NORMALISASI
&
ENTITY RELATIONSHIP DIAGRAM

DIAGRAM - ER

- Database desain juga termasuk diagram ER (Entity-hubungan model). Diagram ER adalah diagram yang membantu merancang database secara efektif dan efisien.

NORMALISASI

- ✓ Proses penerapan peraturan untuk desain database Anda disebut normalisasi database, atau hanya normalisasi.
- ✓ Idenya adalah untuk membantu Anda memastikan bahwa Anda telah membagi item informasi Anda ke dalam tabel yang sesuai
- ✓ Normalisasi dilakukan untuk mengurangi masalah dalam pemakaian database, yakni suatu ketidakkonsistenan (tidak normal) apabila dilakukan penghapusan *(delete)*, pengubahan *(update)* dan pembacaan *(retrieve)* pada suatu basis data.

NORMALISASI

Bentuk normalisasi adalah suatu aturan yang dikenakan pada tabel-tabel dalam **basis data** dan harus dipenuhi oleh tabel-tabel tersebut pada level-level normalisasi.

Ada macam-macam bentuk normalisasi, diantaranya adalah bentuk tidak normal, bentuk normal pertama, bentuk normal kedua dan bentuk normal ketiga.

Bentuk-Bentuk Normalisasi

a. ***Bentuk tidak normal***

Bentuk ini merupakan kumpulan data yang akan disimpan, tidak ada keharusan mengikuti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi dan data dikumpulkan apa adanya.

b. ***Bentuk normal pertama***

Suatu tabel dikatakan dalam bentuk normal pertama (1NF) bila setiap kolom bernilai tunggal untuk setiap baris. Ini berarti bahwa nama kolom yang berulang cukup diwakili oleh sebuah nama kolom (tidak perlu ada indeks dalam memberi nama kolom).

Bentuk-Bentuk Normalisasi

c. ***Bentuk normal Kedua***

- Jika tabel berada dalam bentuk normal pertama, semua kolom bukan kunci primer tergantung sepenuhnya terhadap kunci primer. Suatu kolom disebut tergantung sepenuhnya terhadap kunci primer jika nilai pada suatu kolom selalu bernilai sama untuk suatu nilai kunci primer yang sama. Peraturan ini berlaku bila Anda memiliki kunci primer yang terdiri dari lebih dari satu kolom. Misalnya, Anda memiliki tabel yang berisi kolom berikut, dimana Order ID dan Product ID membentuk kunci primer:

  Order ID (primary key)

  Product ID (primary key)

  Product Name

- Desain ini melanggar bentuk normal kedua, karena Nama Produk tergantung pada Produk ID, tetapi bukan pada ID Pesanan, sehingga tidak tergantung pada seluruh kunci primer. Anda harus menghapus Nama Produk dari table ini. Ini termasuk dalam tabel yang berbeda (Produk).

Bentuk-Bentuk Normalisasi

d. ***Bentuk normal ketiga***

Suatu tabel berada dalam bentuk normal ketiga (3NF) jika tabel berada dalam bentuk normal kedua, setiap kolom bukan kunci primer tidak memiliki ketergantungan secara transitif terhadap kunci primer.

- Bentuk normal ketiga tidak hanya mensyaratkan bahwa setiap kolom non-key tergantung pada seluruh kunci primer, tapi kolom non-kunci tsb menjadi independen satu sama lain. Misalnya, Anda memiliki tabel yang berisi kolom berikut:

  ProductID (primary key)

  Name

  suggested retail price / SRP

  Discount

- Asumsikan bahwa Discount bergantung pada harga eceran yang disarankan/ suggested retail price (SRP). Tabel ini melanggar bentuk normal ketiga karena kolom non-kunci, Discount, bergantung pada kolom lain non-kunci, SRP. Kolom independence berarti bahwa Anda dapat mengubah setiap kolom non-kunci tanpa mempengaruhi setiap kolom lainnya. Jika Anda mengubah nilai dalam kolom SRP, Diskon akan berubah sesuai perubahan pada kolom SRP, sehingga melanggar peraturan itu. Dalam hal ini Diskon harus dipindahkan ke table lain yang mempunyai kunci pada SRP

CONTOH PEMBUATAN DESAIN DATABASE

Pada proses perancangan database dapat dimulai dari dokumen dasar yang dipakai dalam sistem sesuai dengan lingkup sistem yang akan dibuat rancangan databasenya. Berikut ini adalah contoh dokumen mengenai faktur pembelian barang pada PT. Revanda Jaya.

FAKTUR PEMBELIAN BARANG

PT REVANDA JAYA
Jl. Bekasi Timur No. 2
Bekasi Timur

Kode Supplier : G01
Nama Supplier : Gobel Nustra

Tanggal : 07/02/2001
Nomor : 998

Kode	Nama Barang	Qty	Harga	Jumlah
A01	AC Split ½ PK	10	1.350.000	13.500.000
A02	AC Split 1 PK	10	2.000.000	20.000.000
			Total Faktur	33.500.000

Jatuh Tempo Faktur : 09/03/2001

FAKTUR PEMBELIAN BARANG

PT REVANDA JAYA
Jl. Bekasi Timur No. 2
Bekasi Timur

Kode Supplier : S02
Nama Supplier : Hitachi

Tanggal : 02/02/2001
Nomor : 779

Kode	Nama Barang	Qty	Harga	Jumlah
R01	Rice Chocker C3	10	150.000	1.500.000
			Total Faktur	1.500.000

Jatuh Tempo Faktur : 09/03/2001

1. Bentuk Unnormalisasi

- Langkah pertama dalam melakukan normalisasi data adalah dengan membentuk contoh data tersebut diatas dengan membentuk unnormalisasi data

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	Tanggal	Jatuh Tempo	Qty	Harga	Jumlah	Total
779	S02	Hitachi	R02	Rice Chocker C3	02/02/01	09/03/01	10	150000	1500000	1500000
998	G01	Gobel	A01	AC Split ½ PK	07/02/01	09/03.01	10	135000	13500000	33500000
		Nustra	A02	AC Split 1 PK			10	2000000	20000000	

2. Bentuk Normal Pertama (1 NF)

- Bentuklah menjadi bentuk normal pertama dengan memisah-misahkan data pada atribut-atribut yang tepat dan bernilai atomik, juga seluruh *record / baris harus lengkap adanya.*
- *Bentuk* relasi adalah *flat file. Dengan normal pertama kita dapat membuat satu tabel yang terdiri dari 11* Atribut yaitu (No_Faktur, Kode_Supplier, Nama_Supplier, Kode_Barang, Nama_Barang, Tanggal, Jatuh_Tempo, Qty, Harga, Jumlah, Total).

2. Bentuk Normal Pertama (1 NF)

Pada normal pertama tersebut masih terjadi banyak kelemahan, terutama pada proses **ANOMALI *insert, update dan delete berikut ini:***

a). ***Inserting / Penyisipan***

Kita tidak dapat memasukkan kode dan nama supplier saja tanpa adanya transaksi pembelian, sehingga supplier baru bisa dimasukkan kalau ada transaksi pembelian.

b). ***Deleting / Penghapusan***

Bila satu *record / baris di atas dihapus, misal nomor faktur 779, maka berakibat pada* penghapusan data supplier S02 (Hitachi) padahal data tersebut masih diperlukan.

c). ***Updating / Pengubahan***

Kode dan nama supplier terlihat ditulis berkali-kali, bila nama supplier berubah, maka di setiap baris yang ada harus dirubah, bila tidak menjadi tidak konsisten.

Atribut jumlah seharusnya tidak perlu, karena setiap harga dikali kuantitas akan menghasilkan jumlah, sehingga hasilnya akan menjadi lebih konsisten

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	Tanggal	Jatuh Tempo	Qty	Harga	Jumlah	Total
779	S02	Hitachi	R02	Rice Chocker C3	02/02/01	09/03/01	10	150000	1500000	1500000
998	G01	Gobel Nustra	A01	AC Split ½ PK	07/02/01	09/03/01	10	135000	13500000	33500000
998	G01	Gobel Nustra	A02	AC Split 1 PK	07/02/01	09/03/01	10	2000000	20000000	33500000

3. Bentuk Normal Kedua (2 NF)

- Bentuk normal kedua dengan melakukan dekomposisi relasi diatas menjadi beberapa relasi dan mencari kunci primer dari tiap-tiap relasi tersebut dan atribut kunci haruslah unik.
- Melihat permasalahan faktur di atas, maka dapat diambil beberapa kunci kandidat :

(No_Faktur, Kode_Supplier, dan Kode_Barang). Kunci kandidat tersebut nantinya bisa menjadi kunci primer pada relasi hasil dekomposisi.

3. Bentuk Normal Kedua (2 NF)

Relasi Supplier

Kode_Supplier	Nama _Supplier
S02	Hitachi
G01	Gobel Nustra
G01	Gobel Nustra

Relasi Barang

Kode_Barang	Nama_Barang	Harga
R02	Rice Chocker C3	150000
A01	AC Split ½ PK	135000
A02	AC Split 1 PK	2000000

Relasi Faktur

No_Faktur	(Kode_Barang)	(Kode _Supplier)	Tanggal	Jatuh_tempo	Qty
779	R02	S02	02/02/01	09/03/01	10
998	A01	G01	07/02/01	09/03/01	10
998	A02	G01	07/02/01	09/03/01	10

3. Bentuk Normal Kedua (2 NF)

Pada bentuk normal kedua tersebut masih terjadi permasalahan yaitu pada relasi Faktur, yaitu :

1. Atribut Quantitas pada relasi Faktur, tidak tergantung pada kunci utama, atribut tersebut bergantung fungsi pada Kode Barang + no_faktur, hal ini dinamakan ketergatungan transitif dan haruslah dipilah menjadi dua relasi.
2. Masih terdapat pengulangan, yaitu setiap kali satu faktur yang terdiri dari 5 macam barang maka 5 kali juga dituliskan no_faktur, tanggal, dan jatuh_tempo. Hal ini harus dipisahkan bila terjadi penggandaan tulisan berulang-ulang.

4. Bentuk Normal Ketiga (3 NF)

- Bentuk normal ketiga mempunyai syarat, setiap relasi tidak mempunyai atribut yang bergantung transitif, harus bergantung penuh pada kunci utama dan harus memenuhi bentuk normal kedua (2 NF).
- Untuk emmenuhi bentuk normal ketiga (3 NF), maka pada relasi faktur harus didekomposisi (dipecah) lagi menjadi dua relasi yaitu relasi faktur dan relasi transaksi barang, sehingga hasilnya adalah sebagai berikut ini:

4. Bentuk Normal Ketiga (3 NF)

Relasi Supplier

Kode_Supplier	Nama Supplier
G01	Gobel Nustra
S02	Hitachi

Relasi Barang

Kode_Barang	Nama_Barang	Harga
R01	Rice Cooker CC3	150.000
A01	AC Split ½ PK	1.350.000
A02	AC Split 1 PK	2.000.000

Relasi Faktur

No_Faktur	Tanggal	Jatuh_Tempo	Kode_Supplier
779	02/02/2001	09/03/2001	S02
998	07/02/2001	09/03/2001	G01

Relasi Transaksi_Barang

No_Faktur	Kode_Barang	Qty
779	R01	10
998	A01	10
998	A02	10

HASIL NORMALISASI

Kamus Data dari masing-masing relasi tersebut diatas adalah sebagai berikut:

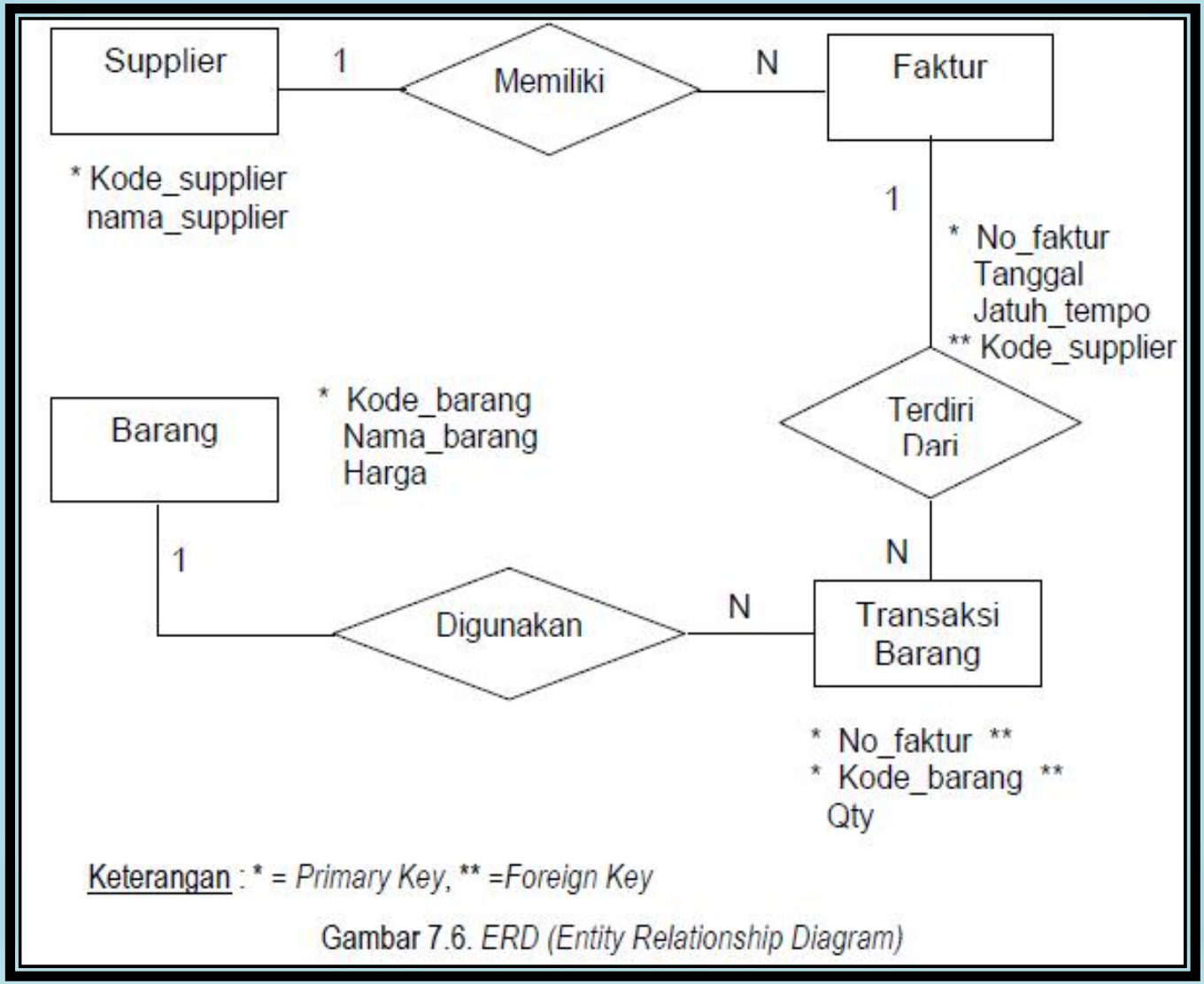
Supllier = { Kode_Supplier, Nama_Supplier }

Barang = { Kode_Barang, Nama_Barang, Harga }

Faktur ={ No Faktur, Kode_Supplier, Tanggal, Jatuh_Tempo }

Transaksi_Barang = { No Faktur, Kode Barang,Qty }

ERD (Entity Relationship Diagram)



Gambar 7.6. *ERD (Entity Relationship Diagram)*

ERD (Entity Relationship Diagram)

Pengertian Hubungan (*Relation) antar pada gambar ERD (entity relationship diagram)* pada gambar di atas adalah sebagai berikut:

a). Supplier ke Faktur relasinya adalah *one to many, artinya adalah satu supplier mempunyai* banyak faktur, faktur punya relasi terhadap supplier.

b). Faktur ke Transaksi Barang relasinya adalah *one to many, artinya adalah satu faktur* mempunyai beberapa transaksi barang (satu faktur terdiri dari satu atau lebih transaksi barang).

c). Barang ke Transaksi Barang relasinya adalah *one to many, artinya adalah satu barang bisa* terjadi beberapa kali transaksi pembelian barang.

Implementasi ERD dalam PLDatabase

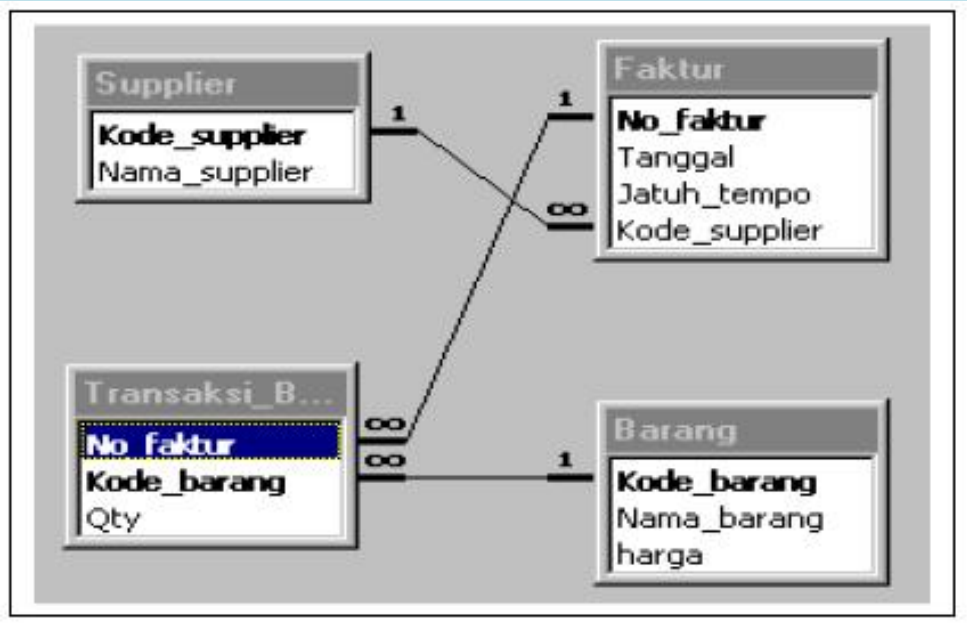
Implementasi ERD (*entity relationship diagram) physical pada contoh diatas, bisa dituangkan* ke dalam database MS-Access aatau SQL Server, seperti terlihat pada gambar beikut ini:

Kode_supplier	Nama_supplier
G01	Gobel Nustra
S02	Hitachi

No_faktur	Tanggal	Jatuh_tempo	Kode_supplier
779	02/02/2001	09/03/2001	S02
998	07/02/2001	09/03/2001	G01

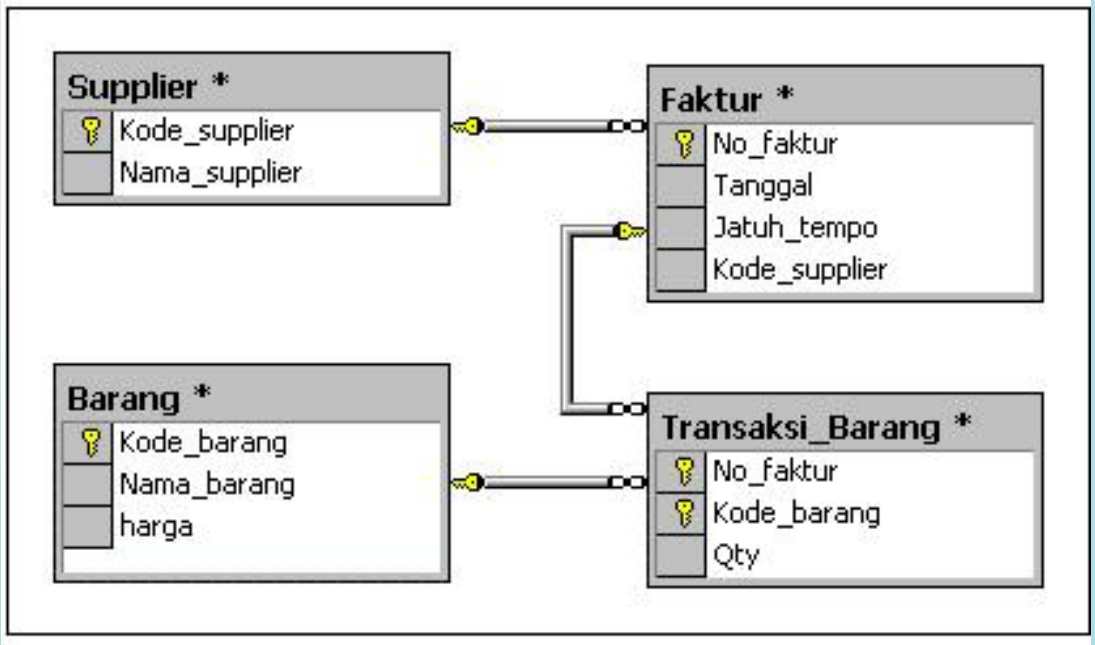
Kode_barang	Nama_barang	harga
A01	AC Split 1/2 PK	1350000
A02	AC Split 1 PK	2000000
R01	Rice Cooker CC3	150000

No_faktur	Kode_barang	Qty
779	R01	10
998	A01	10
998	A02	10



MS-Access 2000

SQL Server 2000



TUGAS

1. Carilah dokumen dasar seperti dalam contoh faktur di atas, setelah saudara temukan, kemudian lakukan langkah – langkah normalisasi data.
2. Setelah model ERD terbentuk, buatlah ERD *physical tersebut dengan menggunakan database MS-Access atau SQL Server.*