

PENGGUNAAN KONEKSI CORBA DENGAN PEMROGRAMAN MIDAS MULTI-TIER APPLICATION DALAM SISTEM RESERVASI HOTEL

Irwan Kristanto Julistiono

Fakultas Teknik Industri, Jurusan Teknik Informatika - Universitas Kristen Petra

e-mail: irwankj@peter.petra.ac.id

Hendra Lamidjan

Alumnus Fakultas Teknik Industri, Jurusan Teknik Elektro - Universitas Kristen Petra

ABSTRAK: Pada makalah ini dibuat suatu sistem multi-tier yang menggunakan teknologi CORBA untuk program reservasi hotel baik dengan web browser maupun program client.

Perangkat lunak yang dipakai sebagai database server adalah SQL server 7.0. Program Client Delphi melalui Corba Connection akan dihubungkan ke Aplikasi server. Dan melalui ODBC Aplikasi Server akan dihubungkan ke SQL Server 7.0. Ada dua buah aplikasi client yaitu yang menggunakan lokal network dan yang menggunakan global network/web browser. Pada pembuatan aplikasi client untuk web browser. Digunakan teknologi activex form pada delphi dimana sistem ini dibuat seperti membuat form biasa, hanya saja memiliki kekurangan pada integrasi dengan bahasa html.

Penggunaan sistem multi-tier dengan Corba ini secara umum memiliki keuntungan selain dapat dikembangkan lebih lanjut juga sistem ini dirancang dengan sistem multi database server, multi middle server, dan multi client dimana dengan hal ini seluruh sistem dapat terkait menjadi satu. Kelemahan sistem ini adalah struktur sistem corba yang rumit sehingga sulit dipahami, sedangkan untuk multi tier itu sendiri dibutuhkan suatu prosedur tersendiri untuk menentukan server mana yang menjadi pilihan dari client.

Kata kunci: CORBA, multi tier

ABSTRACT : *This paper is made from a multi-tier system using corba technology for hotel reservation program for web browser and also client program.*

Client software is connected to application server with Corba Connection and client and application server connect to SQL server 7.0. via ODBC. The are 2 types of client: web client and delphi client. In making web browser client application, we use delphi activex form technology, in where in this system made like making the regular form, but it has shortage in integration with html language.

Multi-pier application using corba system generally has another profit beside it could be developed, this system also stake with multi system database server, multi middle servers and multi client in which with these things all the system can system can be integrated. The weakness of this system is the complicated corba system, so it will be difficult to understand, while for multi-tier it self need a particular procedure to determine which server chossed by the client.

Keywords: CORBA, multi-tier.

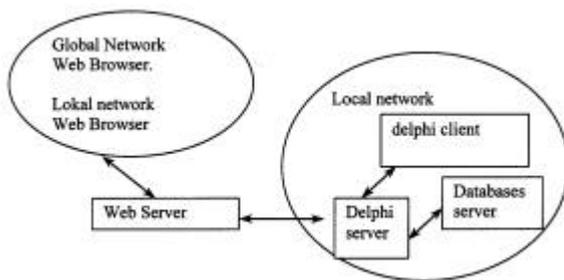
1. PENDAHULUAN

1.1 Latar Belakang Masalah

Semakin meningkatnya pemakaian komputer dewasa ini mendorong meningkatnya pula kebutuhan akan pemakaian system internet. Dan dengan berkembangnya pemakaian internet dewasa ini berkembang pula kebutuhan dan fungsi akan sistem ini. Salah satunya adalah berkembangnya pemakaian *network database system*.

1.2 Perumusan Masalah

Untuk mengatasi hal tersebut diatas maka dibuat *muli-tiered application* yang berdasarkan pada *Common Object Broker Architecture* (CORBA). CORBA merupakan spesifikasi yang diadopsi berdasarkan *Object Manajement Group* (OMG) untuk pengembangan distribusi *object* aplikasi yang kompleks.



Gambar 1. Rangkaian Hubungan antara Client dan Server

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah menerapkan teknologi MIDAS *multi-tier* aplikasi dengan menggunakan koneksi CORBA dengan cara membuat sistem aplikasi reservasi hotel yang berbasis *multi-tier server* dengan menggunakan sistem CORBA pada pemrograman Borland Delphi 4. Sehingga sistem reservasi yang akan dibuat dapat lebih baik, cepat dan lebih aman.

2. TEORI DASAR

2.1 Pengenalan *Object Request Broker*

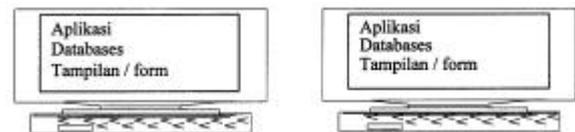
Ada empat bentuk kecenderungan *software* komersial masa depan. Pertama, industri *software* bergerak menjauh dari program aplikasi kecil ke aplikasi terintegrasi menggunakan komponen yang dapat digunakan kembali. Kedua, ada permintaan besar untuk teknologi distribusi untuk menyediakan pengiriman metode jarak jauh dan/atau orientasi pesan *middleware* untuk memudahkan kerjasama aplikasi. Ketiga, ada peningkatan usaha untuk menentukan standart *software* kerangka infrastruktur yang mengijinkan aplikasi untuk bekerja tanpa gangguan melalui *enviroment* yang beragam. Akhirnya, generasi berikutnya dari aplikasi distribusi seperti *video-on-demand*, *teleconferencing*, dan *avionics* membutuhkan *quality-of-service* (QoS) dimana menjamin untuk keadaan *bandwith* tertekan, dan kemudahan kontrol.

Kunci dari teknologi *software* yang sedang trend ini adalah perangkat penengah *distributed object computing* (DOC), perangkat penengah DOC menjembatani lokal dan remote komponen aplikasi dalam lingkungan distribusi yang heterogen.

Tujuan dari perangkat menengah DOC adalah untuk menghilangkan kejenuhan, kemungkinan salah dan aspek yang tidak *portable* dari pengembangan aplikasi dan service distribusi. Dalam kenyataannya perangkat penengah DOC mengotomatiskan tugas-tugas umum *network programming*. Seperti lokasi objek, implementasi awal (server dan pengaktifan objek), membuka selubung dari permintaan *byte* dan *type* parameter seleksi ukuran melalui perbedaan arsitektur (parameter yang memimpin), perbaikan kesalahan dan keamanan. Inti dari perangkat penengah DOC adalah ORB seperti CORBA, DCOM dan JAVA RMI.

2.2 Arsitektur Komputer Database

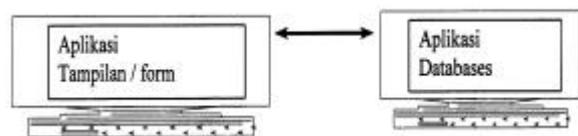
- **Lokal Database**



Gambar 2. Lokal Database Architecture

Pada arsitektur lokal database ini semua informasi data maupun program aplikasi terletak pada satu komputer, sehingga pengolahan database maupun program aplikasi hanya dapat dilakukan oleh satu user saja.

- **Two-tier Arsitektur**



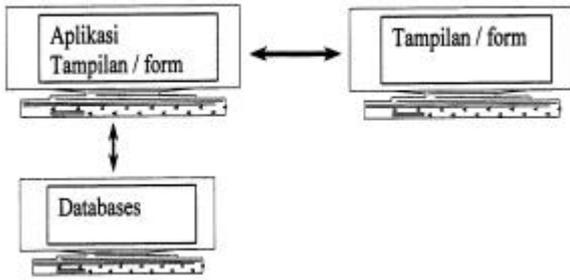
Gambar 3. Two-tier Client/Server Architecture

Pada arsitektur *two-tier client/server* ini database dipisahkan dengan aplikasi *client* sehingga database dapat diakses oleh banyak user selama ada program aplikasi pada komputer yang bersangkutan. Dalam hal ini komputer *client* terjadi proses, untuk menjalankan aplikasi dari program.

- **Three-Tier Architecture**

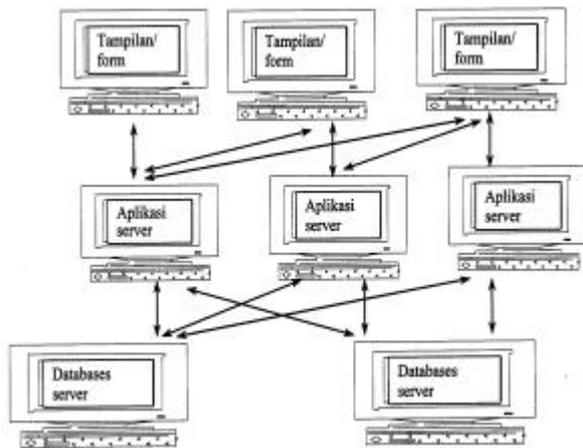
Pada arsitektur *three-tier client server* sudah dipisahkan antara database, aplikasi dan *client*. Di mana dalam hal ini *client* hanya berupa tampilan sedangkan beban

program aplikasi ditanggung oleh komputer *middle server* / aplikasi server. Dan untuk beban database di tangani oleh database server.



Gambar 4. Three-Tier Client/Server Architecture

• Multi-Tier Architecture



Gambar 5. Multi-tier Client/Server Architecture

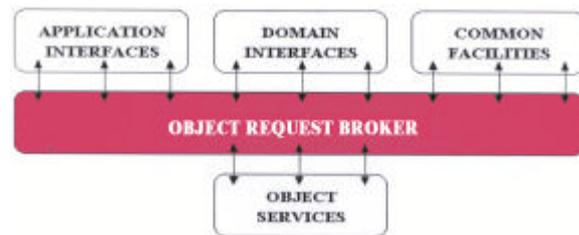
Pada arsitektur multi-tier ini client server ini beban dari database server, maupun aplikasi server dapat ditanggung oleh beberapa komputer / server sehingga lebih ringan dan dapat menangani *client* dengan *system load balancing*. Dalam arsitektur ini *client* dapat meminta data dari beberapa komputer *middle server*, dan tiap komputer *middle server* itu dapat menjalankan permintaan data dari beberapa database server sekaligus. Arsitektur ini lebih diarahkan untuk pengolahan database server dan aplikasi server yang terdistribusi dan digunakan oleh banyak *user client* dari berbagai tempat.

2.3 Sejarah Singkat Corba

Corba adalah singkatan dari *Common Object Request Broker Architecture*. CORBA dibuat oleh OMG (*Object Manage-*

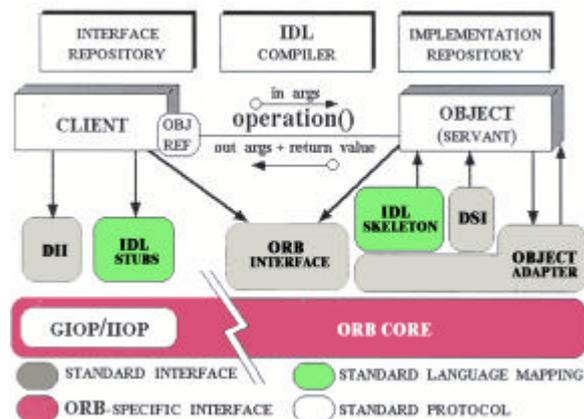
ment Group). Tujuan utama pembuatan teknologi CORBA dari OMG adalah untuk membuat program yang berdasarkan objek orientasi, *cross-platform*, *distributed architecture* yang dapat ditempati oleh server dan *client* dalam jumlah besar dan dalam skala baik bahkan dalam tekanan yang luar biasa. Biasanya pengembang dari CORBA mendesain dengan spesifikasi untuk mendukung jutaan server dan puluhan juta dari simultan user. Teknololgi CORBA memiliki arsitektur yang sangat kuat untuk kebutuhan dari perusahaan yang besar. OMG merupakan gambaran dari spesifikasi CORBA, sedangkan perusahaan seperti *Visigenic* dan *IONA* sebenarnya merupakan implementasi dari CORBA.

• Arsitektur Dasar CORBA Model OMG



Gambar 6. Arsitektur Model OMG

• Arsitektur CORBA ORB



Gambar 7. Arsitektur CORBA ORB

2.4 Perbandingan antara CORBA dengan DCOM dalam Delphi

Dalam banyak cara teknologi ini berjalan secara paralel antara yang satu dengan yang lain dan tugas yang dijalankan juga sangat mirip. Keduanya baik COM dan CORBA menyediakan alat untuk membuat distribusi, dari arsitektur yang berorientasi pada objek

Dengan kata lain keduanya menyediakan alat untuk objek yang berada pada *binary-executable* pada mesin lain. COM dan CORBA memiliki berbagai macam *utilities* yang mendukungnya seperti MTS, ITS, brokers, registeries, dan lainnya. Dan setiap arsitekturnya memiliki kelebihan/keuntungan sendiri-sendiri.

COM, ditujukan pada *Microsoft* dan teknologi yang didasarkan pada *Windows* dimana terintegrasi pada 90 persen mesin dekstop di dunia. Dan tersedia secara gratis yang didukung oleh vendor yang luas.

CORBA, memiliki kelebihan pada penyediaan waktu kerja yang mendukung waktu hidup 24 jam sehari dan 7 hari dalam seminggu. Dan lebih baik dalam mendukung *fail over* dan *load distribution* dibandingkan COM. Dengan menggunakan *protocol standard IIOP*, program yang berdasarkan teknologi CORBA dari berbagai macam vendor, pada hampir semua jenis komputer, *operating system* dan bahasa pemrograman, dari vendor yang sama atau vendor lain, dan jaringan komputer dapat diterjemahkan dengan program dengan teknologi CORBA dari dari berbagai macam vendor, pada hampir semua jenis komputer, *operating system* dan bahasa pemrograman, dari vendor yang sama atau vendor lain dan jaringan komputer lain.

Delphi sekarang menyediakan komponen *Tsimple Object Broker* untuk menangani *load balancing* dan *fail over* untuk objek DCOM. Dan tentunya Corba memiliki pendukung yang lebih baik menangani *operating system* yang lebih luas dibandingkan COM.

Kedua arsitektur ini memiliki pendukung masing-masing dimana mereka mengklaim bahwa arsitektur mereka memiliki paling banyak pendukung. Dimana hal ini COM memiliki pendukung yang luas sedangkan CORBA didukung oleh perusahaan – perusahaan besar seperti SUN, Netscape, IBM dan digunakan oleh group perusahaan–perusahaan besar dimana membutuhkan jaringan data yang terintegrasi dari banyak server dan banyak user.

3. PERENCANAAN SISTEM INFORMASI

3.1 Metode Rancangan Sistem Informasi

Sejalan dengan berkembangnya sistem informasi dewasa ini penggunaan *database system* juga ikut berkembang dengan mengubah penggunaan *lokal database* ke penggunaan *network database* baik dengan server database tunggal yang kemudian terus berkembang ke *multi-tier client server*, dimana terdiri dari banyak database server dan banyak aplikasi server juga banyak *client* yang semuanya terhubung dalam suatu sistem yang terintegrasi. Dalam pembuatan program reservasi hotel ini digunakan database server yang berbasis SQL server 7.0 sedangkan aplikasi server dan *client* servernya menggunakan *delphi client server* 5.0. Dengan demikian data pada database server dapat diakses oleh banyak user dengan data yang *ter-update*.

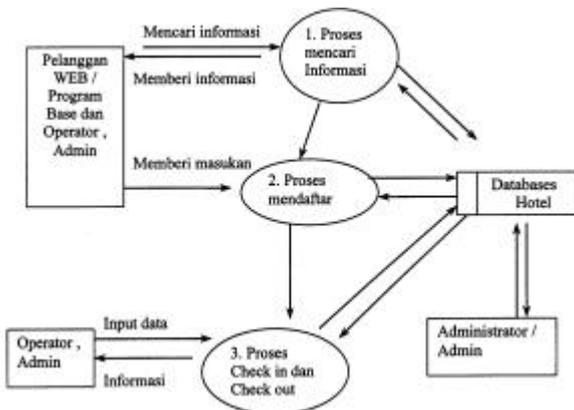
3.2 Sistem Reservasi Hotel

Sistem reservasi hotel dirancang untuk mengatur pemesanan kamar agar dapat melakukan pemesanan sesuai dengan kriteria dari kamar dan juga waktu *check in/check out*, maupun dapat diketahui nomer kamar yang tidak digunakan pada waktu tertentu. Dan sistem reservasi ini digunakan pula untuk memudahkan seseorang melakukan pemesanan kamar hotel dan juga untuk memudahkan operator dari hotel melakukan pengaturan pemesanan kamar hotel sesuai dengan kriteria kamar dan waktu yang diinginkan. Reservasi hotel selain dilakukan oleh operator juga dapat dilakukan melalui web, dalam hal ini sangat tergantung dengan kebijaksanaan perusahaan dalam menyikapi validitas/keamanan melalui web. Tetapi pada umumnya untuk melakukan reservasi melalui web, tetap harus dilakukan reservasi ulang oleh operator hotel. Sehingga fungsi dari web reservasi lebih mirip pada pemberitahuan oleh pelanggan pada pihak hotel, dan transaksi yang sesungguhnya belum benar-benar terjadi, sebelum konfirmasi oleh operator hotel dilakukan dan sebelum pembayaran oleh pelanggan pada pihak hotel dilakukan. Untuk reservasi yang dilakukan oleh operator hotel validitasnya

dapat lebih baik karena diharapkan pada saat reservasi terjadi, dapat langsung dilakukan transaksi/pembayaran kamar yang dipesan. Dan dalam hal ini operator juga dapat melakukan pengecekan langsung tentang identitas pelanggan. Pembuatan suatu sistem reservasi sebuah hotel sangat tergantung pada kebijaksanaan dari manajemen hotel, antara lain boleh tidaknya reservasi dilakukan tanpa adanya pembayaran terlebih dahulu, batas waktu pemesanan kamar, penerapan standard waktu *check in/check out*, kebijaksanaan/metode pembayaran, penting tidaknya pengecekan identitas pelanggan, dan kebijaksanaan-kebijaksanaan lain yang berkenaan dengan waktu – waktu tertentu/spesial.

3.3 Dataflow Diagram Reservasi Hotel

Dataflow diagram memudahkan orang untuk mengenali dan cara kerja dari sistem aplikasi yang dibuat, karena dokumentasi yang dibuat dapat diatur sesederhana mungkin, tanpa perlu menyertakan *field-field* tabel maupun tabel-tabel yang terdapat pada database. Sehingga pada *data flow* diagram ini yang ditampilkan hanyalah proses secara sederhana jalan kerja dari aplikasi yang dibuat nantinya.



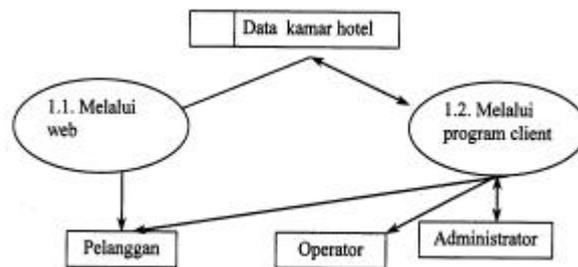
Gambar 8. DataFlow Diagram Reservasi Hotel Level 0

Pada Gambar 8 dapat dilihat bahwa pelanggan dapat melakukan pencarian informasi tentang hotel ataupun dapat melanjutkan ke proses pendaftaran dan *check in/check out* yang semuanya di hubungkan ke dalam database hotel. Dalam hal ini pelanggan yang sudah mengetahui informasi

tentang hotel dapat langsung melakukan proses pendaftaran, *check in* dan *check out*, dalam hal ini transaksi sudah terjadi. Sedangkan pendaftaran melalui internet dimana belum terjadi transaksi hanya berhenti pada proses pendaftaran.

Hubungan antara proses pencari informasi dan database hotel adalah dimana proses meminta informasi yang dibutuhkan pelanggan dan database memberikan informasi yang dibutuhkan pelanggan secara *online*.

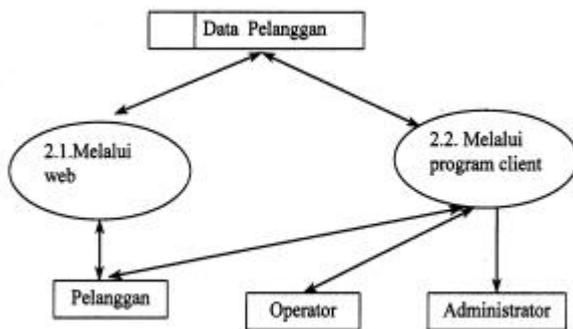
Hubungan antara proses pendaftaran dan database hotel adalah dimana proses daftar memberikan informasi tentang identitas pelanggan dan informasi tentang rencana waktu *check in* maupun rencana *check out*, dalam hal ini database kemudian melakukan pengolahan data untuk memberikan persetujuan pendaftaran. Hubungan antara proses *check in/check out* dengan database hotel adalah dimana proses memberikan informasi tentang waktu *check in*, *check out* dan data tentang kamar yang dipesan sedangkan database melakukan pengolahan informasi yang diberikan dan memberikan informasi tentang biaya dan persetujuan kamar kepada proses untuk diberikan kepada pelanggan.



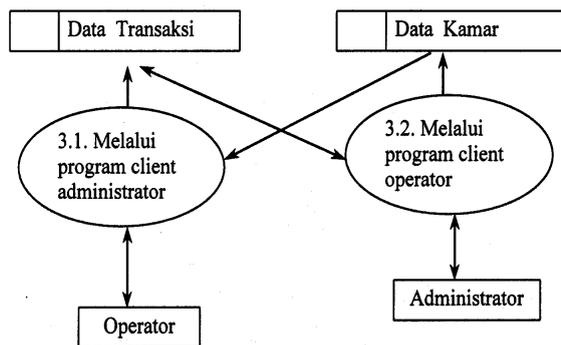
Gambar 9. DataFlow Diagram Level 1 Proses Mencari Informasi

Pada Gambar 9 ini dapat dilihat bahwa database kamar dapat diakses melalui web maupun program *client*, untuk login operator dan pelanggan hanya dapat membaca sedangkan login administrator dapat melakukan baca tulis dan hapus.

Pada Gambar 10 dapat dilihat bahwa database pelanggan dapat diakses melalui web maupun program *client*, untuk login pelanggan hanya dapat membaca sedangkan login operator dan administrator dapat melakukan baca tulis dan hapus.



Gambar 10. DataFlow diagram Level 1
Proses Pendaftaran



Gambar 11. DataFlow diagram Level 1
Check In/Check Out

Pada Gambar 11 dapat dilihat bahwa database transaksi dan kamar dapat diakses melalui program *client* saja, untuk login pelanggan tidak memiliki akses sedangkan login operator dan administrator dapat melakukan baca tulis dan hapus.

4. PENGUJIAN SISTEM JARINGAN

Pada pengujian sistem jaringan ini digunakan sistem jaringan di Universitas Kristen Petra dimana menggunakan *real IP* dan *fake IP*. Sedangkan *operating system* yang diuji adalah *windows 95/98*, *windows NT server*, *Windows NT Workstation*, *Windows 2000 server*. Pada sistem jaringan *multi-tier* ini dibutuhkan database server yang mana dalam hal ini digunakan SQL server 7 database, dibutuhkan juga aplikasi server/middle server dimana dalam hal ini menggunakan aplikasi *delphi client server 5* dan *patch-nya* untuk koneksi Corba. Untuk selanjutnya yang dibutuhkan adalah web server dalam program ini digunakan IIS4 yang terdapat pada *Windows NT server 4 option pack*.

Dalam penelitian ini dikarenakan untuk memudahkan pengaturan dan pengecekan *real time database online* maka untuk dua hal di atas yaitu database server, aplikasi server dibuat dan di set tersendiri pada satu komputer dimana dalam hal ini menggunakan *real IP* 202.43.253.8 dengan nama komputer *thenetone* dengan domain *network* yang berbasis pada OS *Windows NT server 4 service pack 5*. Meskipun demikian pada proses pembuatan aplikasi selanjutnya dapat diubah sesuai dengan kebutuhan dan kondisi yang ada. Dalam hal ini hubungan antara database server dan aplikasi server hanya menggunakan ODBC dimana hal ini merupakan hal umum digunakan pada proses pembuatan database program, yang perlu diperhatikan adalah pada saat penggunaan *32 bit ODBC administrator type* data pada data source yang digunakan harus sama dalam hal ini dipilih SQL server, *login/password* pada SQL server yang mempunyai akses ke database dan lokasi SQL servernya menggunakan *IP address* dan bukan *lokal* sehingga seakan-akan berada pada suatu jaringan. Untuk hal ini yang perlu diperhatikan bahwa *network* komputer haruslah berjalan dengan baik karena database server dianggap pada suatu jaringan dan bukan pada *lokal* komputer.

Sedangkan untuk *client-nya* dapat diakses pada komputer-komputer lain karena tidak dibutuhkan pengaturan yang rumit pada komputer *client*. Untuk *client* yang menggunakan web, hal yang perlu diperhatikan adalah properties pada Internet Explore di *setting* agar boleh menjalankan ActiveX.

5. KESIMPULAN DAN SARAN

Pemakaian *system multi-tier client server* Corba ini juga dapat menjadikan administrator dari server mudah melakukan manajemen karena perbaikan sistem dapat dilakukan pada *middle server*.

Melalui *system multi-tier client server* ini keamanan data dapat di atur melalui banyak tingkatan, mulai dari *database server security*, *program middle server*, bisa juga diatur pada aplikasi *client* yang dipakai baik web based maupun *Windows*.

Untuk *system multi-tier* Corba ini pengaturan database dilakukan pada server database SQL server 7 dan server aplikasi Corba, sehingga *client* dalam hal ini tidak membutuhkan pengaturan dan tidak mengetahui tentang database yang digunakan, letaknya, maupun *system* dan *security* yang dipakai.

Pada pemrograman *Delphi Client Server* 5 tersedia *ActiveX Form* dimana hal ini akan memudahkan programmer untuk membuat halaman/*form Delphi* yang dapat dipakai pada halaman web, tanpa perlu menambahkan script Java, Perl maupun CGI. Pemakaian *ActiveX* pada *delphi* juga mengandung dampak negatif yaitu tidak atau sulitnya integrasi dengan bahasa html itu sendiri, dan dibutuhkannya *setting security* yang lebih rendah agar OCX atau CAB dari *delphi* dapat terinstall di komputer client untuk pertama kalinya.

Pada penelitian ini hanya dilakukan uji coba secara vertikal hubungan multi-tier dalam koneksi corba. Sedangkan untuk hubungan secara horisontal yang melibatkan beberapa database server, middle server dan client dalam jumlah besar tidak dilakukan uji coba. Sedangkan untuk uji coba dua client yang masuk secara bersamaan sudah berhasil dilakukan uji coba.

Dalam sistem reservasi hotel dengan web base sudah dapat dikatakan memenuhi syarat. Dan ditambahkannya fungsi – fungsi yang bermanfaat dalam log informasi maupun *server time* dan *message online*. Untuk sistem dari program *Windows based* belum bisa dikatakan memenuhi syarat untuk dilakukan reservasi yang sesungguhnya, karena tidak adanya aturan–aturan yang jelas dalam proses yang akan dilakukan mulai dari awal sampai dengan akhir dari proses reservasi itu sendiri. Hal tersebut dikarenakan kurangnya data–data informasi yang dibutuhkan dalam proses reservasi yang sesungguhnya dilakukan oleh operator maupun administrator hotel.

Progam penelitian ini secara real hanya dapat diterapkan untuk program *web based*-nya saja sedangkan untuk program *Windows based* perlu dilakukan perbaikan maupun perluasan cakupan dari sistem yang ada sehingga dapat tercipta *system administrasi hotel* yang lebih baik, dengan kata lain tidak dapat hanya dibatasi pada reservasi-nya saja.

DAFTAR PUSTAKA

1. Berson, Alex. *Client/Server Architecture*. McGraw-Hill, Inc, 1992.
2. Calvcrt, Charlic. *Delphi 4 Unlcashed*. Indiana Polis: Sams, 1999.
3. Object Management Group/OMG, Corba Basic. [<http://www.omg.org>]
4. Kate Keahey. A Brief Tutorial on CORBA [<http://www.cs.indiana.edu/hyplan/kksiazck/tuto.html>]
5. Clinton, Scott. Developing Multi-tier Distributed Computing Architecture. [<http://www.borland.com/dclphi/papers/d323/d323tie2.html>], 1999.
6. *Developer's guide: Borland Delphi 4 for Windows 95 and Windows NT*. Scotts Valley, CA: Inprisc Corporation, 1998.
7. Papa, John and Matthew Shepker. *Microsoft SQL Server 7.0 Programming*. USA: Sams, 1999.
8. Pranata, Antony, *Pemrograman Borland Delphi*, Edisi 2, Yogyakarta: Andi, 1989.
9. *Common Object Services Specification*, OMG 95-3-31, 1995.
10. Steve Vinoski, CORBA: Integrating Diverse Applications Within Distributed Heterogrencous Environments, *IEEE Communications Magazine*, February, 1997.
11. Nanbor Wang, Kirthika Parameswaran, and Douglas C. Schmidt, *The Design and Performance of Meta-Programming Mechanisms for Object Request Broker Middleware*, Submitted to the 6^{th} *USENIX Conference on Object-Oriented Technologies and Systems* (COOTS), Dan Antonio, Jan/Feb, 2001. [<http://www.cs.wustl.edu/~schmidt/corba.html>]
12. Applying Patterns to Develop Extensible ORP Middleware (*Updated May 16th*), Appeared in the *IEEE Communications Magazine Special Issue on Design Patterns* (with Chris Cleeland) [<http://www.cs.wustl.ed/~schmidt/corba.html>].