

PERBANDINGAN PEMROGRAMAN TERDISTRIBUSI REMOTE PROCEDURE CALL XML-RPC DENGAN REMOTE METHOD INVOCATION (RMI) STUDI KASUS OPERASI MATRIX PADA CITRA

ON THE COMPARISON OF DISTRIBUTED PROGRAMMING USING REMOTE PROCEDURE CALL XML-RPC AND REMOTE METHO

Daru Puspitaningrum¹, Fazmah Arief Yulianto², Bayu Erfianto³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Suatu program untuk menyelesaikan suatu permasalahan yang kompleks akan lebih mudah diimplementasikan secara modular. Setiap modul dalam program tersebut dapat didesain untuk menangani suatu operasi tertentu. Gabungan fungsional modul-modul suatu program nantinya akan mampu melakukan penyelesaian problem dari suatu program secara utuh. Konsep modularitas pada program memiliki keuntungan lain bagi programmer. Salah satunya adalah untuk menerapkan pemrograman terdistribusi. Dalam pemrograman terdistribusi, modul-modul program tidak harus berada di satu mesin komputer. Modul-modul program dapat diletakkan di beberapa node komputer yang terhubung dalam suatu jaringan. Untuk dapat berkomunikasi satu sama lain, modul-modul dalam program dapat menggunakan berbagai teknologi seperti socket, RMI(untuk Java) , CORBA, message oriented middleware atau pun Remote Procedure Call(RPC). Pada Tugas Akhir ini, pemrograman tedistribusi dibangun menggunakan teknologi Remote Procedure Call (RPC) dan Remote Method Invocation (RMI). Dari berbagai macam implementasi RPC, XML-RPC digunakan sebagai teknologi Remote Procedure Call-nya. Sedangkan implementasi RMI yang digunakan adalah Java-RMI. Hasil dari tugas akhir ini menunjukkan bahwa secara umum Java-RMI memiliki waktu eksekusi yang lebih cepat dibanding program XML-RPC(baik Python ataupun Java). Sedangkan dalam implementasi XML RPC dengan Python dan Java, waktu eksekusi program dengan Java relatif lebih cepat daripada program dengan Python.

Kata Kunci : Pemrograman terdistribusi (Distributed Programming), Remote Procedure Call(RPC), XML-RPC, Server, Client, Remote Method Invocation(RMI).

Abstract

To make a complex program, programmers usually divide the program into some modules in order to ease to be understood, to be solved, and to be built. Each of functionality of the program is usually represented by a module. The comprehensive work of the modules build a complete function of the program itself. Modularity concept of a program may give more than one advantages. One of them is that modularity concept enable programmers to build distributed programming. By using distributed programming, program's modules do not have to be in the same machine. These modules may be spread out in the computer nodes that are connected to the network. To enable them to communicate each other, a technology should be taken to bridge the connections among the modules. Technology such as socket, RMI (for Java), CORBA or Remote Procedure Call (RPC) may become the communication bridge among the modules. In this Final Assignment, a distributed programming is built using Remote Procedure Call and Remote Method Invocation. Among many implementations of RPC, this Final Assignment use XML-RPC as it Remote Procedure Calls. The RMI program is implemented using Java RMI. The result of the final assignment has shown that running distributed program under Java RMI is generally faster than distributed program under XML-RPC (on both Python and Java). However, XMLRPC with Java is relatively faster than Python.

Keywords : distributed programming, Remote Procedure Call (RPC), XML-RPC, Server, Client, Remote Method Invocation(RMI).

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman yang bersifat modular memberikan banyak kemudahan tak hanya kepada pemrogram tetapi juga kepada pengguna program. Program modular pada umumnya membagi-bagi program sesuai dengan fungsionalitas yang harus dipenuhi program.

Masing-masing bagian tersebut direpresentasikan oleh suatu modul program. Gabungan fungsionalitas dari modul-modul tersebut akan membentuk program utuh yang mampu menyelesaikan masalah sesuai kebutuhan. Dengan modul-modul program, suatu program bisa diartikan memiliki bagian-bagian yang independen satu sama lain, tetapi bagian-bagian tersebut mampu digabungkan membentuk fungsionalitas utuh dari program.

Dengan karakteristik program yang modular, program dapat didistribusikan di dalam node-node komputer yang terhubung ke suatu jaringan. Konsep ini disebut dengan pemrograman terdistribusi (*Distributed Programming*). Dalam pemrograman terdistribusi, terdapat beberapa cara agar masing-masing modul atau sub program dapat saling berhubungan. Yaitu dengan socket, single – language distributed object (misalnya pada Java Remote Method Invocation [RMI]), cross-language distributed object (misalnya CORBA), Message-Oriented Middleware, *Remote Procedure Call* (RPC) dan lain-lain.

Dalam tugas akhir ini, program terdistribusi telah diimplementasikan dengan dua cara yaitu dengan *Remote Procedure Call* XML-RPC dan dengan single –language distributed object Remote Method Invocation. Kedua teknologi tersebut merupakan teknologi yang umum digunakan untuk membangun suatu aplikasi terdistribusi.

Teknologi pertama akan digunakan untuk pemrograman terdistribusi ini adalah *Remote Procedure Call*(RPC). Dari beberapa implementasi RPC, tugas akhir ini menggunakan XML-RPC sebagai *Remote Procedure Call*-nya.

Mengapa XML-RPC? Karena XML-RPC merupakan teknologi *Remote Procedure Call* yang tidak terikat platform maupun bahasa pemrograman. Beberapa bahasa pemrograman memiliki library yang bisa ditambahkan untuk membuat suatu mekanisme XML-RPC. XML-RPC juga dinilai cukup stabil dan matang.

Sedangkan untuk teknologi yang kedua yaitu *single –language distributed object* akan diimplementasikan dengan Java-RMI.

1.2 Rumusan Masalah

Tugas Akhir ini mengimplementasikan pemrograman terdistribusi menggunakan XML-RPC dan Java-RMI. Permasalahan dalam Tugas Akhir ini meliputi :

- a. Bagaimana mendesain dan mengimplementasikan suatu pemrograman terdistribusi dalam jaringan.
- b. Bagaimana perbandingan performansi distributed program RMI (Remote Method Invocation) dengan distributed program RPC (*Remote Procedure Call*).
- c. Bagaimana perbandingan eksekusi program terdistribusi menggunakan xml-rpc yang diimplementasikan pada Java dan Python
- d. Bagaimana mengimplementasikan sistem terdistribusi Java RMI dan XML-RPC untuk operasi matrik pada pemrosesan citra.

1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah :

- a. Mengimplementasikan pemrograman terdistribusi dengan teknologi *Remote Procedure Call* (RPC) dengan menggunakan XML-RPC pada Java dan Python

- b. Mengimplementasikan pemrograman terdistribusi dengan teknologi Single-Language Distributed Object dengan menggunakan Java-RMI
- c. Menganalisis perbandingan implementasi pemrograman Java terdistribusi XML-RPC dengan implementasi terdistribusi menggunakan Java-RMI.
- d. Menganalisis perbandingan implementasi XML-RPC pada bahasa pemrograman Java dan Python.

Hipotesa awal yang digunakan pada penelitian ini adalah :

- a. Eksekusi program terdistribusi dengan Java RMI akan relatif lebih cepat dibanding Java-XML RPC
- b. Eksekusi program terdistribusi menggunakan Python relatif lebih cepat dibanding eksekusi program yang sama pada Java.

1.4 Batasan Masalah

Untuk memberikan pedoman yang jelas bagi Tugas Akhir ini maka masalah akan dibatasi. Batasan masalah tersebut adalah :

1. Implementasi akan dilakukan pada platform GNU/Linux.
2. Implementasi dilakukan pada jaringan dengan satu client dan satu server
3. Program akan menggunakan bahasa pemrograman Java dan Python
4. Program yang akan digunakan merupakan program operasi matriks pada citra
5. Operasi matriks yang dilakukan yaitu : View Matrix Server, View Matrix *Client*, Zoom In, Laplacian Operator, Histogram Matrix Server, Histogram Matrix *Client* dan Operasi Boolean AND pada matrik.
6. Matriks merupakan hasil generate bilangan random 0-255.
7. Teori mengenai pemrosesan citra (*Image Processing*) tidak dibahas.
8. Tugas Akhir ini tidak akan membahas mengenai teknologi jaringan dan sekuriti yang digunakan

1.5 Metodologi

Metodologi yang digunakan untuk menyelesaikan masalah tersebut adalah :

1. Studi Literatur

Tahapan ini meliputi pengumpulan data dan sumber-sumber penelitian yang bertujuan untuk mendapatkan gambaran, detail dan dasar teori yang jelas dan valid mengenai *Distributed Programming*, XML-RPC, dan RMI Tahap ini dilakukan selama proses penyusunan Tugas Akhir.

2. Perancangan Sistem

Tahapan ini digunakan untuk merancang program terdistribusi yang akan dibangun. Tahapan ini meliputi perancangan sistem terdistribusi(DS-RMI, DS-XMLRPC) dan program (OOA) yang akan digunakan. Hasil dari tahap ini akan digunakan sebagai cetak biru bagi tahap selanjutnya. Pada metodologi ini juga akan dibahas bagaimana desain DS-RMI dan XML RPC untuk operasi matrik.

3. Implementasi Sistem

Hasil perancangan sistem akan diimplementasikan ke dalam program yang akan dijalankan pada jaringan.

4. Pengujian

Pengujian terhadap hasil implementasi akan dilakukan untuk mengukur parameter pengukuran yaitu *transaction time*, *TCPtime*, *request size*, *response size* dari masing-masing implementasi program terdistribusi. Pengujian terhadap parameter tersebut akan dilakukan dengan *tools* ethereal.

5. Analisis

Dari hasil pengujian, data akan dianalisis. Analisis yang dilakukan berdasarkan perbandingan pengolahan data *transaction time*, *transfer time*, *request size*, *response size* masing-masing program. Dari *transaction time* dan *transfer Time*, dapat ditarik analisis performansi RMI dan XML-RPC dari sisi waktu eksekusi(*execution time*). Sedangkan besar paket yang dikirim masing-masing program akan menjadi dasar analisis perbandingan *request-response size* masing-masing program.

6. Penyusunan Laporan

Hasil penelitian akan di susun menjadi suatu laporan yang meliputi aspek-aspek dalam penelitian yaitu teori dan implementasinya. Pada laporan tersebut akan disimpulkan pula perbandingan hasil penelitian dengan hipotesa awal penelitian.

1.6 Sistematika Penulisan

Tugas Akhir ini disusun berdasarkan sistematika pembahasan sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan pembahasan, metodologi pemecahan masalah dan struktur pembahasan.

BAB II LANDASAN TEORI

Bab ini meliputi konsep umum *Remote Procedure Call* dan XML-RPC dan Remote Method Invocation (RMI) sebagai teknologi yang digunakan untuk pemrograman terdistribusi pada Tugas Akhir ini.

BAB III PERANCANGAN DAN IMPLEMENTASI

Bagian ini meliputi perancangan program yang dilakukan pada tahapan perancangan sistem. Pada bab ini, perancangan program akan direpresentasikan dalam notasi UML (Unified Modelling Language).

BAB IV ANALISIS HASIL IMPLEMENTASI

Memuat hasil tahap pengujian dari implementasi yang telah dilakukan dilanjutkan dengan analisis dari hasil evaluasi tersebut.

BAB V KESIMPULAN & SARAN

Bagian ini berisi kesimpulan terhadap hasil analisis yang dibandingkan terhadap hipotesa awal permasalahan serta saran pengembangan yang dapat dilakukan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Java RMI memiliki *delay* waktu eksekusi yang lebih rendah daripada XML-RPC. Hal ini disebabkan Java-RMI mengirimkan *return value* dalam bentuk byte. Sedangkan XML-RPC mengirim *return value* dalam bentuk XML. *Tag-tag* pada XML menambah besar paket yang harus dikirim XML-RPC. Jika menghendaki pembangunan program terdistribusi dengan bahasa Java saja, maka penggunaan RMI akan lebih menguntungkan. Akan tetapi, jika program terdistribusi yang dikehendaki diharapkan bisa lintas platform, maka penggunaan XML-RPC dapat dipertimbangkan.

Pada operasi matrik ini, implementasi program XML-RPC pada Python lebih sederhana dibandingkan implementasi pada Java. Hal ini disebabkan karena dependensi yang dibutuhkan program Python relatif lebih sedikit dibandingkan pada program Java-XMLRPC. Akan tetapi pengujian membuktikan bahwa hipotesis awal yaitu "*eksekusi program terdistribusi menggunakan Python relatif lebih cepat dibanding eksekusi program yang sama pada Java*" tidak terbukti karena hasil menunjukkan bahwa Python XMLRPC memiliki transaction time dan Transfer Time yang lebih lama daripada Java XML-RPC.

Telkom
University

5.2 Saran Pengembangan

XML-RPC menggunakan XML sebagai *messaging* antara *client* dan server. Penggunaan tag-tag xml pada paket ini menambah besar data yang dikirim. Untuk itu dapat dilakukan pengembangan pada *XML-data compression* dimana library XML-RPC dikembangkan sedemikian rupa sehingga paket XML tidak harus menggunakan *full-tag* XML. Pengembangan di bidang ini tentunya mempertimbangkan hal-hal seperti *trade off* antara *transaction time* dengan *processing time*. Karena jika dilakukan *XML-data compression* maka *transaction time* akan lebih baik. Tetapi mengingat perlu adanya dekompresi yang tentu saja menambah *processing time*, maka *XML-data compression* harus mempertimbangkan *trade off processing time* juga.

Pengamatan dan pengujian pada tugas akhir ini tidak mengaitkan XML RPC *persistent* dan *non-persistent*. Untuk itu, perlu dilakukan pengamatan pula performansi jaringan antara XML-RPC dengan koneksi non-persistent dan XML-RPC koneksi persistent.

DAFTAR PUSTAKA

- [1] Allman, Mark. *An Evaluation of XML-RPC. NASA Glenn Research Center/BBN Technologies.*
- [2] Booch, Grady dkk. *The Unified Modeling Language User Guide.* Addison Wesley. Massachusets.1999.
- [3] Döring , Ralf. *Performance von XML-RPC.* Technische Universität Ilmenau
- [4] Jasnowski, Mike. *Java, XML, and Web Services Bible.* Hungry Minds. New York. 2002
- [5] Jung, Andreas. 2001. *Python: Distributed applications with XML-RPC.* Linux Magazine
- [6] Kadir, Abdul. *Dasar Pemrograman Java™ 2.* Penerbit Andi. Yogyakarta.2004
- [7] Kadir, Abdul. *Dasar Pemrograman Python.* Penerbit Andi. Yogyakarta. 2005
- [8] Kidd, Eric.(2001).*XML-RPC HowTo.*Free Software Foundation
- [9] Mahmoud, Qusay H. *Distributed Programming with Java.* Manning. Greenwich. 2000
- [10] Noprianto. *Python & Pemrograman Linux.* Penerbit Andi. Yogyakarta.2002
- [11] Schildt, Herbert. *The Complete Reference Java TM 2 Fourth Edition.* McGraw-Hill.New York. 2001.
- [12] Srinivasan R. . *RPC : Remote Procedure Call Protocol Specification Version 2. RFC 1831*
- [13] Sun Microsystem. *An Overview of RMI Applications* 2005.
- [14] Susanto, Budi. *Pemrograman Client/Server dengan Java 2.* Elex Media Komputindo. Jakarta. 2003