ANALISIS KECEPATAN AKSES DATA DENGAN MENGGUNAKAN KONEKSI ODBC DAN OLE DB UNTUK EFISIENSI DATABASE

Muhammad Dahria

Program Studi Sistem Informasi, STMIK Triguna Dharma

m.dahria@gmail.com

ABSTRAK: ODBC dibatasi pada pemakaian bersama database relational karena hanya merupakan bahasa standar bagi pengiriman permintaan ke sumber data ODBC. Selain itu juga ODBC bukanlah berorientasi obyek. Pengkodean dan debugging lebih sulit menggunakan ODBC dan lemah pada penerjemahan model obyek ke dalam kode akses data yang lebih banyak.

OLE DB adalah komponen kunci dalam strategi Universal Data Access (UDA) Microsoft. OLE DB mendefinisikan serangkaian antarmuka COM yang membungkus layanan akses data. OLE DB adalah sebuah antarmuka pemrograman tingkat-sistem ke data. OLE DB pada dasarnya serangkaian antarmuka COM yang mengakses data secara langsung.

Beberapa cara untuk megefisienkan sebuah database, sehingga database yang dihasilkan kapasitasnya akan lebih rendah, yaitu : Lakukan pemilihan tipe data yang tepat dan sesuaikan dengan lebar data, apabila data berupa data teks maka hindarkan pemakaian tipe data number, Database yang telah selesai dibuat kemudian dikompres agar kapasitasnya lebih kecil, sehingga space hardisk tidak terlalu terbebani, Sortir data secara *ascending* atau *descending* agar dalam proses searching data akan cepat ditemukan untuk ditampilkan, Penggunaan filter pada Query, misalnya menggunakan klausa ”Like” atau ”Regexp”, Gunakan keneksi data yang sesuai dengan kebutuhan, misalnya untuk akses data dengan *clien server* (jaringan) gunakan koneksi ODBC sedangkan untuk *stand alone* (berdiri sendiri) gunakan koneksi OLE DB.

Kata Kunci: *ODBC, OLE DB, Akses Data*

A. PENDAHULUAN

Dalam melakukan pengembangan perangkat lunak harus berorientasi kepada pemenuhan kebutuhan user dan dapat dipakai sesuai dengan yang direncanakan, sehinga perangkat lunak tersebut akan berkualitas dan lebih efisien dalam hal penggunaan sumber daya sistem tersebut. Untuk itu dalam melakukan pengembangan software terutama untuk aplikasi yang berorientasi database, seorang programmer harus handal dalam menentukan metode koneksi data dengan software yang akan digunakan sebagai basis data, seperti SQL Server atau Microsoft Access. Ada dua metode akses data yang sering digunakan oleh para programmer, yaitu ***Open Database Connection (ODBC)*** dan ***Object Linking Embedding (OLE DB)***. Masing-masing metode akses data tersebut memiliki kekurangan dan kelebihan, terutama dari segi waktu atau lama akses data dan beban kerja CPU pada saat mengakses data tersebut.

Banyak pengguna komputer pada saat ini menggunakan program database, baik itu *Personal Computer* (komputer pribadi) maupun

Computer Network (jaringan) dalam rangka penerapan sistem komputerisasi, program database digunakan untuk menyelesaikan pekerjaan yang berhubungan dengan pengolahan data. Aplikasi pengolahan database saat ini semakin banyak dan semakin berkembang, dalam hal ini pengguna komputer cenderung memilih program database yang cepat dan mudah dalam pengoperasiannya, cepat dalam arti proses dari pada program tersebut maupun pengaksesan datanya.

Salah satu bagian penting adalah kemampuan untuk memanfaatkan teknologi ODBC dan OLE DB. Hal ini merupakan kunci untuk memecahkan berbagai masalah dalam pengolahan data antar *Database Management Sys*tem *(DBMS)*, *Open Database Connection (ODBC)* merupakan jembatan penghubung antara DBMS dengan Developing Tool, misalnya *Microsoft Access* dengan *Visual Basic*.

Object Linking Embedding (OLE DB) adalah salah satu standart database yang digunakan sebagai alat penghubung untuk menghubungkan data, OLE DB merupakan sebuah antarmuka pemrograman tingkat sistem ke data dan mengakses data secara langsung dalam sumber data relasional dan nonrasional misalnya sebuah file teks, grafis atau spreadsheet.

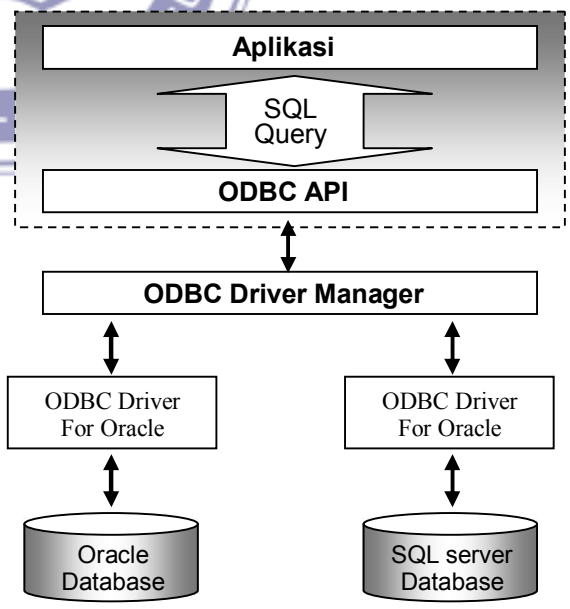
Dari uraian di atas dapat diambil kesimpulan bahwa penentuan metode akses data dalam pengembangan perangkat lunak memiliki peranan yang sangat penting, khususnya untuk perangkat lunak yang berhubungan dengan pengolahan databases dengan kapasitas record yang sangat besar. Yang menjadi permasalahan sekarang ini adalah metode akses data manakah yang lebih cocok digunakan dalam mengakses data untuk mendapatkan akses data yang lebih cepat dengan beban kerja CPU lebih ringan.

B. ANALISA KONEKSI AKSES DATA ODBC

ODBC adalah pendahulu OLE DB dan merupakan salah satu dari upaya pertama untuk menyediakan suatu antarmuka standar ke data, tidak peduli apa flatform database yang ditempatinya. Akses ke data melalui ODBC memerlukan driver yang sesuai bagi suatu sumber data khusus.

ODBC merupakan sebuah antarmuka tingkat rendah ke sumber data pada Visual Basic, karena dapat berkomunikasi langsung dengan driver sehingga koneksi ODBC akan berlangsung cepat namun kompleks dan memiliki sebuah kurva pembelajaran yang tajam. ODBC juga dibatasi pada pemakaian bersama database relational karena hanya merupakan bahasa standar bagi pengiriman permintaan ke sumber data ODBC. Selain itu juga ODBC bukanlah berorientasi obyek. Pengkodean dan debugging lebih sulit menggunakan ODBC dan lemah pada penerjemahan model obyek ke dalam kode akses data yang lebih banyak.

Berikut ini gambar arsitektur ODBC yang melukiskan bagaiman beberapa komponen saling berinteraksi.



Gambar 1. Arsitektur ODBC

API, memanggil fungsi-fungsi ODBC dan menangani koneksi dan diskoneksi dari suatu sumber data serta pengiriman dan penerimaan data.

Driver Manager, menyediakan sebuah daftar sumber data yang tersedia, membuat driver ODBC yang dibutuhkan dan memperantarai permintaan dan hasil antara aplikasi dan driver ODBC.

Driver, memproses semua panggilan fungsi yang dibuat oleh aplikasi dan mengirimkan mereka ke sumber data.

Sumber Data, menyatakan mesin database dan menerima permintaan SQL dari driver, serta mengembalikan hasil data dan pesan.

C. ANALISA KONEKSI AKSES DATA OLE DB

OLE DB adalah komponen kunci dalam strategi Universal Data Access (UDA) Microsoft. OLE DB mendefinisikan serangkaian antarmuka COM yang membungkus layanan akses data. OLE DB adalah sebuah antarmuka pemrograman tingkat-sistem ke data. OLE DB pada dasarnya serangkaian antarmuka COM yang mengakses data secara langsung. OLE DB mengembangkan koneksi di balik ODBC. Sementara ODBC dirancang untuk mengakses data SQL, OLE DB mengakses data dalam sumber relasional dan non relasional, termasuk termasuk database hirarki mainframe, teks, grafis dan spreadsheet.

Bagian ini memberikan suatu penjelasan mengenai OLE DB, namun sebagai seorang pengembang harus mungkin tidak akan menggunakannya secara secara langsung untuk mengakses data. Spesifik OLE DB kompleks dan ADO akan melindungi dari keharusan mempelajari dan menggunakan. Bagaimanapun penting kiranya untuk memperlajari dan memahami bagaimana OLE DB bekerja agar pengguna tahu apa yang akan terjadi di balik layar. Hal ini akan membantu pengembang dalam merancang akses data dan memecahkan masalah yang terjadi.

OLE DB merupakan sebuah API berorientasi panggilan yang menyediakan akses tingkat-rendah ke data dan berdasarkan pada standar COM. Antarmuka OLE DB digunakan sebagai standar untuk mendefinisikan komponen akses data UDA lainnya. OLE DB berisi tiga dari tipe komponen berikut ini:

1. **Penyedia data**, berisi data yang mengeksposnya ke komponen lain. Penyedia data bertanggung-jawab untuk penerjemahan antarmuka OLE DB ke dalam suatu format yang dimengerti oleh sumber data. Contoh sebuah penyedia data adalah OLE DB Provider for SQL Server.
2. **Pemakai data**, menggunakan data yang diekspos oleh penyedia data. Jika aplikasi Visual Basic bersama SQL Server sebagai sumber data, ADO akan menjadi pemakai data.
3. **Service Provider** bekerja sama dengan pemakai data untuk untuk memproses dan mengangkut data. Ini juga disebut sebagai komponen layanan, yang merupakan sebuah tipe dari penyedia layanan. Layanan cursor adalah suatu contoh dari layanan. Layanan ini mendukung dan meningkatkan kemampuan cursor yang ada pada penyedia data. Layanan cursor memungkinkan update batch dan menyediakan properti yang dinamis. Pemilihan cursor yang berbasis-clien dengan sintaks **Recordset.CursorLocation=adUseClient** menjalankan layanan cursor.

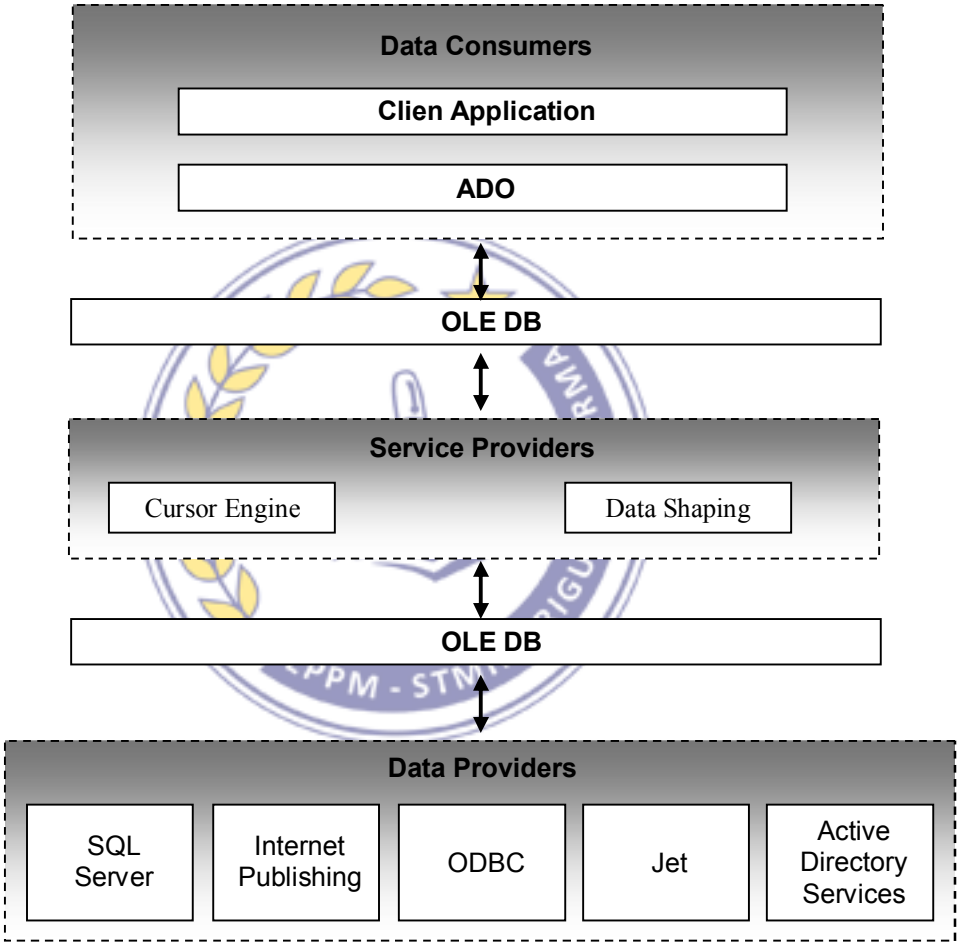
Pada Gambar 2 berikut menggambarkan arsitektur ODBC yang melukiskan bagaimana beberapa komponen saling berinteraksi.

ADO adalah sebuah obyek yang bertempat di atas OLE DB. ADO adalah sebuah API berorientasi-obyek yang menyembunyikan detail membingungkan dari OLE DB. Para pengembang bisa menggunakan metode dan properti yang diekspos oleh obyek ADO untuk mengambil data dan menggunakannya. Obyek-

obyek ini lebih intuitif dan jauh lebih sederhana untuk dipakai dibanding dengan pemrograman OLE DB.

D. EFISIENSI DATABASE

Telah diketahui bahwa kecepatan akses data sangat dipengaruhi oleh penentuan koneksi data baik itu menggunakan ODBC maupun OLE DB. Namun selain itu kecepatan akses data dapat juga dipengaruhi oleh ukuran atau kapasitas database dalam sebuah basis data yang menempati media penyimpan data. Untuk itu perancangan sebuah database harus benar-benar diefisienkan agar kapasitaas yang dihasilkan lebih kecil, sehingga pada saat diakses akan lebih ringan.



Gambar 2. Arsitektur OLE DB

Berikut ini ada beberapa cara untuk megefisienkan sebuah database, sehingga database yang dihasilkan kapasitasnya akan lebih rendah, yaitu :

1. Lakukan pemilihan tipe data yang tepat dan sesuaikan dengan lebar data, apabila data berupa data teks maka hindarkan pemakaian tipe data number.
2. Database yang telah selesai dibuat kemudian dikompres agar kapasitasnya lebih kecil, sehingga space hardisk tidak terlalu terbebani.
3. Sortir data secara *ascending* atau *descending* agar dalam proses searching

data akan cepat ditemukan untuk ditampilkan.
4. Penggunaan filter pada Query, misalnya menggunakan klausa ”Like” atau ”Regexp”.
5. Gunakan keneksi data yang sesuai dengan kebutuhan, misalnya untuk akses data dengan *client server* (jaringan) gunakan koneksi ODBC sedangkan untuk *stand alone* (berdiri sendiri) gunakan koneksi OLE DB.

E. FAKTOR-FAKTOR YANG MEMPENGARUHI KECEPATAN AKSES DATA

Ada beberapa faktor yang mempengaruhi kecepatan akses data, diantarnya :

1. Jumlah program *residence* yang berada di memori, seperti *Service Windows, Anti Virus, Automatic Update*.
2. *Front Side Bus*, merupakan lebar jalur yang dapat dilihat di *Motherboard* sesuai dengan spesifikasinya.
3. Besar database yang tersimpan di dalam hardisk akan mempengaruhi kecepatan akses data.
4. Kapasitas memori yang terpasang di komputer, semakin tinggi kapasitas momori berarti akan semakin cepat dalam memproses akses data.
5. RPM (*Rotation Per Minute*), semakin cepat putaran hardisk, maka akan semakin cepat dan banyak data dapat diakses.

F. SIMPULAN

Beberapa cara untuk megefisienkan sebuah database, sehingga database yang dihasilkan kapasitasnya akan lebih rendah, yaitu :

1. Lakukan pemilihan tipe data yang tepat dan sesuaikan dengan lebar data, apabila data berupa data teks maka hindarkan pemakaian tipe data number.
2. Database yang telah selesai dibuat kemudian dikompres agar kapasitasnya lebih kecil, sehingga space hardisk tidak terlalu terbebani.
3. Sortir data secara *ascending* atau *descending* agar dalam proses searching data akan cepat ditemukan untuk ditampilkan.
4. Penggunaan filter pada Query, misalnya menggunakan klausa ”Like” atau ”Regexp”. Gunakan keneksi data yang sesuai dengan kebutuhan, misalnya untuk akses data dengan *clien server* (jaringan) gunakan koneksi ODBC sedangkan untuk *stand alone* (berdiri sendiri) gunakan koneksi OLE DB.

G. DAFTAR PUSTAKA

Kristiono, Privida. 2008. *Pemrogramman Database Tingkat Lanjut dengan VB 6.* Jakarta: PT. Elex Media Komputindo.

Madcoms. 2003. *Aplikasi Database Visual Basic 6.0 dengan Crystal Reports.* Yogyakarta: CV. Andi Offset.

Manfield, Richard. 2004. *Visual Basic. Net*. Jakarta: PT. Elex Media Komputindo.

Marlissa, Agung. 2005. *Pemrogramman Database Mahir Berbasis Access.* Surabaya: Penerbit INDAH.

Online Training Solution, Inc. 2003. *Microsoft Access Version 2003 Step by Step.* Jakarta: PT. Elex Media Komputindo.

Siebold, Dianne. 2003. *Visual Basic Developer's Guide to SQL Server*. Jakarta: PT. Elex Media Komputindo.

Suarna, Nana. 2003. *Pedoman Panduan Praktis Microsoft Access 2002.* Bandung: CV. Yrama Widya.

Yung, Kok. 2002. *Membangun Database dengan Visual Basic dan Perintah SQL*. Jakarta: PT. Elex Media Komputindo.