

**Komparasi Kinerja ADO dan ODBC dalam memanipulasi RDBMS
Pada Jaringan Model Client-Server**

Gerzon J Maulany, Daniel Parenden
Program Studi Sistem Informasi, Teknik Mesin Fakultas Teknik Universitas Musamus
Merauke - Indonesia

Email : gerzonm@yahoo.com, dparenden@yahoo.com

ABSTRAK

Perkembangan pemrograman jaringan komputer untuk mengakses dan memanipulasi database berkembang dengan cepat, namun demikian pemrograman client/server yang mengakses database masih memperoleh tempat yang sangat berpengaruh, tidak hanya disebabkan oleh masih banyaknya perusahaan skala menengah ke bawah yang menggunakannya tetapi juga karena model pemrograman client/server ini juga dikombinasikan dengan programan multi-tier oleh perusahaan menengah sampai perusahaan berskala Enterprise.

Mengakses database yang berada pada komputer server dari komputer client secara jarak jauh terdapat dua mekanisme yang disediakan oleh Microsoft yaitu ODBC(Open Database Connectivity) dan OLEDB(Objects Linking Embedded Database) yang menggunakan interface ADO (ActiveX Data Object), dua mekanisme ini diciptakan untuk kegunaan yang sama yaitu akses database namun peruntukan detailnya terdapat perbedaan untuk itulah dalam penelitian ini bertujuan untuk membandingkan kinerja dari kedua mekanisme tersebut dalam mengakses RDBMS (Relational Database Management System), sebab penggunaan mekanisme yang tepat akan berpengaruh terhadap unjuk kerja pengaksesan dan berpengaruh pada keluaran dari sistem.

Penelitian dilakukan dengan menganalisis permasalahan koneksi database dan mendisain sistem yang akan dibuat serta dilakukan pengujian untuk mendapatkan hasil dari kedua mekanisme tersebut pada proses pembacaan dan proses penulisan dari dan pada database sehingga diperoleh gambaran kinerja yang dihasilkan oleh kedua mekanisme koneksi dalam mengakses database dan memanipulasinya.

Kata Kunci: *ADO, ODBC, Relational Database*

PENDAHULUAN

Model program untuk mengakses database berkembang dari stand alone, Client/Server dan Multitier, untuk pemrograman model stand alone masih juga dibuat namun saat ini sudah tidak banyak yang menggunakannya, Model Clie/Server adalah model pengembangan selanjutnya yang populer yang dikembangkan dalam jaringan mulai dari LAN(Local Area Network), MAN (Metropolitan Area Network) dan sebagian juga digunakan pada WAN (Wide Area Network), sedangkan model Multitier adalah model pengembangan lanjutan dan terkini yang diimplementasikan pada WAN dengan segala kemajuan teknologi internet yang menyertainya dan telah banyak dikembangkan pada perusahaan berskala Enterprise.

Di Indonesia khususnya banyak perusahaan berskala menengah ke bawah yang masih sangat bergantung pada model client server yang pada gilirannya akan berkembang menjadi model Multitier, untuk itulah maka model Client/Server ini masih banyak digunakan dalam pemrograman yang digunakan pada perusahaan-perusahaan tersebut. Dan dalam pemrograman model Client/Server untuk mengakses database mekanisme koneksi database menggunakan ADO dan ODBC masih sering digunakan.

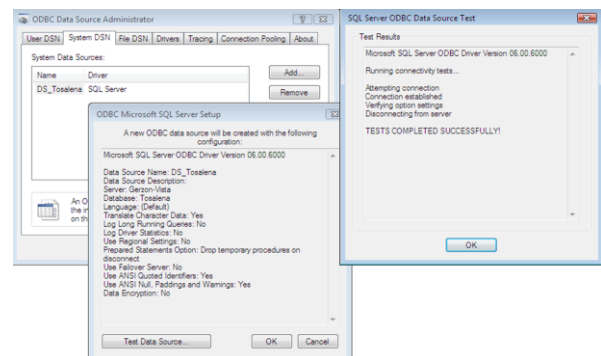
Untuk menggunakan suatu mekanisme dengan baik dan benar maka diperlukan pemahaman untuk yang memadai dalam penggunaan kedua mekanisme tadi, banyak bahasa pemrograman saat ini yang dibandel dengan kemampuan koneksi dan manipulasi database dengan kedua mekanisme ini, namun tidak banyak sumber yang membahas secara detail tentang kinerja yang dihasilkan dari kedua mekanisme pengaksesan database ini, ada asumsi bahwa ODBC lebih cepat dari ADO, atau Akses ADO jauh lebih cepat dari ODBC dikarenakan ODBC adalah mekanisme yang sudah cukup tua dan diperuntukan untuk mengakses jenis database dari berbagai macam vendor yang berbeda-beda. ADO disebut sebagai teknologi masa depan Microsoft yang tidak mengembangkan lagi teknologi API ODBC.

Untuk itulah diperlukan penelitian atau kajian untuk mengetahui kinerja sesungguhnya yang dihasilkan oleh ODBC dan ADO sebagai Database Engine yang mengedikan fasilitas-fasilitas standar pengaksesan dan manipulasi database lebih khusus akan dilakukan pengujian pada model database relational (RDBMS), sehingga nantinya dapat diketahui pada keadaan mana kedua mekanisme ini akan dapat digunakan secara maksimal untuk mendapatkan kinerja optimum.

ODBC (Open Database Connectivity)

ODBC dibuat untuk menyediakan konektifitas ke Database Engine yang terbebas dari aturan bahasa pemrograman atau kekangan database engine tertentu dan dapat dijalankan pada berbagai macam sistem operasi. ODBC adalah bagian dari API (Application Programming Interface) yang menyediakan koneksi dari sebuah aplikasi database ke RDBMS.

Pada sistem operasi Windows milik Microsoft disediakan ODBC Data Source Administrator untuk membuat dan mengatur koneksi secara permanen pada database dengan parameter yang sesuai yaitu computer name / server name/ IP number, User Name, password, port number, Database Name dan Driver Database yang sesuai. Nama DSN (Data Source Name) ini yang dapat digunakan dalam program aplikasi untuk mengkoneksikan program client dengan database Server yang berada pada server tertentu. Seperti terlihat pada gambar dibawah ini DataSource akan menghubungkan program aplikasi dengan database bernama Tosalena yang terkonfigurasi dengan nama Data Source DS_Tosalena menggunakan Driver SQL Server.



Gambar 1. Konfigurasi Data Source Name (ODBC).

Pada dasarnya konfigurasi koneksi Data Source ini dapat juga dilakukan dengan menggunakan koneksi string (String Connection) untuk setiap database sumber yang akan dituju dan driver database yang sesuai. Sistem dengan menggunakan koneksi string ini akan digunakan sepanjang penelitian ini dengan menggunakan file metafile untuk menyimpan data koneksi string tersebut, dan jika akan terjadi perubahan maka file metafile ini akan dapat dihapus untuk membuat koneksi baru yang sesuai dengan konfigurasi yang sesuai.

ADO (ActiveX Data Objects)

ADO adalah objek dari bagian objek COM (Common Object Model) yang dibuat oleh Microsoft untuk mengakses dan manipulasi database yang berada pada sebuah Data Source untuk berbagai macam tipe data sehingga tidak hanya RDBMS saja yang dapat diakses oleh ADO yang merupakan interface dari OLEDB sebagai API untuk mengkoneksikan program client ke database yang berada pada server. Hasil dari koneksi yang dilakukan menggunakan ADO akan dapat diperoleh dengan menggunakan properti-properti yang ada beserta dengan objek ADO tersebut. Setelah objek ADO yang menjadi antarmuka dengan database di server terbentuk maka selanjutnya pengembang dapat melakukan pengaksesan dan pengelolaan database sesuai dengan yang diinginkan.

Relational Database

Dari beberapa jenis Database/Basis data yang ada Relational Database adalah yang paling banyak digunakan sampai saat ini, memang untuk ke depan model data baru yang adaptive akan sangat berkembang dengan sangat pesat. Model Relational Database ini menggunakan kolom dan baris untuk mengidentifikasi suatu entitas data, kolom atau atribut menerangkan item data pada nilainya sedangkan baris atau record/tuple menghimpun semua nilai data dari atribut yang ada guna mendeskripsikan suatu data entitas yang unik.

Relational Database terdiri dari objek-objek berupa tabel, view, store procedure dan skema lainnya yang terhubung dengan aturan tertentu untuk menghasilkan pendeklarasian keterhubungan untuk mendeskripsikan sistem entitas tertentu. Antara penghubung tersebut terdapat kunci-kunci utama yang akan mendefinisikan secara unik dan menjamin suatu record dapat diakses dengan kunci unik tersebut, tetapi juga kunci ini akan menjadi kunci tamu pada tabel anakan yang akan menjamin keintegrasian referensi untuk hubungan yang dibangun.

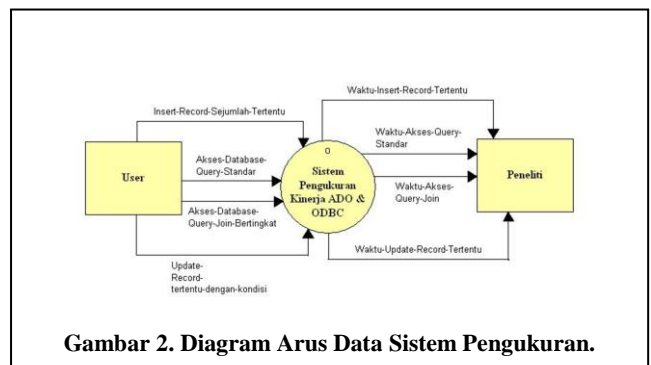
RDBMS yang tersedia saat ini terdapat beberapa yang mendukung pengelolaan database dari kecil, menengah bahkan sampai dengan skala database untuk Enterprise. Vendor-vendor seperti Microsoft, IBM, Oracle dan Sun serta lainnya, telah mengembangkan RDBMS yang tangguh dan realible dalam mendukung proses bisnis berskala besar. Seperti yang akan digunakan dalam penelitian ini yaitu Microsoft SQL Server 2000 (MS SQL Server 2000).

DESAIN SISTEM

Sebelum mendesain sistem ini dilakukan analisis untuk memperoleh spesifikasi perangkat lunak yang akan digunakan untuk mendukung proses pengujian perbandingan kinerja mekanisme akses database menggunakan ODBC dan OLEDB menggunakan ADO. Berikut adalah spesifikasi yang dibutuhkan oleh perangkat lunak yang akan dikembangkan yaitu :

1. Mampu membuat koneksi ke database dengan menggunakan ODBC maupun ADO di dalam suatu sistem
2. Mampu Mengakses database pada server menggunakan perintah SQL untuk meretrieve data, maupun mengubah data serta mengupdate data dengan cara yang sama namun menggunakan masing-masing koneksi baik ODBC maupun ADO.
3. Mampu menghitung waktu yang dibutuhkan untuk melakukan perintah pengaksesan database per periode tertentu sampai hasil data tersebut didapat.
4. Mampu merekam data hasil perhitungan dan menyajikan data tersebut sebagai informasi dalam bentuk grafik untuk memudahkan proses perbandingan.

Software Pengukuran Kinerja dirancang dengan membuat suatu sistem yang dapat mencatat waktu (kinerja) yang dibutuhkan oleh pengguna yang menggunakan komputer sebagai client dan menjalankan sebuah perintah query ke server database yang akan memparse perintah tersebut dan mengembalikan hasil query yang dimaksud. Waktu yang diperlukan termasuk waktu tunggu tersebut yang digunakan sebagai acuan kinerja database engine ADO dan ODBC melaksanakan sebuah permintaan (query). Model Sistem Pengukuran Keinerja tersebut dapat dilihat pada gambar 2



Membuat Koneksi menggunakan ODBC & ADO

Basis koneksi yang akan dibuat tidak menggunakan DSN yang disediakan oleh Windows namun menggunakan kaidah koneksi string (string connection) dimana baik koneksi menggunakan ODBC maupun yang menggunakan ADO tetap menggunakan koneksi yang sama sehingga hasil koneksi yang digunakan tidak ada yang berbeda kecuali fungsi yang digunakan untuk memanfaatkan koneksi ODBC dan objek yang digunakan oleh program untuk mengakses database menggunakan ADO berbeda. Hal ini dimaksudkan untuk mendapatkan parameter dan langkah yang sama dalam hal untuk mengukur dua mekanisme cara pengaksesan yang berbeda. Berikut adalah

```
publi awal,koneksi_ke_server,mstr
public departemen
public objek,RS,PERINTAH
MODIFY WINDOW SCREEN from 0,0 to 47,126
TITLE 'Aplikasi Measurement Performance of ADO
& ODBC' NOCLOSE NOFLOAT NOGROW
MINIMIZE ZOOM
```

```
set exclusive off
set delete on
set date DMY
set century on
set hours to 24
set safety off
set exact on
```

```
IF FILE("KONEKSI.STT")
  RESTORE FROM KONEKSI.STT ADDITIVE
  KON_SANDI=KON_SANDI

KON_NAMA_SERVER=KON_NAMA_SERVER
KON_BASISDATA=KON_BASISDATA

KON_NAMA_PEMAKAI=KON_NAMA_PEMAK
AI
  KON_DRIVER1=KON_DRIVER1
```

```
mstr =
[DATABASE=&KON_BASISDATA;SERVER=&K
```

```
ON_NAMA_SERVER;UID=&KON_NAMA_PEM
AKAI;pwd=&KON_SANDI;DRIVER=&KON_DRI
VER1]
  koneksi_ke_server =
SQLSTRINGCONNECT( mstr)
  IF koneksi_ke_server <= 0
    =MESSAGEBOX("KONEKSI KE
SERVER DATABASE SQL Server 2000 via ODBC
GAGAL", 0+16, "PESAN")
  QUIT
  ELSE
    objek =
CREATEOBJECT("ADODB.CONNECTION")
    objek.open( mstr )
    IF VARTYPE( objek ) # "O"
      =MESSAGEBOX("KONEKSI KE
SERVER DATABASE SQL Server 2000 via ADO
GAGAL", 0+16, "PESAN")
    QUIT
    ELSE
      =MESSAGEBOX("KONEKSI KE
SERVER DATABASE SQL Server 2000 via ADO
BERHASIL ", 0+64, "PESAN")
    ENDIF
    Rs = CREATEOBJECT(
"ADODB.RECORDSET" )
    Perintah = CREATEOBJECT(
"ADODB.Command" )
    Perintah.CommandType = 1
    Perintah.ActiveConnection = objek
    =MESSAGEBOX("KONEKSI KE
SERVER DATABASE SQL Server 2000 via ODBC
BERHASIL ", 0+64, "PESAN")
    DO MENUTAMA.MPR
    READ EVENTS
  ENDIF
  ELSE
    PARAM1=2
    DO FORM KONEKSI.SCX WITH
PARAM1
    READ EVENTS
  ENDIF
```

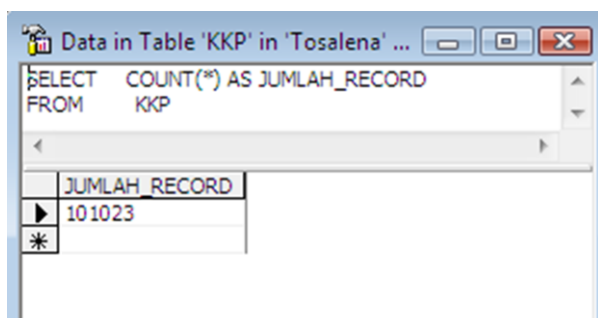
Pengambilan data retrieve

Untuk melakukan Proses Pengakses data standar tetap baik menggunakan mekanisme ODBC maupun ADO tetap menggunakan perintah query

dengan SQL (Structure Query Language) sebagai bahasa perintah yang digunakan untuk mendapatkan hasil dengan kaidah perintah yang sama. Untuk mendapatkan hasil yang cukup representatif dari pengujian maka proses pengambilan data dilakukan delapan kali dengan tingkatan beban yang ditingkatkan dengan kategori yang seimbang.

Proses retrieve dilakukan pada satu tabel yang berada pada satu database tabel yang dimaksud berisi data-data tentang kartu kontrol produksi yang terdiri dari 46 field dan semua field tersebut berisi ini dan tipe-tipe field yang beranekaragam yaitu nvarchar, datetime, money, integer dan logical bit. Dengan jumlah field yang cukup banyak maka proses pengambilan data dari tabel akan menggunakan resource yang cukup untuk diuji.

Untuk jumlah record pada tabel kartu kontrol produksi (KKP) itu jumlah record haruslah lebih dari jumlah maksimal yang jumlah record yang akan diakses dalam sistem ini paling banyak record yang diakses adalah 50.000 record, terlihat pada gambar 2 jumlah record telah melebihi dari data yang dibutuhkan untuk ditampilkan.



Gambar 3. Jumlah record tabel KKP.

Algoritma untuk proses pembacaan record standar dari tabel kkp yang berada pada database adalah :

1. Simpan waktu awal saat ini ke variabel waktu_awal
2. Jalankan perintah query untuk mengakses jumlah record yang diinginkan (5K...50K record)
3. Simpan waktu saat ini sebagai waktu_akhir
4. Waktu yang diperlukan untuk mengakses data tersebut adalah waktu_akhir-waktu awal

Algoritma untuk proses pembacaan record dengan level join yang bertingkat dari tabel kkp yang berada pada database adalah :

1. Simpan waktu awal saat ini ke variabel waktu_awal
2. Jalankan perintah query untuk mengakses data dari tabel KKP dengan join pada tabel 1..11
3. Simpan waktu saat ini sebagai waktu_akhir
4. Waktu yang diperlukan untuk mengakses data tersebut adalah waktu_akhir-waktu awal

Penambahan record baru dan update

Untuk proses penambahan record baru diperlukan proses untuk menambah dengan jumlah record penambahan yang bertingkat dengan tahapan yang mewakili penambahan data secara konstan. Dalam penelitian ini jumlah terkecil yang ditambahkan adalah 5000 record dan yang terbanyak adalah 500000 record dengan setiap proses peningkatan jumlah record yang ditambahkan berbeda sebesar 5000 record, hal ini dilakukan pada proses penambahan menggunakan mekanisme ODBC dan ADO sehingga dapat dilihat perbedaan.

Sedangkan untuk proses pengupdatetan dilakukan harus setelah proses insert sebab yang dilakukan adalah mengubah field tertentu yang telah ditambahkan sebelumnya pada saat proses insert menjadi nilai field baru setelah diproses.

Algoritma untuk proses penambahan adalah sebagai berikut :

1. Simpan waktu awal saat ini ke variabel waktu_awal
2. Jalankan perintah query untuk menambah record pada tabel KKP dan memasukan nilai fieldnya lakukan sesuai dengan banyaknya record yang diinginkan
3. Simpan waktu saat ini sebagai waktu_akhir
4. Waktu yang diperlukan untuk mengakses data tersebut adalah waktu_akhir-waktu awal

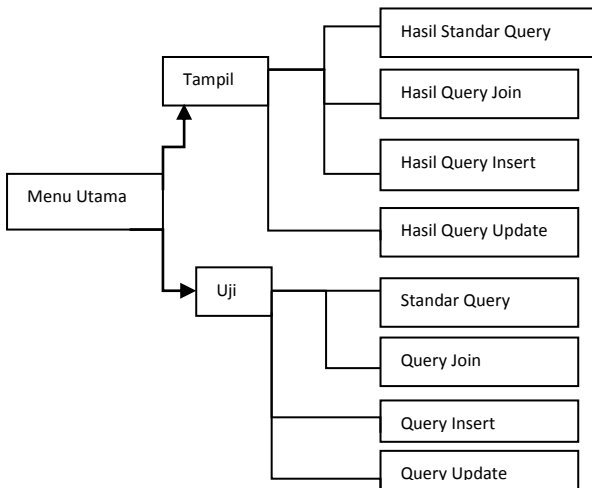
IMPLEMENTASI SISTEM

Sistem perbandingan kinerja ODBC dan ADO ini diimplementasikan pada tipe jaringan LAN dengan menggunakan kabel dan sebuah hub 8 port, kecepatan maksimum yang dapat dicapai untuk proses transfer data di jaringan ini adalah 100MB. Untuk menerapkan model Client/Server digunakan dua buah laptop yang pertama berfungsi sebagai server dengan spesifikasi Intel Core 2 Duo 2GHZ, RAM DDR2 2GB, HDD 250GB, laptop

berikutnya berfungsi sebagai client dengan spesifikasi Intel Dual Core 1,7GHZ, DDR2 2GB,HDD 250GB.

Untuk mengembangkan interface sistem digunakan bahasa pemrograman Microsoft Visual Foxpro 8.0 sedangkan untuk database Server digunakan Microsoft SQL Server 2000 Developer Edition yang mampu mengelolah database sampai 4TB. Sistem operasi pada server digunakan Windows Xp SP2 dan pada sisi client digunakan Windows Vista Ultimate Edition.

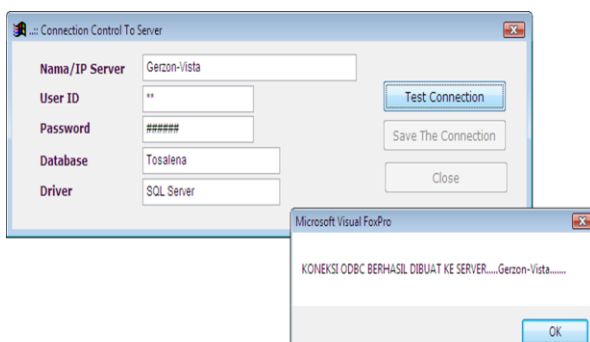
Aplikasi ini mempunyai antarmuka utama sistem sebagai berikut :



Gambar 4. Struktur Menu Aplikasi.

PENGUJIAN SISTEM

Berikut ini adalah proses pengujian untuk aplikasi yang digunakan untuk mengukur kinerja mekanisme pengaksesan database menggunakan ODBC dan ADO. Saat pertama kali sistem dijalankan maka akan diperiksa apakah koneksi (file koneksi.STT) telah ada jika tidak maka akan keluar form pengisian konfigurasi nama server, user name, password , database name dan driver database yang sesuai. Seperti terlihat pada gambar 5. Jika telah terisi dengan parameter yang benar maka selanjutnya koneksi akan diuji apakah dapat dijalankan dengan baik atau tidak.

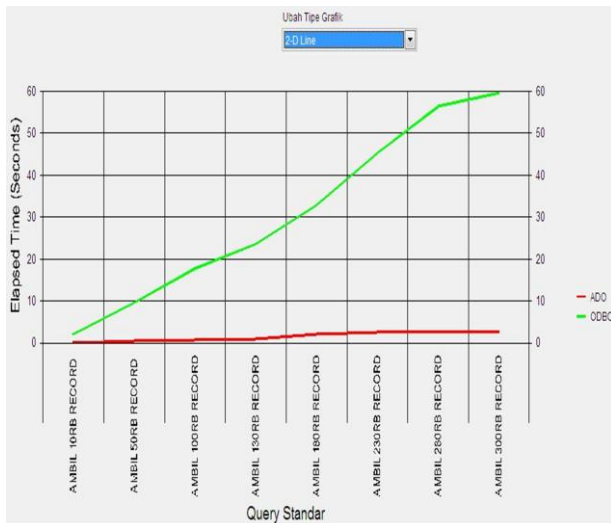


Jika koneksi telah berhasil maka selanjutnya adalah melakukan pengujian untuk query standar yaitu dengan meretrieve record dengan jumlah tertentu akan terlihat waktu yang dibutuhkan untuk melakukan perintah SQL dengan string `msql=[SELECT TOP 10000 * FROM ABSEN]` yang berarti mengambil 10.000 record awal dari tabel Absen nilai 10.000 record akan diganti dengan nilai berikutnya sesuai dengan kebutuhan record yang akan diretrieve sampai dengan 300.000 untuk mendapatkan 300.000 record dengan kenaikan jumlah record setiap tahapan ditambahkan 5000 record. Berikut adalah tambilan proses mengukur dengan ODBC dan ado seperti gambar 5 dibawah ini.

Proses	Time (s)	Detail	Start Time	End Time
Proses ODBC	2.103	DETIK	41504.379	41506.482
Proses ODBC	9.505	DETIK	41509.042	41518.547
Proses ODBC	17.667	DETIK	41525.122	41542.789
Proses ODBC	23.606	DETIK	41545.766	41569.372
Proses ODBC	32.770	DETIK	41601.786	41634.554
Proses ODBC	45.272	DETIK	41653.962	41699.234
Proses ODBC	56.383	DETIK	41703.918	41790.201
Proses ODBC	59.516	DETIK	41789.754	41829.270
Proses ADO	0.126	DETIK	40929.482	40929.588
Proses ADO	0.421	DETIK	40932.694	40933.115
Proses ADO	0.755	DETIK	40958.087	40958.842
Proses ADO	0.957	DETIK	40994.095	40995.052
Proses ADO	2.072	DETIK	41070.233	41072.305
Proses ADO	2.480	DETIK	41132.994	41135.454
Proses ADO	2.669	DETIK	41226.802	41229.471
Proses ADO	2.524	DETIK	41334.585	41337.109

Gambar 6. Form Pengukuran Query Standar.

Hasil dari garfik dari data pengukuran ini dapat dilihat pada form grafik untuk menghasilkan grafik perbandingan seperti kinerja ODBC dan ADO dalam mengeksekusi Query Standar, hal ini dapat dilihat pada gambar 7. Terlihat bahwa waktu yang diperlukan oleh mekanisme koneksi ADO lebih baik dibandingkan dengan ODBC yang pada awalnya tidak terlalu berbeda jauh namun semakin banyak record yang diretrieve maka waktu yang dibutuhkan oleh ODBC semakin banyak.



Gambar 7. Form Grafik Kinerja Query Standar .

Berikutnya adalah proses untuk membandingkan eksekusi query dengan dengan tingkat Join yang bertingkat dengan hubungan antara tabel KKP dengan file lainnya samapi tingkat join terbanyak sebanyak sebelas hubungan dengan tabel KKP terlihat form proses mendapatkan waktu eksekusi query join bertingkat seperti gambar 8 dan gambar 9 menampilkan hasil grafik untuk perbandingan waktu eksekusi query join bertingkat.



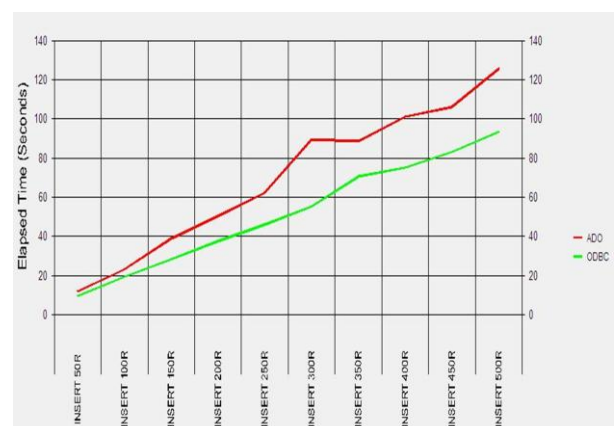
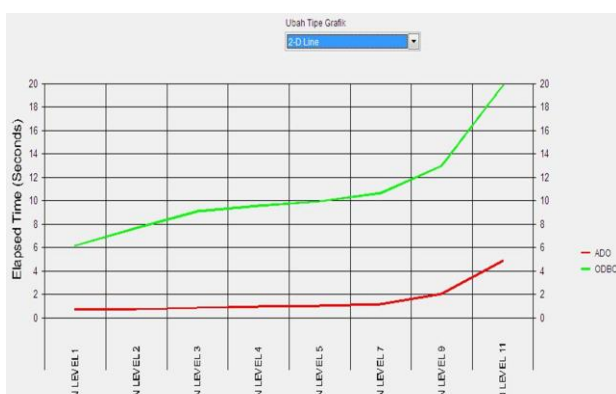
Gambar 8. Form Proses Query Join bertingkat.

Terlihat bahwa kinerja Query Join Bertingkat untuk koneksi yang menggunakan ADO tetap membutuhkan waktu yang lebih sedikit dibandingkan dengan koneksi yang menggunakan mekanisme ODBC.

Proses selanjutnya adalah mengukur kinerja query penambahan record dengan jumlah tertentu mulai dari jumlah record 5000 sampai dengan 500.000 record dengan peringkatan setiap tahapannya adalah 5000 record. Proses pengukuran proses query Insert ini dapat dilihat pada gambar 10 dan hasil grafiknya



Gambar 10. Form Proses Query Insert sejumlah Record.

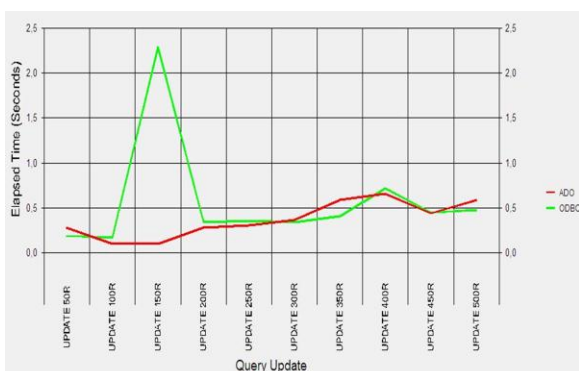


Terlihat untuk grafik kinerja query insert record waktu yang dibutuhkan ODBC lebih baik dibandingkan dengan perintah SQL query insert yang menggunakan mekanisme koneksi ADO.

Pengukuran terakhir dilakukan untuk mengukur waktu yang dibutuhkan dalam proses update sejumlah record yang telah ditambahkan sebelumnya dengan perintah Query Insert. Proses pengukuran Query Update ini dapat terlihat pada gambar 12 dan hasil grafiknya terlihat pada gambar 13.



Gambar 12. Form Proses Query Update sejumlah Record.



Gambar 13. Form Grafik Kinerja Query Update

Hasil pengukuran ini terlihat bahwa proses pengupdatetan / pengubahan data tentulah memerlukan waktu yang jauh lebih sedikit dibanding proses insert/penambahan sebab yang perlu dilakukan adalah mengubah nilai field tertentu bandingkan dengan proses insert / penambahan record baru yang harus menambahkan sebuah record kosong dan mengisi nilai-nilai field kosong tadi dengan nilai tertentu. Hasil menunjukkan bahwa proses pengupdatetan / pengubahan data untuk koneksi menggunakan ADO maupun ODBC tidak berbeda jauh sehingga dapat dikatakan bahwa tidak terlihat pengaruh yang signifikan pada proses update.

KESIMPULAN

Kesimpulan dari hasil penelitian ini adalah sebagai berikut :

1. Program berhasil mengukur tingkatan kinerja antara koneksi database menggunakan mekanisme ODBC dan ADO.
2. Hasil pengukuran menunjukkan bahwa untuk proses-proses yang menggunakan query standar ADO mempunyai kinerja rata-rata 78,76% lebih baik dari ODBC, sedangkan untuk proses yang menggunakan query dengan level join bertingkat ADO memiliki kinerja rata-rata 44,03% lebih baik dibanding ODBC sedangkan untuk proses query insert atau pengubahan dan penghapusan kinerja ODBC rata-rata 20,87% lebih baik dibandingkan kinerja ADO. Dengan demikian dapat dipakai sebagai pertimbangan bagi pengembang aplikasi database model client/server dalam menggunakan mekanisme ODBC atau ADO sesuai dengan keadaan yang dibutuhkan. Terlihat bahwa ADO akan lebih baik dalam proses retrieve record-record tabel, sedangkan ODBC akan lebih baik dalam hal proses penulisan/pengubahan data record pada tabel.

3. Dimungkinkan dalam suatu aplikasi jaringan client/server menggunakan kedua mekanisme ini secara bersamaan.

SARAN

1. Untuk penelitian kinerja ADO dan ODBC dapat dikembangkan untuk pengukuran menggunakan database management system yang lainnya seperti MySQL atau Oracle
2. Perlu dilakukan pengembangan software dengan menggunakan bahasa pemrograman lainnya selain Microsoft Visual Foxpro.

DAFTAR PUSTAKA

1. Marcia Atkins., Andy Kramek, and Rick Schummer. 2002. *MegaFox: 1002 things You Wanted to Know About Extending Visual Foxpro*. Hentzenwerke Publishing, Whitefish bay. USA.
2. Marcia Atkins., Andy Kramek, and Rick Schummer. 2000. *1001 things You Wanted to Know About Visual Foxpro*. Hentzenwerke Publishing, Whitefish bay. USA.
3. Jogiyanto HM, 2000. *Analisis dan Disain Sistem Informasi : Pendekatan Terstruktur teori dan praktek aplikasi bisnis*. Edisi ke 3, Andi Offset .Yogyakarta
4. Kenneth Chazotte. 2004. , *Build Your Own Framework With Visual Foxpro*. Hentzenwerke Publishing, Whitefish bay. USA.
5. Chuck Urwiler, Gary DeWitt, Mike Levy and Leslie Koorhan, 2000. *Client/Server Applications with Visual FoxPro and SQL Server*. Hentzenwerke Publishing, Whitefish bay. USA.
6. Mike Gunderloy and Joseph L. Jordan, 2000. *Mastering SQL Server 2000*. Sybex. USA
7. Sutejo, S.Kom., 2004. *Membuat Program Aplikasi Kearsipan dengan Visual Foxpro 8.0* . Elexmedia Komputindo. Jakarta