

Teknik Perancangan Basis Data

A. Pendahuluan

Perancangan merupakan suatu hal yang sangat penting dalam pembuatan basis data. Permasalahan yang dihadapi pada waktu perancangan yaitu bagaimana basis data yang akan dibangun ini dapat memenuhi kebutuhan saat ini dan masa yang akan datang. Untuk itu diperlukan perancangan basis data baik secara fisik maupun secara konseptualnya. Perancangan konseptual akan menunjukkan entity dan relasinya berdasarkan proses yang diinginkan oleh organisasinya. Untuk menentukan entity dan relasinya perlu dilakukan analisis data tentang informasi yang ada dalam spesifikasi di masa yang akan datang.

Model Konseptual basis data

Perancangan model konseptual basis data dalam sebuah organisasi menjadi tugas dari Administrator basis data. Model konseptual merupakan kombinasi beberapa cara untuk memproses data untuk beberapa aplikasi. Model konseptual tidak tergantung pada aplikasi individual, DBMS digunakan, Hardware komputer dan model fisiknya.

Pada perancangan model konseptual basis data ini penekanan dilakukan pada struktur data dan relasi antara file. Pada perancangan model konseptual ini dapat dilakukan dengan menggunakan model data relasional.

Teknik Model Data Relasional ada 2 yaitu :

- Teknik Normalisasi
- Teknik Entity Relationship

Teknik Normalisasi

Proses normalisasi adalah proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entity dan relasinya. Pada proses normalisasi dilakukan pengujian pada beberapa kondisi apakah ada kesulitan pada saat

menambah/menyisipkan, menghapus, mengubah dan mengakses pada suatu basis data. Bila terdapat kesulitan pada pengujian tersebut maka perlu dipecahkan relasi pada beberapa tabel lagi atau dengan kata lain perancangan basis data belum optimal.

Field (Atribut) Kunci

setiap file selalu terdapat kunci dari file berupa field atau satu set field yang dapat mewakili record. Misalnya Nomor Pokok Mahasiswa (NPM) merupakan kunci dari tabel mahasiswa suatu Perguruan Tinggi, setiap pencarian cukup dengan menyebut nomor mahasiswa tersebut maka dapat diketahui identitas mahasiswa lainnya seperti nama, alamat dan atribut lainnya.

Nomor Pegawai (NIP) bagi data dosen, NIK untuk data karyawan, Kode_Kuliah untuk data Mata kuliah, dan lain sebagainya.

Jenis Atribut Pada Entitas

Atribut yang melekat pada suatu entitas ada bermacam tipe seperti yang akan dijelaskan sebagai berikut :

Atribut Sederhana : atribut sederhana merupakan atribut atomik yang tidak dapat lagi dipecah menjadi atribut lain. Contoh

Entitas mahasiswa mempunyai atribut sederhana berupa NIM, Nama Mahasiswa .

Atribut Komposit : atribut komposit merupakan atribut yang masih dapat dipecah menjadi sub-sub atribut yang masing-masing memiliki arti tersendiri.

Contoh : entitas mahasiswa mempunyai atribut alamat. Alamat disini dapat dipecah menjadi sub atribut seperti nama_kota, kode_pos.

Atribut Bernilai Tunggal: yaitu atribut yang hanya memiliki satu nilai untuk setiap barisnya.

Contoh : entitas mahasiswa mempunyai atribut NPM, Nama, Alamat isi data dari atribut ini hanya boleh diisi dengan 1 data. Setiap mahasiswa hanya memiliki 1 NPM, 1 Nama, 1 Alamat.

Atribut Bernilai Jamak: yaitu atribut yang boleh memiliki lebih dari satu nilai untuk setiap barisnya.

Contoh : entitas mahasiswa mempunyai atribut Hobby isi data dari atribut ini boleh lebih dari 1 data. Mahasiswa Roshita memiliki NPM 13402021 beralamat di Jalan Garuda 32 Yogyakarta memiliki Hobby (Olah Raga, Nyanyi, Masak dan Nonton TV)

Atribut Harus Bernilai: yaitu atribut yang harus memiliki nilai data untuk setiap barisnya. Biasanya atribut seperti ini sudah ditetapkan dalam perancangan tabelnya sehingga jika dalam pengisian dokosongi akan terjadi kesalahan.

Contoh : entitas mahasiswa mempunyai atribut NPM dan Nama_Mahasiswa yang harus diisi datanya, sebab jika tidak diisi akan terjadi kekacauan dalam basis data.

Atribut Bernilai Null: yaitu atribut yang boleh tidak memiliki nilai data untuk setiap barisnya.

Contoh : entitas mahasiswa mempunyai atribut Alamat, Hobby, Nama_Pacar yang boleh untuk tidak diisi tetapi kalau diisi akan lebih baik,

Atribut Turunan: yaitu atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut lain yang berkaitan.

Contoh : entitas mahasiswa mempunyai atribut IPK yang diperoleh dari pengolahan atribut Nilai pada tabel (entitas Nilai) dengan kode NIM mahasiswa yang sama dan diproses sehingga menghasilkan IPK untuk mahasiswa yang bersangkutan..

Kunci Kandidat (Candidate Key)

Kunci kandidat adalah satu atribut atau satu set atribut yang mengidentifikasi secara unik suatu kejadian spesifik dari entity. Satu set atribut menyatakan secara tidak langsung dimana anda tidak dapat membuang beberapa atribut dalam set tanpa merusak kepemilikan yang unik. Jika kunci kandidat berisi lebih dari satu atribut, maka biasanya disebut sebagai composite key (kunci campuran atau gabungan).

Contoh :

File mahasiswa berisi :

- Nomor Pegawai

- No KTP
- Nama Pegawai
- Tempat Lahir
- Tanggal Lahir
- Alamat
- Kota

Kunci kandidat dalam file mahasiswa di atas dapat dipilih sbb :

- Nomor Pegawai
- No KTP
- Nama (tidak dapat dipakai karena sering seseorang punya nama yang sama dengan orang lain)
- Nama + Tanggal Lahir (mungkin bisa dipakai sebagai kunci karena kemungkinan orang dengan nama yang sama dan tanggal lahir yang sama cukup kecil)
- Nama + Tempat Lahir + Tanggal Lahir (dapat dipakai sebagai kunci)
- Alamat dan Kota (bukan kunci)

Kunci Primer (Primary Key)

Primary key adalah satu atribut atau satu set minimal atribut yang tidak hanya mengidentifikasi secara unik suatu kejadian spesifik, tetapi juga dapat mewakili setiap kejadian dari suatu entity.

Setiap kunci kandidat dapat menjadi kunci primer tetapi sebaliknya sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap entity yang ada.

Contoh :

- No Pegawai (karena sifatnya yang unik maka tidak mungkin pegawai mempunyai Nomor Pegawai yang sama).
- No KTP (Bisa dipakai misalnya untuk pegawai yang baru belum mendapatkan nomor pegawai maka bisa digunakan nomor KTP untuk sementara sebagai kunci primer).
- Kode_Kuliah (bisa dipakai untuk data mata kuliah karena kode mata kuliah bersifat unik untuk tiap mata kuliah)

Kunci Alternatif (Alternate Key)

Kunci alternatif adalah kunci kandidat yang tidak dipakai sebagai kunci primer. Kunci alternatif ini sering digunakan untuk kunci pengurutan misalnya dalam laporan.

Kunci Tamu (Foreign Key)

Kunci tamu adalah satu atribut atau satu set minimal atribut yang melengkapi satu hubungan yang menunjukkan ke induknya. Kunci tamu ditempatkan pada entity anak dan sama dengan kunci primer induk yang direlasikan. Hubungan antara entity induk dengan anak adalah hubungan satu lawan banyak (one to many relationship)

Contoh :

File Transaksi Gaji Bulanan

- No Pegawai
- No Bukti
- Tanggal
- Jumlah Gaji Kotor
- Jumlah Potongan
- Jumlah Gaji Bersih
- Jumlah Pajak

Kunci Tamu

- No Pegawai (karena Gaji berhubungan dengan file Pegawai)

Kunci Primer

- No Bukti (karena unik dan mewakili entity)

Kunci Kandidat

- No Pegawai + Nomor Bukti (Unik dan menunjukkan hubungan dengan file Pegawai)

Dalam hubungan dua buah file yang punya relationship banyak lawan banyak maka terdapat 2 kunci tamu pada file konektornya.

Contoh :

File Proyek berisi atribut

- No Proyek

- Tgl Mulai
- Tgl Selesai
- Anggaran

File Pegawai Berisi Atribut

- No Pegawai
- Nama

Hubungan antara file tersebut adalah banyak lawan banyak yaitu satu pegawai mengerjakan lebih dari 1 proyek dan 1 proyek dikerjakan oleh beberapa pegawai maka untuk menunjukkan hubungan tersebut dipakai file konektor yang berisi kunci tamu dari kedua file.

File Proyek Pegawai berisi atribut :

- No Proyek
- No Pegawai
- Jam Kerja

Maka pada file proyek pegawai terdapat kunci tamu yaitu nomor proyek dan no pegawai. Kedua atribut tersebut juga merupakan kunci primer.

Kebergantungan Fungsi

Kebergantungan Fungsi didefinisikan sebagai

Diberikan sebuah relasi R, atribut Y dan R adalah bergantung fungsi pada atribut X dari R jika dan hanya jika setiap nilai X dalam R punya hubungan dengan tepat satu nilai Y dalam R (dalam setiap satu waktu).

File relasi pegawai atribut berisi :

- No Pegawai
- No KTP
- Nama
- Tempat Lahir
- Tgl Lahir
- Alamat
- Kota

Isi dari atribut nama bergantung pada No Pegawai. Jadi dapat dikatakan bahwa atribut nama bergantung secara fungsi pada No Pegawai dan Nomor Pegawai menunjukkan secara fungsi nama. jika anda mengetahui no pegawai maka anda dapat menentukan nama pegawai tersebut. Notasi untuk kebergantungan fungsi ini adalah

No Pegawai → Nama atau

Nama = f(No Pegawai)

Bentuk-Bentuk Normalisasi

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatangannya.

Bentuk Normal Kesatu (1 NF / First Normal Form)

Bentuk Bentuk Normal Kesatu mempunyai ciri yaitu setiap data dibentuk dalam file flat, data dibentuk dalam satu record demi satu record dan nilai dari field berupa "atomic value". Tidak ada set atribut yang berulang ulang atau atribut bernilai ganda (multi value). Tiap field hanya satu pengertian, bukan merupakan kumpulan data yang mempunyai arti mendua. Hanya satu arti saja dan juga bukanlah pecahan kata kata sehingga artinya lain.

Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

Contoh :

Kelas (Kode Kelas, Nama Kelas, Pengajar)

Ini merupakan bentuk 1NF karena tidak ada yang berganda dan tiap atribut satu pengertian yang tunggal

Contoh Data

<i>Kode Kelas</i>	<i>Nama Kelas</i>	<i>Pengajar</i>
1111	Basis Data	Muhamad Ali
2222	Riset Pemasaran	Ahmad Yunani
3333	Pemrograman	Suryo Pratolo

Mahasiswa (NPM, Nama, Dosen Wali, Semester1, Semester2 Semester3)

Mahasiswa yang punya NPM, Nama, Dosen Wali mengikuti 3 mata kuliah. Di sini ada perulangan semester sebanyak 3 kali. Bentuk seperti ini bukanlah 1 NF

Contoh Data :

NPM	Nama	Dosen Wali	Sem 1	Sem 2	Sem 3
1000	Sally Fatimah	Dedy S	1234		3100
1001	Inul Daratista	Ruslan	1234	2109	
1002	Putri Patricia	Denmas		2100	3122

Bentuk 1 NF dari bentuk di atas adalah sbb :

NPM	Nama	Dosen Wali	Semester
1000	Sally Fatimah	Dedy S	1234
1000	Sally Fatimah	Dedy S	3100
1001	Inul Daratista	Ruslan	1234
1001	Inul Daratista	Ruslan	2109
1002	Putri Patricia	Denmas	2100
1002	Putri Patricia	Denmas	3122

Bentuk Normal Kedua (2 NF)

Bentuk Normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk Normal Kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, sehingga untuk membentuk Normal Kedua haruslah sudah ditentukan kunci-kunci field. Kunci field harus unik dan dapat mewakili atribut lain yang menjadi anggotanya.

Dari contoh relasi mahasiswa pada bentuk Normal Kesatu, terlihat bahwa kunci utama adalah NPM. Nama Mahasiswa dan Dosen Wali bergantung pada NPM, Tetapi Kode Semester bukanlah fungsi dari Mahasiswa maka file siswa dipecah menjadi 2 relasi yaitu :

Relasi Mahasiswa

NPM	Nama	Dosen Wali
1000	Sally Fatimah	Dedy S
1001	Inul Daratista	Ruslan
1002	Putri Patricia	Denmas

Dan

Relasi Ambil Kuliah

NPM	Kode Kuliah
1000	1234
1000	3100
1001	1234
1001	2109
1001	2100
1001	3122

Bentuk Normal Ketiga (3 NF)

Untuk menjadi bentuk Normal Ketiga maka relasi haruslah dalam bentuk Normal Kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Artinya setiap atribut bukan kunci harus bergantung hanya pada kunci primer secara menyeluruh.

Contoh pada bentuk Normal kedua di atas termasuk juga bentuk Normal Ketiga karena seluruh atribut yang ada di situ bergantung penuh pada kunci primernya.

Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk Normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk Normal Kesatu dan setiap atribut dipaksa bergantung pada fungsi pada atribut super key.

Pada contoh dibawah ini terdapat relasi Seminar, Kunci Primer adalah NPM + Seminar. Siswa boleh mengambil satu atau dua seminar. Setiap seminar membutuhkan 2 pembimbing dan setiap siswa dibimbing oleh salah satu diantara 2 pembimbing seminar tersebut. Setiap pembimbing hanya boleh mengambil satu seminar saja. pada contoh ini NPM dan Seminar menunjukkan seorang Pembimbing.

Relasi Seminar

NPM	Seminar	Pembimbing
1000	S100	Siska
1001	S100	Sinta
1002	S101	Sukma
1001	S101	Sukma
1003	S101	Akbar

Bentuk Relasi Seminar adalah bentuk Normal Ketiga, tetapi tidak BCNF karena Kode Seminar masih bergantung fungsi pada Pembimbing, jika setiap

Pembimbing dapat mengajar hanya satu seminar. Seminar bergantung pada satu atribut bukan super key seperti yang disyaratkan oleh BCNF. Maka relasi Seminar harus dipecah menjadi dua yaitu :

Relasi Pembimbing

Pembimbing	Seminar
Siska	S100
Sinta	S100
Sukma	S101
Akbar	S101

Relasi Seminar - Pembimbing

NPM	Pembimbing
1000	Siska
1001	Sinta
1002	Sukma
1001	Sukma
1003	Akbar

Penerapan Bentuk Normalisasi

Proses perancangan 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.

No. Fak	Kode Suppl	Nama Suppl	Kode Barang	Nama Barang	Tgl	Jatuh Tempo	Qty	Harga	Jumlah	Total
111	A101	Akbar Comp	DR128	DRAM	07/03/04	07/04/04	10	200	2000	2000
222	B111	Bona Comp	GF52	GForce	10/02/04	15/03/04	10	500	5000	5000
			Ath25	Ath 25			10	700	7000	7000

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 Suplier, Nama Suplier, Kode Barang, Nama Barang, Tanggal, Jatuh Tempo, Quntity, Harga, Jumlah, Total.

No. Fak	Kode Supp	Nama Suppl	Kode Barang	Nama Barang	Tgl	Jatuh Tempo	Qty	Harga	Jumlah	Total
111	A101	Akbar Comp	DR128	DRAM	07/03/04	07/04/04	10	200	2000	2000
222	B111	Bona Comp	GF52	GForce	10/02/04	15/03/04	10	500	5000	5000
222	B111	Bona Comp	Ath 25	Ath 25	10/02/04	15/03/04	10	700	700	700

Namun bentuk Normal Kesatu ini mempunyai banyak kelemahan diantaranya yaitu :

- Penyisipan data
Kode Suplier dan Nama Suplier 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 suplier ditulis berkali-kali (Kode dan Nama). Jika suatu saat terjadi perubahan Nama suplier maka harus mengganti semua record yang ada data supliernya. 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

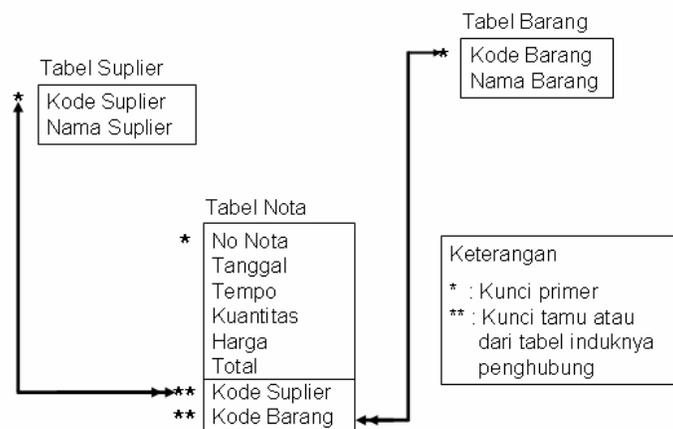
tidak maka dapat mengakibatkan inkonsistensi jika terjadi perubahan harga.

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 Suplier
- Kode Barang

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



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

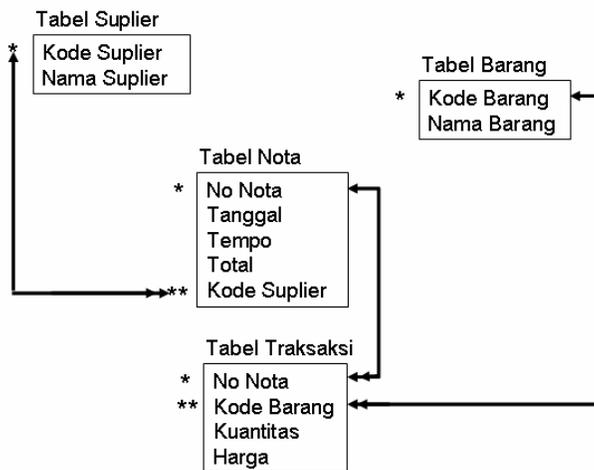
Namun permasalahan masih ada yaitu pada tabel nota

- Field Kuantitas dan Harga tidak bergantung penuh pada kunci primer nomor nota, ia juga bergantung fungsi pada kode barang. Hal ini disebut sebagai kebergantungan yang transitif dan harus dipisahkan dari tabel.

- M= redundansi masih terjadi, yaitu setiap kali satu nota yang terdiri dari 5 macam barang yang dibeli maka 5 kali pula nota dituliskan ke nomor nota, tanggal nota, tempo dan total. Ini harus dipisahkan bila terjadi pengandaan tulisan yang berulang-ulang.

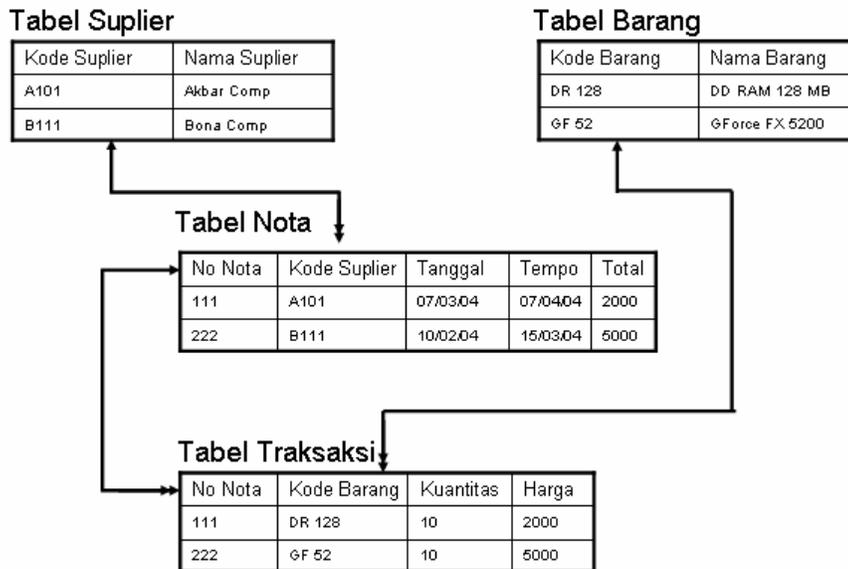
Langkah Ke 4

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



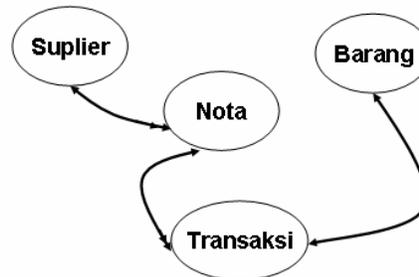
Langkah Ke-5

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 perancangan untuk membuat satu basis data adalah gagal.



Langkah Ke-6 Relasi Antar tabel

Gambarkan hubungan relasi antar file yang ada sbb :



Pengertian relasi di atas adalah

- Satu supplier punya banyak nota
- Nota punya relasi dengan supplier bukan sebaliknya supplier punya relasi terhadap nota.
- Satu nota 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 enty lainnya.

Dengan langkah-langkah perancangan 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