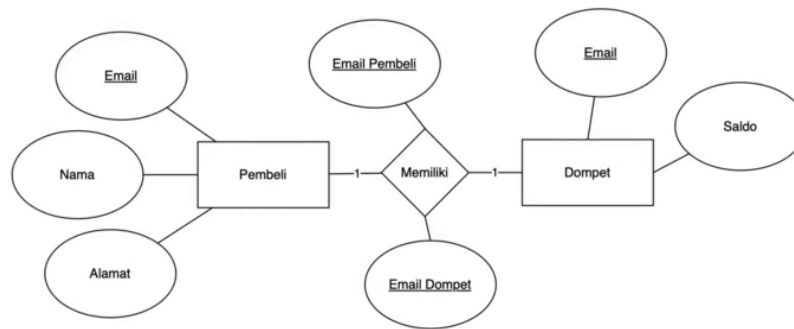


Implementasi Model Data Relational

1. Menambah Relasi One to One

Misal di kasus toko online, kita akan menambahkan Entity Dompot, dimana pembeli hanya bisa memiliki satu dompet, dan satu dompet hanya dimiliki oleh satu pembeli. Pada dompet, kita akan tambahkan Attribute Key Email dan Attribute Saldo.

Diagram : One to One



Pada kasus implementasi relasi One to One, terdapat dua cara. Jika pada kasus tiap Entity dipastikan menggunakan Attribute Key dengan nilai sama, maka kita bisa gunakan Attribute key di tiap Entity sebagai acuan untuk saling berelasi. Namun jika pada kasus tiap Entity bisa memiliki Attribute key yang berbeda, maka kita perlu membuat Attribute foreign key di salah satu Entity yang lebih lemah, dimana Attribute foreign key tersebut harus unique (tidak boleh ada yang sama).

2. Implementasi One to Many atau Many to One

Pada kasus relasi One to Many, implementasinya tergantung relasinya. Jika pada kasus relasinya terdapat Attribute deskriptif, maka diperlukan tabel baru sebagai implementasi dari Relationshipnya. Namun pada kasus relasinya tidak memiliki Attribute deskriptif, maka tidak perlu tabel baru, kita hanya butuh menambah Attribute Foreign Key di Entity yang bersifat Many.

Diagram : Implementasi One to Many

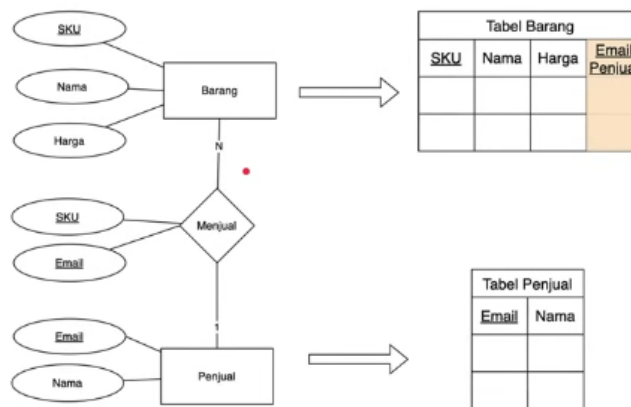


Diagram : Implementasi Many to Many

