

# 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.<sup>[5][11]</sup>

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*)<sup>[3][4][10]</sup>. 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.<sup>[6][10]</sup>

## 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.