



Algoritma Konversi Data Penjualan Microsoft Access Ke Mysql

Muhammad Amrin Lubis
Sistem Informasi, STMIK Indonesia Padang
maloebis@gmail.com

<http://dx.doi.org/10.22202/jei.2014.v1i1.1433>

ABSTRAK

Zombie distro company is a distributor and retailer of garments based in the city of Padang. Currently the company has 14 business units spread across the city of Padang and Pekanbaru. During this time, the company has used to manage the Sales Application transaction data, DBMS used is Microsoft Access. Role of applications that manage the data of purchase, stock, distribution and sales business unit. Each business unit is also conducting a similar difference does not make a purchase to receive the goods but increase the mutation of goods from the head office. Business units everyday to send transaction data to the central office via email. Sales application to important combine all the data so that it can be seen: inventory, daily sales, transfers and purchases. Circumstances such as these continue to run for years so it tends to increase the database size to 40MB. Constraint that arises is diminishing Sales Application performance, the management wants to migrate the database from Microsoft Access to MySQL. Data Conversion Algorithm (AKD) is used to convert sales data. How it works AKD algorithm: (1) equating way connection because both are using different platforms using ODBC, (2) read the table one by one from the old database, (3) change the structure of the table and write it to a new database recorder record, (4) repeat process 1, 2 and 3 until all the tables in the old database to the new database migration. Users can directly use the new database that contains the old sales transaction data. The new database can be accessed by the new Sales Application.

Keywords: *algorithm, ODBC, conversion, migration*

PENDAHULUAN

Distro Zombie merupakan perusahaan distributor dan pengecer garmen berkedudukan di kota Padang. Saat ini distro memiliki 15 unit bisnis dengan 1 kantor pusat dan 14 kantor cabang tersebar di kota Padang dan Pekanbaru, Riau. Terdapat 2600 record data item barang yang dikelola dalam perdagangan ini yang kemudian dikelompokkan menjadi 11 kategori, 42 record data merk, dengan 52 supplier yang menjadi pemasok untuk memenuhi kebutuhan barang pada distro ini. Barang yang didistribusikan berupa pakaian, sepatu, tas, dan asesoris. Jadi, aktifitas Distro ini menerima supply barang dari supplier, mengelola dan menyimpannya di gudang, menjualnya kepada konsumen, dan mendistribusikannya ke unit-unit bisnis. Unit bisnis kemudian mengencernya ke customer dan toko lain.

Distro, setiap mendistribusikan barang ke kantor-kantor cabang selalu menyertakan Surat Jalan sebagai pengantar barang yang dikirim, menjelaskan rincian barang yang dikirim. Unit

bisnis kemudian menambah stok barangnya berdasarkan jumlah barang yang diterima, mengelola, menyimpan dan menjualnya kepada konsumen. Setiap sore hari, setelah aktifitas bisnis ditutup, unit-unit bisnis tersebut mengirim data transaksinya ke kantor pusat melalui email. Admin Distro mengimport data tersebut dan menyatukannya dengan database kantor pusat, dengan demikian tampaklah semua aktifitas bisnis kantor-kantor cabang dan kantor pusat.

Kegiatan perekaman data transaksi, selama tahun 2010 terjadi 718 kali transaksi (terdiri dari 16.020 record item) pengiriman barang dari gudang utama kantor pusat ke kantor-kantor cabang. Mengeluarkan 3.314 faktur transaksi penjualan yang berisi 64.776 record item barang yang dijual, merekam 667 record faktur transaksi pembelian dengan 6.130 record data barang yang dibeli dan 259 record data return pembelian. Disamping data tersebut masih ada data master sebanyak: 2.679 item barang dan 2.440 record harga pokok barang. Keuntungan atau laba distro ini diperoleh dari hasil penjualan tepatnya nilai dari selisih harga jual dengan harga pokok perolehan. Semua data diatas telah disimpan

dalam database yang disebut **posdb.mdb** yang telah dibangun oleh vendor Sangkil Biz.

Database Distro yang ada, dibangun oleh vendor lama, masih dioperasikan, namun saat ini tidak ada lagi kontak antara Distro dengan vendor. *Sizedatabase* tersebut yaitu lebih 40 MB.

Saat ini, perusahaan tersebut menggunakan DBMS Microsoft Access untuk menyimpan data kegiatan perdagangan perusahaan, tentu dengan keterbatasan kemampuan menampung data yang ada dan minimnya fitur database yang dimilikinya.

Permasalahan yang timbul yaitu pihak manajemen ingin meminta bantuan pengembang software lain untuk mengalihkan pemakaian database manajemen sistem DBMS dari Microsoft Access ke MySQL tanpa menyetik ulang data yang ada, tidak ingin terjadi kehilangan data transaksi, dan operasional perusahaan tidak terganggu sewaktu terjadi proses pengalihan *resources*. Tentu dengan harapan hasil konversi data yang akurat. Diinginkan database hasil konversi data **posdb.mdb** ke DBMS MySQL dapat langsung diakses oleh Aplikasi Inventory yang baru.

Mengacu kepada permasalahan yang timbul diatas maka dapat dirumuskan masalah yang dihadapi yakni bagaimana menciptakan *algorithmdata conversion* agar pihak manajemen terbantu pada proses pengalihan pemakaian DBMS dari Microsoft Access ke MySQL dengan menggunakan Aplikasi Inventory yang baru?

Batasan masalah yang ditetapkan dalam artikel ini terlihat dalam lingkup sebagai berikut: penelusuran struktur database yang lama, mempelajari *linking* antar *table*, pengkategorian *table* master dan transaksi, mencari dan memakai *tool* penghubung yang berperan sebagai *connector* untuk kedua DBMS Microsoft Access dan MySQL, mempelajari *business rules* yang telah ditetapkan oleh vendor terdahulu, dan menetapkan *programming* yang dipakai.

KAJIAN LITERATUR DAN PEGEMBANGAN HIPOTESIS

Algoritma adalah kumpulan langkah sistematis untuk memperoleh hasil yang diinginkan [0]. Sebelum sebuah algoritma dijalankan, biasanya ada suatu kondisi awal (*initial state*) yang harus dipenuhi. Kemudian, langkah-langkah ini proses hingga mencapai suatu kondisi akhir (*final state*).

Open Database Connectivity (ODBC) adalah strategi *interface* Microsoft untuk mengakses data dalam lingkungan yang *heterogen*, baik untuk DBMS *relational* maupun non-*relational*. Didasarkan pada spesifikasi *Call Level Interface SQL Access Group*, ODBC menyediakan keterbukaan, vendor yang netral cara akses data yang disimpan dalam banyaknya variasi properti *database* komputer pribadi, *minicomputer*, dan *mainframe* (support.microsoft.com). [Error! Reference source not found.]

ODBC merupakan *universal tool* yang dapat digunakan untuk membangun koneksi keberbagai jenis database management system (DBMS) seperti MySQL, Microsoft Access, SQL Server, Oracle, Postgre, dll. Tool ini diperlukan karena banyak pengembang *software* yang menciptakan *software* sendiri dengan aturan main pengoperasian yang mereka tetapkan. Sering ditemukan adanya perbedaan aturan pengoperasian *tool* tersebut, sehingga pengembang *software* aplikasi harus mencari solusinya. *Programmer* dalam membangun sebuah aplikasi *management information system* (MIS) seperti: Aplikasi Inventori, Sistem Informasi Penjualan dan Laporan Keuangan, dsb, juga menggunakan *programming* dan DBMS yang bervariasi. Untuk menangani keadaan seperti ini diperlukan *software* yang belaku *universal*. Dipilihnya *software* yang bervariasi oleh pengembang *software* aplikasi disebabkan kebutuhan, kehandalan, fitur yang tersedia dan kemudahan untuk mendapatkannya.

Konversi data atau migrasi adalah fase di mana desain yang dikembangkan dalam tahap sebelumnya diuji. Migrasi harus dikembangkan secara menyeluruh. Hal ini karena migrasi dapat dilaksanakan hanya sekali, tidak adacoba kedua. ini adalah mengapa proyek migrasi database memiliki tingkat kegagalan yang tinggi [0]. Idealnya, ambil porsi kecil dari sumber data dan diuji ketika mengembangkan migrasi. Pertama

sebagian data diambil dan semua rutinitas diperiksa untuk fungsi yang tepat. Jika hasilnya memuaskan, volume data yang ada ditingkatkan. Setelah hasil tes keluar, semua perubahan yang diperlukan dibuat untuk migrasi dan merancang untuk membuatnya sempurna. Ini mungkin diperlukan untuk melaksanakan pengujian beberapa kali. Hasil akhir setelah fase ini memberikan proses migrasi yang handal dan cepat.

ODBC yang ada saat ini menyediakan suatu pengaksesan data secara universal, dengan ODBC pengembang aplikasi diperbolehkan pengaksesan secara bersamaan, *concurrently*, *view*, dan *modify* data dari berbagai database yang berbeda [Error! Reference source not found.]. Tidak ada jalan langsung migrasi database dari Microsoft Acces ke MySQL. Oleh karena itu penting untuk menggunakan alat pihak ketiga yang sama datang dengan mengatur sendiri keuntungan dan tantangan [0].

Manfaat

- Faster Database Migration*. Biasanya, suatu perusahaan dengan volume tinggi data mulai TB mungkin lebih suka menggunakan Tools yang tersedia.
- Easy to track migration and test results*. Dengan menggunakan bantuan GUI dan logs konversi melacak dan menguji data bermigrasi lebih baik.

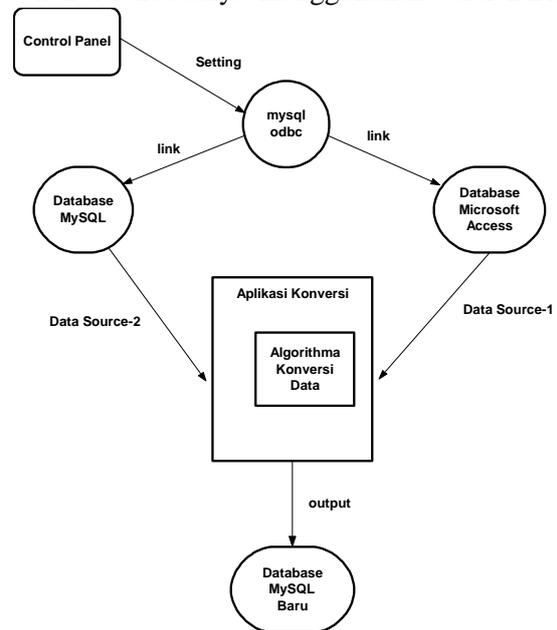
Tantangan

- High Cost of the tool*. Enterprises dengan signifikan rendah mungkin tidak perlu untuk berinvestasi dalam perangkat migrasi yang mahal.
- Database Value conversion mismatch*. Beberapa kunci Bidang Data mungkin memiliki data type pasca *mismatch* migrasi. Semua peralatan pihak ketiga tidak mengurus konversi data type data base JDE, yang menyebabkan data *corruption* [0].

METODE PENELITIAN

Algorithmma selalu menekankan pendekatan penyelesaian masalah dengan

memperhatikan urutan langkah-langkah logis penyelesaian masalah *programming* yang disusun secara sistematis. Gambar 1 metodologi yang digunakan sebagai Algorithmma Konversi Data Database Microsoft Access ke MySQL mengilustrasikan kedua database yang berada dalam sebuah komputer *server*, dengan bantuan ODBC memungkinkan dibangun *link* ke kedua database, dimana kondisi keduanya menggunakan versi DBMS



Gambar 1: Algorithmma Konversi Data Database Microsoft Access dan MySQL

yang berbeda. Diketahui bahwa kedua DBMS Microsoft Access dan MySQL dibangun oleh dua vendor yang berbeda, untuk itu diperlukan pemakai suatu *universal tool* untuk membangun koneksi agar keduanya dapat dikenal oleh Algorithmma Konversi Data, disinilah peran dari *driver mysql odbc*. Ada tiga tahapan secara umum yang dilalui agar proses Konversi Data dapat terjadi. **Pertama**, melalui program *Control Panel*, yang terdapat di *Windows*, kedua database tersebut diseting data sourceny. Hasilnya diperoleh Data Source-1 sebagai alias dari database lama Posdb.mdb, dan Data source-2 sebagai alias dari database baru Distroinventory.pada tahapan ini *linking* kedua database sudah terjadi. **Kedua**, melalui masing-masing data source tersebut diatas database dibaca satu persatu. Berdasarkan struktur database lama, satu persatu tabel dibaca, data yang dikandungnya dikonversi, kemudian data disalinkan ke tabel baru pada dabase baru

Distroinventory. Proses perubahan struktur data dan penggantian penggunaan DBMS disebut migrasi. Ulangi perlakuan langkah yang sama untuk semua tabel yang ada pada database lama Posdb.mdb hingga semua data berhasil dikonversikan ke database baru Distroinventory. Disinilah peran Algorithma Konversi Data diperlukan yaitu menjembatani proses migrasi dari database Microsoft Access ke database MySQL, **Posdb.mdb** ke **Distroinventory**.

Terakhir, terjemahkan Algorithma Konversi Data menjadi program Aplikasi Konversi Data. Uji algorithma tersebut, ketika program Aplikasi dijalankan, perhatikan langkah demi langkah apakah Algorithma Konversi berjalan seperti yang direncanakan? Pastikan data yang menjadi hasil konversi benar dengan cara membandingkan kandungan data tabel yang sama seperti tabel Barang dari kedua database tersebut. Jika benar, maka lakukan proses konversi data untuk semua tabel yang bersumber dari kedua database diatas.

HASIL DAN PEMBAHASAN

Algorithma Konversi Data (AKD) adalah logika program yang dibangun untuk mengkonversi database dari Microsoft Access ke database MySQL. Konversi data memungkinkan dapat terjadi bila kedua database terkoneksi melalui ODBC disebabkan keduanya memiliki struktur database yang berbeda dan keduanya menggunakan DBMS yang berbeda. Konversi data dilakukan per table untuk 22 table yang ada menggunakan AKD dibawah ini.

Algorithma konversi ini memiliki struktur bangun yang terdiri dari: satu algorithma utama yang memanggil satu method AKD. Method AKD memiliki satu struktur program looping yang berfungsi: membaca record data penjualan secara sequential dari database lama, merubah struktur data dan menulisnya ke database baru. Algoritma dimulai dengan: membuat koneksi ke database lama, baca tabel, looping membaca record pada tabel tersebut secara sequential dari awal sampai record terakhir, buat koneksi ke database baru, baca tabel apakah data pada record ini sudah ada dalam tabel baru, jika tidak ada maka

rubah struktur record dan salinkan data ke tabel pada database baru, jika tidak maka abaikan tampilkan pesan "Data ini telah ada". Proses konversi ini berlaku untuk masing-masing tabel yang terdapat pada database lama dan kemudian dikoversikan ke database baru.

Dari 22 table yang dibaca menggunakan algorithma AKD maka diperoleh 100% data berhasil dikonversi dengan waktu kurang dari 5 menit

Sebelum Algorithma AKD dioperasikan pastikan driver *mysql odbc* telah diinstall terlebih dahulu sebelum proses seting dilakukan. Selain ini pastikan semua struktur table pada Data Source-1 sama dengan struktur table pada Data Source-2.

```
// Data Source-1 (DS-1)
// Data Source-1 (DS-2)

Program Utama()

    Inisialisasi var

Connect to DS-1

Connect to DS-2

    Call AKD

End

Function AKD()

tIDorg = input nomor unit bisnis

If tIDorg = "" Then

    Tampilkan pesan "Isi Id Unit

        Bisnis..."

    Exit Sub

End if

Connect to DS-1

read table Source-1.MBarang

Do While Not MBarang End Of File

    Connect to DS-2

    Read tabel Source-2.Mbarang
```

```
If DS-1.idBarang = DS-2.idBarang  
    and DS-1.idUnit=DS-2.idUnit  
    messageshow"Data telah ada"  
  
Else  
    modify struktur table  
    Save data into table MBARANG  
  
Endif  
  
Next record of DS-1  
  
Loop  
  
message "Konversi Data SELESAI....."  
  
End Function
```

Ashish Kumar Mandal, 2012, *DB2 to Oracle database Migration during JDEdwards Upgrade*, Wipro Technologies. All rights reserved

KESIMPULAN

AKD bekerja dengan cara membaca Data Source-1, record demi record menggunakan metoda *access sequential*, mengikuti *business rules* yang ada, kemudian menyimpan data setiap record ke Data Source-2. Proses seperti ini dilakukan berulang hingga semua data pada Data Source-1 disalinkan ke Data Source-2. Pada algoritma AKD setiap data yang akan disimpan sebagai record baru ke Data Source-2, dilakukan pengecekan data terlebih dahulu apakah data yang dimaksud sudah disimpan atau belum, tujuannya untuk menghindari duplikasi data.

PUSTAKA

Munir, Rinaldi., 2008, *Diktat Kuliah IF22153 Matematika Diskrit Program Studi Teknik Informatika*, Institut Teknologi Bandung.

Microsoft Support web site:
<http://support.microsoft.com/>

Arun Swaminathan, 2012, *Database Migration- How hard can it be?*, in International Conference on Information Communication and Management (ICICM 2012), IPCSIT vol. 55 (2012) © (2012) IACSIT Press, Singapore, DOI: 10.7763/IPCSIT.2012.V55.12.