

## Research Article

**Implementasi Java RMI pada Rancang Bangun Tes Toefl Online Berbasis Web**Sri Lestari<sup>1</sup>, Maman Somantri<sup>2</sup>, R. Rizal Isnanto<sup>2</sup>

1. Mahasiswa Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang

2. Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro Semarang

**Abstract**

The advances in information technology today has sparked growth in the number of Internet users. Ease of accessing the internet makes it the best way of presenting data for both individuals and institutions. Similarly UNDIP who have harnessed the Internet to present information such as presenting transcripts of students, teachers' and students' data, etc. However, for organizing the job vacancies to graduates and students, UNDIP is still using the faculty information systems respectively. A better way to facilitate access to the job vacancies is to design an information system that records and organizes all of the job vacancies to students and graduates of UNDIP. Design and implementation will begin with a system requirements analysis which is intended to obtain an appropriate system specifications. Within the design stage, Data Flow Diagram will be used to provide a clear representation of any business function, meanwhile ERD will be used to provide a logical representation of data in this system. The design is then implemented with the PHP programming language and MySQL database and also the addition of java script to create a more dynamic Web. The results of implementation and system testing show that system is running properly and in accordance with the system specifications, particularly system's e-mail and SMS notification for newly inputted job vacancies that meet jobseekers' specifications.

**Keyword :** web-based information system, PHP, MySQL**I. PENDAHULUAN****1.1 Latar Belakang**

Sistem perkuliahan dalam suatu Perguruan Tinggi semakin berkembang dengan dukungan teknologi informasi. Berbagai sistem informasi di bidang akademis dibangun dan dikembangkan pada jaringan yang ada. Sistem ini membantu Perguruan Tinggi dalam pelaksanaan kegiatan akademis seperti proses perkuliahan online atau pada proses administrasi seperti pengisian kartu rencana studi, maupun sebagai pusat basis data mahasiswa, karyawan, serta nilai untuk penyusunan transkrip.

Saat ini sistem informasi yang melakukan pengelolaan lowongan maupun beasiswa di Universitas Diponegoro dapat dikatakan masih belum ada, karena pengelolaannya masih dilaksanakan oleh sistem informasi fakultas. Misalnya untuk Fakultas Teknik Universitas Diponegoro masih menggunakan Sistem Informasi Fakultas Teknik untuk mengelola lowongan dan beasiswa bagi mahasiswa maupun lulusan Universitas Diponegoro. Hal inilah yang menantang penulis untuk menulis penelitiannya dengan judul "Perancangan Career Development Center Undip Berbasis Web".

Diharapkan nantinya dengan adanya sistem ini akan dapat membantu lulusan maupun mahasiswa aktif dalam mencari lowongan pekerjaan, dan sistem ini bisa menjadi sebuah nilai tambah bagi sebuah Perguruan Tinggi terutama bagi Universitas Diponegoro.

**1.2 Tujuan**

Tujuan penelitian ini yaitu membuat sebuah sistem informasi berbasis web sebagai pusat informasi lowongan pekerjaan bagi lulusan maupun mahasiswa aktif Universitas Diponegoro.

**1.3 Batasan Masalah**

Agar tidak menyimpang dari pokok pembahasan, maka permasalahan dari penelitian ini dibatasi pada:

1. Aplikasi yang dibuat masih berupa *prototype* sistem informasi berbasis web
2. Aplikasi dibangun dengan PHP 5.3.5, basis data MySQL 5.0.7, web server Apache 2.2.17, dan mail server Mercury 4.6 serta Now SMS/MMS Gateway v2011.07.05.

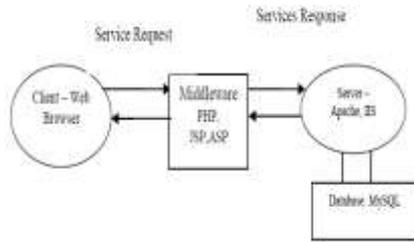
**II. DASAR TEORI****2.1 Sistem Terdistribusi**

Sistem terdistribusi adalah kumpulan komputer individu yang terhubung melalui sistem jaringan komputer dan dilengkapi dengan perangkat lunak sistem terdistribusi untuk membentuk fasilitas komputer terintegrasi yaitu sistem yang komponennya berada pada jaringan komputer yang saling berkomunikasi dan melakukan koordinasi hanya dengan pengiriman pesan (*message passing*).

Aplikasi terdistribusi memberikan manfaat yang cukup besar karena karakteristik yang dimilikinya yaitu bersifat banyak pengguna, penggunaan sumberdaya secara bersama-sama, skalabilitas yang baik, efisien, toleransi terhadap kesalahan, dan transparansi.

**2.2 Client-Server**

*Client-Server* merupakan salah satu dari jenis arsitektur sistem terdistribusi yang terdiri atas 3 jenis arsitektur yaitu 2-tier, 3-tier dan n-tier. Model *client-server* ini banyak dipilih karena kejelasan peran dari komputer yang terhubung dalam sistem jaringan tersebut. Komputer *server* hanya bertugas memberikan layanan kepada komputer *client*. Spesialisasi layanan tersebut akan memudahkan dalam permintaan layanan dan pengamanan. Gambar 1 merupakan arsitektur *client-server*.

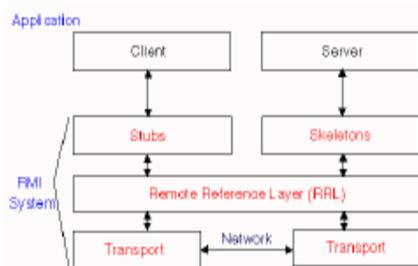


Gambar 1 Arsitektur Model Client-Server

2.3 Java RMI

RMI didefinisikan sebagai sebuah fasilitas standar Java dan merupakan salah satu teknologi sistem terdistribusi yang menangani pemanggilan (*invocation*) suatu metode secara jarak jauh (*remote*) dalam suatu jaringan. Metode yang dipanggil tersebut berada pada *host server* dan dipanggil secara *remote* oleh *host client* pada suatu jaringan. RMI juga disebut sebagai jembatan penghubung antara satu aplikasi dengan aplikasi yang lainnya.

Sistem RMI dibangun atas tiga lapisan yaitu lapis *stub/skeleton*, lapis *remote reference*, dan lapis *transport*. Tiap lapis dibangun dengan menggunakan *interface* khusus dan didefinisikan dengan protokol khusus. Oleh karena itu secara lebih jelasnya dapat dilihat pada Gambar 2 yang menunjukkan arsitektur RMI.



Gambar 2 Arsitektur RMI

Berikut penjelasan dari arsitektur RMI tersebut:

1. Lapis *stub* yang berlaku sebagai *proxy* yang memiliki *remote call*.
2. Lapis *Skeleton* berada pada sisi *server*, bertanggung jawab memanggil implementasi pemanggilan objek yang sesuai bagi jaringan untuk dikirim kembali ke *client*.
3. Lapis *Remote Reference* Merupakan lapis yang tidak terlihat oleh pengguna yang bertanggung jawab mengatur komunikasi antara *client-server* dengan mesin *virtual Java*.
4. Lapis *Transport* Lapis *TCP/IP-based* yang bertanggung jawab dalam mengadakan hubungan antara *server* dan *client*.

Untuk mengetahui cara kerja RMI ditunjukkan oleh Gambar 3.



Gambar 3 Mekanisme kerja RMI

1. Dimulai saat RMI *server* akan mendaftarkan (*bind*) nama objek ke RMI *registry* dengan nama yang unik.
2. RMI *registry* dapat memberikan referensi pemanggilan terpisah objeknya ke RMI *client* untuk melakukan pemanggilan.
3. RMI *client* dapat memanggil metode yang berada di *server*.
4. RMI *server* mengirimkan respon setelah dilakukan pemrosesan ke RMI *client*.

2.4 MySQL

MySQL adalah suatu pemrograman atau sistem manajemen basisdata (kumpulan data yang terstruktur) yang menggunakan dasar bahasa SQL (*Structured Query Language*). SQL merupakan bahasa *query* yang terstandarisasi secara internasional untuk mengakses hubungan basisdata. MySQL dirancang untuk menangani basisdata yang besar dengan cepat, memiliki tingkat keamanan, dan konektivitas yang tinggi.

2.5 TOEFL

TOEFL merupakan tes bahasa Inggris yang digunakan untuk mengetahui tingkat kemampuan seseorang dalam berbahasa Inggris. Nilai tes TOEFL dapat menunjukkan sejauh mana seseorang dalam menguasai bahasa Inggris. Tes TOEFL dibagi dalam 3 (tiga) bagian, yaitu

1. *Listening Comprehension* yang terdiri atas percakapan pendek, percakapan panjang *Mini Talks*.
2. *Structure and Written Expression* yang terdiri atas *Incomplete Sentence Error Recognition*
3. *Reading Comprehension* terdiri atas bacaan-bacaan pendek dan kemudian menjawab pertanyaan yang berdasarkan pada bacaan tersebut.

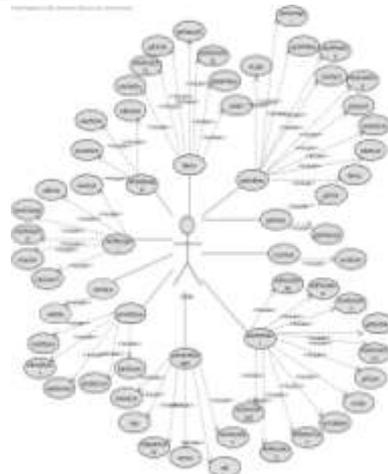
III. PERANCANGAN PERANGKAT LUNAK

1. Analisis dan Perancangan Implementasi Java RMI pada Aplikasi Tes TOEFL Online

Dalam perancangan perangkat lunak ini, digunakan bahasa pemodelan *Unified Modeling Language* (UML) yang terdiri atas diagram *use case*, diagram urutan, diagram kelas, dan diagram komponen.

a. Diagram Use Case

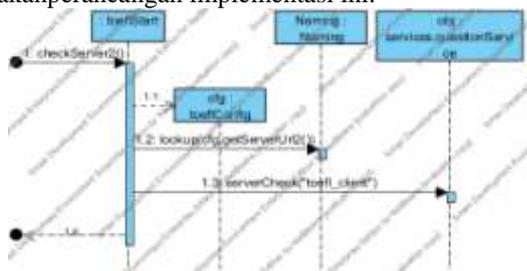
Dalam pengembangan perangkat lunak dalam implementasi Java RMI ini dapat diidentifikasi tiga buah aktor yaitu *client*, *admin*, dan *server*. Gambar 4 menunjukkan salah satu diagram *use case*.



Gambar 4 diagram use case admin

**b. Diagram Urutan**

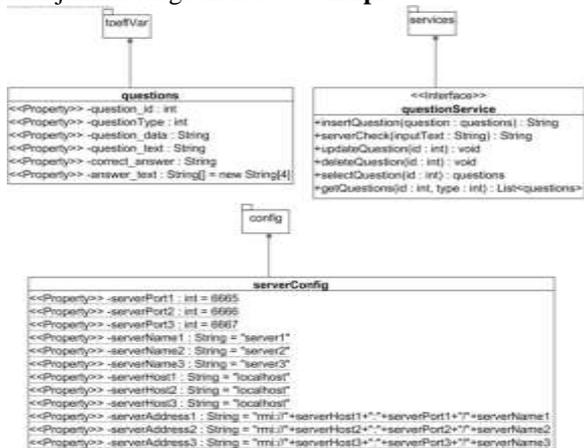
Gambar 5 merupakan diagram **checkServer2** urutan yang menunjukkan salah satu contoh diagram urutan yang digunakan perancangan implementasi ini.



Gambar 5 Diagram Urutan checkServer2

**c. Diagram Kelas**

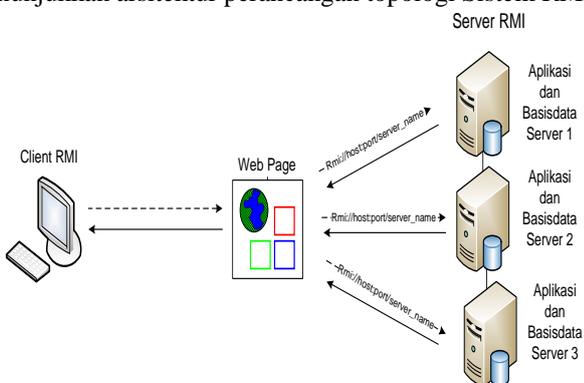
Untuk membangun implementasi Java RMI pada aplikasi tes TOEFL online ini digunakan beberapa diagram kelas yang terdiri atas diagram kelas **toeflApi**, **toefl\_server\_2**, **toefl\_server\_3**, dan **client**. Gambar 6 menunjukkan diagram kelas **toeflApi**.



Gambar 6 toeflApi

**1. Perancangan Topologi Sistem**

Penelitian ini dirancang untuk mengetahui bagaimana Java RMI dibangun dan diimplementasikan. Pada tahap perancangan sistem ini berdasarkan komunikasi sistem terdistribusi Java RMI yang dibangun untuk memanfaatkan layanan model *client-server* dalam suatu jaringan. Gambar 3 menunjukkan arsitektur perancangan topologi Sistem RMI.



Gambar 7 Arsitektur Perancangan Topologi Sistem RMI.

Setiap *client* di lingkungan sistem melakukan permintaan kepada *server* aplikasi, serta *server* aplikasi memberikan respon berupa layanan-layanan sesuai yang dipanggil oleh *client*. Begitu juga jika ada pemanggilan basisdata, aplikasi *server* melakukan permintaan kepada *server* basisdata, kemudian *server* database memberikan data sesuai

permintaan yang dipanggil. Perancangan arsitektur sistem mengimplementasikan arsitektur *three tier*, dimana terdapat dua buah *server*, yaitu *server* basisdata dan *server* aplikasi. *Client* akan memanggil metode ke *server* berdasarkan referensi objek yang diperoleh dari RMI *registry*. Komunikasi antara *client* dan *server* ini melalui *webbrowser* yang telah disediakan. Implementasi pada teknologi RMI ini menggunakan 3 buah basisdata *server* yaitu:

**a. Server 1**

**Server 1** pada RMI *server* ini berupa sebuah antarmuka dan kelas. Antarmuka ini berupa metode yang dipanggil dari *client* yaitu pemanggilan nama objek dan kelas pada *server* yang menyediakan implementasinya. **Server 1** ini merupakan *server* utama yang terdiri dari bank soal *Listening Comprehension*, sebagai penyimpan seluruh data aplikasi tes TOEFL online, dan dapat menjadi bank soal untuk mewakili **Server 1** dan 2 apabila terjadi masalah akses pada *server* tersebut.

**b. Server 2**

**Server 2** mempunyai antarmuka dan kelas yang sama dengan **Server 1**, tetapi **Server 2** ini berfungsi sebagai bank soal berupa basisdata *Structure and Written Expression* pada tes TOEFL online.

**c. Server 3**

**Server 3** juga memiliki antarmuka dan kelas yang sama dengan **Server 1** dan 2, tetapi **Server 3** ini berfungsi sebagai bank soal berupa basisdata *Reading Comprehension* pada tes TOEFL online.

**IV. IMPLEMENTASI, PENGUJIAN, DAN ANALISIS**

**4.1 Implementasi Sistem**

Untuk dapat membangun suatu aplikasi sistem terdistribusi khususnya Java RMI yang diimplementasikan dalam tes TOEFL online dapat dilakukan dengan 7 langkah yaitu definisi *RemoteInterface*, definisi konstruktor untuk objek yang dipanggil, implementasi pemanggilan *remote interface* dan *server*, membuat implementasi untuk setiap pemanggilan *remote method*, membuat dan menginstalasi *security manager*, membuat instansi objek untuk *client*, mendaftarkan objek dengan Java RMI *registry*.

**4.2 Pengujian Sistem dan Skenario Pengujian**

**1. Spesifikasi Perangkat Keras**

Pada pengujian kali ini menggunakan dua buah laptop dimana masing-masing laptop mempunyai peran dan tugas sendiri-sendiri.

Laptop pertama digunakan sebagai *server* utama dan basisdata **Server 1** dengan spesifikasi sebagai berikut.

- a. Prosesor Intel Core i3 2,13Ghz
- b. RAM 2 GB
- c. Harddisk 320 GB

Laptop kedua akan digunakan sebagai **Server 2** dan **Server 3** dengan spesifikasi sebagai berikut.

- a. Prosesor Intel Core 2 D
- b. uo 2 GHz
- c. RAM 2 GB
- d. Harddisk 160 GB

Laptop kedua akan digunakan sebagai *client* dengan spesifikasi sebagai berikut.

- a. Prosesor AMD
- b. RAM 2 GB
- c. Harddisk 320 GB

## 2. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak untuk pengujian sistem yang diperlukan untuk membangun suatu teknologi RMI ini adalah:

- Windows 7 Ultimate
- MySQL
- Apache Tomcat 7.0.14
- Netbeans 7.0.1
- J2EE
- Framework JSF 2.0

### 4.3 Pengujian Sistem

#### 1. Pengujian RMI Server

Pengujian server ini dilakukan dengan menggunakan *host* yang berbeda, dengan IP yang berbeda bertujuan untuk mengecek apakah kode program basisdata server dapat dibuka dalam halaman *web browser* pada *host* berbeda.



Gambar 8 Pengujian Server 1



Gambar 9 Pengujian Server 2



Gambar 10 Pengujian Server 3

#### 2. Pengujian Server Basisdata

Pengujian ini bertujuan untuk melihat status dari basisdata server pada saat tes TOEFL dilakukan. Sebelum tes TOEFL berlangsung sistem melakukan cek server apakah semua server RMI telah aktif, apabila server tersebut aktif maka para member tes TOEFL dapat melakukan tes tersebut. Tes TOEFL pertama adalah *Listening Comprehension*, basisdata soal ini berada pada **Server 1**, tes kedua adalah tes *Structure and Written Comprehension* basisdata soal ini berada pada **Server 2**, dan *Listening Comprehension* berada pada **Server 2**. Misalnya pada saat member melakukan tes TOEFL yang kedua yaitu *Structure and Written Comprehension* tanpa disadari **Server 2** yang menjadi server basisdata soal mati maka tes TOEFL tersebut akan terus berlangsung, dan terjadi perpindahan pembacaan basisdata server dari **Server 2** ke **Server 1** hal ini terjadi karena **Server 1** merupakan server utama yang dapat menjadi server cadangan yang menyediakan basisdata bagi kedua server yang lain bila terjadi gangguan.

#### 4.4 Hasil Pengujian Keseluruhan Sistem

Hasil pengujian keseluruhan sistem dilakukan dengan memberikan kuesioner kepada para responden atas implementasi teknologi java yang diaplikasikan dalam tes

TOEFL online berbasis web. Pengujian tersebut dengan memberi kesempatan kepada para responden untuk mencoba menggunakan aplikasi tes TOEFL online berbasis web yang dibangun berdasarkan teknologi Java RMI, kemudian para responden memberikan penilaian berdasarkan kuesioner yang disediakan dan memberikan kritik serta saran sesuai dengan pengamatan mereka. Hal ini dilakukan untuk memberikan penilaian terhadap kinerja sistem tes TOEFL online berbasis web baik dari waktu tanggap sistem, kelayakan sistem, tampilan antarmuka, dan pengamatan terhadap kinerja teknologi Java RMI tersebut.

## V. PENUTUP

### 5.1 Kesimpulan

Dari perancangan, implementasi, dan pengujian penelitian yang berjudul "Implementasi Java RMI Pada Aplikasi Tes TOEFL Online Berbasis Web", maka dapat diambil beberapa kesimpulan sebagai berikut.

- Teknologi Java RMI ini dapat diaplikasikan dalam Tes TOEFL online berbasis web.
- Secara umum untuk membangun suatu aplikasi sistem terdistribusi khususnya teknologi Java RMI dapat dilakukan dengan beberapa langkah yaitu mendefinisikan *remote interface*, definisi konstruktor, implementasi *remote interface* dan *server*, membuat implementasi tiap metode, pengembangan *client*, membuat dan menginstal *security manager*, membuat instansi objek untuk *client*, memulai RMI registry, menjalankan *server* dan *client*.
- Perlunya sinkronisasi metode dalam komunikasi implementasi RMI khususnya dalam tes TOEFL online pada saat diakses secara bersamaan oleh *client*.
- Dari hasil implementasi teknologi java RMI ini dapat diperoleh beberapa keuntungan diantaranya beban kerja sistem menjadi seimbang karena adanya pembagian fungsi antara RMI server dan RMI client, dengan teknologi RMI kemampuan akses data yang cepat.
- Teknologi Java RMI ini tidak hanya dapat dibangun dalam satu komputer melainkan ke banyak komputer.
- Hasil penelitian berdasarkan hasil kuesioner dari para responden dapat diketahui bahwa sistem ini dapat dinyatakan bagus karena telah memenuhi persentase kisaran penilaian yang telah ditentukan.

### 5.2 Saran

Saran yang dapat digunakan sebagai bahan untuk mengembangkan sistem ini menjadi lebih baik dan untuk masa yang akan datang.

- Perlu dilakukan penelitian lebih lanjut dengan melakukan pengembangan aplikasi yang kompleks dalam implementasi Java RMI.
- Perlu dilakukan penelitian lebih lanjut mengenai penggunaan jaringan internet untuk teknologi Java RMI dan tidak hanya terbatas pada jaringan lokal.
- Perlu dilakukan apakah aplikasi dapat diterapkan lintas sistem operasi, misalnya antara Windows dengan Linux.
- Perlu dilakukan penelitian lebih lanjut tentang keamanan data untuk komunikasi *client-server* pada teknologi Java RMI ini.
- Perlu adanya penambahan kapasitas RAM pada server karena hal ini sangat berpengaruh pada kecepatan akses data antara server dan client.

Daftar Pustaka

- [1] -----, "Java Remote Method Invocation Specification", Sun Microsystem, 2004.
- [2] Sun Microsystems. (2006). Java Remote Method Invocation. Retrieved June 21, 2010, from 3 - RMI System Overview,   
[http://download.oracle.com/javase/6/docs/technotes/gui\\_des/rmi/hello/hello-world.html#2](http://download.oracle.com/javase/6/docs/technotes/gui_des/rmi/hello/hello-world.html#2), Maret 2011.
- [3] Eko Kurniawan Khannedy. "Membangun Aplikasi Client-Server Dengan Java", <http://eecchhoo.wordpress.com/2010/01/27/video-tutorial-membangun-aplikasi-client-server-menggunakan-java/>, Maret 2011.
- [4] ----, "Java Remote Method Invocation", [http://en.wikipedia.org/wiki/JavareMOTE\\_method\\_in\\_vocation](http://en.wikipedia.org/wiki/JavareMOTE_method_in_vocation), Agustus 2011.
- [5] Eko Kurniawan Khannedy. "MSQL dan Java Database Connectivity", <http://eecchhoo.wordpress.com>, Maret 2011
- [6] Eko Kurniawan Khannedy. "Mendeteksi Terputusnya Koneksi dalam Remote Method Invocation", <http://eecchhoo.wordpress.com/2010/10/03/mendeteksi-terputusnya-koneksi-dalam-remote-method-in-vocation.htm>, September 2011.
- [7] -----, "The Java EE 6 Tutorial". Oracle, November 2010.   
<http://docs.jboss.org/hibernate/core/3.3/reference/en/pdf/hibernate-reference.pdf>, Maret 2011
- [8] Christian Bauer and Gavin King. "Java Persistence with Hibernate". Second Edition of Hibernate in Action. 1-932394-88-5. Manning Publications Co. 2007.   
<http://www.manning.com/bauer2>.
- [9] Optimus Prime. "Primfaces User's Guide". Covers 2.2 and 2.2.1 *Prime Teknologi*. 05 Februari. 2011.
- [10] R. Nurrohmah. "Client/Server dengan Java Remote Method Invocation (Java RMI), Sebuah Tutorial", *Jurnal Teknik Elektro dan Komputer Emitor* Vol. 3, No. 2, September 2003.
- [11] Adrian Nathaniel Wikana, Joko Purwadi, Restyandito. "Implementasi RMI Untuk Tes Online Interaktif *Multiuser* Pada LAN", *Jurnal Informatika* Vol.3 No. 2, April 2007.
- [12] Ed Burns and Chris Schalk. "The Complete Reference Java Server Faces 2.0", Mc. Graw Hill, 2010.
- [13] Deborah Philips. "Longman Complete Course The TOEFL Test Preparation For The Computer and Paper Test", Addison-Wesley Longman, 2001.
- [14] R. Ardianto. "Proyek Akhir *Test TOEFL Online Via Web*". Jurusan Teknologi Informasi Politeknik Negeri Surabaya. 2006.
- [15] A. Wollrath, Roger Riggs, and Jim Waldo. "A Distributed Object Model for the Java System". *Proceedings of the USENIX Conference on Object-Oriented Technologies*. Canada, June 1996.