

Penerapan Bentuk Normalisasi

Proses penerapan normalisasi dalam basis data dapat dimulai dari dokumen dasar yang dipakai dalam sistem sesungguhnya. Kadang-kadang basis data dibentuk dari sistem nyata yang mempunyai bentuk masih belum menggambarkan entitas-entitas secara baik. Sebagai contoh basis data yang dibangun dari daftar faktur pembelian sebagai berikut :

PT. Denmas Makmur		FAKTUR PEMBELIAN		
Jl. Sekeloa Utara No 62/152 C				
Bandung				
Kode Pemasok	: A101	Tanggal	: 07/03/2004	
Nama Pemasok	: Akbar Comp.	Nomor	: 111	
Kode	Nama Barang	Jumlah	Harga	Total
A1	DD RAM 128	10	200.000	2.000.000
A2	GForce-FX 5200	10	500.000	5.000.000
A3	Athlon 2500 +	10	700.000	7.000.000
Total faktur				14.000.000
Jatuh Tempo Faktur : 07/04/2004				

Langkah Pertama

Bentuklah menjadi tabel Un-Normalized, dengan mencantumkan semua field data yang ada

NoFaktur	Tgl	KdPemasok	NmPemasok	KdBrg	NmBarang	Qty	Hrg	Total	TotFak	Tempo
111	07/03/04	A101	Akbar Comp	A1	DDRAM128	10	200	2000	14000	07/04/04
				A2	GForce5200	10	500	5000		
				A3	Athlon2500	10	700	7000		
222	10/03/04	B102	Bona Comp	A2	GForce5200	5	500	2500	3700	10/04/04
				A5	DDRAM512	3	400	1200		

Menuliskan semua data yang akan direkam, bagian yang double tidak perlu dituliskan. Terlihat record-record yang tidak lengkap, sulit untuk membayangkan bagaimana bentuk record yang harus dibentuk untuk merekam data tersebut.

Langkah Kedua

Ubahlah menjadi bentuk Normal Kesatu dengan memisahkan data pada field-field yang tepat dan bernilai atomik, juga seluruh record harus lengkap datanya. Bentuk file masih flat.

Dengan bentuk Normal Kesatu ini telah dapat dibuat satu file dengan 11 field yaitu No faktur, Kode Pemasok, Nama Pemasok, Kode Barang, Nama Barang, Tanggal, Jatuh Tempo, Quantity, Harga, Jumlah, Total.

NoFaktur	Tgl	KdPemasok	NmPemasok	KdBrg	NmBarang	Qty	Hrg	Total	TotFak	Tempo
111	07/03/04	A101	Akbar Comp	A1	DDRAM128	10	200	2000	14000	07/04/04
111	07/03/04	A101	Akbar Comp	A2	GForce5200	10	500	5000	14000	07/04/04
111	07/03/04	A101	Akbar Comp	A3	Athlon2500	10	700	7000	14000	07/04/04
222	10/03/04	B102	Bona Comp	A2	GForce5200	5	500	2500	3700	10/04/04
222	10/03/04	B102	Bona Comp	A5	DDRAM512	3	400	1200	3700	10/04/04

Namun bentuk Normal Kesatu ini mempunyai banyak kelemahan diantaranya yaitu :

- Penyisipan data
Kode Pemasok dan Nama Pemasok tidak bisa ditambahkan tanpa adanya transaksi pembelian.
- Penghapusan data
Jika salah satu record dihapus maka semua data yang ada di situ akan terhapus juga.
- Pengubahan data
Data pemasok ditulis berkali-kali (Kode dan Nama). Jika suatu saat terjadi perubahan Nama pemasok maka harus mengganti semua record yang ada data pemasoknya. Bila tidak maka akan terjadi inkonsistensi.
- Redundansi
Field jumlah merupakan redundansi karena setiap harga dikalikan kuantitas hasilnya adalah jumlah, sehingga field ini dapat dibuang. Bila tidak maka dapat mengakibatkan inkonsistensi jika terjadi perubahan harga.

Langkah Ketiga

Pembentukan Normal Kedua dengan mencari field kunci yang dapat dipakai sebagai patokan dalam pencarian dan yang mempunyai sifat yang unik. Melihat kondisi dari permasalahan faktur di atas dapat diambil kunci kandidat sbb :

- No faktur
- Kode Pemasok
- Kode Barang

Buatlah tiga tabel dengan kunci tersebut, lihatlah kebergantungan fungsional field lain terhadap kunci, maka didapatkan tabel sbb :

Tabel Pemasok

KdPemasok	NmPemasok
A101	Akbar Comp
B102	Bona Comp

Tabel Barang

KdBrg	NmBarang	Hrg
A1	DDRAM128	200
A2	GForce5200	500
A3	Athlon2500	700
A2	GForce5200	500
A5	DDRAM512	400

Tabel Faktur

NoFaktur	Tgl	KdPemasok	KdBrg	Qty	Total	TotFak	Tempo
111	07/03/04	A101	A1	10	2000	14000	07/04/04
111	07/03/04	A101	A2	10	5000	14000	07/04/04
111	07/03/04	A101	A3	10	7000	14000	07/04/04
222	10/03/04	B102	A2	5	2500	3700	10/04/04
222	10/03/04	B102	A5	3	1200	3700	10/04/04

Dengan pemecahan seperti di atas maka sebagian dari pertanyaan pengujian pada bentuk normal kesatu yaitu masalah penyisipan, penghapusan dan perubahan dapat dijawab. Data pemasok dapat ditambahkan kapan saja tanpa harus ada transaksi pembelian. Namun permasalahan masih ada yaitu pada tabel faktur :

- Field Kuantitas dan Harga tidak bergantung penuh pada kunci primer nomor faktur, ia juga bergantung fungsi pada kode barang. Hal ini disebut sebagai kebergantungan yang transitif dan harus dipisahkan dari tabel.
- Redundansi masih terjadi, yaitu setiap kali satu faktur yang terdiri dari 5 macam barang yang dibeli maka 5 kali pula faktur dituliskan ke nomor faktur, tanggal faktur, tempo dan total. Ini harus dipisahkan bila terjadi penggandaan tulisan yang berulang-ulang.

Langkah Ke 4

Bentuk normal ketiga mempunyai syarat setiap tabel tidak mempunyai field yang bergantung transitif, harus bergantung penuh pada kunci utama. Maka terbentuklah tabel sbb :

Tabel Pemasok

KdPemasok	NmPemasok
A101	Akbar Comp
B102	Bona Comp

Tabel Barang

KdBrg	NmBarang	Hrg
A1	DDRAM128	200
A2	GForce5200	500
A3	Athlon2500	700
A2	GForce5200	500
A5	DDRAM512	400

Tabel Faktur

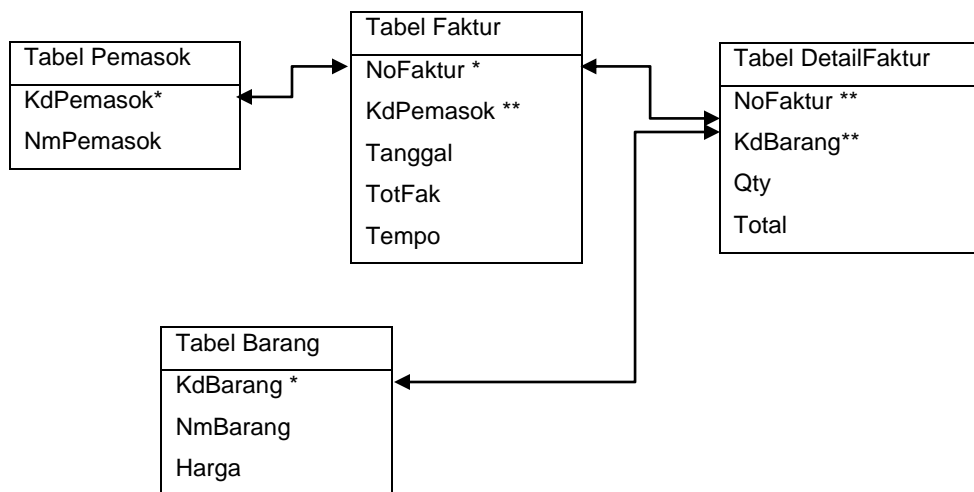
NoFaktur	Tgl	KdPemasok	TotFak	Tempo
111	07/03/04	A101	14000	07/04/04
222	10/03/04	B102	3700	10/04/04

Tabel Detail Faktur

NoFaktur	KdBrg	Qty	Total
111	A1	10	2000
111	A2	10	5000
111	A3	10	7000
222	A2	5	2500
222	A5	3	1200

Langkah Ke-5

Pengujian di sini untuk memastikan kebenaran isi tabel dan hubungan antara tabel tersebut. Ujian bahwa setiap tabel haruslah punya hubungan dengan tabel yang lainnya. Bila tidak ada penghubungan antar tabel maka dapat dikatakan pemodelan untuk membuat satu basis data adalah gagal. Adapun gambaran hubungan relasi antar file yang ada sbb :



Pengertian relasi di atas adalah

- Satu pemasok punya banyak faktur
- Satu faktur punya relasi dengan satu pemasok
- Satu faktur punya banyak transaksi barang
- Satu barang terjadi beberapa kali transaksi pembelian barang.

Langkah Ke-7

Permasalahan di atas hanya terbatas pada satu dokumen Faktur pembelian barang, padahal pada kenyataannya tentu faktur tersebut mempunyai dokumen pelengkap misalnya nota penjualan barang, laporan stok barang, laporan penjualan, laporan pembelian dan masih banyak lagi laporan dan dokumen data entry lainnya. Dengan langkah-langkah seperti di atas maka diperoleh field-field untuk melengkapi tabel-tabel yang ada dalam satu basis data. Misalnya tabel barang dengan bertambahnya field yang lain menjadi

Tabel Barang

Kode Barang
Nama Barang
Harga Beli
Harga Jual
Stok Akhir
Stok Awal Bulan