MEMBANUN APLIKASI CLIENT-SERVER DENGAN *DISTRIBUTED COMPONENT OBJECT MODEL* (DCOM)

Abstraksi

Teknologi client server muncul untuk menjawab semakin besarnya jumlah masalah dan data yang harus diselesaikan. Client-server mengoptimalkan jaringan dan resource komputer yang ada. Teknologi client-server membagi layer menjadi tiga yaitu; *Front-End Component, Back-End Component* dan *Database Component*. Masing-masing component memiliki tugas dan hak yang berbeda-beda. Pembagian komponen ini juga dapat menambah keamanan terhadap data kita, karena user tidak dapat langsung berhubungan langsung ke komponen database. DCOM *(Distributed Component Object Model)* merupakan pengembangan teknologi dari Component Object Model (COM). Pada COM kita melihat bagaimana suatu komponen client saling berinteraksi. Interaksi ini dapat didefinisikan sebagai hubungan secara langsung antara komponen (COM Server) dan COM Client. DCOM memungkinkan membuat aplikasi kita terbagi menjadi beberapa layer.

Keyword : *client-server, component, dcom*

1. Pendahuluan

Latar belakang pemrograman berbasis *client-server* tidak lepas dari sejarah perkembangan teknologi komputer dan kebutuhan yang muncul pada perusahaan besar *(multinasional),* militer, atau perguruan tinggi untuk memperoleh atau berbagi informasi antar bagian.

Teknologi *LAN (Local Area Network)*, dan sistem operasi yang mundukung jaringan membantu perkembangan dalam pemrosesan data. Dari sini muncul permasalahan yaitu dengan banyaknya lalu lintas antar komputer sehinga dapat mengurangi kenerja sistem.

Untuk mengatasi hal tersebut dikembangkan sistem *client-server* yang berbasis pada aturan bahwa komputer server hanya akan mengirimkan data yang dibutuhkan oleh *client* dimana proses penyimpanan data dilakukan pada komputer database server.

Dengan aplikasi *Client-Server* dimngkinkan untuk pembagian tugas antara server dan client, sehingga dapat meningkatkan kenerja terhadap sistem. Selain peningkatan kinerja tingkat keamanan data juga semakin tinggi karena untuk masuk ke level data harus melewati beberapa lapisan *(layer)*.

2. Arsitektur Client- Server

Berdasar pada cara PC client dihubungkan ke komputer server, dikenal dengan dua macam tingkatan arstiektur yaitu model *Two Tier*, *Three Tier,* atau *n-Tier* dengan tiga arsitektur komponen utama yaitu:

1. *Front-End Component*
2. *Back-End Component*
3. *Database Component*

2.1 Front-End Component

Pada bagian ini adalah komponen yang dilihat dan berhubungan langsung dengan *user*. Dalam implementasinya *front-end* biasanya berbentuk *graphical user Interface (GUI)*. Karena bagian ini berhubungan dengan *user* maka dalam sistem arsitektur client-server digunakan untuk form memasukan data dan menampilkan data ke user.

2.2 Back-End Component

Komponen ini tidak dapat dilihat oleh user, tapi memegang peran yang sangat penting di dalam arstitektur *client-server*. Komponen *back-end* berisi *bussiness logic* yang digunakan untuk mengupdate database. *Back-end* akan merespon permintaan dari *front-end* dan mengedit data berdasar aturan bisnis yang ada.

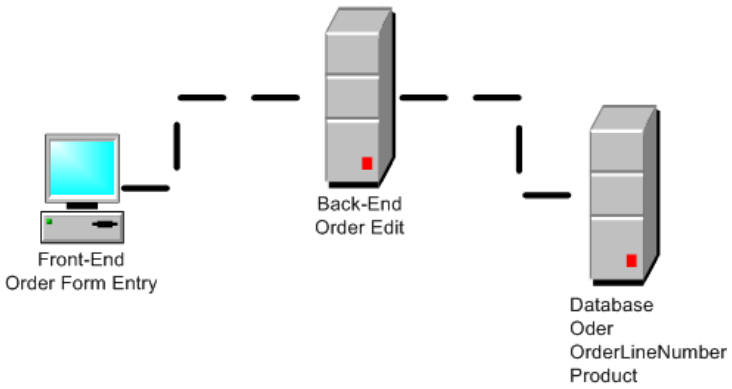
2.3 Database Component

Pada bagian ini berisi informasi data, tapi *end-user* tidak dapat melihat atau berhubungan langsung dengan database. Fungsi utama komponen database adalah menyimpan data. Biasanya dalam komponen ini menggunakan DBMS yaitu singkatan dari *Database Management System* yang berarti sistem manajemen database.

DBMS merupakan paket software yang berfungsi mengkoordinasi dan mengatur semua aktivitas yang yang berhubungan dengan database. Sebuah DBMS berisi koleksi data yang saling berelasi dan set program pengelola untuk menambah data, mengmbil dan menghapus data.

3. Arsitektur *Three Tier*

Pada arsitektur ini akan menempatkan komponen *front-end, back-end* dan database dalam komputer yang berbeda, dimana komputer tersebut saling berhubungan dalam sebuah jaringan.



Gambar 1. Arsitektur Three Tier

3.1 Aplikasi Enterprise

Kata enterprice sebagian orang berarti “besar”, “perusahaan” dan sejenisnya. Sedangkan jika melihat definisi umum yang diberikan di buku *System Design and*

Analysis Method, maka definisi aplikasi enterprise adalah koleksi dari sistem informasi yang terintegrasi secara menyeluruh dan mengakomodasi kesekuruhan fungsi dasar yang dibutuhkan oleh perusahaan. Sehingga dapat diambil kesimpulan bahwa aplikasi enterprise adalah aplikasi komputer yang besar atau sistem aplikasi komputer perusahaan.

Jika diambil acuan dari definisi yang diberikan Microsoft berdasarkan apa yang ada dalam bukunya yang ditunjukan bagi peserta training MCSD yang berjudul *Analysis requirement and defining solution architecture*.

Ada buku yang mendifinisikan bahwa definisi aplikasi enterprise adalah aplikasi bisnis yang besar yang mempunyai kelebihan *multi-user*, *Complex, scalable, distributed* berbasis component dan *mission critical*. Dapat di deploy pada multi-environment seperti IntraNet, InterNet, network lintas latform, bersifat data centric, dan dapat memenuhi syarat sekuritas yang lengkap. Secara singkat dikatakan apliaksi yang kompleks.

4. Teknologi yang Digunakan

4.1 DCOM

DCOM adalah *(Distributed COM)* merupakan suatu teknologi cara melakukan distribusi COM. Istilah COM sama seperti komponen. Tujuan DCOM adalah agar komponen yang telah kita buat dan diregister di suatu komputer dapat di akses oleh banyak komputer lain.

DCOM sendiri bukan merupakan teknologi beru karena teknologi ini banyak dipakai dalam sistem distribusi pada sistem jaringan TCP/IP. Protokol yang digunakan oleh DCOM dapat meliputi:

- Protokol TCP/IP
- Protokol UDP/IP
- Protokol HTTP
- Protokol RPC

Secara umum DCOM menerapkan sistem *Client-Server* sehingga terdapat istilah *DCOM client* dan *DCOM server*.

4.2 Arsitektur DCOM

DCOM merupakan ektensi dari *Component Object Model (COM)*. Pada COM kita melihat bagaimana suatu komponen *client* saling berinteraksi. Interaksi ini dapat didefinisikan sebagai hubungan secara langsung antara komponen *(COM Server)* dan *COM Client*.

Aplikasi client memanggil method yang ada di komponen COM tanpa perantara apapun dan terjadi dalam suatu proses baik pada palikasi client maupun komponen itu sendiri.

Sedangkan pada sistem operasi yang baru proses akan dilindungi dari gangguan proses yang lain sehingga aplikasi client tidak akan langsung mamanggil komponen tetapi melalui interproses yang disediakan oleh sistem operasi.

Ketika client dan sever pada tempat yang berbeda, DCOM akan menggantikan local interprosses yang berkomunikasi dengan sebuah jaringan protokol. Malalui protokol DCOM ini, aplikasi client dapat mengakses DCOM server. DCOM juga menyembunyikan lokasi suatu komponen sehingga disis aplikasi tinggal langsung memanggil method yang ada di komponen DCOM.

Lokasi DCOM yang independen inilah yang membuat penyederhanaan pada penerapan sistem distribusi suatu komponen dan juga meningkatkan performance. Bayangkan kita mempunyai komponen yang banyak dan kemudian dilakukan distribusi ke duatau jaringan LAN, maka hal iini akan meningkatkan laju trafik jaringan sehingga komunikasi data pada jaringan LAN ini akan jauh lebih lambat. Dengan adanya DCOM maka semua komponen didistribusikan dalam suatu protokol DCOM dan proses yang sama. Ketika suatu aplikasi mengakses komponen melalui DCOM maka DCOM akan melakukan proses validasi komponen, tujuanya untuk mengakses apakah komponen yang dipanggil ini ada didalam DCOM.

4.3 *Remote Procedure Call (RPC)*

Cara kerja DCOM adalah dengan menggunakan *Remote Procedure Call* dimana *Object* diletakkan di dalam *Back-End (Server)* dalam bentuk file Dll *(Dynamic linking Library)* dan client akan mengakses object tersebut melalaui media jaringan yang ada.

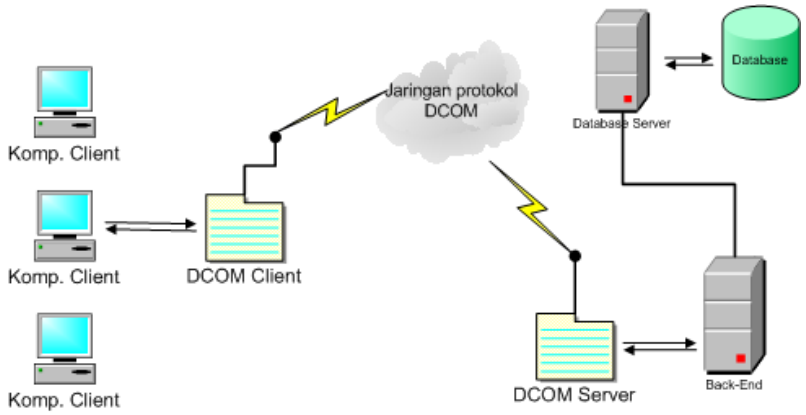
4.4 Cara Kerja Remote DCOM

Salah satu kegunaan DCOM adalah distribusi dan remoting suatu objek ke komponen lain dalam suatu jaringan komputer. Ketika kita akan membuat suatu komponen COM dan komponen ini akan diakses banyak komputer bahkan dalam waktu bersamaan, maka kita harus melakukan sistem distribusi untuk objek yang kita buat belum lagi kita akan menghadapi load balancing karena banyaknya yang mengakses data objek oleh komputer lain. Dengan DCOM ini,kita akan mendapatkan solusi untuk masalah sistem distribusi.

Kita telah ketahui bahwa apliaksi dikatakan berbasis sistem remoting bila apabila aplikasi itu mengakses suatu onjek yang dapat berupa data, suara, informasi dan sebagainya dari suatu komputer yang ada dalam suatu jaringan tertentu. Dalam kasus ini DCOM client akan berfungsi sebagai remoter yaitu objek yang melakukan remote ke komputer serer melalui DCOM Server.

Jika kita perhatikan, kita akan melihat bahwa komunikasi antara komputer dilakukan dan melalui DCOM. Misalkan komputer A meminta data dari database, maka komputer A akan merequest data melalui objek yang ada di DCOM Client. Kemudian DCOM client akan melakukan validasi mengenai komponen objek yang dieksekusi, jika ditemukan objek yang meminta, maka DCOM client akan mengecek sumber objek berasal sebagai contohnya suber objek berasal dari komputer server S, maka DCOM client akan melakukan remoting ke komputer S melalui DCOM server yang dimilikinya. DCOM serve akan mengecek authorisasinya yang dimilki oleh komputer A. Jika komputer A mmpunyai hak akses maka DCOM server akan mengeksekusi sesuai permintaan komputer A dan hasilnya dikembalikan ke DCOM Client. Proses ini akan sama untuk setiap komputer yang ingin melakukan remoting melalui DCOM.

Berikut gambaran arsitekturnya hubungan antara DCOM Server, DCOM Client dan Server Database .



Gambar 2. Arsitektur DCOM Server dan DCOM Client

Pada bagian *Back-End* yang merupakan jembatan antara layer user dan layer data (database) memiliki empat syarat:

- **Atomicity**

  transaksi harus berhasiil atau gagal sama sekali
- **Consistenty**

  Traksaksi harus konsisten secara keseluruhan, memang traksaisi pada awalnya tidak konsisten namun harus konsistm pada alhirnya.
- **Isolation**

  Semua aplikasi yang terlibat traksaksi tidak boleh melihat bagaimana transaksi berlangsung, hanya melihat bagian awal sebagai inisialisasi dan bagian akhir sebagai status apakah transaksi berhasil atau gagal.
- **Durability**

  Ketika transaksi selesai maka update bersfat permanen dan tidak bisa dihapus dengan cara apapun.

Ke empat kriteria diatas dikenal sebagi ACID test. Untuk menghubungkan bisnis layer dengan data layer digunakan teknologi *UDA (universal Data Access).*

Dalam database server kita harus memilih aplikasi yang dapat mendukung untuk DBMS salah satnya adalah: Microsoft SQL Server 2000. SQL Server juga mempunyai kelebihan OLAP. Selain mendukung bahasa SQL standar SQL, SQL Server juga mennunjang T-SQL *(Transact-SQL)* yang merupakan pengembangan dari SQL standart yang ada. Kesemuanya itu sangat mendukung adanya ACID dalam transaksi.

5. Desain dan Implementasi Aplikasi Kepegawaian

5.1 Rencana Tahapan Pembuatan Desain dan Inplementasi Aplikasi

Aplikasi ini dibuat terlebih dahulu dengan *single tier*. Hal ini dilakukan agar mudah dalam penulisan program. Adapun langkah pembuatannya adalah sebagai berikut.

1. Desain server Database
    a. Adapun yang dilakukan adalah mendesain relasi antar tabel (jika ada) dan *Data Definition Language (DDL)*.
2. Medesain aplikasi Server *(Back-End)*
    a. Apkliasi ini nantinya harus dapat dijadikan *ORB (Object Request Broker)* dan dapat di akses dengan *Remote Procedure Call (RPC)*.
    b. menentukan aturan-aturan hubungan antara client dan database.
    c. Aplikasi ini akan dicompile dalam bentuk *.DLL
3. Pembuatan aplikasi *Front-End*.
    a. Pada tahapan akan didesain program dengan kemampuan menampilkan dan pembuatan form untuk client tanpa ada aturan untuk akses database.
    b. Aplikasi ini akan memekai object yang dibuat dilangkah ke 2 secara standalone dahulu dengan tuujuan untuk mempercepat proses pembuatan program.
4. Proses distribusi
    a. Pada proses ini akan diterapkan aplikasi *Three Tier* murni, karena akan dipisah masing-masing komponen dalam komputer yang terpisah tapi terkoneksi dengan jaringan.
    b. Selain melakukan proses distribusi aplikasi (*.exe) perlu dilakukan proses distribusi untuk *Object Class*-nya. Hal ini dilakukan agar object di server dapat diakses oleh client secara RPC.

5.2 Pembuatan Server Database

Dalam hal ini *Database Management System* (DBMS) yang dipakai oleh penulis adalah Microsoft SQL Server 2000.

Database dalam aplikasi ini disusun dengan *DDL (Data Definition Language)* sebagai berikut:

```
CREATE DATABASE company
USE company
CREATE TABLE pegawai (
     nik char (4) PRIMARY KEY,
     nama varchar (50) NOT NULL ,
     alamat varchar (100) NULL )
```

Jika digambarkan dalam bentuk tabel dapat dilihat pada gambar dibawah ini.

Nama Field	Type	Panjang
Nik	Char	4
Nama	Varchar	50
Alamat	Varchar	1000

Ket: Primary Key : Nik

5.3 Aplikasi *Back-End (aCOMClassDb)*

Untuk level aplikasi Back-End ini penulis menggunakan bahasa pemrograman Microsoft Visual Basic 6.0 dan teknologi *DCOM (Distributed COM)*. Apliksi ini berfungsi sebagai *ORB (Object request Broker)* yang akan mengatur transaksi antara Client dan Database dengan metode *RPC (Remote Procedure Call)*. Client tidak dapat secara langsung mengakses Database Server kecuali melalui aplikasi ini. Adapun Class diagram aplikasi Back-End adalah sebagai berikut.

Nama Class : **COMClassDB**
Method: rLogin rFind rShow rUpdate rDelete

Semua aturan bisnis untuk mengakses database didesain dalam aplikasi *Back-End*, perintah-perintah SQL untuk memanipulasi data (DML) dibuat diobject ini yang nantinya akan diakses oleh client secara remote.

5.3.1 Kontruksi *Class Object Back-End*

Nama Object yang akan dibuat adalah COMClassDb, adapun kode programnya adalah sebagai berikut:

```
Option Explicit
Dim StrConn As String
Public Conn As ADODB.Connection

Public Function rLogin(rUid As String, rPwd As String) As Boolean
On Error Resume Next
'On Error GoTo errHandler
Dim objX As ObjectContext

Set objX = GetObjectContext()
Set Conn = New ADODB.Connection

StrConn = "Provider=SQLOLEDB.1;Password=rPwd;Persist Security Info=True;User
ID=rUid;Initial Catalog=company;Data Source=192.168.0.10"
Conn.ConnectionString = StrConn
Conn.Open

If Conn.State = adStateOpen Then
  rLogin = True
Else
  rLogin = False
End If
End Function

Public Function rFind(ByVal vnik As String) As Boolean
Dim RsFind As New ADODB.Recordset
Dim SQLShow As String
Dim Stat As Boolean

SQLShow = "SELECT * FROM pegawai WHERE nik='" & vnik & "'"
Set RsFind = Conn.Execute(SQLShow)

If RsFind.EOF Then
  rFind = False
Else
  rFind = True
End If
End Function

Public Function rShow() As ADODB.Recordset
Dim SQLShow As String
Dim Stat As Boolean

Set rShow = New ADODB.Recordset
rShow.CursorLocation = adUseClient
SQLShow = "SELECT * FROM pegawai"
rShow.Open SQLShow, Conn, 1, 3
End Function
'
Public Function rInsert(vnik As String, vnama As String, valamat As String)
As Boolean
'On Error Resume Next

Dim SQLInsert As String
```

```
SQLInsert = "INSERT INTO pegawai VALUES ('" & vnik & "','" & valamat & "','"
& valamat & "')"
Conn.Execute (SQLInsert)
If Err.Number > 0 Then
    rInsert = False
Else
    rInsert = True
End If
End Function
'
Public Function rUpdate(vnik As String, vnama As String, valamat As String)
As Boolean
On Error Resume Next
Dim SQLUpdate As String

SQLUpdate = "UPDATE pegawai SET nik='" & vnik & "',nama='" & vnama &
"',alamat='" & valamat & "' WHERE nik='" & vnik & "'"
Conn.Execute (SQLUpdate)

If Err.Number > 0 Then
    rUpdate = False
Else
    rUpdate = True
End If
End Function
'
Public Function rDelete(vnik As String) As Boolean
On Error Resume Next

Dim SQLDelete As String

SQLDelete = "DELETE FROM pegawai WHERE nik='" & vnik & "'"
Conn.Execute SQLDelete

If Err.Number > 0 Then
    rDelete = False
Else
    rDelete = True
End If
End Function
```

Setelah selesai dalammenulis program,maka langkah selanjutnya adalah meng-Compile Object tersebut menjadi bentuk file (*.DLL), agar dapat dipakai untuk membuat program Client. Setelah di compile akan menghasilkan file COMClassDb.dll.

5.4 Aplikasi *Front-End*

Untuk mempermudah pembuatan GUI pada Client, maka untuk kita buat dulu dalam bentuk standalone. Setelah selesai semua maka baru kita buat dalam bentuk Three Tier.

Aplikasi Front-End berisi *Graphical User Interface (GUI)* yang berfungsi untuk menampilkan database dan form untuk pengolahan database yang dilakukan oleh user.

Dalam *Front-End* hanya berisi GUI yang berbentuk penampilan data dan Form untuk mengolah data.

Aplikasi ini tidak dapat berhubungan langsung dengan database. Untuk mengambil data dari database harus melaui aplikasi *Back-End*, dengan mengakses Object Class yang dibuat kita buat dalam *Back-End*. Untuk dapat memakai Object Class maka perlu dilakukan registrasi object dulu dengan cara mengetikkan:

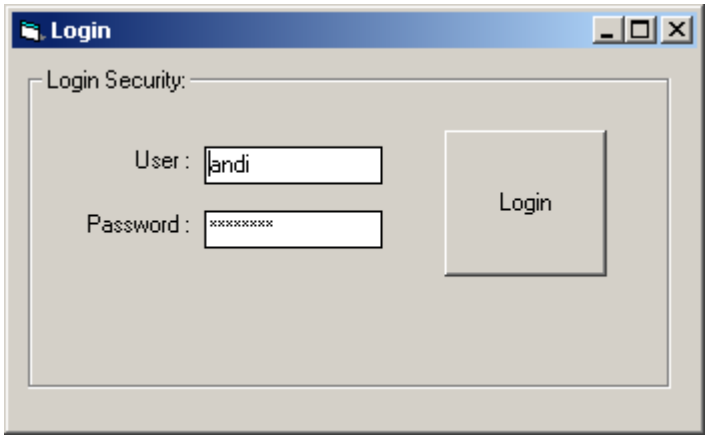
```
regsvr32 “lokasi object\ COMClassDb.dll”
```

5.4.1 Modul Inisialisasi DCOM

```
Dim Stat As Boolean
Public objDb As Object

Public Sub objInit()
Set objDb = CreateObject("aCOMClassDb.COMClassDb")
End Sub
```

5.4.2 Form Login



Gambar 3. Form Login

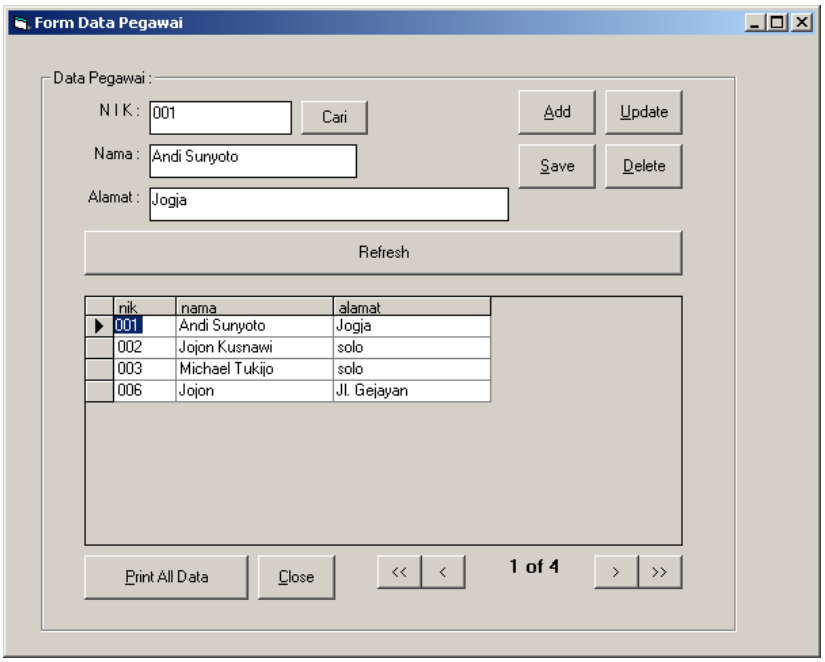
```
Private Sub cmdLogin_Click()
'On Error Resume Next
Dim Stat As Boolean

MousePointer = vbHourglass
Stat = objDb.rLogin(txtUser, txtPassword)
MousePointer = vbDefault
If Stat Then
  MsgBox "Login Sukses !!!" & vbCrLf & "Anda Berhak Masuk Aplikasi Ini",
vbInformation + vbOKOnly, "Informasi"
  frmPegawai.Show
  Unload Me
Else
  MsgBox "Login Gagal !!!", vbCritical + vbOKOnly, "Informasi"
End If
```

```
End Sub

Private Sub Form_Load()
Call objInit
End Sub
```

5.4.3 Form Olah data karyawan



Gambar 4. Form olah data karyawan

```
Dim WithEvents rsData As ADODB.Recordset

Private Sub cmdAdd_Click()
txtNik.Text = ""
txtNama.Text = ""
txtAlamat.Text = ""
End Sub

Private Sub cmdClose_Click()
Set objDb = Nothing
End
End Sub

Private Sub cmdDelete_Click()
Dim pesan As String
Dim konfirm As Integer

konfirm = MsgBox("Anda Yakin Akan Menghapus Data ini ?", vbQuestion +
vbYesNo, "Konfirmasi")
If konfirm = vbYes Then
  MousePointer = vbHourglass
  pesan = objDb.rDelete(txtNik)
  MousePointer = vbDefault
  If pesan = False Then
```

```
    MsgBox "Gagal melakukan pengahapusan data" + vbCrLf + Err.Description,
vbInformation + vbOKOnly, "Informasi"
  End If
  cmdRefresh_Click
End If
End Sub

Private Sub cmdFind_Click()
If objDb.rFind(txtNik) Then
  MousePointer = vbHourglass
  rsData.Find "nik='" & txtNik & "'", , , 1
  MousePointer = vbDefault
Else
  MsgBox "Data tidak ada", vbInformation + vbOKOnly, "Informasi"
End If
End Sub

Private Sub cmdFirst_Click()
rsData.MoveFirst
End Sub

Private Sub cmdLast_Click()
rsData.MoveLast
End Sub

Private Sub cmdNext_Click()
rsData.MoveNext
End Sub

Private Sub cmdPrevious_Click()
rsData.MovePrevious
End Sub

Private Sub cmdPrintAll_Click()
With DataReport1.Sections("Section1").Controls
  .Item("txtnik").DataField = rsData(0).Name
  .Item("txtnama").DataField = rsData(1).Name
  .Item("txtalamat").DataField = rsData(2).Name
End With
'DataReport1.Refresh
DataReport1.Sections("Section2").Controls.Item("lblTgl").Caption = "Tanggal
Print: " & Format(Date, "ddd, dd - mmmm - yyyy")
Set DataReport1.DataSource = rsData
DataReport1.Sections("Section5").Controls.Item("Label5").Caption = "Jumlah
data : " & rsData.RecordCount
DataReport1.Show
End Sub

Private Sub cmdRefresh_Click()

Set rsData = objDb.rShow()
Set DataGrid1.DataSource = rsData
End Sub

Private Sub cmdSave_Click()
Dim pesan As String
MousePointer = vbHourglass
pesan = objDb.rInsert(txtNik, txtNama, txtAlamat)
MousePointer = vbDefault
If pesan = False Then
  MsgBox "Gagal melakukan penambahan data" + vbCrLf + Err.Description,
vbInformation + vbOKOnly, "Informasi"
End If
```

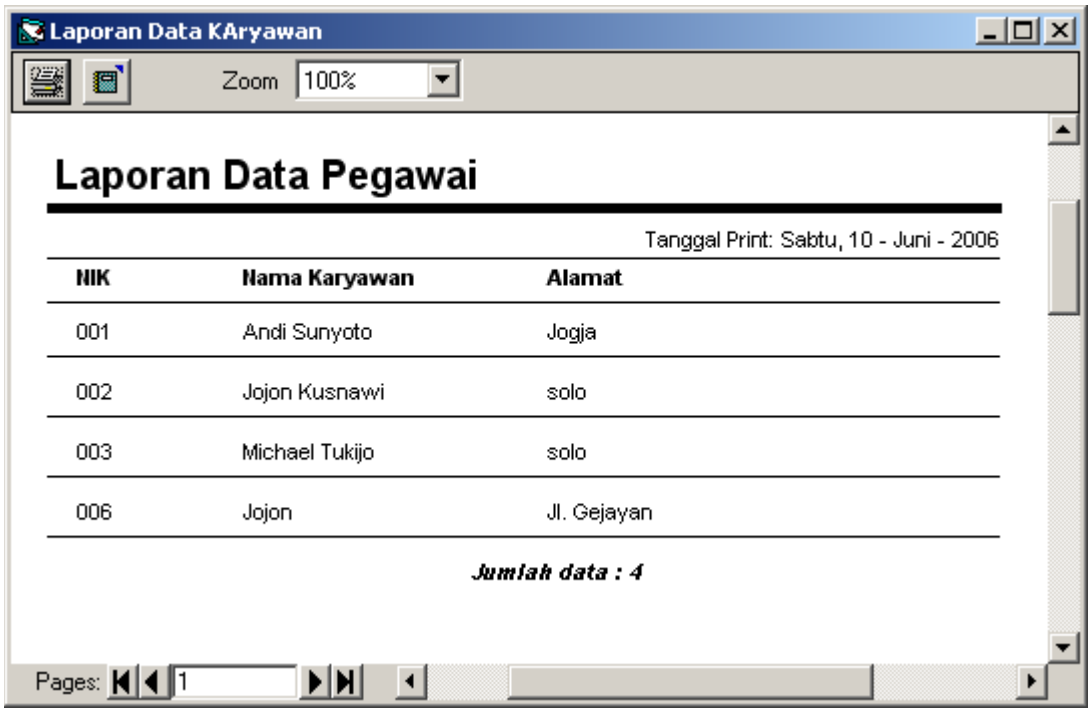
```
cmdRefresh_Click
End Sub

Private Sub cmdUpdate_Click()
Dim pesan As String
Dim xnik As String
MousePointer = vbHourglass
pesan = objDb.rUpdate(txtNik, txtNama, txtAlamat)
MousePointer = vbDefault
xnik = txtNik
If pesan = False Then
  MsgBox "Gagal melakukan pengupdatan data" + vbCrLf + Err.Description,
vbInformation + vbOKOnly, "Informasi"
End If
cmdRefresh_Click
rsData.Find "nik='" & xnik & "'", , , 1
End Sub

Private Sub Form_Load()
Call cmdRefresh_Click
End Sub

Private Sub rsData_MoveComplete(ByVal adReason As ADODB.EventReasonEnum,
ByVal pError As ADODB.Error, adStatus As ADODB.EventStatusEnum, ByVal
pRecordset As ADODB.Recordset)
If pRecordset.EOF Then
  pRecordset.MoveLast
ElseIf pRecordset.BOF Then
  pRecordset.MoveFirst
End If
txtNik.Text = pRecordset(0)
txtNama.Text = pRecordset(1)
txtAlamat.Text = pRecordset(2)
lblRecod.Caption = pRecordset.AbsolutePosition & " of " &
pRecordset.RecordCount
End Sub
```

5.4.4 Laporan data Seluruh karyawan



6. DISTRIBUSI APLIKASI

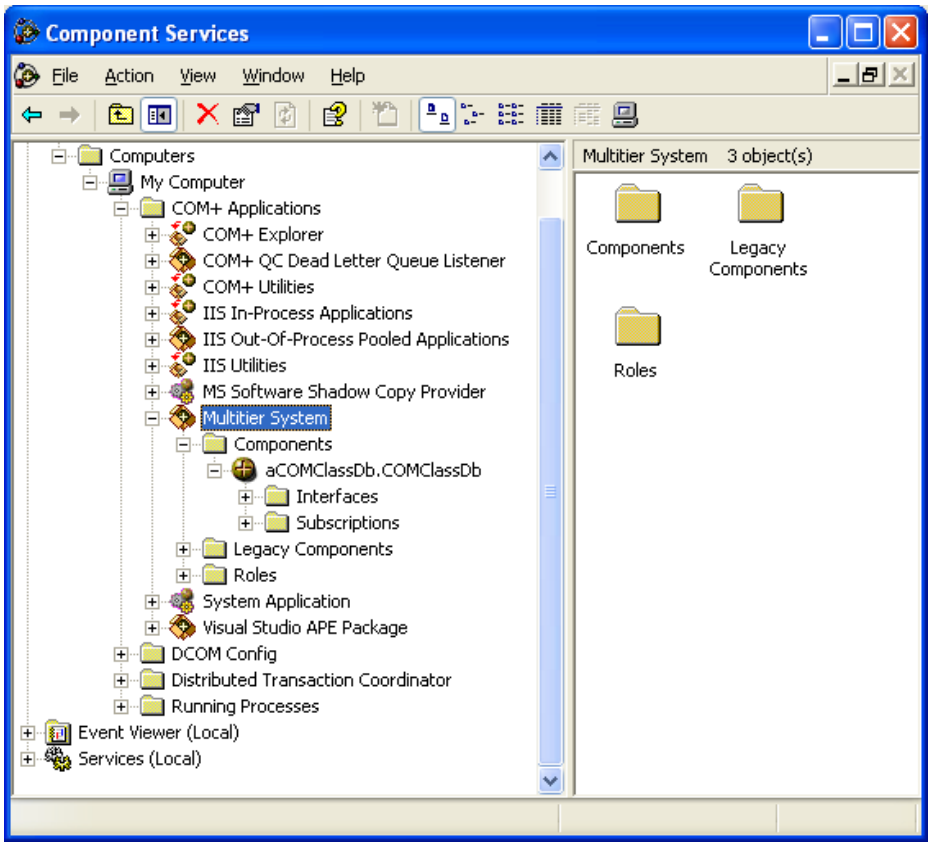
Terlebih dulu kita siapkan tiga komputer yang terhubung dalam jaringan yang akan digunkan untuk Server database, apikasi *Back-End*, dan *Client (Front-End)*.

6.1 Distribusi Server Database

Setelah kita tentukan komputer untuk Server database, dan asumsikan mesin untuk database (MS SQL Server 2000) telah terinstall., langkah selanjutnya adalah menjalankan file DDL (Data Definition language) yang kita buat dalam Microsoft SQL Server.

6.2 Distribusi Back End

Hasil dari kode program pada *Back-End* diatas kita compile terlebih dahulu dengan nama *COMClassDb.dll* agar dapat kita jadikan sebagai Object server dan berfungsi sebagai ORB, maka file tersebut didaftarkan ke *Componen Service*, dengan cara: **Start – Program – Control Panel – Administratif Tool – Component Service** sehingga seperti terlihat pada gambar dibawah ini. Selanjutnya dilakukan proses diatas maka DCOM server telah terbentuk. DCOM ini akan bertindak sebagai *ORB (Object Request Broker)*.



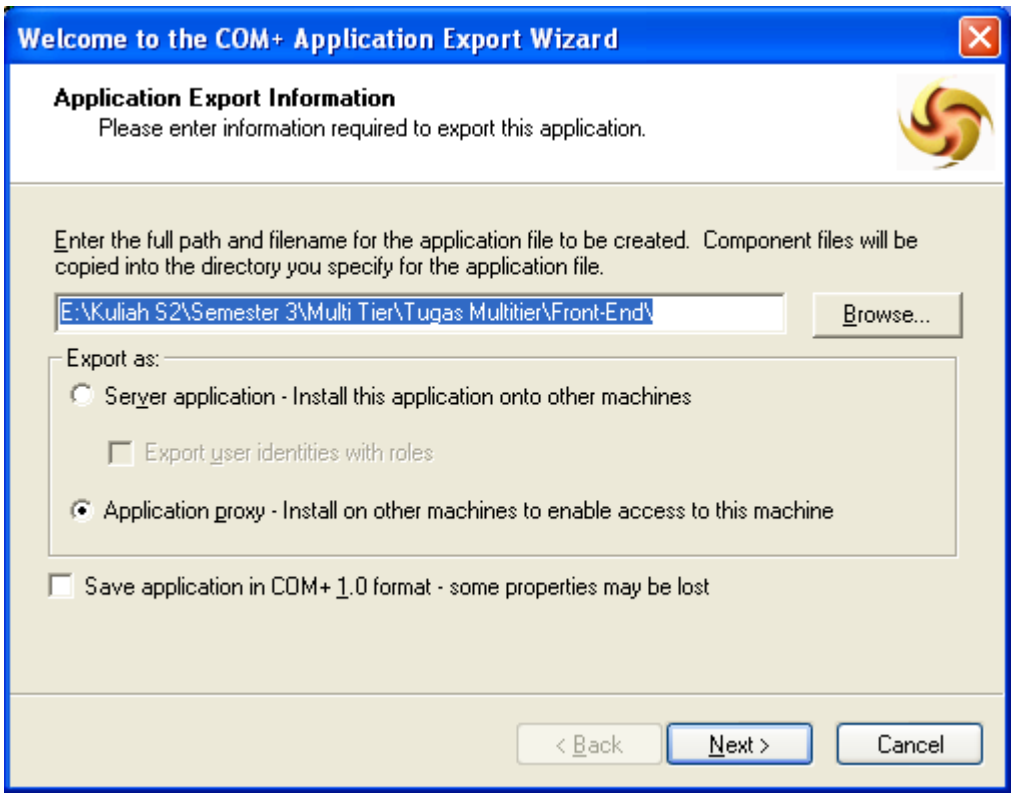
Gambar 5. Konfiguarsi Componen Service pada Windows XP

6.3 Distribusi *Front-End*

Kode program yang dibuat dalam *Front-End* di kompile yang akan menghasilkan aplikasi bernama Karyawan.exe setelah tidak ada kesalahan maka kita lakukan proses Deployment.

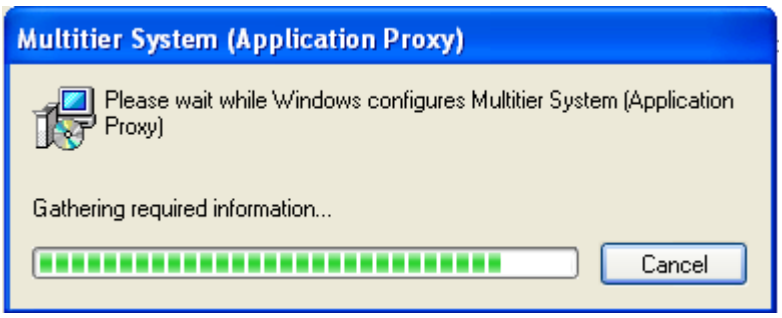
Hasil Deplotment kita install di komputer client yang diinginkan. Untuk dapat memanggil Object Class (DCOM Server) di Back-End perlu dilakukan distribusi DCOM, yang selanjutnya disebut sebagai DCOM Client.

DCOM client dibuat dengan cara meng-Export bentuk DCOM Server ke DCOM Client dengan memilih sebagai Application Proxy yang berbentuk file berektensi (*.msi).



Gambar 6. Export dari COM Server ke COM Client

Setelah program Client di compile, maka langkah selanjutnya adalah mengninstall DCOM client (*.msi) yang duperoleh dari hasil extraxt DCOM dari Server.



Gambar 7. Gambar Instalasi DCOM Client (*.msi)

Semua proses seledai dilaksanakan, maka *Front-End* siap untuk dijalankan.

7. Kesimpulan

Aplikasi three tier membagi membagi sebuah aplikasi menjadi tiga bagian yaitu: Front-End, Back-End dan Database. Pembagian tersebut dapat meningkatkan kinerja dan meningkatkan tingkat keamanan terhadap data, karena user tidak dapat berhubungan langsung dengan database.

Teknologi *Distributed COM* (DCOM) digabung dengan Microsoft Visual Basic 6.0 dan Microsoft SQL Server dapat diterapkan untuk membangun aplikasi three tier.

8. Daftar Pustaka

Balena, Francesco,*"Programming Microsoft Visual Basic 6.0"*, Microsoft Press, Redmond, Washington, 1999.

Jonshon, J, Eric, *"The Complete Guide to Client/Server Computing"*, Prentice Hall, United State Of America, 2001.

Kurniawan, Agus, "*Pemrograman COM, DCOM, dan COM+ dengan Visual Basic 6.0"*, PT. Elek Media Komputindo, Kelompok Gramedia, Jakarta 2003.

Pattison, Ted, "*Programming Distributed Applications with COM and Microsoft Visual"*, Microsoft Press, Redmond, Washington, 1998.

Setiabudi, Djoni. H.Ir. M.Eng, Gunawan, Ibnu. S.T, *"Belajar Sendiri Database Terdistribusi dengan Visual Studio 6"*, PT. Elek Media Komputindo, Kelompok Gramedia, Jakarta 2003.

Web Site http://support.microsoft.com, *"How To Troubleshoot DCOM for Visual Basic Client/Server Applications"*, Microsoft Help and Support",Tanggal Access 10 Juli 2006.