

PENERAPAN ARSITEKTUR MULTI-TIER DENGAN DCOM DALAM SUATU SISTEM INFORMASI

Kartika Gunadi, Irwan Kristanto Julistiono

Fakultas Teknologi Industri, Jurusan Teknik Informatika – Universitas Kristen Petra
e-mail: kgunadi@peter.petra.ac.id,

Heliman Prasetya Tantama

Alumnus Fakultas Teknologi Industri, Jurusan Teknik Elektro – Universitas Kristen Petra

ABSTRAK: Penerapan sistem informasi menggunakan *two-tier architecture* mempunyai banyak kelemahan : penggunaan kembali komponen, skalabilitas, perawatan, dan keamanan data. *Multi-tier Client-Server architecture* mempunyai kemampuan untuk memecahkan masalah ini dengan DCOM teknologi.

Perangkat lunak ini dapat dibuat menggunakan *Delphi 4 Client/Server Suite* dan *Microsoft SQL Server 7.0* sebagai perangkat lunak database. Aplikasi program multi-tier ini dibagi menjadi tiga partisi. Pertama adalah aplikasi client menyediakan presentasi servis, kedua aplikasi server menyediakan servis aplikasi, dan ketiga aplikasi database menyediakan database servis. Perangkat lunak aplikasi multi-tier ini dapat dibuat dalam dua model, yaitu client/server windows model dan client/server web model dengan *activeX Form* teknologi.

Pada penelitian ini ditemukan bahwa pembuatan arsitektur multi-tier dengan DCOM teknologi memiliki banyak keuntungan seperti, memusatkan logika aplikasi pada middle-tier, membuat thin client aplikasi, mendistribusikan beban proses data pada beberapa mesin, meningkatkan keamanan dengan kemampuan menyembunyikan data, cepatnya perawatan dengan tanpa instalasi database driver pada setiap client.

Kata kunci: *multi-tier clien-server, architecture, DCOM, active form.*

ABSTRACT: Information System implementation using *two-tier architecture* result lack in several critical issues : reuse component, scalability, maintenance, and data security. The multi-tiered client/server architecture provides a good resolution to solve these problems that using DCOM technology .

The software is made by using *Delphi 4 Client/Server Suite* and *Microsoft SQL Server V. 7.0* as a database server software. The multi-tiered application is partitioned into thirds. The first is client application which provides presentation services. The second is server application which provides application services, and the third is database server which provides database services. This multi-tiered application software can be made in two model. They are Client/Server Windows model and Client/Server Web model with *ActiveX Form Technology*.

In this research is found that making multi-tiered architecture with using DCOM technology can provide many benefits such as, centralized application logic in middle-tier, make thin client application, distributed load of data process in several machines, increases security with the ability in hiding data, dan fast maintenance without installing database drivers in every client.

Keywords: *multi-tier clien-server, architecture, DCOM, active form.*

1. PENDAHULUAN

Bentuk Sistem Informasi dengan teknologi client/server *two-tier* yang mempunyai banyak kelemahan menyebabkan banyak perusahaan mulai berpindah dari teknologi *client/server two-tier* arsitektur menjadi *three-tier/multi-tier* arsitektur yang menyediakan kemampuan perluasan lebih besar, perawatan yang lebih rendah dan mudah, meningkatkan keamanan, dapat

melakukan penggunaan kembali komponen, dan mendistribusikan beban.

Teknologi *multi-tier* ini dapat dibuat dengan beberapa macam bahasa pemrograman, dan *database server* yang mendukung. Dalam proyek ini dipakai bahasa pemrograman *Delphi 4.0 client/server* karena sifatnya yang visual, berorientasi *object*, open arsitektur dengan banyak komponen pendukung dan yang terpenting adalah memiliki komponen-komponen yang

mendukung teknologi multi-tier ini, misalnya komponen DCOM. Sedangkan untuk database server, dalam proyek ini dipilih MS SQL Server 7.0 karena memang penggunaannya bagus diperuntukkan untuk teknologi client/server.

Aplikasi Multi-tier dapat berupa client/server window model ataupun web model. Untuk window model, biasa dipakai untuk jaringan lokal, karena prosesnya lebih cepat dibanding web model, dan juga dapat dipakai untuk aplikasi yang sangat kompleks. Sedangkan web model untuk aplikasi intranet ataupun internet, tapi banyak dipakai untuk aplikasi internet, karena teknologinya yang memungkinkan untuk itu, dengan sisi client hanya memerlukan universal client yang disebut web browser. Dalam proyek ini dengan bantuan komponen DCOM akan dibuat client/server window model berbentuk 3-tier, dan web model berbentuk 4-tier.

2. DASAR TEORI

2.1 Pembagian aplikasi dalam multi-tier

Dalam desain aplikasi perangkat lunak, terdapat beberapa pembagian aplikasi yang ada yaitu :

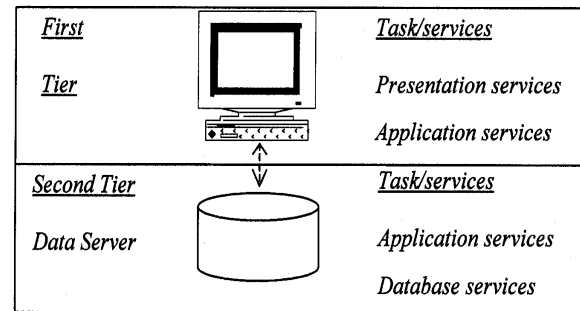
- a. *Presentation processing logic*
Ini adalah bagian dari aplikasi yang berinteraksi dengan device seperti *end user terminal* atau *workstation*..
- b. *Application logic*
 - b1. *Business processing logic*
Adalah bagian dari aplikasi yang menggunakan data input untuk mendayagunakan *business task*.
 - b2. *Database processing logic*
Adalah bagian dari aplikasi yang memanipulasi data diantara aplikasi. Manipulasi data dalam *Relational DMBS* dilakukan oleh *Structured Query Language*.
- c. *Database processing*
Ini adalah proses aktual dari database data yang didayagunakan oleh *DBMS*.

Dengan adanya pembagian aplikasi ini, maka langkah selanjutnya adalah menentukan bagaimana mendistribusikan komponen aplikasi antara client-client dan server-

server. Komponen-komponen tersebut akan didistribusikan dalam suatu client/server processing model, yaitu dapat berupa arsitektur two-tier, three-tier, hingga n-tier. Berikut ini adalah contoh gambar dari tiap client/server processing model:

1. Two-tier architecture

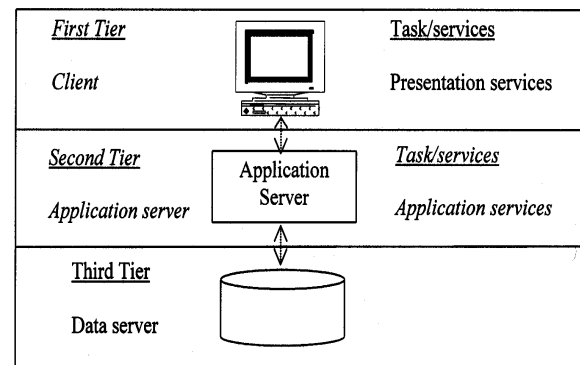
Two-tier Client/Server menyuguhkan basis/dasar pembagian tugas.



Gambar 1. Two-tier Client/Server Architecture

2. Three-tier architecture

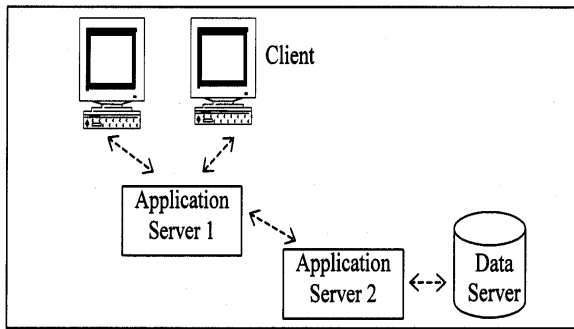
Three-tier atau Multi-tier arsitektur menyuguhkan aplikasi besar yang scalability, perawatan rendah/sedikit, dan meningkatkan reuse komponen.



Gambar 2. Three-tier Client/Server Architecture

3. Multi-tier/n-tiers architecture

Multi-tier architecture menyuguhkan bentuk three-tier yang diperluas dalam model fisik yang terdistribusi. Application Server dapat mengakses Application Server yang lain untuk mendapat data dari Data Server dan mensuplai servis ke Client Application.



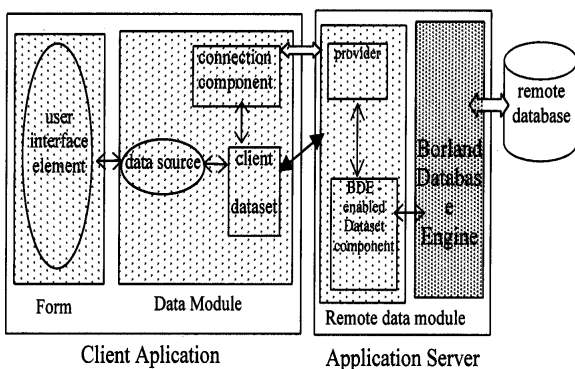
Gambar 3. Multi-tier Client/Server Architecture

2.2 Teknologi Midas

MIDAS (*Multi-tier Distributed Application Service Suite*) adalah teknologi buatan Delphi untuk mengimplementasikan multi-tier arsitektur.

Komputasi multitier database memungkinkan kita untuk mempartisi aplikasi, sehingga kita dapat mengakses data pada Server Middle tanpa memiliki perlengkapan database yang menyeluruh pada komputer lokal. Dan memungkinkan untuk melakukan sentralisasi proses dan mendistribusikan beban proses dalam suatu jaringan. Multi-tier aplikasi melakukan sharing data dan berkomunikasi satu sama lain melalui Lokal Area Network atau bahkan melalui Internet. Pada bentuk yang sederhana, kadangkala disebut “three-tiered model”, Multi-tier Aplikasi terdiri dari 3 bagian utama: *Aplikasi Client*, *Server aplikasi*, dan *remote database server*.

Di bawah ini adalah skema dari MIDAS multitier /three –tier arsitektur :



Gambar 4. Multi-tier Database Architecture

Pada gambar ini, tampak bahwa *Client* aplikasi menggunakan data control yang terkoneksi ke datasource ke satu atau lebih

client dataset component untuk mendapatkan data dan menampilkan atau melakukan editing. Setiap client dataset berkomunikasi dengan application server melalui *Iprovider Interface* yang merupakan bagian dari *remote data module* pada application server. Aplikasi client dapat menggunakan beberapa macam protokol (TCP/IP, DCOM, MTS, atau CORBA) untuk menjalankan komunikasi. Protokol yang digunakan tergantung dari *connection component* yang digunakan pada client. dan tipe dari remote datamodule yang digunakan pada application server.

MIDAS adalah teknologi yang memindahkan dataset dari TTable atau TQuery objek pada server menjadi TclientDataset object pada client. Tclient Dataset menyerupai dan berlaku seperti TTable atau TQuery komponen, kecuali bahwa ia tidak perlu dihubungkan ke BDE. Pada kasus ini Tclientdataset mendapat data dari membuka paket data variant dari server.

MIDAS mengijinkan penggunaan semua standar Delphi komponen, namun sisi client tidak perlu diinstall Borland Database Engine (BDE), ODBC atau berbagai perangkat lunak middleware lainnya. Hal ini membuat letak dari server database dapat disembunyikan dari sisi client. Pada sisi client hanya diperlukan satu file berukuran kecil, kurang lebih 211 KB yang bernama dbclient.dll, tapi file ini termasuk sangat kecil jika dibandingkan dengan banyak megabyte file yang diperlukan untuk instalasi BDE.

3. PERENCANAAN SISTEM

3.1 Perencanaan Awal Sistem

Perencanaan awal yang dimaksudkan disini meliputi perencanaan *data flow* diagram, perencanaan *flow chart*, perencanaan entity relationship diagram, dan perencanaan struktur tabel.

Perencanaan data flow diagram dan flow chart dimaksudkan untuk memudahkan pembuatan program dengan membuat skema aliran input, proses, dan output dengan simbol – simbol yang sederhana dan mudah dimengerti, selain itu dapat digunakan untuk

dokumentasi suatu sistem karena merupakan media komunikasi yang baik.

Perencanaan Entity Relationship Diagram /ERD diperlukan agar dapat melihat jelas hubungan antara entity-entity dan mengetahui apakah bersifat obligatory atau non-obligatory. Hal ini berguna untuk merevisi atau melakukan perubahan pada sistem database yang sudah ada. Kemudian setelah itu perlu dilakukan perencanaan struktur tabel sehubungan dengan rancangan ERD yang telah dibuat. Dalam perencanaan struktur tabel ini, perlu ditentukan field-field apa saja yang diperlukan pada masing-masing database, berapa besar field, dan field yang merupakan primary key.

3.2 Perencanaan Aplikasi Program

Dalam proyek ini terdapat program aplikasi client windows, program aplikasi server, dan program untuk web server dengan teknologi ActiveX Form yang menggunakan client web browser.

Untuk program aplikasi client, berisi program yang merupakan presentation logic, meliputi tampilan form dan data, manipulasi data, penghapusan dan penyimpanan data, serta pencarian data. Program ini akan melakukan koneksi ke application server yang dituju dengan bantuan protokol DCOM. Dalam program ini tampak bahwa fungsi-fungsi query digantikan oleh suatu komponen bernama clientdataset. Client-dataset akan terhubung dengan application server lewat iprovider interfaces untuk menjalankan query ke database server. Misalnya untuk melakukan penambahan data dilakukan dengan prosedur berikut ini:

```
Procedure TFormData1.BitBtn3Click(Sender:
TObject); begin
  If pagecontrol1.ActivePage= tabsheet1
  then datamodule1.ClientDataSet1.
  insert;
  if pagecontrol1.activepage=tabsheet2
  then datamodule1.clientdataset2.
  insert;
  if pagecontrol1.activepage= tabsheet3
  then datamodule1.clientdataset4.
  insert;
end;
```

Kemudian data-data yang telah dimanipulasi meskipun telah di-post tidak akan dikirim ke database server, tapi disimpan sementara dalam change log lokal sampai

dilakukan perintah applyupdate, seperti contoh berikut :

```
procedure TForm2.BitBtn1Click(Sender:
TObject);
begin
  if (application.MessageBox('Save it
now?', 'Warning', MB_YESNO)=IDYES) then
    begin
      datamodule1.clientdataset5.Apply
      Updates(-1);
      datamodule1.clientdataset7.apply
      updates(-1);
      datamodule1.clientdataset8.apply
      updates(-1);
    end;
end;
```

Untuk program pada aplikasi server, akan memuat semua perintah-perintah query atau application logic. Komponen-komponen query ini terletak pada remote data module yang nantinya akan diekspor untuk membentuk provider. Fungsi untuk mendapatkan provider pada tiap query akan di-generate otomatis oleh delphi setelah sebuah query diekspor ke remote data module. Seperti contoh berikut ini :

```
function Tdm1.Get_Query1: IProvider;
begin
  Result := Query1.Provider;
end;
```

Untuk program aplikasi Web Server dibuat dengan membuat *ActiveForm*, yaitu kontrol *ActiveX* yang dapat diletakkan pada halaman html. Pembuatan program *Active Form* ini seperti pembuatan program pada form biasa. Setelah rancangan *ActiveForm* selesai, maka langkah selanjutnya adalah membuat file html yang memuat form tersebut dengan melakukan *Web Deploy* dan mengisi variabel yang diperlukan, seperti misalnya target direktori file OCX di server Web, target URL dari file OCX di server Web, dan directory dari file html yang akan menunjuk ke *ActiveForm* ini. Kemudian setelah proses *Web Deploy* selesai, akan terbentuk dua file, yaitu file OCX dan file html. Bagian dari file html, yang secara otomatis dihasilkan pada waktu melakukan deploy untuk mengakses file OCX, tampak pada program di bawah ini :

```
<p align="center">
<object classid="clsid:ACFB10E2-0BAF-
11D4-ACA0-00A0C989F570"
codebase="http://natanael/cehomepage/
default.ocx#version=1,0,0,0"
width="547" height="381">
```

```
align="center" hspace="0" vspace="0">
</object>
```

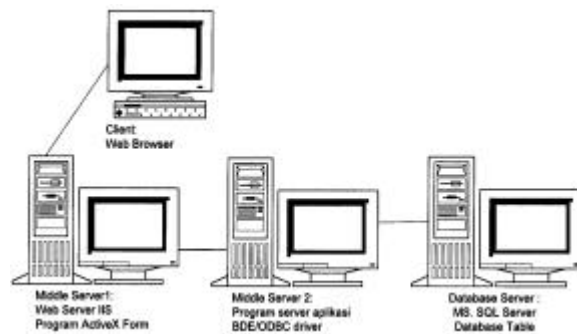
Pada program ini, CLSID yang diperlihatkan menspesifikasikan GUID dari object. Sedangkan baris bertuliskan code-base menunjukkan alamat tempat pada sistem dimana file OCX berada, atau merupakan Target URL yang diberikan pada waktu melakukan *deploy*.

4. IMPLEMENTASI SISTEM

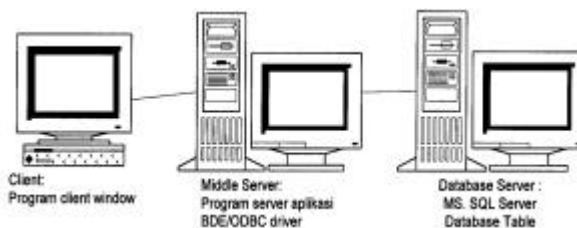
Untuk membuat sistem multitier ini, komputer client dapat menggunakan Windows 95/98 ataupun Windows NT Workstation. Sedangkan untuk middle server dan database server digunakan Windows NT Server. Namun perpaduan terbaik untuk sistem keamanan adalah menggunakan Windows NT Workstation dan Windows NT Server. Selain itu perlu dilakukan seting pada protokol-protokol komunikasi yang dipakai untuk menghubungkan masing-masing tier, yaitu seting protokol DCOM, TCP/IP, dan BDE/ODBC driver database. Pada proyek ini dibuat dua macam program dengan model yang berbeda, yaitu client/server windows model dan client/server web model. Window model memberikan akses yang lebih cepat, namun lingkupnya terbatas, sedangkan Web model memberikan akses dengan lingkup internet dengan menggunakan client web browser.

Untuk menjalankan program client /server web model, pada komputer client diinstal web browser (Universal Client), pada middle server 1 diinstal program aplikasi ActiveX Form, dan Web Server IIS, pada middle server 2 diinstal program aplikasi server dan BDE/ODBC driver, sedangkan pada database server adalah MS SQL Server dan database table. Sistem 4-Tier ini dapat digambarkan pada gambar 5.

Untuk menjalankan program client server window model, pada komputer client perlu diinstal program aplikasi client, kemudian pada middle server diinstall program aplikasi server dan ODBC driver, sedang pada database server diinstal program SQL Server dan database-database yang digunakan. Sistem 3-tier ini dapat dilihat pada gambar 6.



Gambar 5. Skema Koneksi Jaringan Web Model (4-tier)



Gambar 6. Skema Koneksi Jaringan Client Server Window 3-Tier

5. KESIMPULAN

1. Dengan menggunakan aplikasi multi-tier database, maka logika aplikasi dapat dipusatkan pada middle-tier, sehingga memudahkan untuk melakukan kontrol terhadap client-client yang mengakses middle server dengan mengatur seting pada dcomcnfg.
2. Dengan menggunakan aplikasi multi-tier, maka database driver seperti BDE/ODBC untuk mengakses database hanya perlu diinstal sekali pada middle server, tidak perlu pada masing-masing client.
3. Pada aplikasi multi-tier, logika bisnis pada middle-tier dapat digunakan lagi untuk mengembangkan aplikasi client lain, sehingga mengurangi besarnya program untuk mengembangkan aplikasi lain. Selain itu meringankan beban pada tiap-tiap mesin karena program terdistribusi pada beberapa mesin.

DAFTAR PUSTAKA

1. Berson, Alex. *Client/Server Architecture*. McGraw-Hill, Inc, 1992.
2. Calvert, Charlie. *Delphi 4 Unleashed*. Indiana Polis: Sams, 1999.
3. Clinton, Scott. *Developing Multi-tier Distributed Computing Architecture*. [<http://www.borland.com/delphi/papers/d323/d323tie2.html>], 1999.
4. *Developer's guide : Borland Delphi 4 for Windows 95 and Windows NT*. Scotts Valley, CA: Inprise Corporation, 1998.
5. Gruber, Martin. *Understanding SQL*. Sybex / Tech Asian Edition, 1990.
6. Papa, John and Matthew Shepker. *Microsoft SQL Server 7.0 Programming*. USA: Sams, 1999.
7. Pranata, Antony, *Pemrograman Borland Delphi*, Edisi 2, Yogyakarta: Andi, 1998.
8. Sampurna, *Belajar Sendiri Membuat Homepage dengan html*, Elekmedia Komputindo, Jakarta, 1996.