

NORMALIZATION

KELAS PROGRAMMING FULL STACK DEVELOPER

MITRA PELATIHAN



Jabar Digital Academy

digitalacademy.jabarprov.go.id

2024

BAB II

NORMALISASI

A. Tujuan

- a. Peserta didik dapat mengimplementasikan normalisasi pada database

B. Perlengkapan

- a. Modul 2. Normalisasi.
- b. Alat Tulis
- c. Komputer dengan sistem operasi Windows bila diperlukan
- d. Aplikasi Draw.io atau Diagram Designer sejenisnya bila diperlukan

C. Materi

Dalam merancang sebuah database, kita memerlukan *Entity Relationship Diagram* (ERD) untuk menggambarkan struktur entitas, atribut, dan hubungan antar entitas dalam suatu database. ERD membantu dalam pemahaman visual tentang bagaimana data terkait dan bagaimana mereka saling berinteraksi dalam sistem.

Dalam pembuatan ERD, terkadang kita menyadari bahwa beberapa data dapat digunakan berulang kali. Misalnya, dalam tabel user, atribut seperti domisili dapat memiliki nilai yang redundan jika tidak dinormalisasikan. Hal ini dapat mengakibatkan penyimpanan data yang tidak efisien dan mempengaruhi kinerja database secara keseluruhan. Oleh karena itu, normalisasi database penting untuk mengatasi masalah seperti ini dengan meminimalkan redundansi data dan memastikan struktur database yang efisien dan konsisten.

ID	Name	Email	Domisili
1	Ahmad	Ahmad@gmail.com	Kota bandung
2	Randi	Randi@gmail.com	Kota bandung

Contoh tabel yang tidak dinormalisasikan.

Pentingnya normalisasi database adalah untuk mengurangi redundansi data, menghindari anomali dalam manipulasi data, dan memastikan struktur database yang efisien dan konsisten. Dengan melakukan normalisasi, kita dapat meminimalkan kemungkinan kehilangan data, meningkatkan kinerja query, dan memudahkan pemeliharaan database secara keseluruhan.

1) Apa itu Normalisasi?

Normalisasi adalah salah satu pendekatan logical design dari suatu database relational berdasarkan himpunan data dengan tingkat ketergantungan fungsional dan keterkaitan yang tinggi sedemikian sehingga menghasilkan struktur tabel yang normal. Tujuan dari normalisasi database adalah meminimalisasi redundansi (pengulangan data), memudahkan identifikasi entitas, mencegah terjadinya anomali data. Secara garis besar proses normalisasi model data dapat diringkas sebagai berikut:

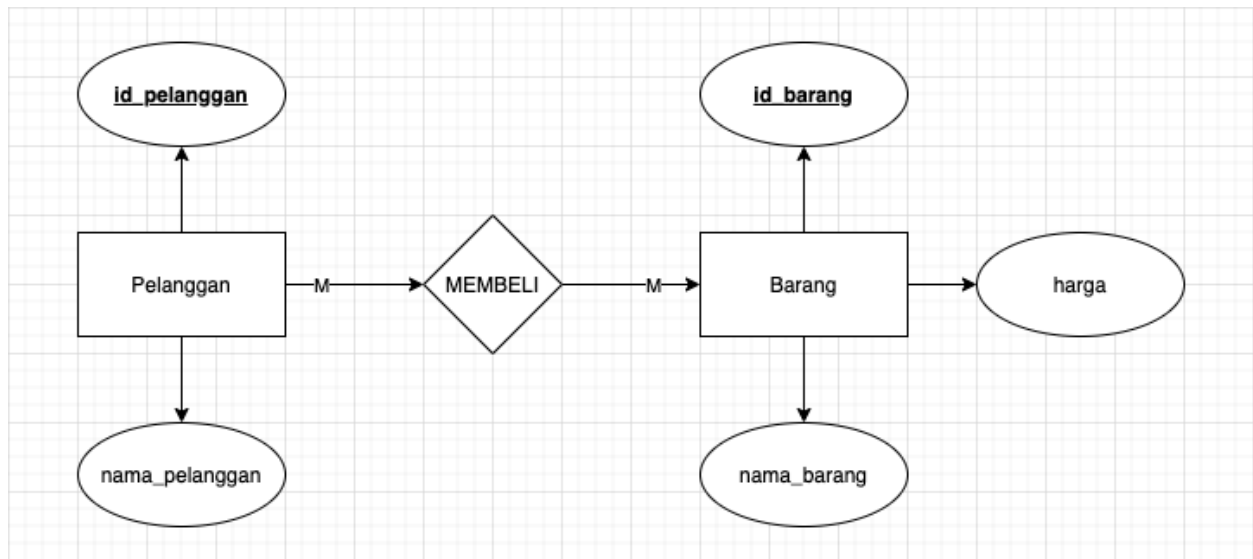
- Temukan entitas-entitas utama dalam model data.
- Temukan hubungan atau relasi antara setiap entitas.
- Tentukan atribut yang dimiliki masing-masing entitas.

2) Database Yang Tidak Normal

Rancangan desain database dikatakan tidak normal apabila terdapat relasi antar entitas yang berjenis *Many-to-Many*. Sehingga terdapat duplikasi data yang akan berulang ketika data yang sama pada entitas tersebut berinteraksi.

3) Teknik Normalisasi ERD

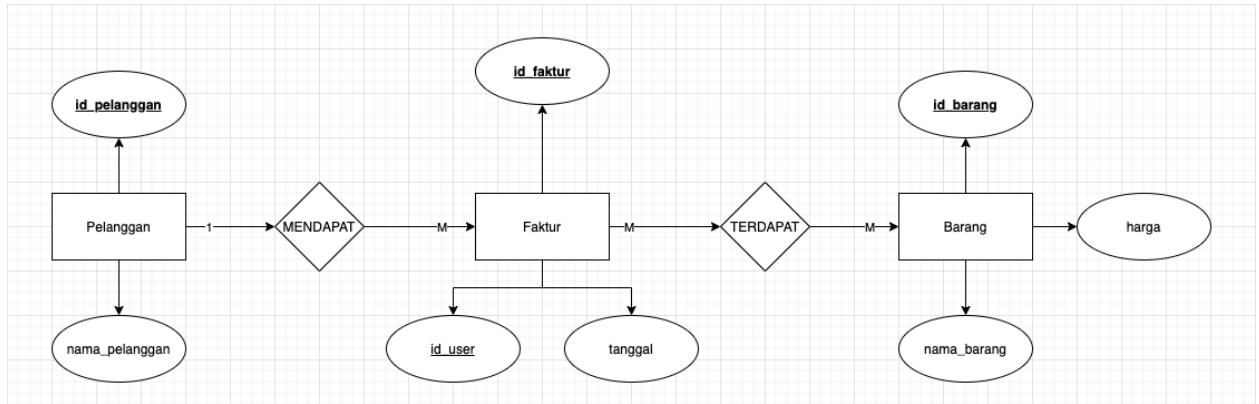
Pada pembahasan sebelumnya, kita membahas bahwa relasi *Many-to-Many* antar entitas tidak diperbolehkan dalam desain ERD kita. Berdasarkan pembahasan tersebut, kita akan membuat ERD transaksi antara pelanggan dan barang. Desain ERD awal kita adalah sebagai berikut:



Gambar 6.1. ERD Transaksi Pelanggan dan Barang

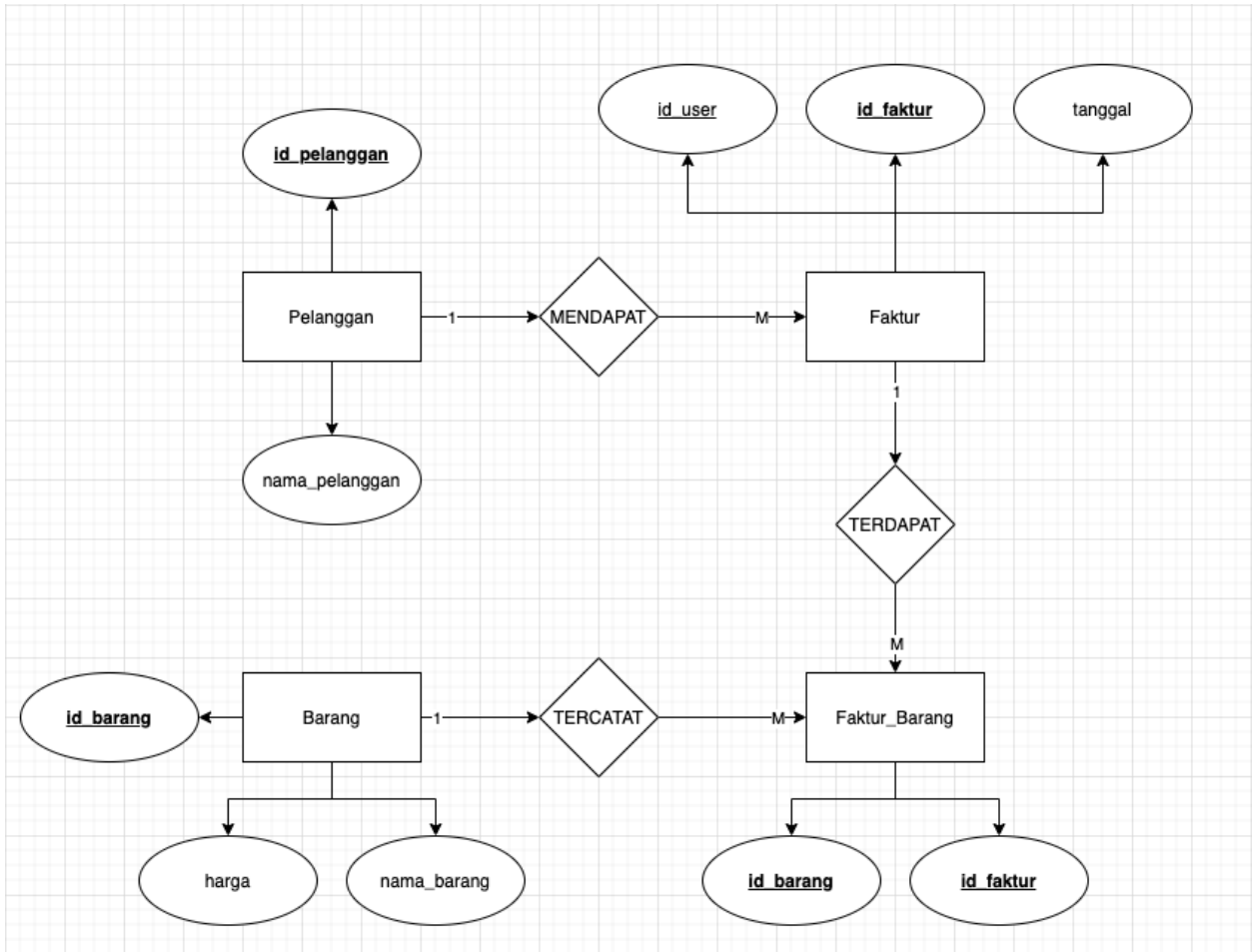
Pada gambar di atas, terlihat bahwa relasi antara entitas pelanggan dan barang adalah *Many-to-Many*. Untuk melakukan normalisasi pada relasi tersebut, langkah-langkahnya adalah sebagai berikut:

1. Ubah relasi “membeli” menjadi sebuah objek Faktur, dimana Faktur adalah objek pencatatan terhadap setiap transaksi yang berlangsung. Sehingga diagram kita akan menjadi sebagai berikut:



Gambar 6.2. ERD Transaksi Pelanggan dan Barang

2. Pada proses sebelumnya, kita telah melakukan normalisasi pada ERD kita. Sehingga, menghasilkan entitas baru bernama Faktur. Sehingga relasi antara entitas Pelanggan dan Faktur menjadi One-to-Many. Sehingga, bisa kita katakan relasi antara dua entitas tersebut sudah **normal**.
3. Akan tetapi, relasi antara entitas Faktur dan Barang ternyata masih didapati jenis relasi *Many-to-Many*. Kenapa demikian, dikarenakan satu faktur bisa terdapat jenis **banyak barang**. Dan satu jenis barang dapat tercatat dalam **banyak faktur**.
4. Sehingga kita perlu melakukan normalisasi pada relasi entitas Faktur dan Barang. Dengan membuat sebuah entitas baru dengan nama Faktur_Barang. Kemudian, desain ERD kita akan menjadi sebagai berikut:



Gambar 6.3. Desain ERD Transaksi Pelanggan dan Barang

5. Pada gambar di atas, entitas Faktur_Barang akan memiliki dua *Primary Key*. Hal ini bermaksud untuk membuat nilai kombinasi kedua data tidak terulang. Sehingga dalam satu faktur hanya terdapat barang yang bersifat *unique*.
6. Dalam gambar tersebut, sudah tidak didapati relasi dengan jenis Many-to-Many. Sehingga dapat kita simpulkan bahwa desain database kita sudah normal.

4) Teknik Normalisasi Tabel

Bentuk tidak normal akan terlihat seperti pada gambar di bawah, kita dapat melakukan normalisasi dengan melakukan beberapa tahap normalisasi berikut:

id_pelanggan	nama_pelanggan	tipe_membership	keterangan_membership	id_barang	nama_barang	harga_barang	jumlah	id_faktur	tgl_faktur
0001	Bambang	G	Gold	1101	Kopi	5000	2	0101210001	1/1/2021
				1102	Gula	3000	1		
0002	Jatmiko	P	Platinum	1101	Kopi	5000	5	0101210002	1/1/2021
				1102	Gula	3000	4		

Gambar 6.4. Tabel Tidak Normal

Keterangan: Pada kolom kode barang terdapat redudansi atau duplikasi data.

a) Bentuk Normal Pertama (1NF)

Pada bentuk normal pertama atau disebut 1NF, setiap data bersifat atomik, yaitu setiap baris hanya mempunyai satu nilai data. Contoh bentuk normal pertama adalah sebagai berikut:

id_pelanggan	nama_pelanggan	tipe_membership	keterangan_membership	id_barang	nama_barang	harga_barang	jumlah	id_faktur	tgl_faktur
0001	Bambang	G	Gold	1101	Kopi	5000	2	0101210001	1/1/2021
0001	Bambang	G	Gold	1102	Gula	3000	1	0101210001	1/1/2021
0002	Jatmiko	P	Platinum	1101	Kopi	5000	5	0101210002	1/1/2021
0002	Jatmiko	P	Platinum	1102	Gula	3000	4	0101210002	1/1/2021

Gambar 6.5. Bentuk Normal Pertama (1NF)

b) Bentuk Normal Kedua (2NF)

Pada bentuk normal kedua atau disebut 2NF. Pada bentuk normal kedua ini, tidak diperbolehkan adanya *partial functional dependency*. Sehingga, pada bentuk normal ini data harus dipisah berdasarkan *Primary Key*. Pada contoh dibawah ini bentuk normal kedua menghasilkan 4 tabel. Contoh bentuk normal kedua:

id_pelanggan	nama_pelanggan	tipe_membership	keterangan_membership	id_barang	nama_barang	harga_barang
0001	Bambang	G	Gold	1101	Kopi	5000
0002	Jatmiko	P	Platinum	1102	Gula	3000
id_faktur	tgl_faktur	id_pelanggan	id_barang	jumlah		
0101210001	1/1/2021	0001	1101	2		
0101210002	2/1/2021	0002	1102	1		
			1101	5		
			1102	4		

Gambar 6.6. Bentuk Normal Kedua (2NF)

c) Bentuk Normal Ketiga (3NF)

Dalam bentuk normal ini, tidak diperbolehkan adanya *partial transitive dependency* pada sebuah tabel. Yaitu, apabila terdapat suatu atribut yang tidak bergantung pada *primary key* tapi bergantung pada field yang lain maka atribut-atribut tersebut perlu dipisah ke tabel baru. Contohnya adalah pada keterangan_membership, yang bergantung pada tipe_membership. Sehingga, pada bentuk normal ketiga akan menjadi sebagai berikut:

id_pelanggan	nama_pelanggan			id_barang	nama_barang	harga_barang
0001	Bambang			1101	Kopi	5000
0002	Jatmiko			1102	Gula	3000
tipe_membership	keterangan_membership			id_faktur	id_barang	jumlah
G	Gold			0101210001	1101	2
P	Platinum			0101210001	1102	1
				0101210002	1101	5
				0101210002	1102	4
id_faktur	tgl_faktur	id_pelanggan				
0101210001	1/1/2021	0001				
0101210002	2/1/2021	0002				

Gambar 6.7. Bentuk Normal Ketiga (3NF)

Pada pembahasan di atas, normalisasi dilakukan ketika database sudah memiliki nilai. Sehingga kita dapat memvisualisasikan ketidaknormalan dan kenormalan dalam sebuah relasi antar entitas atau tabel.