Jurnal Ilmiah KURSOR
Menuju Solusi Teknologi Informasi
Vol. 4, No. 2, 2008
ISSN 0216 - 0544

# PENGEMBANGAN SISTEM *E-AUCTION* UNTUK BARANG ELEKTRONIK BERBASIS PADA ARSITEKTUR CORBA

**Nunut Priyo Jatmiko*, Waskitho Wibisono, IGP Remon AP**
Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember
Gedung Teknik Informatika, Kampus ITS, Jl. Raya ITS, Sukolilo, Surabaya – 60111
email : *nunutpj@cs.its.ac.id

**Abstract**

*The information technology offers a lot of benefits which are the flexibility of place and the effectiveness of time. The information technology could be implemented as a medium between the seller and the trader, so the transaction become much more flexible and can cover broader space.*

*Based on the benefits that offer by the information technology the writer try applying the information technology in one particular form of trading which is calls auction. Auction generally performed in a particular place and in particular time. The technology, that will be applied in this auction system is CORBA, which is a software architecture based on client-server paradigm. With the implementation of CORBA the heterogeneity between client and server can be handled. Both, the client and the server can be implemented in different hardware, different operating system and also different location. However, both of them still can communicate to each other.*

*Hopefully the implementation of CORBA in auction able to omit binding of place and time and also can extend the scope of the auction.*

***Keywords*** *: CORBA ,middleware, ORB*

## 1. Pendahuluan

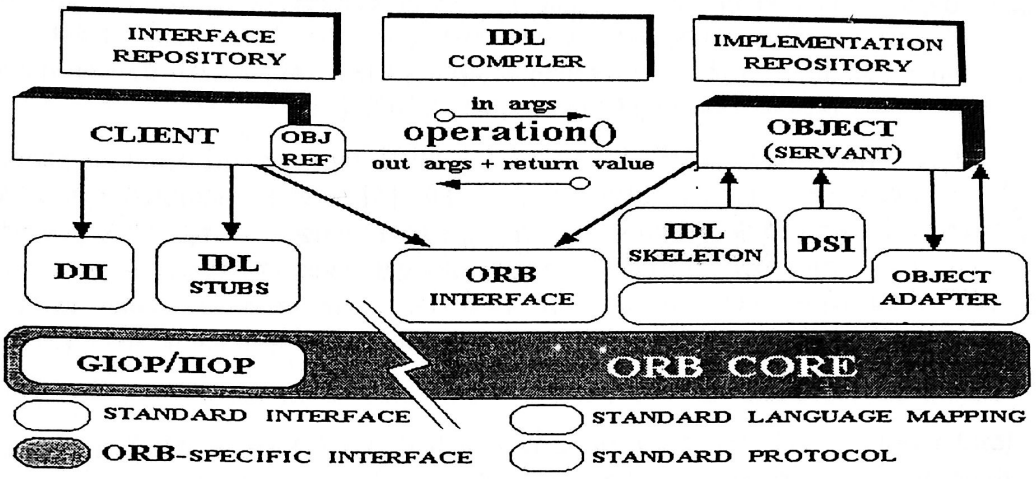
Pelelangan merupakan suatu bentuk perdagangan yang unik, pesertanya pun terbatas pada orang-orang yang datang ke tempat pelelangan. Oleh karena itu pelelangan sangatlah terikat oleh tempat dan waktu, karena para peserta pelelangan haruslah berada di suatu tempat dan waktu tertentu untuk dapat mengikuti suatu pelelangan.

Oleh karena itu pada penelitian ini dicoba untuk dibuat suatu sistem pelelangan online dengan memanfaatkan teknologi CORBA (*Common Object Request Broker*). CORBA merupakan suatu arsitektur *software* yang berorientasi pada object, dengan CORBA ini memiliki kemampuan untuk menangani heterogenitas yang dimiliki aplikasi-aplikasi yang membangun suatu sistem, sehingga meskipun aplikasi-aplikasi tersebut dibuat dengan bahasa pemrograman yang berbeda, dan beroperasi pada lingkungan yang berbeda semua aplikasi tersebut tetap dapat saling bekerjasama (interoperabiiitas). Interoperabilitas dalam CORBA terjadi karena arsitektur CORBA mengimplementasikan *middleware* yang disebut dengan ORB (*Object Request Broker*), ORB inilah yang menjadi jembatan penghubung antara aplikasi-aplikasi yang dibangun dengan bahasa pemrograman yang berbeda, ORB lah yang melakukan pelacakan *object* aplikasi *server* dan juga meng-*invoke method* yang terdapat pada *object* tersebut. Aplikasi *client* dan *server* tidak berhubungan secara langsung dengan ORB, melainkan melalui *stub* dan *skeleton*, kedua perantara ini dikenal dengan sebutan *proxy*. Baik *Stub* maupun *Skeleton* merupakan hasil *generate* dari IDL, IDL berisikan definisi tentang *interface* apa yang tersedia pada *client* dan *server*. Catatan bahwa IDL hanya digunakan untuk menggambarkan *interface*, dan bukan sebuah implementasi.

## 2. CORBA

CORBA (*Common Object Request Broker*) merupakan suatu standar sistem terdistribusi yang dikembangkan oleh OMG (*Object Management Group*), yaitu sebuah konsorsium yang terdiri lebih dari 800 perusahaan untuk membantu dalam pemrograman objek-objek terdistribusi. CORBA

merupakan sebuah cara bagi objek – objek untuk melakukan interoperabilitas lintas jaringan. Sistem yang dibangun dengan standarisasi CORBA memiliki interoperabilitas yang tinggi karena sistem tersebut nantinya akan mampu menghadapi heterogenitas baik dari segi bahasa pemrograman yang digunakan maupun juga heterogenitas *platform* [1]. Hal inilah yang menjadi kelebihan CORBA karena CORBA mampu mengkomunikasikan sistem yang memiliki perbedaan – perbedaan tersebut. Berikut dapat dilihat gambar arsitektur CORBA .




Gambar 1. Arsitektur CORBA

CORBA memiliki paradigma *client-server*, dimana *client* akan mengakses *method – method* yang disediakan oleh *server*. *Method* yang terdapat pada *server* seolah–olah diakses secara lokal oleh *client*, hal ini dikarenakan adanya *client stub* dan *server skeleton* yang merupakan hasil *generate* dari IDL yang berisikan *interface – interface method* yang terdapat di *server* [4]. Arsitektur CORBA memiliki bagian utama yaitu perangkat lunak perantara (*middleware*) yang disebut ORB (*Object Request Broker*), IDL (*Interface Definition Language*), DII (*Dynamic Invocation Interface*), IR (*Interface Repositories*), OA (*Object Adapters*) yang bersama – sama membentuk suatu *software bus* yang memfasilitasi interoperabilitas dan integrasi dari suatu sistem yang memiliki heterogenitas bahasa pemrograman dan platform. Berikut penjelasan mengenai bagian – bagian utama CORBA

- ORB : sering disebut juga *middleware* yaitu merupakan suatu jembatan atau penghubung antara dua aplikasi atau lebih yang memiliki perbedaan. ORB memungkinkan suatu *object* yang berada di *client* mengirim *message* ke suatu *method* yang ter-*encapsulated* oleh suatu *object* yang berada di *server*
- IDL : Dalam IDL hanya menyebutkan tentang metode – metode beserta parameter - parameter yang digunakan metode tersebut, tidak menjelaskan detail ataupun implementasi dari metode – metode tersebut. Melalui IDL,implementasi object tertentu memberitahukan kepada clientnya operasi – operasi apa saja yang disediakan oleh *object* tersebut dan juga scara meng-*invoke* operasi tersebut.
- DII : merupakan suatu metode pemanggilan (invoke) operasi – oprasi yang terdapat pada object server. Dengan menggunakan DII sistem menjadi lebih fleksibel sebab *client* dapat mengenali operasi – operasi baru pada *server* tanpa terlebih dahulu harus melakukan perubahan pada IDL *Interface*.
- IR : merupakan komponen dari ORB yang menyimpan definisi interface, melalui IR aplikasi *client* dapat menentukan lokasi dari *object server*, mengetahui informasi tentang parameter – parameter yang digunakan oleh *object* tersebut, dan akhirnya membuat suatu *request* ke *object server* tersebut melaui ORB.
- OA : merupakan jalan utama bagi suatu implementasi *object* untuk mengakses *service – service* yang disediakan oleh oleh ORB

## 2.1. CORBA dan JAVA

- J2SE 1.4 menyertakan ORB dan 2 buah programming model yang dapat digunakan untuk mengimplementasikan *framework* CORBA. Berikut penjelasan mengenai 2 programming model tersebut :
- RMI-IIOP : memungkinkan untuk mengimplementasikan *framework* CORBA melalui RMI API. RMI-IIOP menggunakan ORB dan juga IIOP sebagai protokol komunikasi, sehingga dapat dibuat suatu aplikasi menggunakan bahasa pemrograman JAVA dan kemudian digunakan rmic *compiler* untuk men-*generate* code yang dapat digunakan untuk menghubungkan aplikasi kita dengan aplikasi lain yang dibuat dengan bahasa pemrograman lain yang juga mendukung CORBA melaui IIOP [2]
- JAVA IDL : terdiri dari JAVA CORBA ORB dan juga idlj *compiler* yang digunakan untuk men-*generate stub* dan *skeleton* dari IDL [3]. Untuk menggunakan JAVA IDL terlebih dahulu harus didefinisikan *remote interface* dengan menggunakan OMG IDL, barulah kemudian *interface* tersebut di compile menggunakan idlj *compiler*. Hasil *compile* dari OMG IDL ini akan menghasilkan *file stub* dan *skeleton* dalam versi java, file inilah yang selanjutnya akan menjadi pedoman dalam pembuatan aplikasi tersebut [5].

## 2.2. CORBA dan .NET

Implementasi Corba di .NET tidak semudah di Java. Karena .NET tidak menyediakan kelas yang dapat langsung dipakai (di Java tinggal dipakai idlj.exe atau rmic.exe untuk *generate class* dan *stub* dari sebuah idl). Dan untuk sistem ini penulis menggunakan library .NET untuk CORBA yang disebut dengan IIOP.NET yang dapat didownload di http://iiop-net.sourceforge.net/. IIOP.NET merupakan suatu *remoting channel .NET* yang menggunakan IIOP sebagai protocol komunikasinya dimana IIOP merupakan protocol komunikasi standar dari CORBA [2]. IIOP.NET mengubah data – data dari sistem .NET menjadi data – data yang sesuai dengan spesifikasi CORBA dan sebaliknya, sehingga memungkinkan aplikasi CORBA yang dibangun menggunakan IIOP.NET dapat berkomunikasi dengan aplikasi CORBA lainya yang dibuat dengan bahasa pemrograman yang berbeda.

## 3. Arsitektur Sistem

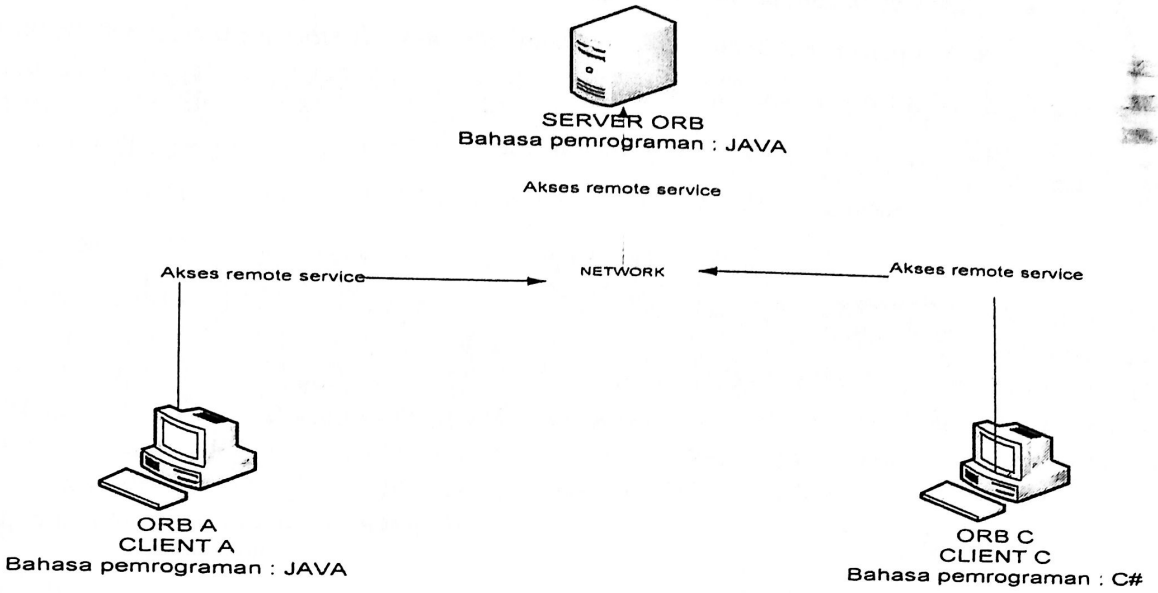
Terdapat dua aplikasi yang dibangun pada sistem ini. Yaitu aplikasi *client* dan aplikasi *server*. Aplikasi *client* merupakan aplikasi yang digunakan oleh *user* yang bertindak sebagai penjual maupun penawar, melalui aplikasi inilah seorang *user* akan mendaftarkan suatu item ataupun melakukan penawaran terhadap item – item tertentu. Aplikasi *client* ini diimplementasikan dengan 2 bahasa pemrograman yaitu JAVA dan C#.NET, aplikasi ini akan terpusat pada satu *server* yang diimplementasikan dengan bahasa pemrograman JAVA. Komunikasi antara *client* yang diimplementasikan dengan C# dengan *server* yang diimplementasikan menggunakan JAVA dapat terjadi berkat penggunaan *framework* CORBA.

- Skenario pencarian pelelangan

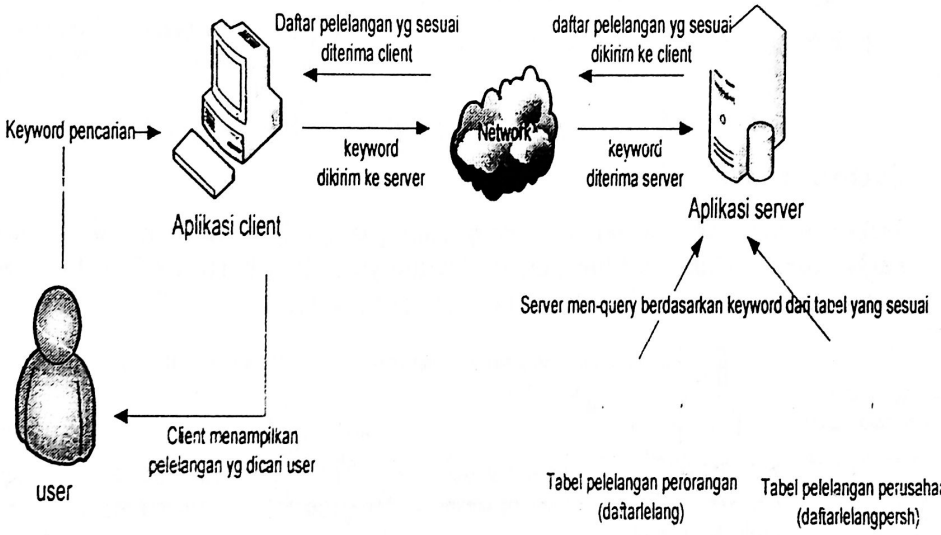
  Proses pencarian pelelangan dilakukan dengan menggunakan *keyword* yang dimasukkan *user*. *Keyword* ini nantinya akan dicocokkan dengan nama – nama dari pelelangan yang terdaftar pada *database* pelelangan. Pencarian hanya dilakukan pada pelelangan yang memiliki status dibuka, jika suatu pelelangan sudah ditutup maka pelelangan itu tidak akan diikutkan dalam pencarian. Pencarian pelelangan diilustrasikan pada Gambar 3.

- Skenario penawaran

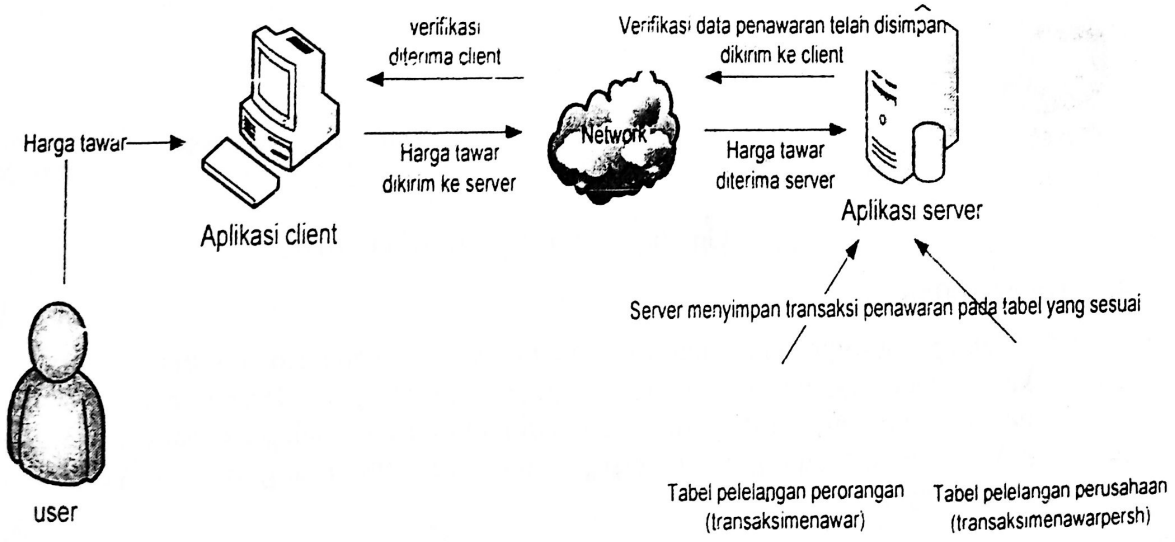
  Proses penawaran selalu diawali dengan proses pencarian pelelangan, sebab dari hasil pencarianlah seorang *user* dapat memilih item mana yang akan dia tawar. Dalam melakukan penawaran seorang harus memperhatikan harga tawar terendah dari suatu item, *user* harus memberikan harga tawar lebih tinggi dari harga tawar terendah, barulah penawaran *user* tersebut akan dimasukkan dalam *database*.




Gambar 2. Arsitektur sistem pelelangan



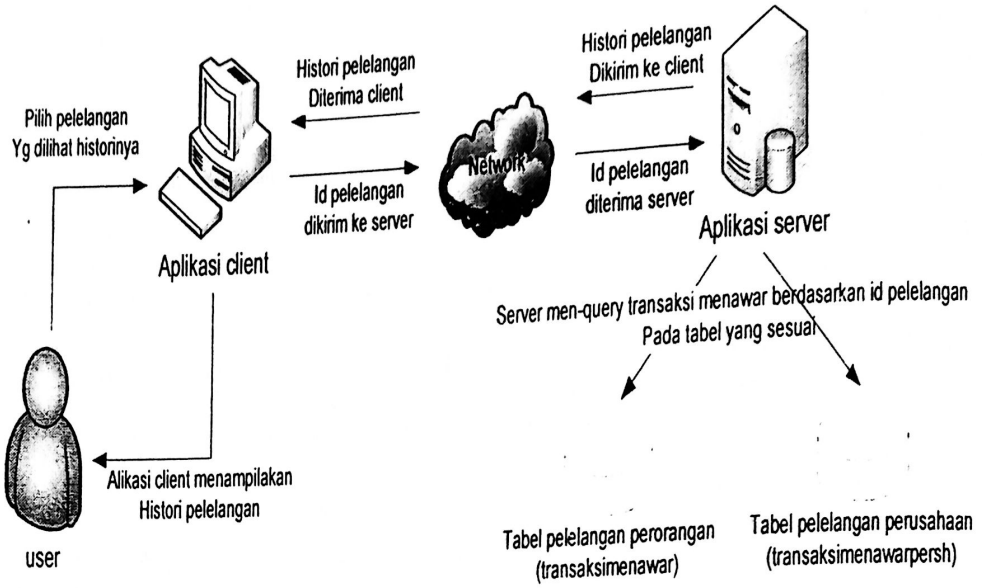

Gambar 3. Pencarian pelelangan




Gambar 4. Proses penawaran

- Skenario melihat histori pelelangan

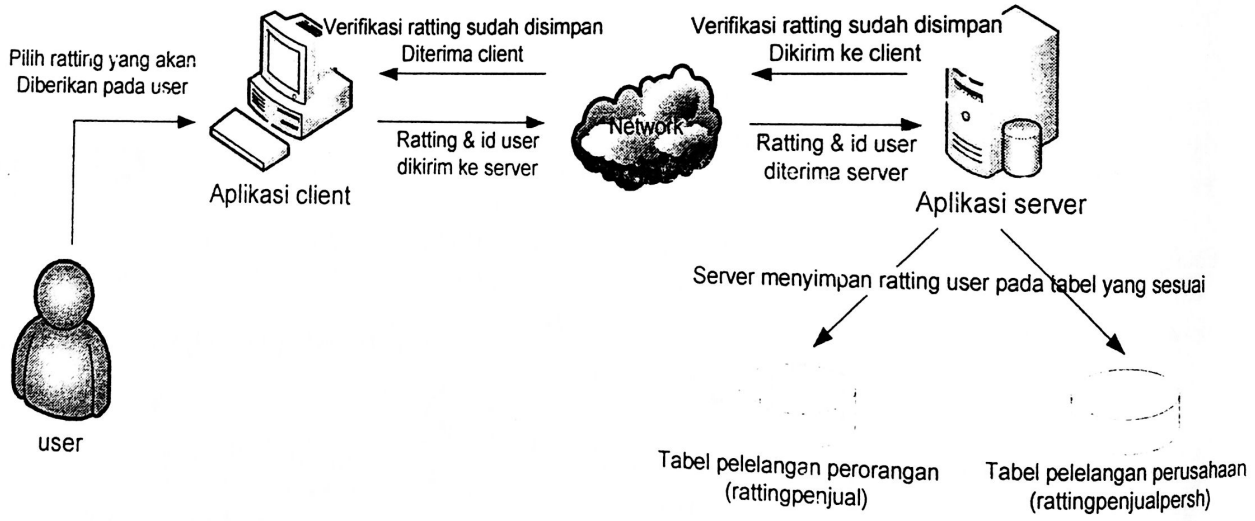
  Setiap pelelangan akan memiliki histori transaksi. *Histori* ini merupakan suatu laporan yang berisi informasi tentang penawaran – penawaran yang pernah terjadi pada pelelangan yang bersangkutan. Dari histori ini dapat diketahui *user* siapa saja yang pernah melakukan penawaran dan harga tawar yang diajukan oleh masing – masing *user* tersebut.




Gambar 5. Proses lihat histori pelelangan

- Skenario pemberian *ratting*

  Dalam sistem pelelangan ini setiap *user* yang telah menyelesaikan suatu pelelangan dapat saling memberikan *ratting* kepada lawannya dalam bertransaksi. Penjual dapat memberikan *ratting* pada pembeli dan demikian juga sebaliknya.




Gambar 6. Proses pemberian *ratting*

## 4. Kesimpulan

1. Pelelangan sangat tepat untuk diimplementasikan pada suatu sistem yang bersifat *online*, karena akan sangat mempermudah peserta pelelangan, dengan adanya pelelangan yang bersifat *online* akan mengurangi keterikatan peserta terhadap tempat dan waktu pelaksanaan pelelangan, disamping itu pelelangan juga memiliki ruang lingkup yang lebih luas dari sebelumnya.

2. Penggunaan CORBA pada sistem dapat memberikan interoperabilitas pada sistem, karena dengan CORBA memungkinkan sistem pelelangan memiliki 2 aplikasi *client* yang diimplementasikan dengan bahasa pemrograman yang berbeda, sehingga pengembangan sistem ini nantinya tidak terlalu terikat pada satu bahasa pemrograman.

## 5. Saran

Saran yang dapat diberikan untuk pengembangan ke depan pada sistem pelelangan online adalah proses pembayaran pasca pelelangan. Proses pelelangan pada sistem ini sebaiknya melibatkan suatu sistem pembayaran, misalnya dengan menggunakan kartu kredit, yaitu ketika suatu pelelangan telah dimenangkan oleh seorang user maka sistem secara otomatis akan melakukan pemindahan saldo dari kartu kredit user yang memenangkan pelelangan ke rekening dari penjual. Sehingga yang bisa berpartisipasi pada peelangan ini hanyalah user yang memiliki kartu kredit yang valid.

## Daftar Pustaka


[1] Irfan Pyarali, Douglas C.Schmidt, *An Overview of the CORBA Object Adapter*.

[2] ELCA, Januari 2004. *Interoperability with IIOP.NET*.

[3] Qusay H. Mahmoud, *Distributed Programming with JAVA*, Manning Publications Co, 2000.

[4] Marko Boger, *JAVA in Distributed System Concurency, Distribution and Persistence*, John Wiley and Sons Ltd, 2001.

[5] JavaTM 2 SDK, *Standard Edition Documentation*.