

MENGELOLA HISTORI DATA DENGAN PENDEKATAN DENORMALISASI DATABASE

Damar Widodo

Program Studi Manajemen Informatika
STMIK Jenderal Achmad Yani Yogyakarta

dmr_wdd@yahoo.com

Abstrak

Pengelolaan histori data pada sistem informasi sangat penting guna mengelola data masa lalu dan menjaga konsistensi informasi. Seringkali konsistensi informasi tidak bisa di capai apabila proses pengolahan datanya melibatkan data yang sering berubah. Untuk mencapai tujuan tersebut terdapat berbagai pendekatan. Salah satu pendekatan untuk mengelola data histori yaitu dengan pendekatan denormalisasi database.

Kata Kunci: Sistem Informasi, Data, Informasi, Histori, Database, Denormalisasi.

1. Pendahuluan

Pada organisasi modern, data dan informasi menjadi asset yang berharga. Informasi di pergunakan untuk mendukung pengelolaan organisasi (misalnya organisasi bisnis dan organisasi pemerintahan). Oleh karena itu, pengelolaan data harus mampu memberi gambaran operasional pada saat ini dan mampu menyediakan informasi yang terkait dengan kejadian masa lalu.

Dalam pengembangan sistem, seringkali data masa lalu tidak terkelola dengan baik sehingga informasi kejadian masa lalu tidak bisa ditemukan kembali. Di samping itu, perubahan terhadap suatu data dapat mengakibatkan inkonsistensi informasi. Inkonsistensi informasi harus dihindari.

Perancangan database memiliki peranan yang penting untuk menghindari hilangnya data masa lalu dan inkonsistensi informasi. Untuk mengelola data masa lalu bisa menggunakan tabel histori atau melakukan *copy* terhadap *field* data yang akan dicatat historinya. Tulisan ilmiah ini akan menguraikan pengelolaan data masa lalu untuk menjaga konsistensi informasi dengan pendekatan *copy field* terhadap data yang akan dihistorikan dengan mengambil sampel penanganan histori harga pada suatu transaksi penjualan barang.

2. Dasar Teori

2.1. Sistem Informasi

Sistem Informasi adalah sistem terintegrasi yang menyediakan informasi yang bermanfaat bagi penggunaannya (McLeod and George, 2007). Dalam

pengolahan data, sistem informasi akan memanfaatkan basis data. Basis data dipergunakan untuk mengelola data guna mendukung sistem.

2.2. Data dan Informasi

Data adalah fakta yang mendiskripsikan fenomena tertentu (Stephen, Meave, dan Amy, 2007). Dengan demikian, data akan mewakili suatu objek tertentu, misalnya: harga dan jenis barang. Dalam pengolahan informasi, data merupakan bahan mentah (*raw material*) untuk diproses oleh sistem guna menghasilkan informasi.

Informasi adalah data yang telah diproses menjadi bentuk yang lebih berarti bagi penerima (Stephen, Meave, dan Amy, 2007). Informasi yang berguna harus berkualitas. Informasi berkualitas adalah informasi yang memenuhi kriteria tertentu yaitu akurat, valid, dan tersaji tepat waktu. Keakuratan dan kevalidan informasi yang dihasilkan dari proses transaksi yang melibatkan beberapa data yang nilainya sering mengalami perubahan membutuhkan penanganan khusus dalam perancangan *database*.

2.3. Fungsi *Database* Dalam Suatu Aplikasi

Database adalah mekanisme yang dipergunakan untuk menyimpan data atau informasi (Ryan dan Ronald, 2001). Sistem *database* merupakan bagian penting pada sistem informasi. *Database* diperlukan untuk mengelola sumber informasi. Untuk mengelola sumber informasi tersebut yang pertama kali dilakukan adalah merancang suatu *database*. Perancangan *database* diperlukan agar informasi yang ada pada organisasi tersebut dapat digunakan secara maksimal.

Tujuan Perancangan *Database*

- 1) Untuk memenuhi kebutuhan akan informasi dari pengguna dan aplikasi
- 2) Menyediakan struktur informasi yang natural dan mudah di mengerti oleh pengguna
- 3) Mendukung kebutuhan pemrosesan dan beberapa obyek kinerja dari suatu sistem *database*

Histori data merupakan salah satu kebutuhan pengguna yang harus disediakan oleh *database*.

2.4. Histori Data

Dalam pengembangan sistem informasi, salah satu prinsip yang harus di pertahankan yaitu sistem informasi tidak diperbolehkan menghilangkan data atau

informasi. Oleh karena itu diperlukan pengelolaan terhadap data masa lalu. Sebagai contoh dalam proses transaksi tidak hanya dibutuhkan untuk mengetahui status transaksi tetapi dibutuhkan juga histori transaksi. Histori transaksi akan menjelaskan transaksi masa lalu yang melibatkan *field* data yang sering berubah. Histori data akan mendukung histori transaksi.

2.5. Denormalisasi *Database*

Denormalisasi *database* adalah pelanggaran aturan normalisasi terhadap *database* yang telah normal. Denormalisasi *database* diperlukan untuk meningkatkan performa pengaksesan data pada *database*. *Database* yang telah normal adalah *database* yang redundansi datanya minimum sehingga data yang disimpan tidak mengalami kerancuan dalam proses pengaksesan. (Ryan dan Ronald, 2001)

Perbedaan normalisasi dan denormalisasi adalah terletak pada redundansi data dan kompleksitas *query*. Pada redundansi data normalisasi lebih strik atau harus dihilangkan seminimal mungkin. Proses normalisasi ada sisi positif dan sisi negatif. Sisi negatif yang mungkin muncul yaitu apabila akan mengakses data dalam suatu *database* seringkali membutuhkan *query* yang kompleks. Berbeda dengan denormalisasi, di sini tidak terlalu memikirkan tentang data yang redundan sehingga dalam mengakses data lebih cepat dan *query* seringkali lebih sederhana.

Fungsi utama denormalisasi dalam *database* yaitu untuk menjaga performa sistem. Sebagai contoh apabila menilik lebih lanjut tentang proses pengaksesan yang dilakukan pada *database* relasional. Pada saat data akan diakses berada dalam suatu tabel yang berisi 100 baris (100 *record*) akan memiliki performa yang berbeda dibandingkan data yang akan diakses berada pada tabel yang berisi 10 juta baris (10 juta *record*). *Response time* akan terasa sangat beda sampai data yang akan diakses ditemukan. Apabila pengaksesan data dilakukan pada beberapa tabel yang setiap tabel berisikan jutaan data dan yang diinginkan hanya sebagian saja maka *response time* semakin lama. Oleh karena itu, diperlukan denormalisasi untuk menjaga kestabilan performa suatu sistem.

Denormalisasi *database* di samping untuk meningkatkan performa sistem juga dapat dipergunakan untuk melakukan penganan terhadap histori data. Dalam denormalisasi *database* redundansi data tidak seketat pada pendekatan

normalisasi sehingga pendekatan *copy field* terhadap suatu data untuk tujuan pencatatan data masa lalu dapat di terapkan.

3. Pendekatan Denormalisasi untuk Penanganan Histori Data

Dalam pengembangan sistem, salah satu fungsi histori data antara lain yaitu untuk menjaga keakuratan transaksi. Menjaga histori data guna keperluan keakuratan transaksi menjadi hal yang kritis ketika transaksi yang ditangani ketika dalam pengembangan sistem, untuk memenuhi kebutuhan histori data dapat dilakukan dengan beberapa cara. Misalnya pada sistem penjualan barang. Dalam sistem ini terdapat terdapat tabel barang dan tabel transaksi penjualan. Tabel barang dipergunakan untuk menyimpan data barang dan tabel transaksi untuk menangani proses transaksi. Tabel barang (*TBarang*) memiliki beberapa *field* yaitu *Kd_Barang*, *Nm_Barang*, *Harga*, dan *Stok_Barang*. Tabel ini dinyatakan sebagai *Tbarang* (*Kd_Barang*, *Nm_Barang*, *Harga*, *Stok_Barang*). Tabel transaksi dipergunakan untuk mencatat proses transaksi. Tabel transaksi dinyatakan sebagai *TPenjualan* (*Kd_Kwitansi*, *Kd_Barang*, *Tgl_penjualan*).

Tabel barang memuat data harga barang. Data harga sering mengalami perubahan nilai. Perubahan nilai ini akan berakibat pada proses transaksi yang melibatkan harga yang berbeda-beda. Ketika suatu transaksi terjadi maka harus bisa dibedakan transaksi dengan harga tertentu yang berlaku pada saat terjadinya transaksi. Penangan data transaksi berada di tabel penjualan.

Tabel penjualan memiliki *field* *Kd_Kwitansi*, *Kd_Barang*, dan *Tgl_Penjualan*. Ketika terjadi transaksi maka tabel penjualan hanya akan menangani transaksi sesuai dengan harga terakhir. Perancangan *database* ini mengakibatkan sistem tidak dapat membedakan dua transaksi dengan harga yang berbeda. Apabila terjadi perubahan harga, transaksi yang melibatkan harga yang semula tidak akan tercatat karena laporan transaksi akan selalu merujuk pada harga yang terakhir setelah berubah.

Dalam pengembangan sistem hal ini harus dihindarkan. Salah satu pendekatan yang bisa dipergunakan yaitu dengan mencatat *field* harga pada tabel penjualan. Pendekatan ini akan membuat tabel tidak normal sehingga pendekatan histori data dengan metode ini adalah pendekatan dengan denormalisasi *database*. Dengan pendekatan ini, tabel penjualan akan berubah menjadi *TPenjualan* (*Kd_Kwitansi*, *Kd_Barang*, *Tgl_penjualan*, *Harga*). Pendekatan ini akan menyebabkan redundansi *field* harga yang terdapat pada *TBarang* dan *TPenjualan*. Namun demikian, pendekatan ini masih dibenarkan untuk menangani

histori data. Dengan perubahan desain *database* ini maka sistem akan mencatat transaksi yang merujuk pada harga sesuai transaksi sehingga bisa dicatat perbedaan transaksi sesuai dengan perubahan harga. Oleh karena itu, konsistensi informasi karena perubahan *field* harga dapat dijaga.

4. Kesimpulan

Data histori diperlukan untuk mengetahui transaksi masa lalu. Untuk mendapatkan konsistensi terhadap transaksi masa lalu, data histori perlu dikelola dengan baik. Pengelolaan data histori dapat dilakukan dengan berbagai pendekatan. Salah satu pendekatan yang bisa dilakukan yaitu dengan denormalisasi *database*. Terdapat banyak cara melakukan denormalisasi *database*. Salah satu metode yang bisa dipergunakan yaitu dengan melakukan *copy* terhadap *field* tertentu yang akan dikelola sebagai data histori. Dengan *copy field* tertentu yang akan dihistorikan maka *field* tersebut akan redundan pada beberapa tabel dan menyebabkan terjadi denormalisasi *database*. Namun demikian, redundansi yang terkendali masih bisa dibenarkan untuk kepentingan pengelolaan data histori.

Daftar Pustaka

- Kendal, E.K. & Kendal, J.E., 2002, *Analisis Dan Perancangan Sistem*, Prentice Hall, New Jersey.
- Stephen, D., Cumming, M., & Philips, A., 2007, *Management Information Systems For The Information Age*, McGraw Hill, New York.
- Stephen, R.K. & Plew, R.R., 2001, *Database Design*, SAMS, Indiana.