BAB 7

PENERAPAN BENTUK NORMALISASI

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 — Tanggal : 07/02/2001
Nama Supplier : Gobel Nustra — 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 — Tanggal : 02/02/2001
Nama Supplier : Hitachi — 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

Gambar 7.1. Faktur pembelian barang

Sehubungan dengan dokumen dasar tersebut, tahapan yang harus dilakukan untuk melakukan normalisasi data adalah sebagai berikut:

1. Bentuk Unnormalisasi

Langkah pertama dalam melakukan normalisasi data adalah dengan membentuk contoh data tersebut didtas dengan membentuk unnormalisasi data, dengan cara mencantumkan semua atribut data yang ada apa adanya seperti terlihat berikut ini :

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	

Gambar 7.2. Relasi faktur unnormalisasi

Pada relasi diatas adalah dengan menuliskan semua data yang ada yang akan direkam, data yang *double* tidak perlu ditulis. Terlihat baris / *record* yang tidak lengkap. Sulit dibayangkan bagaimana bentuk baris yang harus dibentuk untuk merekam data itu.

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).

Sehingga hasil daripada pembentukan normal pertama (1 NF) adalah sebagai berikut ini :

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

Gambar 7.3. Relasi memenuhi 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 (merupakan atribut turunan) seharusnya tidak perlu, karena setiap harga dikali kuantitas akan menghasilkan jumlah, sehingga hasilnya akan menjadi lebih konsisten.

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.

Dengan melihat normal pertama, kita dapat mendekomposisi menjadi tiga relasi berserta kunci primer yang ada yaitu : relasi Supplier (***Kode_Supplier***), relasi Barang (***Kode_Barang***), dan

Relasi Faktur (***No_Faktur***). Dengan melihat ketergantungan fungsional atribut-atribut lain terhadap atribut kunci, maka didapatkan 3 (tiga) relasi sebagai berikut :

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

Gambar 7.4. Relasi memenuhi 2 NF

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 Barang Kode_Supplier, Tanggal, Jatuh_Tempo, Qty }

Dengan pemecahan relasi di atas, maka untuk pengujian bentuk normal kesatu (1 NF) yaitu *insert, update*, dan delete akan terjawab. Kode dan nama supplier baru dapat masuk kapanpun tanpa adanya transaksi pada tabel faktur. Demikian pula untuk proses update dan delete untuk tabel Supplier dan Barang.

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:

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

Gambar 7.5. Relasi memenuhi 3 NF

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

Supllier = { Kode_Supplier**, Nama_Supplier }**

Barang = { Kode_Barang**, Nama_Barang, Harga }**

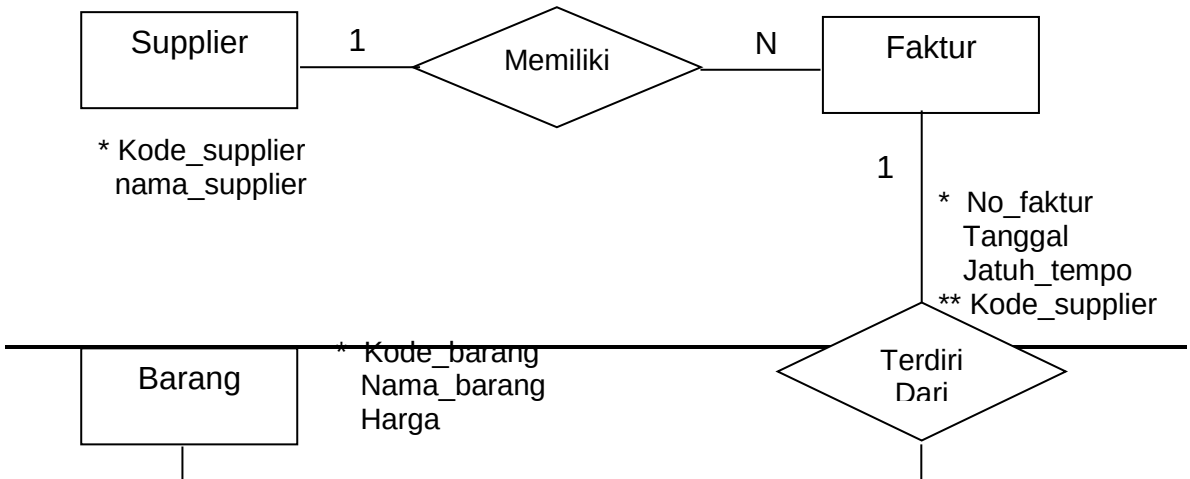
Faktur = { **No Faktur**, Kode_Supplier, Tanggal, Jatuh_Tempo }

Transaksi_Barang = { **No Faktur**, **Kode Barang** Qty }

Primary key pada relasi Supplier adalah Kode_Supplier, *Primary key* pada relasi Barang adalah Kode_Barang, *Primary key* pada relasi Faktur adalah No_Faktur dan Foreign key nya adalah Kode_Supplier, *Primary key* pada relasi Transaksi_Barang adalah No_Faktur, Kode_Barang dan keduanya juga menjadi *Foreign key*.

5. ERD (Entity Relationship Diagram)

Gambaran hubungan antar entitas / relasi yang terbentuk, adalah seperti terlihat pada gambar berikut ini :



Keterangan : * = *Primary Key*, ** =*Foreign Key*

Gambar 7.6. *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.

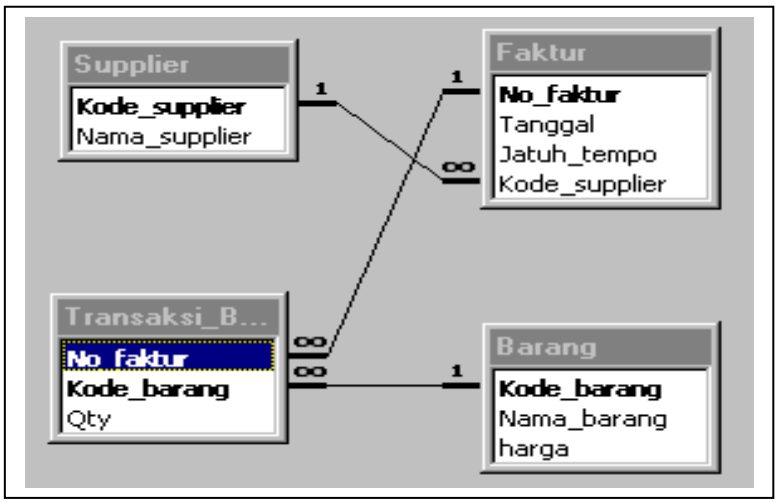
Implementasi ERD (*entity relationship diagram*) *physical* pada contoh diatas, bisa dituangkan kedalam database MS-Access aatau SQL Server, seperti terlihat pada gambar beikut ini:

Kode_supplier	Nama_supplier
G01	Gobel Nustra
S02	Hitachi

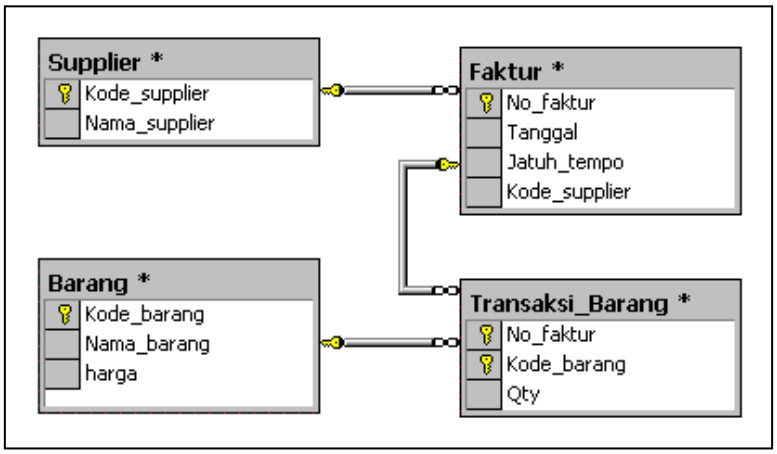
No_faktur	Tanggal	Jatuh_tempo	Kode_supplier
779	02/02/2001	09/03/2001	S02
998	07/02/2001	09/03/2001	G01

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



Gambar 7.7. ERD dengan *database* MS-Access 2000



Gambar 7.8. ERD dengan *database* SQL Server 2000

Pertanyaan Soal

1. Carilah dokumen dasar seperti dalam contoh faktur diatas, 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 ?.