

## PENDAHULUAN

### Latar Belakang

*Service oriented architecture* (SOA) merupakan sebuah konsep arsitektur perangkat lunak yang mendefinisikan penggunaan layanan untuk memenuhi kebutuhan suatu perangkat lunak. Layanan ini tidak hanya dapat digunakan oleh sistem yang menaunginya namun dapat digunakan juga oleh sistem lain yang berbeda, sehingga integrasi antarsistem dapat dicapai. Dibandingkan dengan arsitektur berorientasi objek terdistribusi, SOA lebih sesuai untuk mengintegrasikan sistem yang heterogen dan lebih mudah beradaptasi dengan perubahan lingkungan.

*Web service* merupakan salah satu bentuk implementasi SOA yang dapat memberikan banyak keuntungan bagi sebuah organisasi. *Web service* menggunakan seperangkat teknologi standar terbuka yang memungkinkan sebuah aplikasi dapat berkomunikasi dan menyediakan layanan bagi aplikasi lain melalui jaringan komputer. *Web service* memberikan kemudahan upaya integrasi antarsistem dengan latar belakang teknologi yang heterogen. Sebuah aplikasi berbasis teknologi *Web service* dapat menyediakan data maupun fungsi tertentu bagi aplikasi lain meskipun berbeda sistem operasi, perangkat keras, maupun bahasa pemrograman yang digunakan untuk membangunnya.

Keunggulan *web service* ini dapat dimanfaatkan untuk memecahkan permasalahan integrasi sistem informasi dalam suatu organisasi, misalnya institusi pendidikan tinggi. Institusi pendidikan merupakan sebuah organisasi yang di dalamnya terdapat beberapa entitas/unit yang memiliki fungsi khusus. Setiap unit dapat memiliki satu atau lebih sistem perangkat lunak yang berbeda-beda. Dalam menjalankan suatu fungsi tertentu, suatu perangkat lunak bisa jadi memerlukan data atau fungsi dari unit lain. Sebagai contoh untuk menjalankan salah satu proses bisnis atau layanan yang dimiliki oleh unit administrasi akademik, yaitu pengelolaan wisuda, diperlukan data bebas pustaka dari perpustakaan dan bebas SPP dari bank. Tanpa SOA, data ini diperoleh secara manual sehingga menjadi tidak efisien. Dengan adanya *web service*, data dari perpustakaan dan bank dapat diperoleh dengan mudah.

Penelitian ini menerapkan *service oriented architecture* menggunakan *Web service* pada pengembangan sistem informasi akademik. Sistem ini memiliki fungsi utama mengelola data mahasiswa dan akademik. Penerapan SOA memungkinkan sistem informasi akademik dapat diintegrasikan dengan sistem di perpustakaan dan bank untuk menyelesaikan proses bisnis tertentu.

### Tujuan

Tujuan penelitian ini adalah menerapkan konsep SOA pada pengembangan sistem informasi akademik sehingga memiliki *interoperability* lintas *platform* dan mampu berkomunikasi dengan unit lain dengan menyediakan *service* yang dapat diakses oleh aplikasi lain.

### Ruang Lingkup

Ruang lingkup penelitian ini adalah:

- 1 Sistem informasi akademik mengelola data terbatas pada mahasiswa dan yang terkait langsung dengan perkuliahan.
- 2 Uji *interoperability* dengan sistem eksternal dilakukan dengan mengembangkan sistem yang dapat mengkonsumsi *service* dengan lingkungan pengembangan yang berbeda, tidak dengan memanfaatkan sistem yang ada pada entitas/unit yang bersangkutan.

### Manfaat

Sistem yang dikembangkan dalam penelitian ini diharapkan memberikan gambaran pengembangan sistem informasi akademik di lingkungan institusi pendidikan. Penggunaan *web service* memungkinkan sistem mampu menyediakan *service* bagi sistem lain. Penyediaan *service* dapat meningkatkan efisiensi pengembangan sistem dalam lingkungan yang heterogen yaitu multi entitas dan multi *platform*. Sistem ini bersifat *interoperability* lintas *platform* sehingga memudahkan proses integrasi sistem secara keseluruhan.

## TINJAUAN PUSTAKA

### *Service Oriented Architecture*

*Service oriented architecture* atau SOA didefinisikan sebagai kebijakan, praktek, kerangka kerja yang memungkinkan fungsionalitas aplikasi disediakan dan dikonsumsi sebagai seperangkat *service* pada sebuah unit yang sesuai dengan kebutuhan

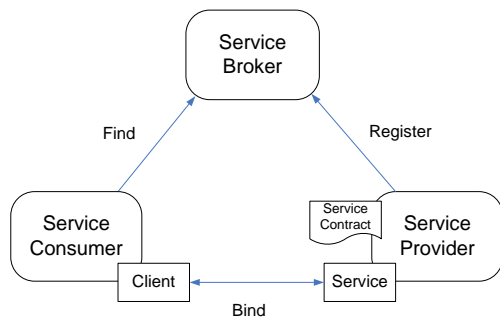
Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:  
 a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.  
 b. Pengutipan tidak merugikan kepentingan yang wajar IPB.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

*service customer*. *Service* dapat digunakan, dipublikasikan, ditemukan, dan diabstraksikan menggunakan standar antarmuka (Sprott & Wilkes 2004).

SOA menggambarkan pola yang membantu sebuah aplikasi *client* terhubung pada sebuah *service*. Pola tersebut menyajikan mekanisme yang digunakan untuk menggambarkan sebuah *service*, mempublikasikan dan menemukan *service*, dan berkomunikasi dengan *service*. Pola tersebut digambarkan pada Gambar 1 (Manes 2003).



Gambar 1 Pola SOA.

**Web Service**

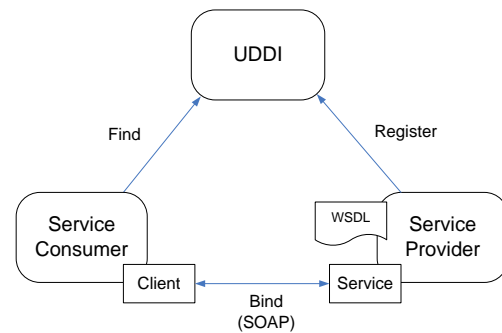
Sebuah *web service* adalah sebuah sistem perangkat lunak yang didesain untuk mendukung interaksi *machine-to-machine* yang dapat beroperasi melalui suatu jaringan. *Web service* memiliki sebuah antarmuka yang digambarkan dalam sebuah format yang dapat diproses mesin (khususnya WSDL). Sistem lain berinteraksi dengan *Web service* dengan cara yang ditentukan oleh deskripsi *Web service*. Deskripsi ini menggunakan pesan SOAP yang disampaikan menggunakan HTTP dengan sebuah serialisasi XML yang dengan berhubungan standar web lain yang. (W3C 2004)

**Komponen Web Service**

Komponen utama arsitektur *web service* adalah *service provider*, *service registry*, dan *service requester*. Sebuah *service* adalah sebuah aplikasi yang tersedia untuk digunakan oleh *requester* yang sesuai prasyarat awal yang ditetapkan oleh *service provider*. *Web service* dapat disusun dengan berbagai *service* lain menjadi *service* atau aplikasi baru. Berbagai *service* disebar pada suatu tempat pada *web* oleh *service provider*. Sebuah *service* tertentu yang disebut *registry* menyediakan dukungan untuk mempublikasikan dan menemukan *service*.

*Registry* merupakan sebuah tempat penyimpanan deskripsi *service* yang dapat dicari dimana *service provider* mempublikasikan *service*-nya. Sebuah bahasa deskripsi digunakan untuk mendeskripsikan *web service*. Fungsionalitas dan kebijakan akses dicatat dan diterbitkan dengan sebuah *registry*. Berbagai *service* digunakan melalui sebuah jaringan dengan menggunakan informasi yang disimpan dalam deskripsi *service* (Menasce & Almeida 2002).

*Web service* bersandar pada pola SOA. Pemetaan pola SOA dalam *web service* beserta komponennya dapat dilihat pada Gambar 2 (Manes 2003).



Gambar 2 Pemetaan pola SOA dalam *web service*.

**Protokol Web Service**

*Web service* berdasarkan pada sekumpulan protokol kunci. Protokol-protokol ini merupakan blok bangunan *web service platform*. Protokol utama adalah (Turban et al 2005):

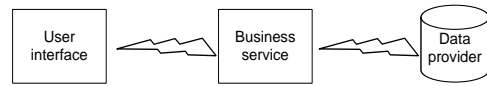
- XML  
*Extensible Markup Language* membuat *web service* lebih mudah bertukar data antaraplikasi yang bervariasi dan untuk mengesahkan dan menerjemahkan data tersebut. Sebuah dokumen XML menggambarkan sebuah *web service* dan memasukkan detail informasi bagaimana *web service* dapat dijalankan.
- SOAP  
*Simple Object Access Protocol* adalah seperangkat aturan yang memfasilitasi XML untuk melakukan pertukaran data antaraplikasi jaringan. SOAP mendefinisikan sebuah standar umum yang mengizinkan berbagai *web service* yang berbeda untuk saling beroperasi. SOAP merupakan spesifikasi *platform independent* yang mendefinisikan

bagaimana pesan dapat dikirimkan di antara dua sistem perangkat lunak melalui penggunaan XML. Pesan tersebut khususnya mengikuti pola *request/response*.

- WSDL  
*Web Service Description Language* digunakan untuk menciptakan dokumen XML yang menggambarkan tugas yang dilakukan oleh *web service*. WSDL mendefinisikan antarmuka yang dari *web service*.
- UDDI  
*Universal Description Discovery and Integration* memungkinkan untuk pembuatan direktori publik atau privat. Direktori *web service* ini dapat dicari. UDDI merupakan *registry* dari deskripsi *web service*.
- Protokol Keamanan  
Standar keamanan yang masih dalam tahap pengembangan adalah *Security Assertion Markup Language (SAML)*. Standar ini digunakan untuk autentikasi dan otorisasi. Standar keamanan lainnya adalah XML Signature, XML Encryption, XKMS, dan XACML.

#### Arsitektur Model Aplikasi *Three-Tier*

Berbicara mengenai suatu aplikasi, ada beberapa komponen yang terlibat di dalamnya, yaitu *user interface*, *business service*, dan *data provider*. Lapisan *user interface* merupakan bagian aplikasi yang berinteraksi dengan *user*. Lapisan ini menampilkan semua data yang diperlukan dengan menggunakan objek-objek *window*. Lapisan ini menerima masukan dan modifikasi *user* terhadap data dengan menggunakan objek-objek *window*. Lapisan *business service* mengendalikan semua data yang diakses dan meng-*update* data yang ada dalam basis data. Lapisan ini biasanya dapat digunakan oleh modul-modul lain dalam aplikasi. Lapisan *data provider* bertanggung jawab menyimpan data dan menyediakan data yang akan diberikan ke lapisan *user interface*. Arsitektur model aplikasi *three-tier* memisahkan lapisan *user interface*, *business service*, dan *data provider* dalam bagian yang berbeda sebagaimana terlihat pada Gambar 3 (Hadiwinata 2003).



Gambar 3 Model aplikasi *three-tier*.

#### *Linear Sequential Model*

Pada metode *Linear Sequential Model*, proses pengembangan sistem dibagi menjadi 4 tahap utama, yaitu tahap analisis, tahap perancangan, tahap implementasi dan tahap pengujian. Antara tahap yang satu dengan tahap yang lain saling berhubungan. Pada saat berada pada suatu tahap, dapat melanjutkan ke tahap selanjutnya apabila tahap tersebut dianggap selesai atau kembali ke tahap sebelumnya jika terdapat perubahan atau penambahan proses baru. Tahapan metode *Linear Sequential Model* dapat dilihat pada Gambar 4 (Pressman 2001).



Gambar 4 Metode *linear sequential model*.

- Tahap Analisis  
Kegiatan pada tahap analisis adalah melakukan analisis untuk menentukan kebutuhan (*requirement*) dalam proses pengembangan sistem. Setelah diperoleh kebutuhan dalam pembangunan suatu sistem aplikasi, kemudian disusun diagram konteks. Hasil dari tahap ini adalah deskripsi umum sistem, analisis kebutuhan, dan diagram konteks.
- Tahap Perancangan  
Kegiatan pada tahap perancangan adalah menerjemahkan kebutuhan-kebutuhan yang didefinisikan dalam tahapan analisis ke dalam bentuk model presentasi sistem aplikasi. Pada tahap ini dirancang arsitektur perangkat lunak, antarmuka, input, proses dan output dalam menggunakan aplikasi.
- Tahap Implementasi  
Kegiatan pada tahap implementasi adalah mengimplementasikan rancangan yang telah disusun pada tahap perancangan menjadi instruksi-instruksi program. Dalam tahap implementasi, antarmuka dibuat agar dapat dimengerti untuk memudahkan pengguna dalam menggunakan sistem ini.



1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar IPB.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB.

- Tahap Pengujian

Hasil implementasi kemudian diuji dan dievaluasi, selanjutnya pengujian dibandingkan dengan hasil uji yang diharapkan. Apabila tidak sesuai dengan yang diharapkan akan dilakukan perbaikan kemudian diuji kembali, sampai memenuhi hasil yang diharapkan.

### METODE PENELITIAN

Pengembangan sistem informasi akademik akan menerapkan SOA dengan metode pengembangan *linear sequential model*.

#### Analisis Kebutuhan

Tahap ini dilakukan untuk mengidentifikasi kebutuhan sistem yang akan dikembangkan dan tipe pengguna sistem. Kebutuhan meliputi kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional dimodelkan dengan pembuatan diagram *use case* dan instrumen lain untuk memudahkan pemahaman dalam tahap selanjutnya. Kebutuhan sistem dipengaruhi oleh tipe pengguna sistem.

#### Perancangan

Aktivitas dalam tahap perancangan meliputi perancangan basis data yang diwujudkan dengan pembuatan *Entity Relationship Diagram* (ERD). Pembuatan diagram *Unified Modelling Language* (UML) untuk pemodelan objek dalam sistem baik objek yang berfungsi sebagai penyedia *service* maupun eksekutor proses bisnis, serta pemodelan arsitektur sistem.

#### Implementasi

Tahap ini dilakukan untuk menransformasi model pada tahap perancangan menjadi kode-kode yang bisa dieksekusi komputer untuk setiap *tier* sistem. Untuk *data provider*, wujud implementasinya adalah kueri berbasis pemrograman yang dapat dieksekusi oleh DBMS. Untuk *Business service*, wujud implementasinya adalah kumpulan kelas yang saling bekerjasama untuk mencapai tujuan tertentu. Kumpulan kelas tersebut digunakan untuk pembuatan *service*. Untuk *user interface* wujud implementasinya adalah antarmuka yang berinteraksi langsung dengan pengguna serta meminta *service*. Berbagai *service* yang dihasilkan dipublikasikan ke server UDDI.

### Pengujian

Tahap ini dilakukan dengan memeriksa apakah semua *service* telah memenuhi spesifikasi kebutuhan yang telah ditentukan melalui antarmuka yang telah dibuat. Jika sudah tidak ditemukan kesalahan pada seluruh *service*, suatu perangkat lunak akan dibuat untuk uji *interoperability*. Perangkat lunak ini dibuat dengan lingkungan pengembangan yang berbeda dan mampu mengonsumsi *service* yang telah dibuat.

### Lingkungan Implementasi

Lingkungan implementasi yang digunakan adalah sebagai berikut:

- Perangkat lunak: Microsoft Windows Server 2003 SP 2, IIS Versi 6, Microsoft SQL Server 2005, Microsoft Visual Studio 2005, Microsoft Office Visio 2003, Crystal report 11, VMWare Workstation 4.5.2, Microsoft Windows XP SP2, WOS Portable 2 (PHP 5.1.6, Apache 2, Mysql 5), Zend Studio 5, Mozilla Firefox 2.
- Perangkat keras: Prosesor Intel Pentium IV 2.8 GHz, RAM 1430 MB, *harddisk* 80 GB, *keyboard*, *mouse*, dan monitor.

### HASIL DAN PEMBAHASAN

#### Analisis kebutuhan

Tahap analisis kebutuhan meliputi identifikasi pengguna, membuat daftar kebutuhan fungsional serta pemodelannya, dan mengidentifikasi kebutuhan non fungsional.

#### Identifikasi Pengguna Sistem

Pengguna sistem dikategorikan menjadi tiga, yaitu mahasiswa, pegawai staf akademik departemen, dan staf kantor Administrasi dan Jaminan Mutu Pendidikan (AJMP). Setiap kategori pengguna memiliki hak akses yang berbeda terhadap fitur sistem. Setiap pengguna sistem memiliki identitas pengenal dan *password* yang unik dan disimpan dalam basis data. Identitas pengenal dan *password* digunakan tatkala akan memasuki sistem.

#### Kebutuhan Fungsional

Sistem yang dikembangkan harus mampu melakukan operasi terhadap data dalam basis data. Operasi meliputi penambahan, penghapusan, pengubahan dan pencarian yang biasa disingkat CRUD (*Create, Read, Update, Delete*). Data yang dikelola meliputi data