

MIGRASI DATABASE DENGAN MENGGUNAKAN SSIS (SQL SERVER INTEGRATION SERVICE) DI PT XYZ INDONESIA

Banni Satria Andoko *

Abstract : PT. INDONESIA XYZ is the largest Direct Selling company in Indonesia that have strong growth in comparison with other Direct Selling companies in Indonesia. With the rapid development of PT XYZ INDONESIA, then decided to update its sales application that can not follow the process of starting a thriving business. In addition the needs of data storage capacity becomes very large. To be able to meet that need, so the new system required a more reliable and faster in performance. By looking at the problem, we needed a system / application that can move and manipulate the old data so it can be used by the new system. Database migration process is designed using one of the facilities provided by MS SQL 2005, the facility is named SSIS (SQL Server Integration Service).

Keyword : Data Migration, SSIS, ETL, MsSQL 2005, RDBS, SQL Query.

Banyak perusahaan yang memiliki tempat penyimpanan data yang sangat besar menggunakan relasi *database*. Sistem Relasi Database (RDBS) terdiri atas relasi dari DBMS, skema relasi dan database itu sendiri. Sebuah *database* dibangun dari relasi dimana setiap elemen-elemennya terdiri dari sekelompok atribut-atribut *atomic*. Setiap elemen dapat berhubungan melalui *foreign key* (kunci tamu). Hirarki dan relasi diimplementasikan dengan penggabungan (*join*). Sebuah aplikasi, biasanya dibangun dengan menggunakan bahasa pemrograman C, VB dan Delphi yang dapat mengakses *database* dengan menggunakan bahasa SQL yang sudah dideklarasikan.

Relasi data RDBS yang begitu kompleks tidak dapat dimodelkan secara

langsung ke dalam pemrograman berorientasi objek. Solusinya adalah dengan menggunakan *object-oriented database system*. Dengan menggunakan sistem ini, data diakses melalui *object form* sehingga menghasilkan struktur data yg berbeda dengan sistem RDBS.

Perbedaan tersebut mengakibatkan struktur data lama tidak dapat lagi digunakan untuk menunjang kebutuhan dari sistem baru. Sistem aplikasi baru yang dibangun dengan dasar pemrograman berorientasi objek menghasilkan skema/struktur *database* baru yang berbeda jauh dengan struktur data lama.

PT. XYZ INDONESIA membutuhkan sebuah sistem informasi yang lebih menunjang akan kebutuhan-kebutuhan tersebut. Sistem lama yang sudah tidak

* Dosen STMIK Pradnya Paramita Malang

dapat lagi menunjang akan kebutuhan tersebut, harus digantikan dengan sistem baru yang memberikan fleksibilitas dan kehandalan yang diinginkan. Hal ini menyebabkan teknologi lama yang sudah diterapkan/ diimplementasikan juga harus ikut berubah. Bagi perusahaan yang memiliki tingkat data transaksi yang sangat banyak serta kepentingan akan data historikal yang tinggi, mengakibatkan kebutuhan akan kapasitas penyimpanan data serta kehandalan dalam pencarian data (*query*) juga semakin tinggi. Padahal dalam sebuah *database* jika data yang disimpan sudah banyak, akan mengakibatkan proses pengolahan data semakin lambat. \Salah satu solusi yang dapat diterapkan adalah dengan mengganti sistem lama dengan sistem yang baru. Baik itu dalam segi aplikasi sebagai antarmuka (*interface*) maupun *database* server sebagai pusat penyimpanan data.

PT. XYZ INDONESIA adalah sebuah perusahaan *Direct Selling* yang menjual produk yang terbuat dari plastik. Sebagai perusahaan *direct selling* yang pemasarannya sudah mencakup hampir seluruh wilayah Indonesia, maka sebuah sistem/ aplikasi penjualan dibutuhkan untuk dapat membantu kelancaran bisnis perusahaan tersebut. Dengan seiringnya waktu, PT XYZ INDONESIA tumbuh dan berkembang semakin besar. Distributor-

distributor yang tersebar di wilayah Indonesia semakin banyak, penjualan meningkat, anggota semakin besar dan perubahan-perubahan dari proses bisnis yang mengakibatkan sistem lama yang digunakan sudah tidak lagi sesuai dengan kebutuhan bisnis dan tidak lagi mampu menampung data yang sudah semakin besar. Sehingga PT XYZ INDONESIA memutuskan untuk mengembangkan/ membuat sistem baru dengan menggunakan teknologi terbaru yang diharapkan dapat mengatasi permasalahan dari sistem lama.

Aplikasi yang digunakan saat ini adalah sebuah aplikasi yang dibangun dengan menggunakan bahasa pemrograman *Visual Basic 6.0*, *MSSQL 2000* sebagai *database* server dan *Crystal report 6* sebagai *reporting tools*. Sistem lama tersebut akan dikembangkan dengan aplikasi berbasis web yaitu *Visual Studio 2005*, sedangkan untuk *database server*nya diganti dengan versi lebih tinggi yaitu *MSSQL 2005* begitu juga dengan *Reporting tools*nya menjadi *Crystal Reports XI*.

Perubahan yang terjadi cukup signifikan, baik dari sisi aplikasi maupun data. Aplikasi lama dibangun menggunakan pendekatan DFD sedangkan sistem baru menggunakan pendekatan OOP. Hal ini mengakibatkan struktur dari data mengalami perubahan yang sangat drastis. Sehingga struktur data lama tidak dapat lagi

digunakan. Sedangkan data – data yang terdapat didalamnya masih tetap harus ada dan berlanjut didalam sistem baru nantinya.

Tidak semua data dari sistem lama yang akan dipindah. Data yang akan dipindah adalah data master, data summary, data stok dan data historis. Sedangkan untuk data transaksi tidak dipindah. Alasan data transaksi tidak dipindah adalah dikarenakan data transaksi sudah dikumpulkan menjadi satu dan diperhitungkan kedalam data summary.

Sebagai perusahaan *direct selling*, PT. XYZ INDONESIA memiliki sistem penjualan yang berbeda dengan perusahaan dagang lainnya. Product PT. XYZ INDONESIA tidak diperjualbelikan di toko, supermarket maupun showroom. Sistem penjualannya adalah menjual secara langsung kepada konsumen maupun anggota-anggota yang sudah bergabung dengan XYZ melalui distributor.

Dari transaksi penjualan terdapat perhitungan pencapaian-pencapaian dari *salesforce* yang menyebabkan data *historical* menjadi sangatlah penting. Data-data *historical* lainnya adalah data jenjang karir *salesforce* dan prestasi-prestasi yang pernah dicapai oleh para *salesforce*. Jenjang karir dalam PT. XYZ INDONESIA adalah Distributor (DB), General Manager (GM), Manager (Mgr), Team Capten (TC) dan Dealer (Dlr).

Agar konsistensi aplikasi lama dengan aplikasi baru dapat terjaga, maka dibutuhkan sebuah aplikasi lain untuk dapat memindahkan/ migrasi data lama ke dalam data baru.

Jika kita melihat Struktur organisasi diatas, maka dengan jelas dapat kita lihat bahwa PT. XYZ INDONESIA memiliki banyak Distributor yang tersebar di hampir seluruh wilayah Indonesia. Dan perpindahan sistem lama menjadi sistem baru dilakukan secara bertahap, tidak secara serentak. Saat ini Distributor PT. XYZ INDONESIA mencapai jumlah 72 distributor. Masing-masing distributor memiliki datanya sendiri-sendiri. Dengan melihat permasalahan itu maka, dibutuhkan sebuah sistem/ aplikasi/ *service* yang dapat melakukan proses migrasi data secara otomatis, mudah dan cepat.

Jika kita lihat dari uraian di atas maka dapat kita ketahui permasalahan dari data yang digunakan oleh sistem baru:

- Bagaimana memetakan struktur data lama dengan struktur data baru
- Bagaimana cara memindahkan data lama ke data baru yang dapat digunakan dengan mudah

Desain Database

Desain *database* adalah merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya,

tersimpan dalam suatu pusat komputer yang dapat di rubah dengan perangkat lunak tertentu untuk memanipulasinya; *database* merupakan salah satu komponen yang penting dalam sistem informasi karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Sistem basisdata (*database sistem*) ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam – macam di dalam suatu organisasi.

Database

Di dalam proses pembuatan sistem ini, *database* dibuat untuk memenuhi kebutuhan aplikasi, *database* berisikan *file-file*, namun *database* tidak memanipulasi *file-file* tersebut. Untuk memanipulasi *file-file* yang ada di dalam *database* maka digunakan *database manajemen sistem* (DBMS), jadi DBMS adalah “suatu piranti lunak yang digunakan untuk membatasi (*define*), menciptakan (*create*), memelihara (*maintain*), dan mengontrol (*control*) akses ke *database*”, (Conolly, 2002:134).

Recovery control sistem, yang memperbaiki *database* menjadi sebuah *state* konsisten yang sebelumnya menyusul kegagalan pada perangkat keras dan perangkat lunak yang digunakan

Pengertian SSIS (SQL Server Integration Service) ETL

SSIS (*SQL Server Integration Service*) adalah sebuah servis yang terdapat di dalam aplikasi *SQL Server 2005*. SSIS merupakan pengembangan dari DTS (*Data Transformation Service*) yang terdapat di dalam aplikasi *SQL Server 2000*. Secara umum cara kerja dari DTS dan SSIS hampir sama, jika pada DTS bahasa pemrogramannya terbatas pada pemrograman *Active X* dan bahasa *Visual Basic*. Maka pada SSIS ini juga ditambahkan dengan bahasa XML, Visual .NET dan *java*.

SSIS juga dapat berfungsi sebagai *service/tools* untuk data *mining*. Dan dapat pula berfungsi sebagai pengumpul data (*data collector*).

Studi Literatur

Studi literatur ditujukan untuk memperoleh dasar bagi pemecahan masalah atau pencapaian tujuan penelitian ini. Melalui tahap ini diperoleh acuan-acuan, seperti model-model atau hasil-hasil penelitian terdahulu yang berkaitan dengan kebutuhan data, sebagai dasar pengembangan atau pembentukan model serta pengambilan data yang harus dilakukan.

Dalam memecahkan permasalahan tersebut, penulis menggunakan beberapa

metode penelitian, seperti mengumpulkan data-data yang akan dipindahkan, mengumpulkan informasi mengenai cara kerja SSIS dalam melakukan migrasi *database*. Studi pustaka mengenai pengumpulan data dan informasi yang dibutuhkan dalam penulisan ini, penulis mendapatkan dari berbagai literatur buku-buku serta berbagai website di internet yang menginformasikan berbagai teori yang terkait dengan pembahasan penelitian ini.

Analisa Permasalahan Yang dihadapi Dalam Proses Migrasi Database

Permasalahan yang dihadapi dalam migrasi *database* adalah :

- o Struktur data yang berbeda antara sistem lama dengan sistem baru
- o Banyak data yang harus dikalkulasi ulang untuk disesuaikan dengan sistem baru
- o Data yang dipindahkan sangat banyak
- o Pemindahan data dapat dilakukan secara *incremental*

Analisa Kebutuhan Data

Menganalisa kebutuhan data dari setiap tabel sehingga mengurangi data sampah/ data yang tidak dipergunakan lagi oleh sistem. Merekayasa data-data, sehingga data yang dimigrasi sudah sesuai dengan kebutuhan dari sistem baru. Menggabungkan beberapa tabel dengan

proses *query* sehingga menjadi satu tabel yang sesuai dengan kebutuhan data baru.

Perancangan Input, Proses, dan Output

Migrasi *database* dilakukan untuk memindahkan data lama ke dalam data baru, untuk memastikan data yang dipindahkan sudah benar dan sesuai dengan kebutuhan sistem baru dan tetap menghasilkan kebutuhan informasi yang sama dengan sistem lama. Data yang dipindahkan tidaklah sedikit dan data-data itu berasal dari tiap-tiap distributor dimana pada saat implementasi sistem baru pada tiap-tiap distributor, maka prosesnya haruslah cepat dan benar agar tidak mengganggu proses bisnis.

Tempat dan Waktu Penelitian

Penelitian ini dilakukan PT. XYZ INDONESIA dengan lama penelitian selama 1 tahun sejak Januari 2008 – Desember 2008.

HASIL DAN PEMBAHASAN

Langkah-langkah Analisis Data

Langkah-langkah dalam tahap analisis data ini hampir sama dengan langkah-langkah yang dilakukan dalam mendefinisikan proyek sistem yang akan dikembangkan ditahap perencanaan. Perbedaannya terletak pada ruang lingkup tugasnya yang lebih terinci. Tahap-tahap

analisis yang harus dilakukan antara lain :

Mengidentifikasi Data

Mengidentifikasi data merupakan langkah pertama yang dilakukan dalam tahap analisis data, Ketersediaan data *historic* dan data master harus dipastikan lengkap dan tidak memiliki masalah. Permasalahan inilah yang menyebabkan sasaran dari migrasi tidak dapat tercapai. Oleh karena itulah pada tahap analisis data, langkah pertama yang harus dilakukan adalah mengidentifikasi terlebih dahulu data-data yang dibutuhkan. Dan memetakannya kedalam data baru. Jika kebutuhan akan data baru tidak dapat diberikan oleh satu tabel maka penulis harus mencari tabel lain untuk memenuhinya.

Memahami Struktur Data

Inti dari *service* ini adalah memetakan data lama kedalam data baru. Pemetaan data lama dengan data baru haruslah benar dan sesuai dengan kebutuhan sistem baru. Sehingga pada saat sistem baru berjalan, tidak ada lagi kesalahan kalkulasi dikarenakan data tidak lengkap ataupun tidak sesuai. Sehingga proses bisnis dapat berjalan dengan lancar kembali.

Menganalisa data

Pada tahapan ini penulis haruslah

mempelajari kebutuhan data baru. Kekurangan data yang tidak disediakan oleh data lama, haruslah dapat disediakan dalam proses migrasi. Struktur data baru yang menggunakan aplikasi *hibernate*, dimana setiap tabel haruslah memiliki sebuah *unique field* sebagai identitasnya. Sehingga dalam proses migrasi dibutuhkan kunci generator untuk meng-*generate key* tersebut. Selain itu juga terdapat perubahan letak *field* data yang disebabkan adanya perubahan desain sistem.

Membuat Laporan Hasil Analisis

Setelah proses analisis data ini selesai dilakukan, maka tugas berikutnya adalah membuat laporan hasil analisa kebutuhan sistem baru.

Pemetaan Tabel

Dalam rancangan ETL migrasi data ini telah disesuaikan dengan kebutuhan-kebutuhan untuk sistem baru yang telah sesuai dengan proses bisnis. Penulis telah memetakan tabel-tabel dari sistem lama kedalam sistem baru, dalam pemetaan tabel-tabel tersebut, penulis mengelompokkannya menjadi 3 kelompok jenis data. Kelompok-kelompok tersebut adalah :

Kelompok data Master

Yang termasuk dalam katagori data

master adalah data-data yang menjadi acuan atau sebagai data utama yang harus dimiliki oleh sistem lama maupun sistem baru. Data-data ini mencakup identitas & kelompok dari *Salesforce* yang akhirnya digunakan sebagai daftar keanggotaan dari suatu distributor. Selain dari itu, tabel-tabel

master ini juga digunakan sebagai acuan / *primary key* dari tabel-tabel yang lainnya.

Tabel-tabel lama yang termasuk ke dalam data master berikut pemetaannya kedalam tabel baru adalah seperti pada table1.

Tabel 1: Tabel-tabel lama yang termasuk ke dalam data

Tabel Lama	Staging Tabel	Tabel Baru
Tbldealer	tblDuoSynSalesForce	Mstsalesforce Mstsalesforcetrxprofile Mstsalesforceimage Mstsalesforcedistributor
Tblunit	tblDuoSynUnit	Mstunit Mstunitdistributor
TblEzyLivingmember	tblDuoSynELC	MstEzylivingmanagement
TblStock TblStockDetail	stgStockDetail	trxStock trxStockD trxStockSpecial trxStockDistributor trxStockHistory

Kelompok Tabel Data Historical

Pada kelompok ini, tabel-tabel terdiri dari data-data *performance*/pencapaian yang pernah dicapai oleh *salesforce*/anggota-anggota yang bergabung dengan XYZ. Data-data ini dibutuhkan

sebagai salah satu data penunjang kalkulasi untuk menentukan pencapaian oleh *salesforce* dimasa yang akan datang.

Yang termasuk kedalam tabel data historical adalah seperti pada table 2

Tabel 2: Tabel data *historical*

Tabel Lama	Staging Tabel	Tabel Baru
Tblbestsalesforceperformance	None	Mstbestsalesforceactivity
tblPOSHistory tblDealerSummaries	tblDuoHist	Mstsalesforcehistory

Tabel Lama	Staging Tabel	Tabel Baru
tblDealerHist		
tblVanguardSummary tblDealesummaries	tblDuoBonusVanguard	trxBonusOverrideList trxBonusOverrideListDistributor trxBonusPromoteUpBabyManagerList trxBonusPromoteUpBabyManagerListDistributor trxBonusVanguardList trxBonusVanguardListDistributor trxManagerMaintenanceBonusList trxManagerMaintenanceBonusListDistributor trxClaimStarDealerList trxClaimStarDealerListDistributor

Kelompok Tabel Data Summary

Kelompok tabel ini bisa dibilang juga sebagai kumpulan tabel pencapaian-pencapaian dari *salesforce* maupun unitnya. Pencapaian-pencapaiannya dibagi menjadi *summary* mingguan dan *summary* bulanan. Data-data disini lebih banyak diambil dari 1

tabel utama dari data lama dan dipecah menjadi beberapa tabel di system baru. Data *summary* disini juga digunakan sebagai salah satu data yang dibutuhkan untuk kalkulasi pencapain dari *salesforce* dimasa yang akan datang seperti pada table 3.

Tabel 3: Data summary digunakan sebagai salah satu data yang dibutuhkan untuk kalkulasi pencapain dari *salesforce* dimasa yang akan datang

Tabel Lama	Staging Tabel	Tabel Baru
tblDealerSummaries	tblDuoUnitActivityWTD	trxUnitActivityWTD
tblVanguardSummary	tblMonthlyUnitActivity	trxMonthlyUnitActivity
	tblsfactivityWTD	trxSalesforceActivityWTD
	tblMonthlysactivity	trxMonthlySalesforceActivity

Desain SSIS / ETL

SSIS/ ETL adalah sebuah service yang dibangun menggunakan Visual Studio 2005. service ini merupakan salah satu fasilitas yang dimiliki oleh MsSQL Server 2005. pada versi sebelumnya yaitu MsSQL 2000, service ini bernama DTS (Data

Transformation Service). Pada versi tersebut DTS dan SQL server berjalan pada 1 service. Sedangkan pada MsSQL 2005 servicenya sudah dipisahkan.

Desain ETL sangatlah mirip dengan desain Flowchart. Sehingga jika kita sudah

terbiasa flowchart maka untuk desainnya kita tidak akan menemukan kesulitan yang berarti. Sedangkan pemrogramannya, kita dapat menggunakan bahasa pemrograman yang cukup beragam. Karena service ini dibangun menggunakan Visual Studio 2005, maka kita bisa menggunakan beberapa pilihan bahasa yaitu VB, C#, Active X bahkan kita juga dapat menuliskan SQL Statement secara langsung.

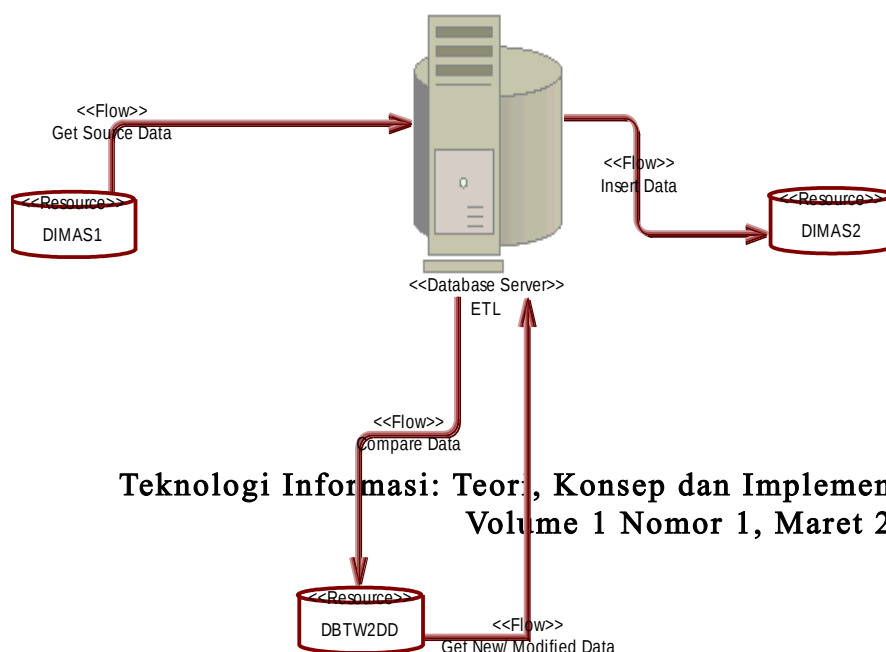
Migrasi data dapat digambarkan seperti pada gambar 1. Jika diperinci lagi, maka proses *flowchart* dari ETL adalah seperti gambar 2.

Dari gambar 2 maka dapat kita implementasikan secara langsung kedalam SSIS. Dalam SSIS ada beragam cara dalam mengeksekusi suatu *query* maupun sebuah *command*. Dalam penelitian ini telah dicoba meminimaliskan pemrograman dan memperbanyak perintah SQL. Selain memindahkan beban proses aplikasi, penelitian ini juga meminimaliskan kesalahan dalam pemrograman.

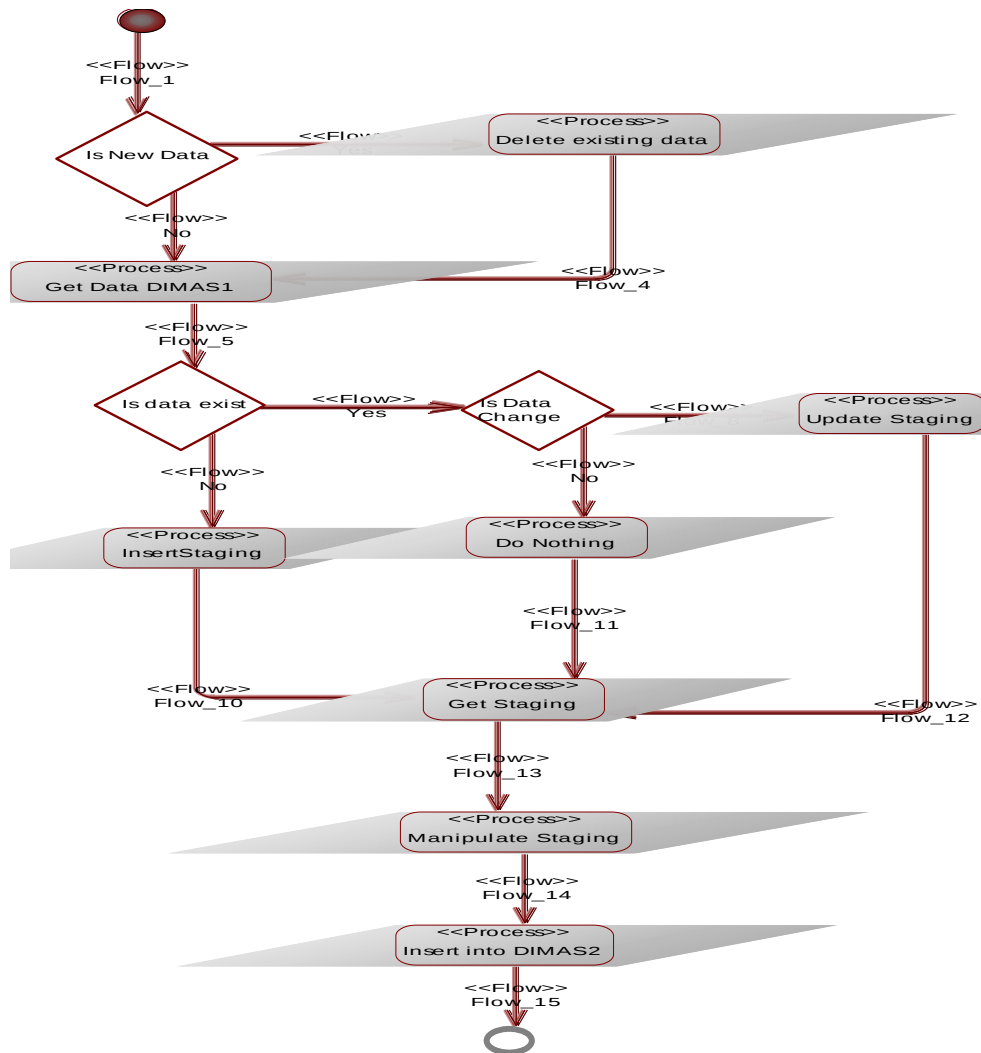
Peneliti membuat sebuah service yang hanya memanggil *Store Procedure* dan melakukan proses pada sisi SQL Server. Sedangkan pemrograman hanya menyiapkan statemen-statement yang akan dipergunakan oleh *Store Procedure* tersebut. Selain statement pemrograman juga dibutuhkan untuk melempar parameter-parameter yang dibutuhkan oleh *Store Procedure* tersebut. Terkadang penulis juga langsung menuliskan perintah SQL tanpa harus dimodifikasi terlebih dahulu. Hal ini hanya berlaku untuk statement yang sudah memiliki global variable dan tidak memerlukan pergantian parameter.

Metode Fetch Function

Metode yang digunakan mengalami beberapa perubahan yang cukup signifikan. Perubahan tersebut merupakan salah satu perubahan yang berhubungan dengan performa dari aplikasi. Dengan perubahan metode tersebut performa dari aplikasi meningkat hingga lebih dari 50%.



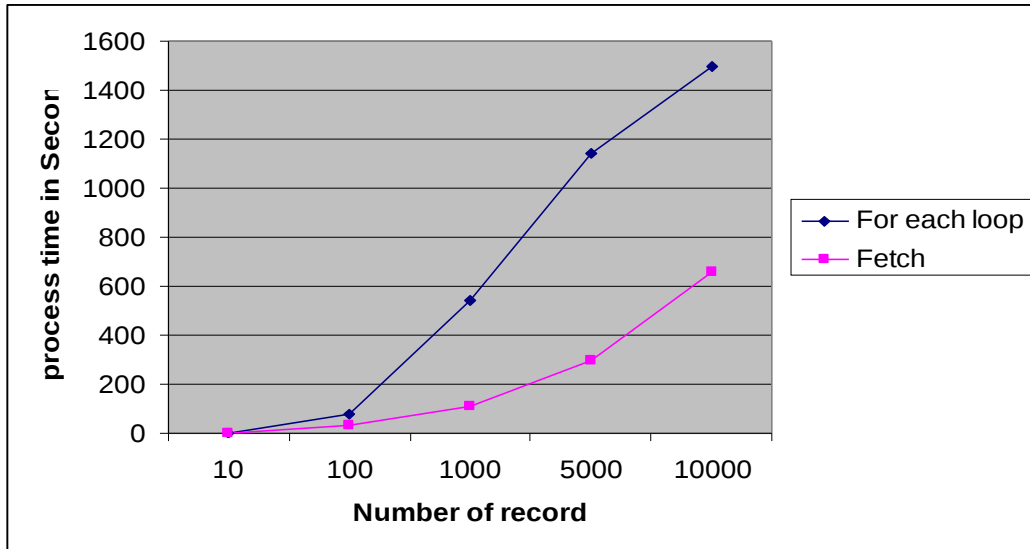
Gambar 1: Gambar Server konektivitas migrasi database



Gambar 2: Diagram Alur Migrasi Data

Metode yang diubah adalah metode pemrogramannya, tetapi jika jumlah data perulangan. Sebelumnya menggunakan yang di proses mencapai puluhan ribu maka metode perulangan yang disediakan oleh proses tersebut akan membutuhkan waktu *tools* dari Visual Studio yaitu komponen *for each loop*. Komponen tersebut memang yang sangat lama dan terkadang dapat lebih mudah dan sederhana dalam menyebabkan komputer berhenti dikarenakan *overflow*. Peneliti

menggantinya dengan metode *Fetch* tidak pernah mengalami *overflow*. *function* yang disediakan oleh MSSQL 2005. dan ternyata performa yang dihasilkan meningkat tajam dan aplikasi



Gambar 3: Grafik peningkatan performa

Kesimpulan

Setelah melakukan pemetaan data dan membuat aplikasi migrasi data di PT. XYZ INDONESIA yang memiliki lebih dari 80 Distributor yang tersebar di hampir seluruh wilayah Indonesia dan memiliki proses transaksi tinggi setiap harinya. Maka proses migrasi data sangatlah penting untuk mendukung proses migrasi sistem. Dari pembahasan tersebut dapat tarik kesimpulan sebagai berikut :

1. Proses migrasi data awal menjadi dasar dari implementasi sistem baru. Semakin cepat migrasi data, maka semakin cepat juga implementasi sistem baru di distributor.

2. Migrasi data awal yang sederhana dan mudah dalam penggunaan, sehingga aplikasi ini dapat digunakan oleh user non-IT.
3. Dengan menggunakan aplikasi, maka kesalahan maupun kekurangan data dapat diminimalisasikan, dikarenakan semua pemetaan data sudah didesain di dalam aplikasi.
4. Aplikasi ini dapat berkembang dan mudah dalam kostumisasi dikarenakan aplikasi ini lebih banyak menggunakan *Store Procedure* yang merupakan procedure/ fungsi yang terdapat ditempat yang berbeda, sehingga kita tidak perlu membongkar aplikasi.

Saran

Sebagai sebuah aplikasi, SSIS sangatlah mudah untuk dibangun dan digunakan. Masih banyak fitur-fitur yang dimiliki oleh aplikasi ini. Salah satu keunggulannya adalah Aplikasi ini terhubung dengan aplikasi SMTP *E-Mail*, Akses langsung ke berbagai jenis server, dapat membaca dan menulis serta memodifikasi file excel, dan desain yang mirip dengan *flowchart*.

Dengan adanya aplikasi ini maka penulis mengharapkan agar semakin banyak orang yang dapat mengenalnya, membuatnya serta mempergunakannya secara maksimal.

DAFTAR PUSTAKA

- Andrew J. Brust, Stephen Forte. 2006. *Programming Microsoft® SQL Server™ 2005*. USA : Microsoft Press
- Andrew Watt. 2005. *Microsoft.SQL.Server.2005.For.Dummies.(2006).BBL*. Hoboken: Wiley Publishing, Inc
- Conolly, Thomas & Begg, Carolyn. (2002). *Databases Systems : A Practical Approach to Design, implementation and management*. Third Edition. Addison Wesley, England.
- Gordon, Steven. R. (1996). *Information System : A Management Approach*. Hartcourt Brace Company, Orlando.
- Itzik Ben-Gan, Dejan Sarka and Roger Wolter, 2006. *Inside Microsoft SQL Server 2005*. USA : Microsoft Press A Division of Microsoft Corporation
- Jogiyanto HM. 1993. *Analisis Dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta : Andi Offset
- MCLeod, Jr. Raymond. (2001). *Sistem Informasi Manajemen, Edisi ke-7, Jilid-1*. Terjemahan Hendra Teguh, SE, AK. PT.Prenhallindo, Jakarta.
- Michael Otey, 2005. *Microsoft SQL Server 2005 New Features*. California : McGraw-Hill/Osborne
<http://msdn.microsoft.com/>
- Ray Rankins, Paul Bertucci, Chris Gallelli, Alex T. Silverstein, et al, 2007. *Ms SQL Server 2005 Unleashed*. USA : Sams Publishing
- Rick F. van der Lans. 2006 *Introduction to SQL: Mastering the Relational Database Language, Fourth Edition/20th Anniversary Edition*: Addison Wesley Professional
<http://www.sqlis.com/>
<http://sqlug.be/>
- Zhao Hui Tang And Jamie MacLennan. 2005. *Data Mining with SQL Server2005*. Canada : Wiley Publishing Inc