

ANALISIS ARSITEKTUR DYNAMIC WEB SERVICE COMPOSITION DENGAN MENGGUNAKAN TEKNIK DECENTRALIZED DATAFLOW

Raditya Syahrozi¹, Kemas Rahmat Saleh Wiharja², Yanuar Firdaus A.w.³

¹Teknik Informatika, Fakultas Teknik Informatika, Universitas Telkom

Abstrak

Saat ini web service telah menjadi syarat wajib dalam dunia bisnis. Dengan menggunakan web service pertukaran informasi dan kolaborasi antar perusahaan yang bergerak dalam bidang E-Business menjadi lebih terjamin. Untuk menyelesaikan sebuah task, maka web service cukup mengambil informasi service yang berkaitan di Internet.

Sebuah service tidak pernah cukup untuk menyelesaikan sebuah task. Solusinya adalah dengan mengkomposisikan dua atau lebih service yang saling berkaitan yang sering disebut dengan Web Service Composition (WSC).

Dalam tugas akhir ini penulis menyarankan sebuah framework web service dengan menggunakan teknik decentralized dataflow, yaitu selain menggunakan Internet untuk menjelajah service, juga terdapat Web Service DataBase (WSDB), dan UDDI Registry yang menyimpan service yang berkaitan. Dengan kata lain dengan menggunakan teknik ini, ketersediaan informasi service menjadi meningkat.

Akan tetapi prioritas utama aliran data hanya pada WSDB karena nilai throughput yang tinggi dan response time yang lebih cepat dibandingkan dengan dua aliran data lainnya yaitu Internet dan UDDI Registry.

Kata Kunci : Web Service, Scalability, Response Time, WSDB, UDDI Registry.

Abstract

Currently web service has become a mandatory requirement in business world. By using web service, information exchange and collaboration among companies engaged in e-business becomes more secure. Efficient use of time and cost can be achieved by implementing a web service to build a new application. To complete a task, then the web service simply take the related information service in the Internet.

A service is never enough to complete a task. The solution is composed of two or more interrelated services. This technique is known as Web Service Composition (WSC).

In this final task, I suggest a web service framework using decentralized dataflow technique, which in addition to using the Internet to browse the service, there is also a Web Service DataBase (WSDB), and UDDI Registry that store the related services. In other words, using this technique, the level of availability of information services increases.

However, dataflow priority only on WSDB because its high throughput and fast response of request compared to Internet and UDDI Registry.

Keywords : Web Service, Scalability, Response Time, WSDB, UDDI Registry.

1. Pendahuluan

1.1. Latar Belakang

Saat ini web service menjadi sangat populer di perusahaan yang bergerak dalam bidang e-business. Banyak keuntungan yang dapat diambil dari pemanfaatan web service sebagai alat tukar informasi pada perusahaan tersebut. Salah satunya adalah dapat mereduksi biaya dan waktu untuk berkomunikasi dan berkolaborasi dengan consumer, supplier, dan dealer. Kehadiran teknologi web service kini menjadi sebuah tren yang memberikan dampak yang baik bagi pertumbuhan dan daya saing perusahaan. Teknologi web service dapat mengkoordinasi interaksi proses bisnis dari perusahaan yang berbeda. Dengan menggunakan web service, membangun sebuah aplikasi menjadi lebih cepat dengan cara mengkomposisi dua atau lebih aplikasi yang telah ada sebelumnya.^{[5][11]}

Permasalahannya adalah teknik web service yang sekarang digunakan masih mengandalkan service – service individual yang telah didesain sebelumnya. Faktanya permintaan user sangat kompleks dan tidak selalu sama dari waktu ke waktu, maka dari itu diperlukan sebuah komposisi dari service dengan operator tunggal yang bersifat dinamis berdasarkan personalisasi dari user yang dikenal dengan teknik dynamic web service composition. Tantangan lainnya adalah ketersediaan informasi tentang web service yang disimpan dalam WSDB (*Web Service DataBase*) sangat terbatas.

Maka dari itu diperlukan sebuah teknik untuk dapat mencari service lain yang terkait dengan service yang disimpan dalam WSDB tersebut untuk meningkatkan level *availability* dari sebuah service yaitu dengan menggunakan middleware (*UDDI Registry*)^{[3][4][10]}. Teknik ini dinamakan dengan teknik *decentralized dataflow*, dimana service – service tersebar ke web service yang berbeda. Kemudian service tersebut disimpan kedalam sebuah register. User atau *Service Requester* dapat mencari service dengan hanya mengambil register dari *UDDI Registry*, kemudian *UDDI Registry* mencari service yang terkait pada setiap web service. Teknik ini dapat mengurangi *tight-coupling* atau ketergantungan antara user dengan web service yang menyediakan service tersebut. Dengan demikian masalah yang sering dihadapi pada *traffic* web service yaitu *bottleneck* juga dapat dihindari.^{[6][10]}

1.2. Perumusan Masalah

Mengacu pada latar belakang diatas, beberapa masalah yang akan diselesaikan pada Tugas Akhir ini adalah :

- a. Bagaimana cara mengimplementasikan dynamic web service composition dalam menyediakan informasi penjualan real estate untuk perangkat lunak berbasis android?
- b. Bagaimana proses pertukaran informasi yang diterima dan dikirimkan oleh web service yang disimpan dalam WSDB dan *UDDI Registry*?

- c. Apakah dalam implementasinya arsitektur dynamic web service composition menggunakan teknik *decentralized dataflow* dapat memenuhi kebutuhan user dari segi fungsionalitas dan non-fungsionalitas?

1.3. Batasan Masalah

Dalam pengerjaan tugas akhir ini, permasalahan dibatasi dalam beberapa hal yaitu:

- a. Implementasi dilakukan pada server lokal (localhost).
- b. Sistem operasi android yang digunakan adalah Android 4.1, dan Google APIs Level API : 16.
- c. Implementasi web service, *UDDI Registry*, dan *Web Service Database* menggunakan bahasa pemrograman *PHP Object Oriented* dan *MySQL*.
- d. Service yang diambil dari Internet seperti Google Map, menggunakan library google android map, dan Google Map API key yang langsung di-embed kedalam aplikasi *client* android.
- e. Tidak membahas skenario *tight-coupling* dan *bottleneck* pada web service.
- f. Proses pengiriman dan penerimaan data service menggunakan *parsing JSON* : *JSONArray* untuk data list, *JSONTokener* untuk data string, dan *JSONObject* untuk data image.
- g. Hanya menggunakan *RESTful Web Service*.
- h. *Client* web service yang akan digunakan adalah *client* android emulator yang berjalan pada *Android Virtual Machine*.
- i. Tidak membahas aspek *Semantic Web*, seperti *meta-level*, dan *domain knowledge*.

1.4. Tujuan

Adapun tujuan dari pengerjaan Tugas Akhir ini antara lain adalah:

- a. Agar dapat mengimplementasikan teknologi dynamic web service composition dalam membangun layanan yang dapat menyediakan informasi penjualan real estate untuk digunakan dalam perangkat lunak berbasis android.
- b. Agar dapat menganalisis pertukaran informasi yang diterima dan dikirimkan oleh web service yang disimpan dalam *WSDB (Web Service DataBase)*, dan *UDDI (Universal Description, Discovery, and Integration) Registry*.
- c. Agar dapat membuktikan bahwa dalam implementasinya, arsitektur dynamic web service composition yang menggunakan teknik *decentralized dataflow* memenuhi atribut fungsionalitas dan non-fungsionalitas dalam melayani kebutuhan service pada user.

1.5. Hipotesis

Berdasarkan rumusan masalah dan tujuan Tugas Akhir, maka hipotesa yang dapat diasumsikan adalah:

- a. Arsitektur dynamic web service composition dapat diimplementasikan kedalam aplikasi android dengan menggunakan IDE Eclipse, RESTful Web Service, dan HttpClient sebagai library yang disediakan Eclipse untuk menghubungkan server web service dengan *client*.
- b. Pertukaran informasi yang dikirimkan dan diterima oleh web service dapat dianalisis dengan menggunakan tool yang disediakan oleh WizTools.org yaitu RestClientUI 3.1 dan JMeter.
- c. Teknik *decentralized dataflow* dapat meningkatkan *availability* service dari arsitektur dynamic web service composition.

1.6. Metodologi Penyelesaian Masalah

Metode yang dipakai untuk menyelesaikan masalah ini adalah:

- a. Studi Pustaka dan Literatur
Melakukan pencarian secara mandiri dari berbagai sumber mengenai web service, web service composition, dan dynamic web service composition, sehingga diperoleh identifikasi dan metode penyelesaian masalah sesuai dengan tujuan Tugas Akhir ini.
- b. Konsultasi dan Diskusi
Konsultasi dan diskusi dengan pembimbing TA mengenai konsep dasar dynamic web service composition, teknik *decentralized dataflow*, dan *Quality of Service* dalam menentukan performansi non-fungsional dari web service, serta proses pertukaran informasi service dari *service requester* ke service provider dan sebaliknya.
- c. Analisis Kebutuhan Sistem
Melakukan perancangan arsitektur dynamic web service composition dengan menggunakan empat buah RESTful Web Service sebagai server yang berjalan pada localhost dengan menggunakan AppServ. Tiga diantaranya merupakan *Web Service DataBase*, dan satu web service dirancang sebagai UDDI API.
- d. Implementasi
Membangun sebuah aplikasi *client* berbasis android dengan menggunakan IDE Eclipse Versi Juno dengan Android SDK 4.1 dan Google API Level 16.
- e. Desain Skenario Uji
Menguji hasil simulasi dengan menjalankan program *client* berbasis android pada Android Virtual Device Manager untuk mengambil service dari empat buah RESTful Web Service.
- f. Testing dan Analisis
Menganalisis performansi fungsional dan non-fungsional dari arsitektur dynamic web service composition tersebut dengan menggunakan tool tambahan yang disediakan oleh WizTools.org yaitu RestClientUI 3.1 dan JMeter.
- g. Penyusunan Laporan Tugas Akhir

Membuat dokumentasi dari semua tahapan pengerjaan tugas akhir ini, mulai dari dasar teori yang digunakan, arsitektur yang disarankan, proses pembuatan web service dan aplikasi *client*, hingga hasil dari pengujian yang dilakukan.

1.7. Sistematika Penulisan

Tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

BAB I Pendahuluan

Bab ini menguraikan tugas akhir ini secara umum, meliputi latar belakang masalah, perumusan masalah, tujuan, batasan masalah, dan metode yang digunakan, serta hipotesis penelitian.

BAB II Dasar Teori

Bab ini membahas mengenai uraian teori yang berhubungan dengan web service composition dan dynamic web service composition.

BAB III Perancangan dan Implementasi

Bab ini berisi perancangan arsitektur dynamic web service composition dengan menggunakan teknik *decentralized dataflow* dan analisis kebutuhan dari arsitektur tersebut serta masalah-masalah yang ada di dalamnya. Dari tahap analisis kebutuhan kemudian dilanjutkan ke tahap implementasi.

BAB IV Pengujian dan Analisis

Bab ini membahas mengenai pengujian hasil implementasi yang telah dilakukan pada bab sebelumnya. Pengujian dilakukan dengan melakukan beberapa skenario untuk memperoleh pengaruh yang diberikan implementasi arsitektur tersebut terhadap *Quality of Service* (QoS).

BAB V Kesimpulan dan Saran

Bab ini berisi kesimpulan dari penulisan Tugas Akhir ini dan saran-saran yang diperlukan untuk pengembangan lebih lanjut.

5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan analisis dan pengujian terhadap arsitektur dynamic web service dengan menggunakan teknik *decentralized dataflow* dapat disimpulkan:

1. Dari hasil pengujian pada kasus uji ketersediaan data pada web service menunjukkan bahwa implementasi *UDDI Registry* sebagai salah satu aliran data yang terdesentralisasi dapat meningkatkan level *availability* suatu service. Jika service yang *direct* tidak terdapat pada WSDB maka sistem akan langsung mencari service yang terkait di Internet melalui sebuah middleware yaitu *UDDI Registry*.
2. Dari hasil pengujian diketahui bahwa nilai *throughput* pada *UDDI Registry* lebih rendah dibandingkan dengan nilai *throughput* pada WSDB. Dengan jumlah maksimal *request* yang dapat ditangani oleh *UDDI Registry* adalah 22 *request* dalam satu detik sedangkan WSDB dapat menangani 68 *request* dalam satu detiknya. Dengan kata lain, tingkat *scalability* dari WSDB lebih tinggi dibandingkan dengan *UDDI Registry*.
3. Dari hasil pengujian diketahui bahwa proses pengambilan data *image* menggunakan *UDDI Registry* lebih lama dibandingkan dengan pengambilan data *image* pada WSDB karena pada *UDDI Registry* terdapat proses tambahan berupa pembacaan kode HTML, pengekstrakan url *image* dari *response data*, serta proses pengubahan url *image* menjadi data *image* dengan menggunakan *cURL Function*. Dengan waktu rata-rata yang dibutuhkan untuk mengambil data *image* pada *UDDI Registry* sebesar 1.4875 detik yaitu melebihi dari batas *response time* dikatakan pengguna terganggu (1 detik) berdasarkan buku *usability engineering*^[10].
4. Dari ketiga analisis terhadap aspek QoS web service yaitu *Availability*, *Scalability*, dan *Response Time*, dapat disimpulkan bahwa *UDDI Registry* hanya dapat digunakan sebagai alternatif aliran data pada arsitektur dynamic web service composition dan tidak dapat dijadikan sebagai prioritas utama.

5.2. Saran

Untuk pengembangan arsitektur web service, penulis menyarankan:

1. Perlu diteliti lebih lanjut mengenai implementasi *UDDI Registry* agar tidak hanya mencari service yang berkaitan di Internet jika service tersebut tidak terdapat pada WSDB tetapi juga menyimpan service tersebut dalam WSDB untuk penggunaan selanjutnya
2. Perlu diteliti lebih lanjut mengenai *aging factor* dalam menentukan batas waktu yang dimiliki oleh service dari hasil pencarian oleh *UDDI Registry* dapat disimpan dalam WSDB.
3. Perlu diteliti lebih lanjut mengenai sumber aliran data pada arsitektur web service selain WSDB, *UDDI Registry*, dan Internet.
4. Perlu diteliti lebih lanjut mengenai penggunaan software yang paling sesuai selain menggunakan aplikasi *client* android agar dapat mensimulasikan arsitektur web service dengan lebih baik lagi.

Daftar Pustaka

- [1] Antony, D., dan V.S. Ananthanarayana. 2010. *“Dynamic Web service composition Based on Operation Flow Semantics”*. India : International Journal of Computer Applications (0975-8887).
- [2] Bottaro, A., Eric Simon, Stephane Seyvos, dan Anne Gerodolle. 2007. *“Dynamic Web Service on a Home Service Platform”*. France : France Telecom R&D.
- [3] Charif, Y., dan Nicolas Sabouret. 2007. *“Dynamic Web Service Selection and Composition : An Approach based on Agent Dialogues”*. Paris : Laboratoire d’Informatique de Paris 6.
- [4] D., Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, dan D. Orchard. 2004. *Web Services Architecture*. W3C Working Group Note.
- [5] G., Frederico, dan Jose M. Parente. 2010. *“An Infrastructure for Evolving Dynamic Web Service Composition”*. Brasil : Technological Institute of Aeronautics (ITA).
- [6] H., Khan, M. Younus Javed, Saba Bashir, Aihab Khan, dan Malik Sikandar Hayat Khiyal. 2010. *“QoS Based Dynamic Web Services Composition & Execution”*. Pakistan : National University of Science & Technology.
- [7] Kim, A. 2005. *Comparison of WSQM & QoS FE*. FWSI Functional Elements SC document repository.
- [8] Leander, R. 1998. *“Throughput Limitations for Web Services”*. (online). (<http://smallbusiness.chron.com/throughput-limitations-services-38675.html>, diakses Desember 2012).
- [9] N., Sabouret. 2002. *“Representing, requesting and reasoning about actions for active components in human-computer interaction”*. Technical Report 2002-09, LIMSI-CNRS.
- [10] Nielsen, J. 1993. *Usability Engineering*. Penerbit Morgan Kaufmann.
- [11] S., Jayampathi. 2010. *“Data Aggregation through Web Service Composition in Smart Camera Networks”*. Atlanta : Georgia State University.
- [12] Sirin, E., J. Hendler, dan B. Parsia. 2003. *“Semi-Automatic Composition of Web Services using Semantic Descriptions”*. France : ICEIS ’03.
- [13] T., Yu, dan Kwei-Jay Lin. 2005. *“A Broker-Based Framework for QoS-Aware Web Service Composition”*. California : University of California, Irvine.
- [14] Zhang, D, Minder Chen, dan Lina Zhou. 2005. *Dynamic and Personalized Web Services Composition in E-Business*. Journal of Computer Information Systems.