PENGELOLAAN PERSEDIAAN BANYAK SATUAN DENGAN RDBMS VISUAL FOXPRO

Katon Wijana

Abstrak

Persediaan barang dalam dunia bisnis, banyak yang mempunyai satuan yang dapat dikonversikan menjadi satuan-satuan yang lebih kecil atau lebih besar. Pada sebuah toko Alat Tulis Kantor (ATK) misalnya, pensil dibeli dari supplier dalam satuan Karton, sedangkan ketika dijual bisa dalam satuan Dus yang berisi 12 Batang atau bahkan dijual eceran per Batang. Bukan hanya pensil saja yang mempunyai satuan dapat dikonversikan, tetapi kertas, buku tulis, buku gambar, penggaris, dan sebagainya hampir tiap jenis barang mempunyai beberapa satuan barang.

Dalam Sistem Informasi Berbasis Komputer (SIBK), jenis barang dikenal melalui kode barang yang diberikan kepadanya, masalah duplikasi data akan timbul ketika basis data dirancang dengan cara membuat kode barang yang berbeda untuk tiap-tiap satuan barang dengan nama barang sama. Selain itu masalah tersebut, masalah yang sering timbul pada SIBK persediaan barang adalah tidak akuratnya data persediaan karena adanya kegagalan yang tidak terduga, misalnya pada saat pencatatan transaksi sudah berhasil tersimpan, tetapi pada saat baris program yang melakukan update stok komputer mengalami gangguan, maka akibatnya data mengenai persediaan barang tidak sesuai dengan data transaksi yang ada.

Dengan melakukan perancangan basis data yang cermat, masalah duplikasi data dapat direduksi, integritas data antar tabel dapat terjaga, serta manipulasi data menjadi tepat sasaran. Untuk mengurangi resiko terpotongnya suatu proses, DBMS menyediakan fitur untuk memanggil fungsi secara otomatis ketika terjadi manipulasi data pada suatu tabel, yaitu yang disebut Trigger. Dengan Trigger suatu perintah manipulasi data tertentu akan dapat lebih terjamin keberhasilannya.

Kata kunci: *Sistem Informasi, RDBMS, Trigger*

1. Pendahuluan

Mengelola persediaan barang yang beraneka ragam tidaklah mudah, apalagi tiap ragam mempunyai beberapa macam satuan. Dengan adanya banyak transaksi dalam dunia bisnis, pengelolaan persediaan harus menggunakan Teknologi Informasi. Teknologi Informasi yang banyak digunakan untuk mengelola data yang sangat banyak adalah RDBMS yang baru dapat berjalan dengan efektif dan efisien, jika basis data dirancang dengan baik. Selain itu, pemanfaatan fitur RDBMS yang tepat akan mendukung kualitas dari informasi yang dikelola.

Visual FoxPro (VFP) adalah *Full Featured Software Application Development Tool* yang sangat handal pada kelas RDBMS berbasis file (*file system*). Untuk pengelolaan informasi dalam sekala kecil menegah, RDBMS berbasis file masih merupakan pilihan yang tepat. Selain karena alasan harga perangkat lunak yang lebih ekonomis, RDBMS berbasis file juga tidak membutuhkan perangkat keras dengan spesifikasi tinggi, yang berarti juga dapat menghemat biaya. VFP selain mempunyai RDBMS, juga menyediakan *GUI Creator, Report Generator, View Designer, Menu Designer* dan fitur-fitur lain yang membuat *developer* merasa dapat melakukan apa-saja yang diinginkan.

Pada kesempatan ini, akan dieksplorasi fitur-fitur RDBMS VFP, yaitu *referential integrity* dan *trigger* yang dapat mendukung pembuatan SIBK persediaan barang banyak satuan yang cukup handal.

2. Permasalahan

Bagaimana membangun aplikasi sistem informasi yang efisien dan efektif serta mempunyai resiko kegagalan sekecil mungkin dalam mengelola persediaan dengan banyak satuan.

3. Dasar Teori

Ada banyak model yang dapat digunakan untuk merancang basis data, pada pembahasan ini akan digunakan ***Entity Relationship Diagram*** **(ER-D)**. Dimana ***Entity*** adalah sesuatu yang dapat dibedakan, mempunyai identitas, dalam dunia data biasanya diberi kunci primer. ***Entity*** bisa berupa orang, barang, pekerjaan, faktur, nota atau apa saja yang dapat dibuat daftar yang diberi identitas.

Entity Relationship Diagram (ERD) adalah suatu model konseptual yang dipakai untuk menggambarkan data dan relasinya sehingga informasi yang harus disimpan dalam suatu sistem informasi menjadi lebih jelas. ERD biasanya digunakan pada saat awal desain Sistem Informasi, berupa gambar *Entity* (4 persegi panjang) dan garis-garis yang merelasikan diantara keduanya.

a. Konsep Entity Relationship (Cardinality) (Silberschats, 2002, hal 42)

- **One to One Relationship**
  Hubungan antara *Entity* pertama dan *Entity* kedua adalah satu berbanding satu.

Gambar 1. One to One Relationship

- **One to Many atau Many to One Relationship**
  Hubungan antara *Entity* pertama dan *Entity* kedua adalah satu berbanding banyak atau banyak berbanding satu.

Gambar 2. One to Many Relationship

- **Many to Many Relationship**
  Hubungan *Entity* pertama dan *Entity* kedua adalah banyak berbanding banyak.

Gambar 3. Many to Many Relationship

b. Langkah-Langkah Perancangan ER

Sumber awal data teknik perencanaan *database* dengan ER adalah *data dictionary* (kumpulan data), dari data tersebut dilakukan :

- Memilih kelompok **atribut** yang sama untuk dijadikan sebuah **entitas** dan menentukan ***primary key*** dengan syarat unik dan mewakili entitas.
- Menggambarkan *Cardinality* dari ER diagram berdasarkan analisa relasi yang didapat. Relasi yang terjadi dapat *One to One, One to Many* dan *Many to Many Relationship*
- Membentuk SKEMA DATABASE atau LRS (*Logical Record Structure*) berdasarkan ER diagram
  - Bila relasi *One to One* maka *foreign key* diletakkan pada salah satu dari 2 entitas yang ada atau menyatukan ke dua entitas tersebut.
  - Bila relasi *One to Many* maka foreign key diletakkan di entitas yang *Many*
  - Bila relasi *Many to Many* maka dibuat "file konektor" yang berisi 2 *foreign key* yang berasal dari kedua entitas

c. Fungsi

Fungsi adalah sekumpulan program yang diberi nama. Dengan menulis program dalam sebuah fungsi, maka jika suatu proses yang ada pada fungsi tersebut dibutuhkan dapat dipanggil tanpa harus menulis ulang semua kode program yang ada pada fungsi tersebut. Selain itu, dengan menulis program dalam sebuah fungsi, program tersebut dapat dipanggil melalui *trigger* yang ada pada mesin database. Bentuk umum fungsi dalam VFP adalah :

```
Parameter <var1>,...
...statement_1
...statement_2
...
...statement_n
[return <value>]
```

4. Pembahasan

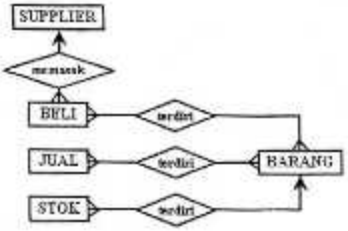
a. Langkah-Langkah Perancangan ER

- Memilih kelompok **atribut** yang sama untuk dijadikan sebuah **entitas** dan menentukan ***primary key*** dengan syarat unik dan mewakili entitas. Entitas dapat berupa Look up table, dapat juga berupa transaksi. Pada kasus ini bisa ditemukan :

  Look up table :
  - BARANG
  - SUPPLIER

  Transaction table :
  - BELI
  - JUAL

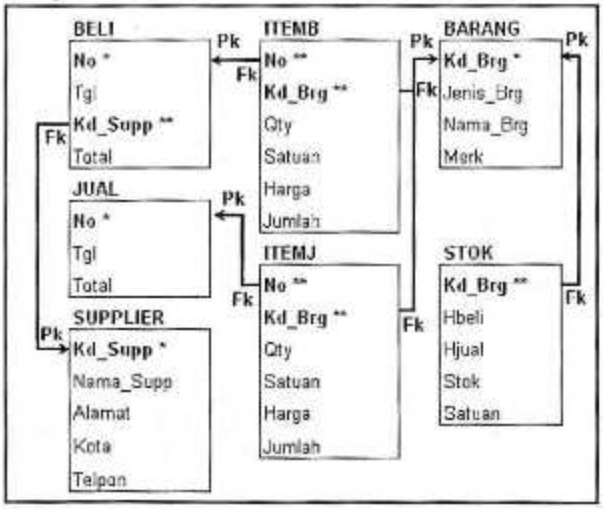
- Menggambarkan *Cardinality* dari ER diagram berdasarkan analisa relasi yang didapat. Relasi yang terjadi dapat *One to One, One to Many* dan *Many to Many Relationship*. Pada kasus ini Sebuah barang mempunyai beberapa satuan stok, sehingga ditemukan entity lemah (***weak entity***) yang diberi nama STOK, sehingga ER-Diagram diperoleh sebagai berikut:



Gambar 4. ER-Diagram Stok

- Membentuk SKEMA DATABASE atau LRS (*Logical Record Structure*) berdasarkan **ER-D**
  - Relasi One to One pada kasus ini tidak ada.
  - Relasi One to Many pada kasus ini terjadi pada relasi antara **Supplier** dan **Beli**, karena Entitas **Beli** yang Many, maka **Foreign Key** diletakkan pada entitas **Beli**. Selain itu juga pada relasi antara **Barang** dan **Stok**, karena entitas **Stok** yang many, maka Foreign Key diletakkan pada entitas **Stok**.
  - Untuk relasi many to many perlu dibuat "file konektor" yang berisi 2 foreign key yang berasal dari kedua entitas. Pada kasus ini antara entitas **Beli** dan **Barang** merupakan relasi many to many oleh karena itu akan dibuat sebuah file connector bernama **ItemB**. Demikian juga antara entitas **Jual** dan **Barang** merupakan relasi many to many oleh karena itu akan dibuat sebuah file connector bernama **Item.J**.

b. Skema Diagram



Gambar 5. Skema Diagram Stok

c. **Struktur tabel**

SUPPLIER

Field	Type	Width	Keterangan
Kd_Supp	Character	3	Primary Key
Nama_Supp	Character	40	
Alamat	Character	50	
Kota	Character	30	
Telpon	Character	30	

BARANG

Field	Type	Width	Keterangan
Kd_Brg	Character	5	Primary Key
Jenis_Brg	Character	40	
Nama_Brg	Character	50	
Merk	Character	30	

STOK

Field	Type	Width	Keterangan
Kd_Brg	Character	5	Regular Fk
Hbeli	Numeric	10	
Hjual	Numeric	10	
Stok	Numeric	10	
Satuan	Character	10	

Gambar 6. Struktur tabel SUPPLIER, BARANG dan STOK

BELI

Field	Type	Width	Keterangan
No	Character	10	Primary Key
Tgl	Date	8	
Kd_Supp	Character	3	Regular Fk
Total	Numeric	10	

ITEMB

Field	Type	Width	Keterangan
No	Character	10	Regular Fk
Kd_Brg	Character	5	Regular Fk
Qty	Numeric	5	
Harga	Numeric	10	
Satuan	Character	10	

Gambar 7. Struktur tabel BELI dan ITEMB

JUAL

Field	Type	Width	Keterangan
No	Character	10	Primary Key
Tgl	Date	8	
Total	Numeric	10	

ITEMJ

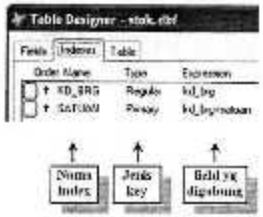
Field	Type	Width	Keterangan
No	Character	10	Regular Fk
Kd_Brg	Character	5	Regular Fk
Qty	Numeric	5	
Harga	Numeric	10	
Satuan	Character	10	

Gambar 8. Struktur tabel JUAL dan ITEMJ

d. Integritas Data

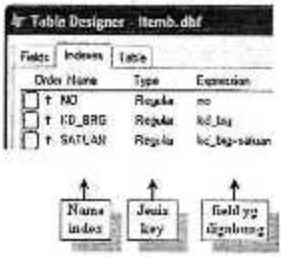
Pada kasus ini transaksi pembelian (perubahan data pada tabel ITEMB) akan melakukan update data Stok pada tabel STOK. Demikian juga transaksi penjualan (pada ITEMJ) juga akan melakukan update Stok. Masalah akan timbul karena yang digunakan sebagai kunci tamu pada tabel ITEMB adalah field **Kd_Brg** dan pada tabel STOK juga field **Kd_Brg**, maka keduanya sama-sama 'many' dan ini tidak mungkin bisa dilakukan penjagaan integritas data (Hoffer, 2007) dengan hanya kunci tersebut.

VFP mendukung pembuatan kunci utama maupun kunci tamu dengan menggunakan lebih dari sebuah field, yang dikenal sebagao ***Composite Key***. Maka untuk menjaga integritas antara data pada ITEMB dengan STOK dan ITEMJ dengan STOK perlu dibuat *composite key* pada tabel stok dengan cara menggabungkan field Kd_Brg dan Satuan menjadi sebuah kunci baru, pada contoh ini dinamakan **Satuan**.



Gambar 9. Desain tabel Stok.dbf

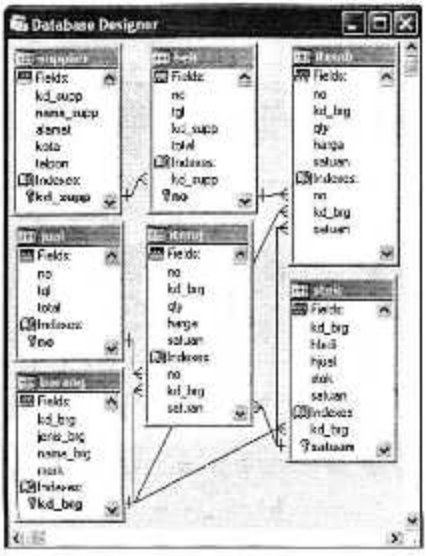
Demikian pula pada tabel ITEMB dan ITEMJ dapat dibuat *composite key* yang akan digunakan sebagai kunci tamu sebagai berikut :



Gambar 10. Desain tabel itemb.dbf

e. Mengatur Referential Integrity

Membuat kunci primer dan reguler pada VFP belum bisa dikatakan membuat integritas data dalam database, namun pembuatan key hanya tahap mempersiapkannya saja. Untuk membuat Referential Integrity pada VFP dapat dilakukan melalui **Database Designer**, yaitu dengan cara mendrag field ***Primary Key*** (bergambar kunci) ke Field ***Regular Key*** yang bersangkutan sebagai berikut :



Gambar 11. Desain Database

Setelah itu masuk ke menu Database → *Edit Referential Integrity* kemudian atur ***rule*** yang diinginkan, misalnya seperti di bawah ini :

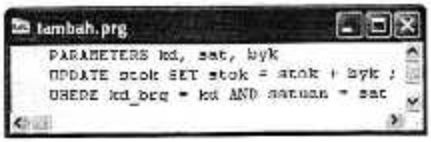
Tabel 1. Pengaturan Rule

Parent Tag	Child Tag	Update	Delete	Insert	Parent Tag	Child Tag
barang	itemb	Restrict	Restrict	Restrict	kd_brg	kd_brg
barang	itemj	Restrict	Restrict	Restrict	kd_brg	kd_brg
barang	stok	Cascade	Cascade	Restrict	kd_brg	kd_brg
beli	itemb	Restrict	Restrict	Restrict	no	no
jual	itemj	Restrict	Restrict	Restrict	no	no
stok	itemb	Cascade	Cascade	Restrict	satuan	satuan

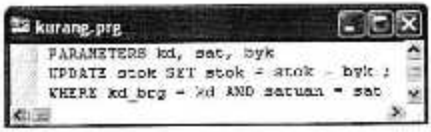
Dengan mengatur ***rule-rule*** ini maka integritas data pada RDBMS VFP sudah terjamin dengan baik.

f. Fungsi Manipulasi Stok

Untuk memanipulasi Stok bisa dibuat fungsi yang melakukan manipulasi tabel STOK pada field Stok, fungsi yang dibuat cukup sederhana, yaitu melakukan update berdasarkan kunci **Kd_Brg** dan **Satuan** yang berasal dari tabel ITEMB atau ITEMJ dan besarnya perubahan diperoleh dari field **Qty**. Untuk itu fungsi yang dibuat bernama Tambah() dan Kurang() masing-masing mempunyai 3 parameter sebagai berikut:



Gambar 12. Tambar.prg



Gambar 13. Kurang.prg

Pada kedua fungsi tersebut perbedaan hanya ada pada operator + dan -, sedangkan nilai-nilai yang nanti bakal diterima dari tabel ITEMB atau ITEMJ akan ditangani parameter **kd**, **sat** dan **byk**.

g. Trigger

VFP memang bukan mesin database besar, namun yang menarik adalah bahwa RDBMS VFP juga menyediakan fitur **Trigger** (Kramek, 2000, hal 230) seperti database server besar lainnya.

Seperti namanya, ***trigger*** adalah pemicu, dan pemicu ini dikaitkan dengan suatu tabel, yaitu jika terjadi manipulasi data (*insert / update/ delete*) pada suatu tabel dapat dilakukan pemanggilan fungsi tertentu.

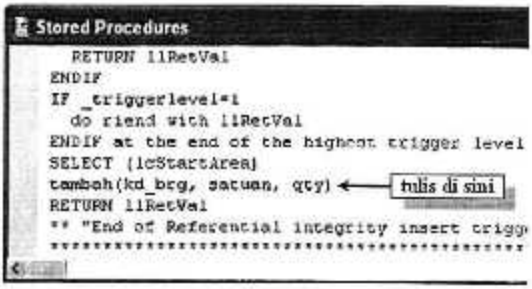
Trigger ini pulalah yang digunakan oleh VFP dalam melakukan ***Referential Integrity***, sehingga setiap ada manipulasi data akan selalu diperiksa keabsahannya berdasarkan rancangan yang dibuat oleh pemakai.

Gambar 14. Penggunaan trigger

Oleh karena pada tabel ITEMB, untuk trigger insert sudah digunakan oleh *Referential Integrity*, maka fungsi tambah tidak langsung ditulis pada Tabel Designer ini, melainkan disisipkan pada fungsi yang dipanggil tersebut.

Tampak pada gambar 14 bahwa pada tabel **ITEMB** pemicu (**trigger**) saat terjadi **insert** sudah diisi dengan fungsi `__ri_insert_itemb()`, yaitu fungsi yang dibangkitkan oleh **VFP** ketika dibuat *Referential Integrity*. Karena itu, karena diperlukan untuk memanggil fungsi

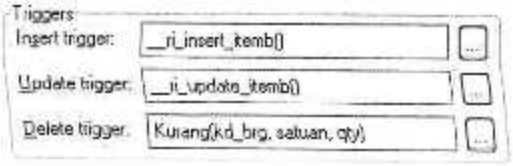
lain pada *trigger* yang sama, hal ini dapat dilakukan dengan memasukkan fungsi yang akan dipanggil (yaitu tambah.prg) ke dalam **_ri_insert_itemb()**. Untuk itu masuk ke dalam Stored Procedures dengan cara klik Store Procedures → Modify kemudian sisipkan program di akhir fungsi tersebut :



Gambar 15. Prosedur penyimpanan

Perlu diketahui bahwa kd_brg, satuan dan qty merupakan nilai field tabel ITEMB yang sedang ditambahkan dan nilai tersebut yang akan digunakan fungsi Tambah() untuk melakukan update stok.

Untuk menjalankan fungsi **kurang()** dapat dituliskan langsung pada pemicu yang ada pada tabel *designer* karena pada tabel ITEMB pemicu pada kejadian **delete** belum digunakan. Berikut ini adalah cara menuliskan fungsi kurang pada tabel designer:



Gambar 16. Penulisan fungsi kurang

Dengan demikian, setiap ada penambahan (insert) maupun penghapusan (delete), stok barang akan selalu menyesuaikan. Selain pada tabel ITEMB, trigger ini juga harus dipasang pada tabel ITEMJ. Sehingga kegagalan perubahan stok berdasarkan transaksi bisa diminimalkan.

5. Kesimpulan

Kehadalan suatu Sistem Informasi Berbasis Komputer sangat tergantung dari kemampuan pengembang dalam merancang basis data. Semakin banyak fitur RDBMS yang dimanfaatkan akan meningkatkan keamanan integritas data.

Memakai RDBMS berbasis file tidak harus mengorbankan keselamatan data, sebab RDBMS berbasis file juga menyediakan fitur-fitur yang cukup untuk kebutuhan manajemen basis data.

Dengan membuat fungsi dan memakai fitur trigger, jika ada kegagalan transaksi akan menyebabkan kegagalan trigger yang bersangkutan, sehingga konsistensi data transaksi dan data Stok akan lebih terjamin.

Daftar Pustaka

Kramek, Andy, 2000, *1001 Things You Always Wanted to Know About Visual FoxPro*, Hentzenwerke Publishing.

Silberschats, 2002, *Database System Concepts*, Mc Graw Hill.

Hoffer, Jeffrey A., 2007, *Modern Database Management*, Pearson Education.