

# **ActiveX Data Object (ADO) dengan OLE DB sebagai pengganti Data Access Object (DAO) dengan ODBC dan JET dalam Microsoft Visual Basic 6**

Miczard L Riupassa – L2F 098 638 - [ambon54@hotmail.com](mailto:ambon54@hotmail.com)

Pembimbing I – Ir. Kodrat IS, MT

Pembimbing II – Agung Budi P, ST.MIT

## **Abstrak**

*Pemrograman mengenai basisdata dengan bahasa program Microsoft Visual Basic 6 selalu menggunakan akses data dan ada 2 akses data yang biasa digunakan, yaitu DAO (Data Access Object) dan ADO (ActiveX Data Object). ADO bersama dengan OLE DB adalah komponen kunci dalam Universal Data Access (UDA) Microsoft. Kelebihan utama ADO dengan DAO adalah memberikan kemudahan bagi semua jenis kebutuhan akses data dan mampu mereduksi penggunaan memori koneksi sehingga cocok dalam jaringan yang memiliki lalu lintas data yang besar dan transaksi yang tinggi.*

*Tugas akhir ini menggunakan obyek Connection dan obyek Recordset dari ADO versi 2.7 dan obyek Database dan obyek Recordset dari DAO versi 3.6 dengan bahasa program Microsoft Visual Basic 6. Sistem operasinya yang digunakan adalah Microsoft Windows 2000 Advance Server dan basisdata adalah Microsoft SQL Server 7 dan Microsoft Access. Pembahasan mengenai ADO dimulai dari hierarki model objek, perbedaannya dengan DAO, yaitu koneksi dan sekuritas, hierarki obyek, koneksi asinkron, data binding, Flex Grid dan pengaruh koneksi terhadap basisdata server (SQL Server 7). Dengan pembahasan ini, penyusun mengharapkan kepada pengembang program aplikasi perangkat lunak basisdata Microsoft Visual Basic 6 mendapatkan referensi yang tepat untuk menggunakan atau beralih ke akses data ADO.*

## **I. Pendahuluan**

### **1.1. Latar Belakang**

Akses data adalah salah satu komponen yang penting dalam pengembangan aplikasi perangkat lunak basisdata. Berbagai macam perusahaan menciptakan teknologi akses data dalam perangkat lunak yang intinya adalah mengambil dan mempergunakan data tersebut secara efektif dan efisien.

Microsoft adalah salah satu perusahaan yang mengeluarkan teknologi baru dalam akses data yaitu ActiveX Data Object (ADO). Pada awalnya, ADO diciptakan sebagai pengembangan teknologi akses data sebelumnya, yaitu DAO (Data Access Object). Selain itu ADO mengacu pada paradigma pengaksesan data Microsoft yang terakhir yang disebut OLE for Database (OLE DB), yaitu dengan menggunakan beberapa API (Application Programming Interface) dari setiap jenis basisdata yang digunakan untuk menghubungkan ke akses tingkat rendah basisdata tersebut. OLE DB juga memiliki standar berdasarkan dengan standar COM (Component Object Model)<sup>[9]</sup>.

ADO bersama dengan OLE DB menjadi komponen kunci dalam strategi UDA (Universal Data Access) Microsoft karena OLE DB mampu menyediakan akses ke berbagai jenis sumber data baru seperti: sistem *e-mail*, format grafik atau sumber dari internet lainnya, selain itu ADO

membutuhkan memori yang sedikit daripada DAO, sehingga lebih cocok untuk sistem jaringan komputer yang memiliki lalu lintas data yang besar dan transaksi yang tinggi<sup>[12]</sup>.

### **1.2. Maksud dan Tujuan**

Tujuan dari tugas akhir ini membahas performansi ADO sebagai suatu akses data untuk memanipulasi informasi pada basisdata menggantikan akses data DAO dan menunjukkan perbedaan antara akses data ADO dengan akses data DAO melalui program-program yang akan dibuat, sehingga pengembang program aplikasi perangkat lunak basisdata Microsoft Visual Basic 6 mendapatkan referensi yang tepat untuk menggunakan atau beralih ke akses data ADO untuk mendapatkan hasil manipulasi data yang efektif dan efisien.

### **1.3. Pembatasan Masalah**

Adapun Ruang lingkup pembuatan tugas akhir ini adalah sebagai berikut:

1. ADO yang digunakan adalah versi 2.7 dan DAO yang digunakan adalah versi 3.6.
2. Obyek ADO yang digunakan adalah obyek Connection dan obyek Recordset.
3. Obyek DAO yang digunakan adalah obyek Database dan obyek Recordset.

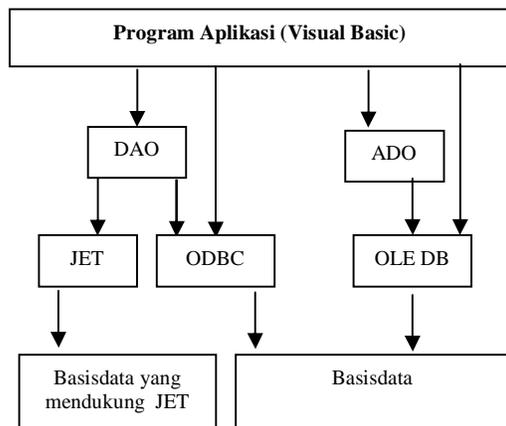
4. Pembahasan mengenai jumlah komponen Data Binding ADO dan Data Binding DAO.
5. Pembahasan mengenai perbandingan performansi ADO dengan DAO sebagai data akses.
6. Sistem operasi yang digunakan adalah Microsoft Windows 2000 Advance Server.
7. Basisdata yang digunakan adalah Microsoft SQL Server 7 sebagai salah satu contoh jenis basisdata server dan Microsoft Access 2000 sebagai salah satu contoh jenis basisdata lokal.
8. Kelebihan fitur ADO dibandingkan DAO dan sebaliknya tidak dibahas dalam tugas akhir ini.

## II. Dasar Teori

### 2.1. DAO dan ADO sebagai akses data

#### 2.1.1. Akses data

Akses data adalah obyek yang berfungsi untuk menghubungkan antara program aplikasi perangkat lunak basisdata dengan basisdata yang akan digunakan. Oleh karena itu, akses data merupakan obyek yang mutlak digunakan dalam membuat suatu program aplikasi perangkat lunak basisdata. Contoh dari akses data adalah DAO dan ADO. Gambar 2.1 menunjukkan mekanisme DAO dan ADO dalam mengakses basisdata.



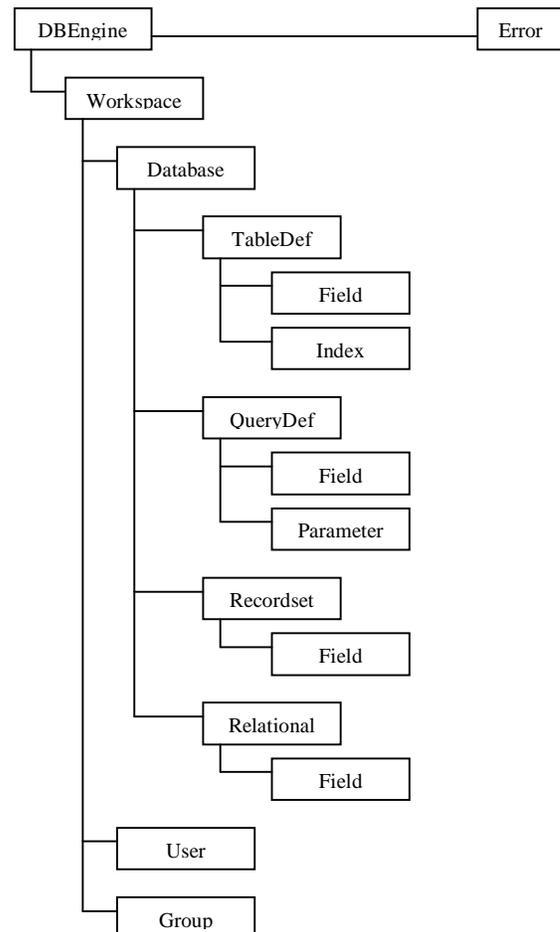
Gambar 2.1. Mekanisme DAO dan ADO dalam mengakses basisdata

#### 2.1.2 DAO (Data Access Object)

DAO adalah perintis akses data. Pada awalnya DAO hanya menggunakan provider JET untuk mengambil data dari basisdata Microsoft Access (file-file berekstensi .mdb) atau dari basisdata lainnya seperti ISAM, FoxPro dan dBase lalu selanjutnya DAO berkembang sebagai akses data yang dapat berhubungan dengan ODBC.

DAO sebagai akses data memiliki keterbatasan dalam mengakses basisdata yaitu, hanya dapat mengakses basisdata ber-provider JET atau yang memiliki driver ODBC. Hal ini membatasi pengaksesan terhadap jenis basisdata baru. (Contohnya adalah basisdata Oracle versi-versi yang baru). Selain itu, keterbatasan DAO adalah hubungan koneksi secara asinkron pada DAO hanya dapat dibuat jika menggunakan hubungan dengan provider ODBC saja.<sup>[12]</sup>

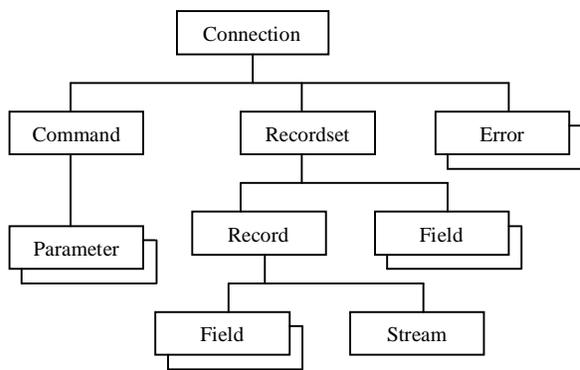
Jumlah obyek DAO adalah 16 buah dan obyek DAO memiliki sifat hierarki, sehingga dari kedua alasan tersebut membuat DAO kurang efektif pada setiap penggunaannya. Gambar 2.2 menunjukkan model obyek-obyek dari DAO dan menunjukkan tingkatan obyek-obyek DAO. Langkah pertama yang harus dilakukan jika menggunakan akses data DAO adalah membuat dahulu obyek DBEngine (tingkat paling atas) kemudian membuat Workspace (tingkat di bawahnya) dan seterusnya sampai ke tingkat paling bawah. Sebaliknya, ADO memungkinkan membuka sebuah obyek Recordset langsung tanpa secara eksplisit membuat obyek Connection.<sup>[9]</sup>



Gambar 2.2. Hierarki dari obyek DAO

### 2.1.3. ADO (ActiveX Data Object)

ADO adalah akses data yang bertempat di atas OLE DB seperti yang ditunjukkan pada Gambar 2.3 pada lampiran. ADO bersifat berorientasi pada obyek karena ADO menyembunyikan kerumitan OLE DB dalam mengakses sebuah basisdata. ADO terdiri dari obyek yang fleksibel karena obyek tersebut terdiri dari properti, metode dan event sendiri. Jadi, penggunaan properti, event dan metode yang ada pada obyek-obyek ADO jauh lebih mudah daripada harus memanggil OLE DB yang akan digunakan.



Gambar 2.4. Hierarki dari obyek ADO

Gambar 2.4 menunjukkan hierarki obyek ADO dan obyek ADO yang ditunjukkan dengan kotak ganda berarti obyek tersebut memiliki koleksi informasi. Model hierarki obyek ADO lebih datar jika dibandingkan dengan model hierarki obyek DAO. Walaupun ADO ditampakan sebagai sebuah hierarki dengan sebuah obyek tunggal di atasnya, namun ada obyek tersebut bisa berdiri sendiri-sendiri (independen) seperti obyek Recordset.

## 2.2. Obyek-obyek koneksi basisdata DAO dan ADO

### 2.2.1. Penggunaan obyek-obyek DAO untuk mengakses basisdata

Pembuatan koneksi ke basisdata menggunakan DAO dapat dilakukan dengan menggunakan dua provider, yaitu provider JET dan provider ODBC. Seperti penjelasan sebelumnya yang menerangkan tentang hierarki obyek DAO yang memiliki banyak tingkatan maka, langkah pertama yang harus dilakukan jika menggunakan DAO adalah membuat dahulu obyek DBEngine (tingkat paling atas) kemudian membuat obyek Workspace (tingkat di bawahnya) lalu

membuat obyek Database dan terakhir adalah membuat obyek Recordset. Hal ini membuat penggunaannya DAO lebih rumit karena penggunaan suatu obyek harus didahului oleh obyek yang berada di atasnya.

Namun dua obyek paling atas dalam hierarki DAO, yaitu obyek DBEngine dan obyek Workspace memiliki nilai default yang jika tidak dideklarasikan maka menggunakan provider JET, karena DAO diciptakan untuk memaksimalkan fitur dari provider JET bukan untuk provider ODBC.

### 2.2.2. Penggunaan obyek-obyek ADO untuk mengakses basisdata

Setiap aplikasi perangkat lunak basisdata selalu melakukan 3 buah proses yang berurutan, yaitu:<sup>[7]</sup>

1. Membuat hubungan ke basisdata.
2. Menjalankan perintah-perintah manipulasi data ke basisdata.
3. Mengambil informasi dari basisdata

Obyek Connection, obyek Recordset dan obyek Command adalah obyek ADO yang mempresentasikan suatu hubungan ke basisdata, namun dalam tugas akhir ini dibatasi hanya obyek Connection dan obyek Recordset.

#### 2.2.2.1. Obyek Connection

Obyek Connection digunakan untuk membuka koneksi dan menjaga koneksi tersebut ke sumber data seperti Microsoft SQL Server 7. Microsoft Visual Basic 6 mempunyai berbagai cara yang berbeda untuk membuka koneksi ke sumber data, yaitu:<sup>[3]</sup>

- a. Data Environment.
- b. Data View Window.
- c. ADO Data Control.
- d. Obyek.Connection ADO

Dari keempat cara tersebut yang mendukung koneksi hubungan dengan akses data DAO hanya Data Environment dan Data View Window. Setiap obyek dalam akses data ADO berhubungan dengan obyek Connection ini (gambar hierarki obyek ADO pada Gambar 2.4), karena setelah obyek Connection ini membuat koneksi ke sumber data maka obyek lain dapat melakukan tugasnya masing-masing.

#### 2.2.2.2. Obyek Recordset

Obyek Recordset berfungsi sebagai penyusun record dari sumber data. Obyek Recordset ADO bersifat fleksibel karena properti source yang memiliki 7 macam argumen untuk mendapatkan record, yaitu:

1. Obyek Command yang diberikan ke Obyek Recordset
2. Pernyataan SQL
3. Nama tabel
4. Nama prosedur tersimpan
5. URL yang diakses
6. Nama file yang dari Recordset yang dipertahankan.
7. Nama obyek Stream dari Recordset yang dipertahankan.

### III. Perbandingan antara ADO dengan DAO

#### 3.1. Koneksi dan Sekuritas

##### 3.1.1. Koneksi dan Sekuritas ADO

Koneksi dan sekuritas pada ADO selalu menggunakan string koneksi. String koneksi ini menampilkan informasi komponen-komponen yang diperlukan untuk membuat koneksi ke suatu sumber data.

Koneksi pada ADO tergantung pada provider yang disediakan oleh OLE DB. Selain itu ADO dapat berkoneksi langsung dengan suatu alamat URL dan jenis koneksi ini tidak dimiliki oleh DAO. Penggunaan ADO untuk koneksi ke basisdata SQL Server 7 lebih cepat dibandingkan koneksi DAO karena ADO dapat langsung berhubungan dengan ke SQL Server 7 melalui provider OLE DB untuk SQL Server lalu, sedangkan DAO harus melewati prosedur penggunaan provider ODBC. Gambar mekanisme ADO dalam mengakses data SQL Server ditunjukkan pada Gambar 2.3 pada lampiran.

ADO memiliki perbedaan provider untuk basisdata ber-provider JET seperti Microsoft Access. *User Id* yang berisi nama pemakai dan *pwd* yang berisi password dari pemakai tersebut yang disertakan pada string koneksi tidak bisa membuka password pada basisdata seperti pada basisdata lainnya melainkan harus menggunakan tambahan string pada string koneksi, yaitu *JET OLEDB:Database Password=password*.

##### 3.1.2. Koneksi dan Sekuritas DAO

Koneksi DAO ke basisdata SQL Server 7 harus menggunakan provider ODBC. Koneksi ke SQL Server dengan menggunakan provider ODBC memiliki kelemahan, yaitu akses ke basisdata lambat. Koneksi DAO dengan menggunakan provider ODBC memerlukan selang waktu tertentu untuk koneksi yang tergantung dari keadaan komputer server-nya dan apabila koneksi itu dipaksakan untuk dijalankan maka akan memberikan sebuah kotak pesan kesalahan. Pesan kesalahan tersebut isinya adalah sebuah proses lain yaitu proses perhubungan antara ODBC dengan

basisdata sedang dilakukan sehingga pengeksekusian program tidak dapat dijalankan.

Sekuritas DAO dalam menggunakan provider JET tidak dibedakan dengan menggunakan provider ODBC seperti pada ADO.

#### 3.2. Hierarki Obyek

##### 3.2.1. Hierarki Obyek ADO

ADO memiliki hierarki yang lebih datar dibandingkan dengan hierarki obyek akses data DAO dan ada obyek tersebut yang bisa berdiri sendiri, yaitu obyek Recordset dan obyek Command. Kelebihan hierarki obyek ADO ini memberikan kemudahan bagi setiap pengembang dalam membuat aplikasi programnya.

##### 3.2.2. Hierarki Obyek DAO

Hierarki Obyek DAO mempunyai banyak tingkatan. Hal ini membuat penggunaannya DAO lebih rumit karena penggunaan suatu obyek harus didahului oleh obyek yang berada di atasnya. Namun dua obyek paling atas dalam hierarki DAO yaitu obyek DBaseEngine dan obyek Workspace memiliki nilai default yang jika tidak dideklarasikan maka menggunakan provider JET karena akses data DAO lebih mendukung fitur dari provider JET bukan untuk provider ODBC.

#### 3.3. Koneksi asinkron

##### 3.3.1. Koneksi asinkron ADO

Koneksi asinkron dalam program aplikasi basisdata yang kompleks sangat diperlukan karena pada koneksi asinkron memperbolehkan beberapa *query* dapat dilayani dalam waktu yang bersamaan. Obyek Connection adalah obyek yang menentukan sebuah jenis koneksi termasuk koneksi asinkron atau koneksi sinkron dan koneksi asinkron ADO dapat digunakan untuk setiap provider OLE DB yang ada. Apabila obyek Connection tidak mendeklarasikan jenis koneksi yang akan digunakan maka menggunakan nilai default-nya yaitu koneksi sinkron.

##### 3.3.2. Koneksi asinkron DAO

Kegunaan koneksi asinkron DAO sama dengan koneksi asinkron ADO namun yang membedakannya adalah koneksi asinkron DAO pemakaiannya terbatas hanya bisa digunakan jika menggunakan koneksi dengan provider ODBC saja. Selain itu perbedaannya adalah pembuatan koneksi asinkron DAO ditentukan pada obyek Recordset tidak pada obyek Database karena obyek Database hampir dapat disamakan dengan obyek Connection pada ADO. Kerugian jika menggunakan hubungan asinkron dengan DAO adalah penggunaan koneksi dengan provider ODBC yang

aksesnya lambat, sehingga dapat mempengaruhi kinerja dari program aplikasi yang dibuat.

### **3.4. Data binding**

#### **3.4.1. Data binding ADO**

Data binding adalah kumpulan dari beberapa komponen yang dapat berhubungan dengan suatu basisdata. ADO memiliki komponen data binding yang terintegrasi keseluruhan komponennya dengan mudah. Komponen dari data binding itu terdiri dari:

- a. ADO Data Control 6.0
- b. Chart Control 6.0
- c. DataGrid Control 6.0
- d. DataList Control 6.0 (terdiri dari DataList dan DataCombo)
- e. DataRepeater Control 6.0
- f. Hierarchical FlexGrid Control 6.0

ADO Data Control 6.0 merupakan komponen utama karena dalam komponen ini memiliki 2 properti yang tidak dimiliki oleh komponen lain yaitu properti ConnectionString yang fungsinya membuat koneksi ke sumber data dan properti RecordSource yang fungsinya mengambil atau menunjukkan data-data dari tabel yang akan dimanipulasi. Kelima komponen data binding lain mengambil data dari ADO Data Control 6.0 dengan mengisi properti DataSource dengan nama dari ADO Data Control 6.0.

#### **3.4.2. Data binding DAO**

Komponen data binding DAO memiliki jumlah lebih sedikit dibandingkan dengan ADO, yaitu 4 buah. Keempat komponen itu adalah:

- a. Data Control
- b. Data Bound Grid Control 5.0
- c. Data Bound List Control 6.0 (terdiri dari DBList dan DBCombo)
- d. FlexGrid Control 6.0

Data Control sama seperti ADO Data Control yaitu komponen yang utama karena memiliki properti DatabaseName yang fungsinya menunjuk nama file dari basisdata dan properti RecordSource yang fungsinya untuk menunjukkan nama tabel yang akan digunakan. Ketiga komponen data binding yang lain terintegrasi dengan Data Control untuk mendapatkan data dengan mengisi properti DataSource dengan nama dari Data Control.

Pembuatan grafik pada DAO dapat mempergunakan obyek dari Excel dengan cara memanggil obyek Microsoft Excel Chart. Pembuatan grafik dengan obyek ini tidak mempunyai hubungan langsung dengan Data Control seperti pada ADO Chart Control 6.0. Jadi,

masing-masing data harus dimasukkan satu per satu kemudian dibuat bentuk grafiknya.

### **3.5. Flex Grid**

#### **3.5.1. Flex Grid pada ADO**

Salah satu komponen dari data binding adalah Flex Grid. Flex Grid adalah salah satu obyek yang berfungsi hanya untuk menampilkan isi data secara baris dan kolom dari data yang ada, jadi data yang ditampilkan pada Flex Grid tidak bisa dimanipulasi (read-only) seperti halnya Data Grid. Flex Grid pada ADO namanya adalah Hierarchical FlexGrid Control 6.0 dan sesuai namanya Flex Grid ini memiliki kemampuan untuk menampilkan data secara hierarki. Menampilkan data secara hierarki maksudnya adalah satu atau lebih field menjadi kunci terhadap field yang lain karena ada salah satu field yang dihubungkan dengan field yang sejenis yang berasal dari tabel yang sama atau dari tabel yang berbeda. Kemampuan ADO untuk mengelompokkan data ini dinamakan *Data Shaping*. Syarat yang harus dilakukan apabila akan menggunakan fasilitas data shaping ini adalah menggunakan provider MSDDataShape yang dideklarasikan pada string koneksi.

#### **3.5.2. Flex Grid pada DAO**

Sama seperti komponen data binding ADO, DAO pun mempunyai Flex Grid yaitu FlexGrid Control 6.0. Perbedaan antara kedua jenis Flex Grid ini adalah FlexGrid Control 6.0 tidak memiliki kemampuan untuk menampilkan data secara hierarki seperti yang dapat dilakukan oleh Hierarchical FlexGrid Control 6.0 sehingga FlexGrid Control 6.0 hanya menampilkan data seperti yang ditampilkan oleh Data Grid saja.

### **3.6. Pengaruh koneksi terhadap basisdata server (SQL Server 7)**

Pada bagian ini akan membahas pengaruh pembuatan perangkat lunak dengan menggunakan akses data DAO dan ADO terhadap SQL Server 7. Pengaruh terhadap SQL Server 7 yang akan diuji adalah memori yang digunakan untuk membuat dan menjaga koneksi dan jumlah *query* yang diterima oleh SQL Server 7 per satuan waktu (detik). Pembahasan mengenai pengaruh ini akan dibantu oleh program tambahan dari SQL Server 7 yaitu Performance Monitor yang fungsinya adalah memonitor keadaan yang terjadi ketika program aplikasi dijalankan lalu menampilkannya dalam bentuk grafik.

Performance Monitor adalah perangkat lunak yang memiliki kemampuan untuk memonitor berbagai macam mekanisme kerja suatu perangkat

lunak atau perangkat keras dan yang digunakan dalam pengujian kedua macam koneksi ini adalah Connection Memory dan Batch Request. Definisi dari Connection Memory adalah total memori dinamis dari server yang sedang digunakan untuk membuat dan mempertahankan koneksi sedangkan definisi dari Batch Request adalah jumlah *query* yang diterima oleh server basisdata.

Pengujian performa ini dengan membuat dua buah program, yang pertama adalah program koneksi dengan provider ODBC menggunakan DAO dan yang kedua adalah program koneksi dengan provider ODBC menggunakan ADO. Kedua program ini dibuat berbeda menurut masing-masing akses datanya namun agar tidak mempengaruhi hasil data yang diperoleh dari Performance Monitor maka dibuat seidentik mungkin dan keidentikkan keduanya adalah menggunakan koneksi ODBC ke SQL Server 7, mengakses DSN ZipCode, menggunakan basisdata ZipCode.mdf, *query* yang diberikan sama, besar dan isi dari form windows pun sama.

Contoh tampilan dari pemantauan Connection Memory dan Batch Request ditunjukkan pada Gambar 4.1 pada lampiran. Pada gambar tersebut terlihat hasil dari pemantauan adalah grafik dan 5 buah nilai hasil koneksi namun yang digunakan sebagai bahan analisis yang digunakan hanya 2 nilai hasil saja yaitu minimum, maksimum. Setelah program tersebut dijalankan satu persatu maka hasil dari pemantauan yang dilakukan oleh Connection Memory ditunjukkan pada Tabel 4.1 dan hasil dari pemantauan yang dilakukan oleh Batch Request ditunjukkan pada Tabel 4.2

Keadaan minimum pada kedua tabel tersebut adalah nilai yang paling rendah dari hasil pemantauan sedangkan keadaan maksimum adalah nilai paling besar dari hasil pemantauan. Dalam pencatatan hasil antara tahap satu ke tahap berikutnya selalu didahului oleh pembersihan tampilan layar grafik yang juga menyebabkan pengulangan pemantauan dari awal kembali. Hal ini dilakukan agar hasil pemantauan dari tahap sebelumnya tidak mempengaruhi pencatatan hasil yang sedang dilakukan.

Pada Tabel 4.1 dapat dianalisa bahwa besar memori koneksi minimum pada program koneksi dengan DAO sama besar dengan program koneksi dengan ADO, yaitu besarnya memori koneksi minimum pada saat SQL Server Service di aktifkan karena kedua program tersebut menggunakan basisdata SQL Server 7 yang telah diaktifkan sebelumnya oleh SQL Server Service. Besarnya memori koneksi maksimum pada program koneksi dengan DAO menunjukkan hasil besar daripada program koneksi dengan ADO

sehingga ini membuktikan bahwa ADO adalah solusi akses data yang tepat untuk pembuatan perangkat lunak sistem basisdata yang memiliki lalu lintas data yang besar dan transaksi yang tinggi karena dapat menghemat penggunaan memori koneksi dari server.

**Tabel 4.1.** Tabel hasil pemantauan Connection Memory

Keadaan	Minimum (byte)	Maksimum (byte)
Pada saat SQL Server Service dimatikan	0	0
SQL Server Service di aktifkan	72	72
Menjalankan program koneksi dengan DAO	72	94
Menjalankan program koneksi dengan ADO	72	112

Pada Tabel 4.2 dapat dianalisa bahwa besar permintaan (*request*) minimum pada program koneksi dengan DAO sama besar dengan program koneksi dengan ADO yaitu nol permintaan tiap detik, karena kedua program yang dijalankan tersebut tidak meminta *query* secara terus-menerus tiap detik jadi, setelah selang beberapa detik hasilnya adalah nol sedangkan Performance Monitor selalu memperbaruinya hasil pemantauannya tiap detik. Besarnya permintaan maksimum pada kedua program tersebut menunjukkan hasil yang sangat kontras antara koneksi menggunakan DAO dengan koneksi menggunakan ADO, yaitu ADO memiliki jumlah lebih kecil hingga 2,74% dari jumlah DAO. Hasil ini membuktikan bahwa ADO mempunyai kemampuan untuk meminimalkan jumlah permintaan yang dikirimkan ke server sehingga dapat menghemat sumber daya server dan jaringan komputer yang digunakan.

**Tabel 4.2.** Tabel hasil pemantauan Batch Request

Keadaan	Minimum (per detik)	Maksimum (per detik)
Pada saat SQL Server Service dimatikan	0	0
SQL Server Service di aktifkan	0	0
Menjalankan program koneksi dengan DAO	0	48,927
Menjalankan program koneksi dengan ADO	0	1,009

## IV. Kesimpulan dan Saran

### 4.1. Kesimpulan

Adapun kesimpulan yang dapat ditarik dari hasil analisa sebagai berikut:

1. Koneksi dan sekuritas ADO menggunakan string koneksi sedangkan DAO menggunakan metode OpenDatabase.
2. ADO dapat berkoneksi langsung dengan suatu alamat URL.
3. Koneksi ADO ke basisdata SQL Server 7 lebih cepat dibandingkan koneksi DAO karena ADO dapat langsung berhubungan dengan ke SQL Server 7 melalui provider OLE DB untuk SQL Server, sedangkan DAO harus melewati prosedur penggunaan provider ODBC.
4. Hierarki obyek ADO yang lebih datar dibandingkan dengan hierarki obyek DAO sehingga, pemograman menggunakan ADO lebih sederhana dibandingkan dengan menggunakan DAO.
5. Koneksi asinkron ADO dapat dilakukan setiap provider OLE DB sedangkan DAO hanya hanya bisa dilakukan apabila menggunakan provider ODBC saja (provider JET tidak mendukung koneksi asinkron).
6. Komponen ADO Data Binding berjumlah 6 buah sedangkan DAO Data Binding hanya berjumlah 4 buah.
7. ADO Data Control 6.0 merupakan komponen utama dalam ADO Data Binding sedangkan Data Control merupakan komponen utama dalam DAO Data Binding.
8. Flex Grid ADO (Hierarchical FlexGrid Control 6.0) memiliki kemampuan untuk menampilkan data secara hierarki.
9. ADO mempunyai kemampuan untuk menghemat penggunaan memori koneksi dan mengatur agar jumlah permintaan yang dikirimkan ke server sekecil mungkin sehingga dapat menghemat sumber daya server dan jaringan komputer yang digunakan.
10. Penggunaan ADO sangat bergantung pada ketersediaan provider OLE DB sedangkan penggunaan DAO bergantung pada ketersediaan provider ODBC.
3. Penjelasan mengenai cara pemakaian dan kegunaan ADO Data Binding dan DAO Data Binding tidak diberikan secara mendetail.
4. Pembahasan selanjutnya adalah perbedaan antara ADO dengan ADO.NET.

#### Daftar Pustaka

1. Alam J Agus M, **“Manajemen Database dengan Microsoft Visual Basic 6”**, Elex Media Komputindo, Jakarta, 2000.
2. Callahan, Evan, **“Microsoft Access 2000 Visual Basic for Application”**, Elex Media Komputindo, Jakarta, 2000.
3. Gunderloy, Mike, **“Visual Basic Developer’s Guide To ADO”**, Sybex, 1999.
4. Halvorson, Michael, **“Microsoft Visual Basic 6.0 Step By Step”**, Microsoft Press, Redmond US, 1998.
5. Jung, David dan Kent, Jeff, **“Visual Basic Annotated Archives”**, Osborne, 1999.
6. Romalho, Jose, **“SQL Server 7”**, Elex Media Komputindo, Jakarta, 2000.
7. Petroustos, Evangelos, **“Menguasai Pemograman Database dengan Visual Basic 6 - Buku 1”**, Elex Media Komputindo, Jakarta, 2002.
8. Petroustos, Evangelos, **“Menguasai Pemograman Database dengan Visual Basic 6 - Buku 2”**, Elex Media Komputindo, Jakarta, 2002.
9. Siebold, Dianne, **“Visual Basic Developer’s Guide To SQL Server”**, Sybex, 2001.
10. MSDN Visual Studio 6.
11. SQL Server 7 Books Online.
12. [www.microsoft.com/data](http://www.microsoft.com/data)
13. [www.planetsourcecode.com](http://www.planetsourcecode.com)
14. [www.sql-server-performance.com](http://www.sql-server-performance.com)
15. [www.sybex.com](http://www.sybex.com)
16. [www.vbcity.com](http://www.vbcity.com)

Disusun oleh Miczard L Riupassa – L2F 098 638  
Mahasiswa Jurusan Teknik Elektro Universitas  
Diponegoro.

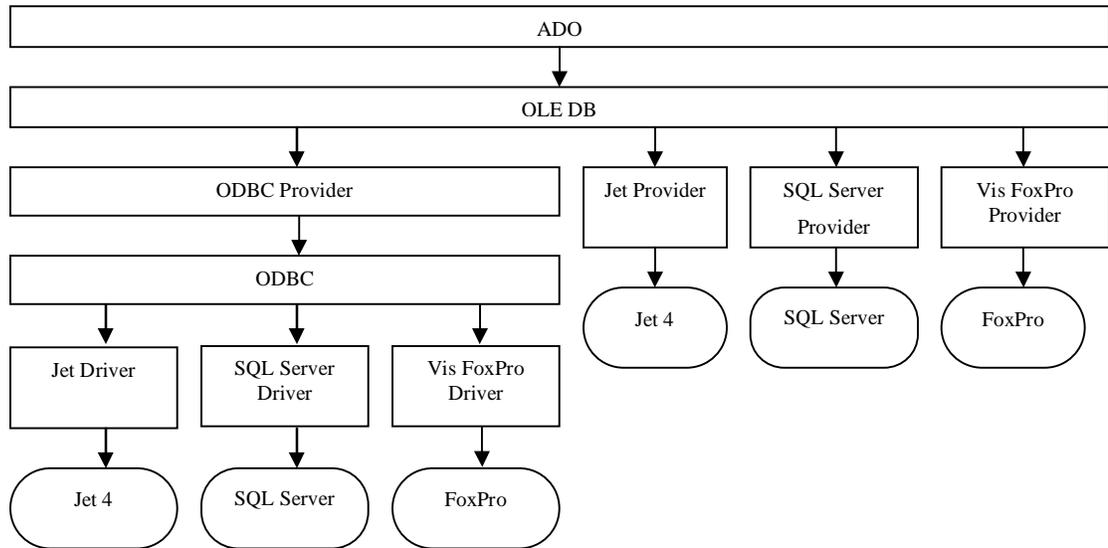
#### 4.2. Saran

1. Pembahasan mengenai perbedaan antara ADO dengan DAO didahului dengan pembahasan mengenai perbedaan antara OLE DB dengan ODBC.
2. Pengujian performance ADO dan DAO akan mendapatkan hasil yang lebih tepat jika dilakukan pada jaringan komputer sebenarnya (dalam LAN atau WAN).

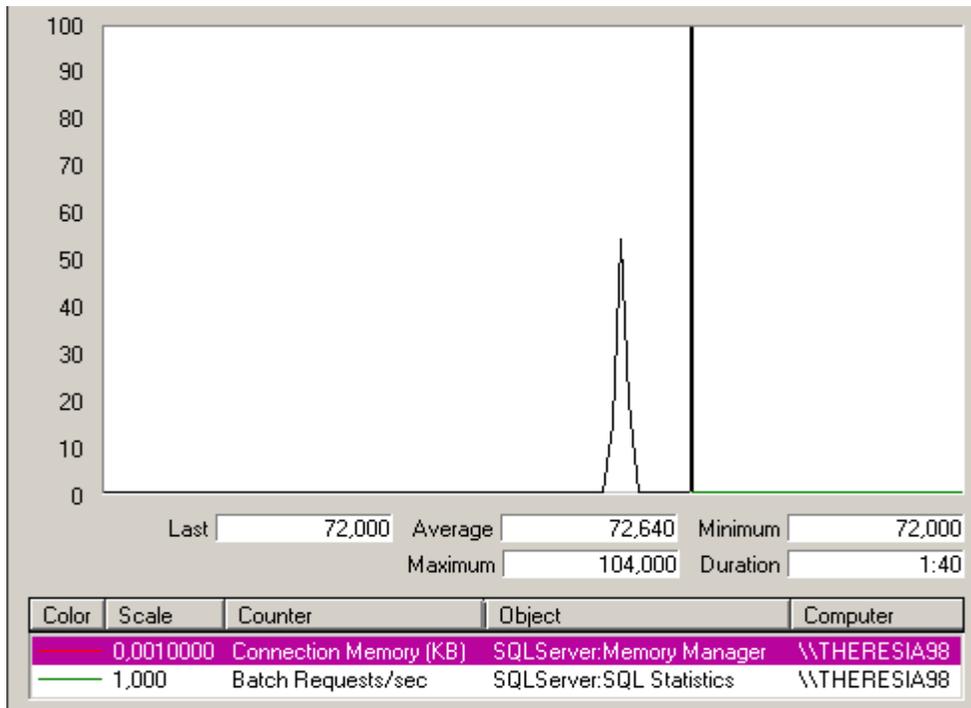
Menyetujui  
Dosen Pembimbing

Agung Budi P, ST  
NIP. 132 137 932

## LAMPIRAN



Gambar 2.3. Posisi ADO dalam mengakses basisdata



Gambar 4.1. Contoh tampilan dari pemantauan Connection Memory dan Batch Request.