

Web Services untuk Sinkronisasi Data Learning Management System dengan Sistem Informasi Akademik di Universitas Budi Luhur

Utomo Budiyanto¹, Khabib Mustofa²

Fakultas Teknologi Informasi, Universitas Budi Luhur Jakarta

Jl. Ciledug Raya Petukangan Utara Jakarta Selatan 12260¹

Jurusan Ilmu Komputer dan Elektronika FMIPA Universitas Gadjah Mada, Yogyakarta

Sekip Utara, Yogyakarta, 55521²

Email: utomo.budiyanto@budiluhur.ac.id¹, khabib@ugm.ac.id²

Abstrak

E-Learning digunakan oleh institusi pendidikan untuk menunjang proses pembelajaran. *Learning Management System* (LMS) yang digunakan untuk melakukan pengaturan *E-Learning* sangat beragam dari membuat sendiri maupun menggunakan LMS yang berbayar hingga gratis (*open source*). Institusi pendidikan juga memiliki sistem informasi akademik yang digunakan untuk transaksi akademik seluruh sivitas akademiknya. Universitas Budi Luhur menggunakan perangkat lunak *Moodle* sebagai LMS serta memiliki sistem informasi akademik yang digunakan oleh mahasiswa dan dosen dalam proses belajar mengajar. Problem yang dihadapi saat ini adalah bagaimana mengintegrasikan LMS dengan sistem informasi akademik yang ada, sehingga setiap sistem dapat menghasilkan informasi yang akurat dan konsisten. Masalah ini muncul disebabkan masing-masing sistem memiliki *database* dan *server* yang terpisah. Hasil dari paper ini adalah usulan strategi yang digunakan untuk mengintegrasikan data antar sistem menggunakan *web service*.

Kata kunci: *E-Learning*, *Learning Management System*, *Web Service*

1. Pendahuluan

Universitas Budi Luhur mempunyai visi menjadi universitas unggul dengan standar mutu tertinggi, yang dilandasi kecerdasan dan keluhuran budi, ditopang teknologi informasi yang tertuang dalam rencana strategis universitas [1]. Sebagai institusi pendidikan yang berawal dari Akademi Ilmu Komputer pada tahun 1979, Universitas Budi Luhur memiliki dan mengembangkan sistem informasi untuk membantu proses administrasi serta proses belajar mengajar. Pengembangan aplikasi ini ditangani oleh Direktorat Teknologi Informasi.

Saat ini terdapat aplikasi untuk menunjang proses belajar mengajar, yaitu 1) aplikasi webdosen yang digunakan oleh pengajar untuk melakukan transaksi pengajaran dari absensi mahasiswa hingga menginputkan nilai semester, 2) aplikasi webstudent yang digunakan oleh mahasiswa untuk membantu dalam proses belajar mengajar, 3) aplikasi *e-learning* yang digunakan oleh pengajar dan mahasiswa dalam pembelajaran secara *online*.

Pembelajaran secara *online* digunakan sebagai pelengkap dari pembelajaran tatap muka dengan tujuan untuk meningkatkan kemampuan mahasiswa. Dengan tersedia secara *online* mahasiswa dan pengajar dapat memanfaatkannya tanpa terbatas ruang dan waktu, hanya membutuhkan koneksi internet untuk dapat terhubung dengan pembelajaran ini.

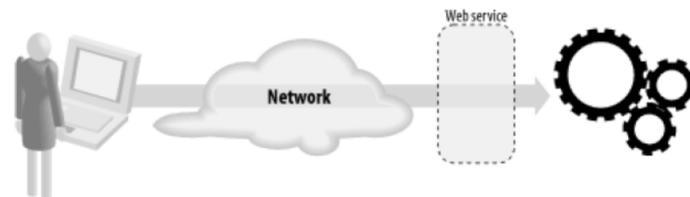
Perkembangan teknologi yang pesat baik dari sisi *hardware* maupun *software* sangat membantu dalam berbagai hal, termasuk dunia pendidikan. *E-learning* sendiri pada dasarnya adalah integrasi dari berbagai teknologi [2], dalam dekade terakhir *e-learning* telah berkembang secara dramatis [11]. Banyak perangkat lunak yang dikembangkan untuk mendukung *e-learning* dari yang berbayar hingga tersedia gratis (*open source*). Universitas Budi Luhur menggunakan perangkat lunak LMS *open source Moodle* (<http://www.moodle.org>) untuk manajemen pembelajaran. Moodle memiliki fitur yang lengkap untuk *e-learning* dan digunakan oleh banyak institusi pendidikan di dunia.

Problem yang dihadapi sekarang adalah bagaimana mengintegrasikan LMS dengan sistem informasi akademik yang ada, sehingga setiap sistem dapat menghasilkan informasi yang akurat dan konsisten. Masalah ini muncul disebabkan masing-masing sistem memiliki *database* dan *server* yang terpisah. Dengan kemampuan interoperabilitas dari *web services*, akan disusun strategi untuk

mengintegrasikan sistem akademik yang sudah berjalan dengan LMS Moodle yang digunakan sebagai perangkat lunak *e-learning* untuk menyelesaikan masalah ini.

2. Konsep Web Service

Pengertian *web services* menurut [3] sebagai berikut *Web service* adalah interface yang dapat diakses melalui jaringan untuk fungsionalitas aplikasi, yang dibangun menggunakan standard teknologi internet, seperti terlihat pada gambar 1. Dengan kata lain, jika sebuah aplikasi dapat diakses melalui jaringan menggunakan kombinasi protokol seperti HTTP, XML, SMTP, atau Jabber, maka itu adalah *Web service*.



Gambar 1. *Web service* memungkinkan akses aplikasi menggunakan standard teknologi Internet

Web service adalah sebuah antarmuka yang terletak di antara kode aplikasi dan pengguna kode tersebut, berperan sebagai lapisan abstrak yang memisahkan *platform* dan rincian spesifik Bahasa pemrograman tentang bagaimana kode aplikasi sebenarnya dipanggil. Lapisan standar ini berarti bahwa setiap bahasa yang mendukung *web service* dapat mengakses fungsionalitas aplikasi, seperti terlihat pada gambar 1 dan 2.



Gambar 2. Web services menyediakan lapisan abstrak antara kode aplikasi dengan aplikasi pengguna

Layanan web yang kita lihat dikembangkan di Internet saat ini adalah situs web HTML. Dalam hal ini, layanan aplikasi seperti mekanisme untuk publikasi, mengelola, mencari, dan mengambil konten diakses melalui penggunaan protokol standar dan format data: HTTP dan HTML. Aplikasi *client* (*web browser*) yang memahami standar-standar ini dapat berinteraksi dengan layanan aplikasi untuk melakukan tugas-tugas seperti memesan buku, mengirim kartu ucapan, atau membaca berita.

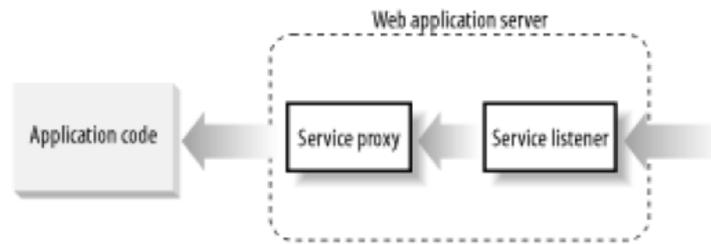
Karena abstrak disediakan oleh antarmuka berbasis standar, tidak menjadi masalah ketika layanan aplikasi yang ditulis dalam Java dan *browser* yang ditulis dalam C++, atau layanan aplikasi dikembangkan pada *platform* Unix sementara *browser* yang digunakan pada *platform* Windows. Layanan web memungkinkan untuk interoperabilitas lintas *platform*.

Interoperabilitas adalah salah satu manfaat utama yang diperoleh dari implementasi layanan web. Solusi berbasis Java dan Microsoft Windows biasanya sulit untuk diintegrasikan, tapi lapisan layanan web antara aplikasi dan klien dapat menghilangkan hambatan ini.

Web service merupakan kerangka pengiriman pesan. Satu-satunya persyaratan yang ditempatkan pada layanan web adalah bahwa hal itu harus mampu mengirim dan menerima pesan menggunakan beberapa kombinasi dari protokol Internet standar. Bentuk yang paling umum dari layanan web adalah untuk memanggil prosedur yang berjalan pada server, dalam kasus ini mengkodekan pesan "Panggil subrutin ini dengan argumen ini," dan "Berikut adalah hasil dari panggilan subrutine."

Pada gambar 3 terlihat bagian dari layanan web. Kode aplikasi menampung semua logika bisnis dan kode untuk melakukan hal-hal (daftar buku, menambahkan buku ke keranjang belanja, pembayaran buku, dll). *Service Listener* berbicara protokol transport (HTTP, SOAP, Jabber, dll) dan menerima permintaan masuk. *Service Proxy* menerjemahkan permintaan tersebut menjadi panggilan menjadi ke

dalam kode aplikasi. *Service Proxy* kemudian mengkodekan respon untuk *Service Listener* untuk membalas, namun ada kemungkinan langkah ini diabaikan.



Gambar 3. Web service terdiri dari beberapa komponen kunci

Komponen *Service Proxy* dan *Service Listener* dapat berupa aplikasi mandiri (misalnya *TCP server* atau *HTTP server daemon*) atau dapat berjalan dalam konteks beberapa jenis lain dari server aplikasi. Yang perlu diingat adalah layanan web tidak memerlukan lingkungan server untuk menjalankannya. Layanan Web dapat digunakan di mana saja dimana standard teknologi internet dapat digunakan. Ini berarti bahwa layanan web dapat dikelola atau digunakan oleh apa pun dari *farm server Application Service Provider* yang luas hingga PDA.

Dalam arsitektur *Web service* seperti pada gambar 4, *Service Provider* mempublikasikan deskripsi layanan yang ditawarkan melalui *Service Registry*. *Service Consumer* mencari *Service Registry* untuk mencari layanan yang memenuhi kebutuhan mereka. *Service Consumer* dapat berupa seseorang atau sebuah program.



Gambar 4. Web service arsitektur

Arsitektur *Web service* diimplementasikan melalui lapisan dari lima jenis teknologi yang disusun diatas satu dengan lainnya seperti pada gambar 5.



Gambar 5. Web service stack

Discovery

Lapisan *Discovery* menyediakan mekanisme bagi konsumen untuk mengambil deskripsi *Provider*. Salah satu mekanisme *discovery* yang tersedia telah diakui secara luas adalah proyek *Universal Description, Discovery, and Integration (UDDI)*. IBM dan Microsoft telah bersama-sama mengusulkan alternatif untuk UDDI, yaitu *Web Services Inspection Language (WSInspection)*.

Description

Ketika *Web service* diimplementasikan, harus membuat keputusan yang mendukung pada setiap tingkatan untuk *network*, *transport* dan *packaging protocols*. Penjelasan dari layanan yang mewakili

keputusan sedemikian rupa sehingga *Service Consumer* dapat menghubungi dan menggunakan layanan ini.

Web Service Description Language (WSDL) adalah standar de facto untuk memberikan gambaran. Lainnya pendekatan yang kurang populer, termasuk penggunaan *W3C Resource Description Framework* (RDF) dan *DARPA Agent Markup Language* (DAML), keduanya menyediakan kemampuan mendeskripsikan layanan web yang jauh lebih kaya (tapi jauh lebih kompleks) daripada WSDL.

Packaging

Untuk data aplikasi yang akan dipindahkan di sekitar jaringan oleh lapisan transport, harus dikemas dalam format yang dapat dipahami oleh semua pihak (istilah lain untuk proses ini adalah serialisasi dan *marshalling*). Meliputi pilihan tipe data yang dipahami, pengkodean nilai-nilai dan sebagainya.

HTML merupakan bentuk dari format *packaging*, tetapi tidak nyaman untuk dibawa bekerja karena HTML sangat terkait dengan penyajian informasi daripada maknanya. XML merupakan dasar dari sebagian besar format *packaging Web service* ini karena dapat digunakan untuk mewakili arti dari data yang ditransfer, juga karena parser XML sekarang tersedia di mana-mana. Ada beberapa format yang digunakan berbasis XML, antara lain XML-RPC dan SOAP.

Transport

Lapisan *transport* mencakup berbagai teknologi yang memungkinkan komunikasi langsung antar aplikasi di atas lapisan *network*. Teknologi tersebut termasuk protokol seperti TCP, HTTP, SMTP, dan Jabber. Peran utama dari lapisan *transport* adalah untuk memindahkan data antara dua lokasi atau lebih pada jaringan. *Web service* dapat dibangun di atas hampir semua protokol *transport*.

Pemilihan protokol *transport* sebagian besar didasarkan pada kebutuhan komunikasi *Web service* yang diimplementasikan. HTTP misalnya, menyediakan dukungan *firewall* yang paling ada dimana-mana tetapi tidak menyediakan dukungan untuk komunikasi *asynchronous*. Di sisi lain, Jabber sementara ini tidak standar, mampu menyediakan jalur komunikasi *asynchronous* yang baik.

Network

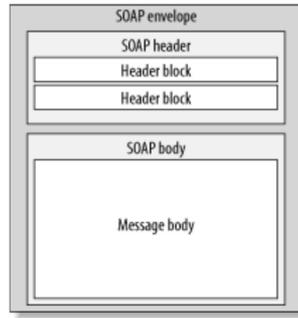
Lapisan *network* dalam tumpukan teknologi *Web service* sama persis dengan lapisan *network* dalam Jaringan Model TCP/IP yang menyediakan kemampuan komunikasi dasar, peng-alamatan dan rutean.

2.1 SOAP

SOAP (*Simple Object Access Protocol*) adalah XML, yang memiliki makna SOAP adalah sebuah aplikasi dari spesifikasi XML. SOAP sangat bergantung pada standar XML seperti XML *Schema* dan XML *Namspaces* untuk definisi dan fungsinya

Tempat SOAP pada tumpukan teknologi *Web service* adalah sebagai protokol standar untuk paket pesan yang dibagikan oleh aplikasi. Spesifikasi pendefinisian tidak lebih dari sebuah amplop berbasis XML yang sederhana untuk informasi yang ditransfer, dan satu set aturan untuk menerjemahkan aplikasi dan tipe data *platform* tertentu ke dalam representasi XML. Desain SOAP membuatnya cocok untuk berbagai aplikasi *messaging* dan integrasi pola. Hal ini memberikan kontribusi besar dalam kepopulerannya.

Sebuah pesan SOAP terdiri dari sebuah amplop berisi *header* opsional dan *body* yang diperlukan seperti pada gambar 6. *Header* berisi blok informasi yang relevan dengan bagaimana pesan tersebut diproses. termasuk *routing* dan pengaturan pengiriman, pernyataan otentikasi atau otorisasi dan konteks transaksi. Tubuh berisi pesan yang sebenarnya harus disampaikan dan diproses. Apa pun yang dapat dinyatakan dalam sintaks XML bisa masuk tubuh pesan.



Gambar 6. Struktur pesan SOAP

Contoh pesan *request* SOAP

```
<s:Envelope
  xmlns:s="http://www.w3.org/2001/06/soap-envelope">
  <s:Header>
    <m:transaction xmlns:m="soap-transaction"
      s:mustUnderstand="true">
      <transactionID>1234</transactionID>
    </m:transaction>
  </s:Header>
  <s:Body>
    <n:getQuote xmlns:n="urn:QuoteService">
      <symbol xsi:type="xsd:string">
        IBM
      </symbol>
    </n:getQuote>
  </s:Body>
</s:Envelope>
```

Contoh pesan *response* SOAP

```
<s:Envelope
  xmlns:s="http://www.w3.org/2001/06/soap-envelope">
  <s:Body>
    <n:getQuoteResponse
      xmlns:n="urn:QuoteService">
      <value xsi:type="xsd:float">
        98.06
      </value>
    </n:getQuoteResponse>
  </s:Body>
</s:Envelope>
```

3. Penelitian Terkait

Beberapa penelitian terkait pemanfaatan *web services* dengan *e-learning* adalah:

Penggunaan sistem manajemen pembelajaran seperti Moodle mendukung proses pendidikan. Tapi hasil terdapat redundansi dalam membuat *course*, *account* siswa, pendaftaran dan lain-lain di *database* registrasi universitas dan *database* Moodle. Solusi asinkronisasi antara *database* Moodle dan *database* pendaftaran mahasiswa disajikan untuk mengurangi overhead ini. Dengan menyediakan *script* yang dapat melakukan sinkronisasi semua pendaftaran pengguna dalam *database* penerimaan universitas dan *database* Moodle. Moodle dikonfigurasi untuk mengotentikasi pengguna dari server LDAP dan secara otomatis membuat *account* bagi siswa yang tidak memiliki *account* Moodle. Kemudian, setelah pengguna log in ke Moodle, Moodle memeriksa *database* untuk entri pendaftaran. Jika siswa tersebut terdaftar dalam *course*, Moodle secara otomatis akan mendaftarkan siswa dalam *course* yang sesuai. Moodle juga dapat membuat *course* secara otomatis. Hasil uji coba menunjukkan validitas dan kebenaran dari metodologi yang diusulkan [5].

Biasanya lembaga pendidikan tinggi (IHE) mengalami kesulitan untuk memperoleh informasi yang terintegrasi dari sistem informasi mereka, karena kebanyakan dari mereka masih bekerja dengan sistem terisolasi dan tanpa prospek untuk menyediakan layanan terpadu. Dalam konteks ini *Service Oriented Architecture* (SOA) dan, khususnya *Web Services* adalah pendekatan yang menjanjikan untuk integrasi sistem informasi dengan menerapkan layanan dapat digunakan kembali dan *interoperable*. Kurangnya integrasi antara manajemen akademik yang ada dan sistem pembelajaran jarak jauh merupakan masalah bagi lembaga pendidikan tinggi yang mengarah ke duplikasi data dan kurangnya informasi koherensi. Kontribusi untuk penyelesaian beberapa masalah ini dengan menerapkan model integrasi dari dua sistem yang digunakan di sebagian besar lembaga pendidikan tinggi: Moodle dan Sistem Manajemen Akademik. Model dibuat melalui *Web Services* memungkinkan Moodle untuk mendapatkan informasi secara otomatis dari sistem manajemen akademik. Dengan cara ini berkontribusi pada penurunan yang signifikan dari tugas-tugas administrasi Moodle, meningkatkan kualitas informasi yang diberikan kepada pengguna, menghemat waktu, tenaga dan mengurangi jumlah kesalahan yang melekat pada tugas-tugas manual dan manipulasi informasi. Kontribusi juga dihasilkan dalam beberapa cara untuk peningkatan kualitas pendidikan yang diajarkan di lembaga-lembaga pendidikan tinggi karena membebaskan guru dari beberapa tugas administratif, yang memungkinkan mereka untuk mencurahkan lebih banyak waktu untuk fungsi utama mereka. Di sisi lain juga memberikan siswa dengan informasi yang lebih baik tentang data akademik mereka. Fitur yang disertakan dalam model memungkinkan untuk memperoleh sistem manajemen akademik dengan semua informasi yang diperlukan yaitu: membuat *account*; membuat *course*; membuat kurikulum, asosiasi pengguna untuk *course*; *import* kalender [6].

Pengurangan biaya merupakan masalah utama dalam pembangunan sistem *e-learning*. Komponen perangkat lunak *web services* memiliki karakteristik dapat digunakan kembali dan dapat saling dipertukarkan, oleh karena itu dapat mengurangi pemborosan sumber daya pendidikan, serta biaya pengembangan sistem yang lebih rendah. Digunakan komponen *web services* yang dikembangkan dengan Microsoft .NET dan XML teknologi untuk membangun sebuah *platform* pengajaran dengan spesifikasi standar, yang memungkinkan pengembang sistem untuk secara cepat membangun sebuah sistem *e-learning* berbasis pada *web services* [1].

Learning Management Systems (LMS) lebih penting dalam *e-learning* dan pendidikan di masa depan. LMS menyediakan fungsi dan layanan pendidikan untuk mengelola peserta didik dan isi pembelajaran. Dengan pertumbuhan internet, berbagai konten pembelajaran dan sistem belajar yang berbeda didistribusikan secara independen. Dengan demikian, mengelola informasi yang terdistribusi dan menerapkan interoperabilitas antara sistem pembelajaran menjadi lebih menantang. *Web services* adalah platform implementasi yang paling inovatif dan mapan untuk komunikasi antara aplikasi. Diusulkan desain arsitektur komunikasi untuk LMS termasuk interaksi *web services*, desain layanan yang esensial dan pengembangan aplikasi. *Web services* dan aplikasi yang dirancang menggunakan standar struktur data LMS. Struktur data ini disusun kembali ke dalam bentuk baru yang tergantung pada kebutuhan aplikasi remote. Desain arsitektur komunikasi *web services* membuat standar LMS mampu menghubungkan LMS dan aplikasi lain secara bersama. Arsitektur komunikasi yang diusulkan juga membuat LMS standar untuk mengaktifkan fitur interoperabilitas. Arsitektur komunikasi *web services* untuk LMS yang diusulkan untuk menambahkan interoperabilitas LMS standar yang ada untuk dapat dibagikan dan mendefinisikan layanan esensial yang dibutuhkan untuk interaksi antara LMS dan sistem lain dalam rangka untuk memberikan materi dari LMS standar ke aplikasi lain. Prototipe telah diimplementasikan pada LMS Moodle dengan mengintegrasikan *web services*. Selain itu, manfaat dari pengembangan *web services* untuk LMS berguna dalam *e-learning* terutama dalam administrasi LMS [7].

Satu hal yang perlu dipahami adalah bahwa *e-learning* pada dasarnya adalah integrasi dari berbagai teknologi. Teknologi saat ini telah matang dan kita dapat menemukan standar yang berbeda untuk *e-learning*. Teknologi baru seperti agen dan *web services* yang menjanjikan hasil yang lebih baik. Diusulkan sebuah arsitektur *e-learning* yang tergantung pada sistem multi-agent dan *web services*. Teknologi komunikasi ini akan membuat arsitektur yang lebih kuat, terukur dan efisien. Arsitektur berorientasi *service* untuk *e-learning* akan sangat meningkatkan interoperabilitas, skalabilitas, kemudahan perawatan, fleksibilitas, kustomisasi dan interaktivitas dalam lingkungan *e-learning* heterogen. Menggunakan *web services* dan teknologi agen cerdas memudahkan untuk mem-program

dan menjaga *services* di *server*. Agen pengguna pada sisi klien dapat melakukan banyak tugas bagi pengguna dan terus membantu dalam pembelajaran [2].

E-learning telah berkembang secara dramatis dalam dekade terakhir. Dengan munculnya semakin banyak teknologi canggih seperti *web services*, pengembangan yang ringan, fleksibel dan sistem *e-learning* cerdas menjadi kenyataan. Diusulkan sebuah arsitektur baru untuk sistem *e-learning* berbasis pada *web services* dan agen cerdas. Arsitektur ini menyediakan model integrasi yang fleksibel di mana semua komponen pembelajaran dan aplikasi yang tidak terikat dan dapat didistribusikan di Internet. Selain itu, melalui penggunaan agen, konten pembelajaran dapat secara cerdas disesuaikan agar sesuai dengan konteks dan kebutuhan belajar khusus untuk pengguna [11].

4. Pembahasan

4.1 Sistem informasi akademik dan *e-learning* di Universitas Budi Luhur

Sistem informasi akademik yang ada di Universitas Budi Luhur yang berhubungan dengan proses belajar mengajar terbagi atas 1) aplikasi webdosen (<http://webdosen.budiluhur.ac.id>) seperti terlihat pada gambar 7 yang digunakan oleh pengajar untuk proses belajar mengajar yang memiliki fasilitas yang berhubungan dengan pembelajaran sbb: a) absen dan berita acara b) entry nilai c) history mengajar d) jadwal mengajar dosen e) pilih hari pengganti f) daftar SAP, untuk masuk ke dalam aplikasi ini pengajar harus memasukan nomor induk dan passwordnya. Aplikasi ini menggunakan *script* PHP dan *database* ORACLE dan MySQL.



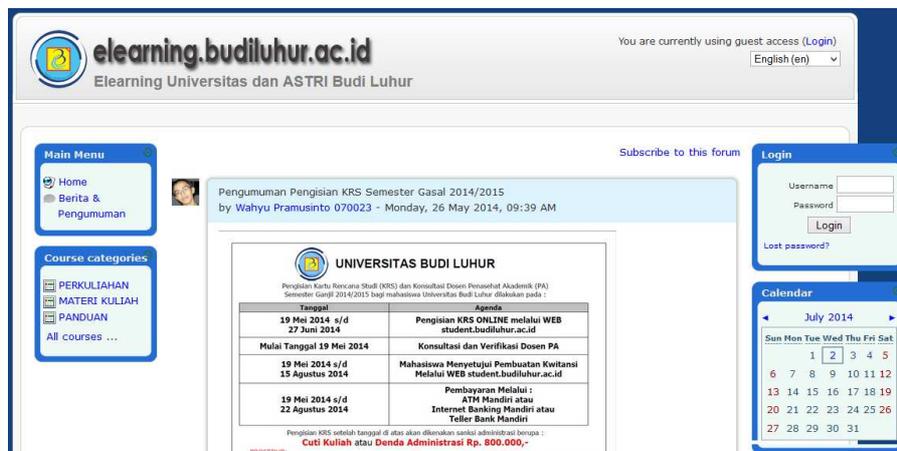
Gambar 7. Aplikasi webdosen (<http://webdosen.budiluhur.ac.id>)

2) aplikasi webstudent (<http://student.budiluhur.ac.id>) seperti pada gambar 8, digunakan oleh mahasiswa dalam proses perkuliahan, fasilitas yang berhubungan dengan proses belajar mengajar adalah sbb: a) jadwal kuliah b) HSK (Hasil Studi Kumulatif) c) nilai semester. Untuk dapat mengaksesnya mahasiswa harus memasukan NIM dan passwordnya. Aplikasi ini menggunakan *script* PHP dan *database* ORACLE dan MySQL



Gambar 8. Aplikasi webstudent (<http://student.budiluhur.ac.id>)

3) aplikasi *e-learning* (<http://elearning.budiluhur.ac.id>) seperti gambar 9, digunakan untuk melengkapi pembelajaran tatap muka secara langsung. Untuk masuk ke dalam aplikasi ini pengajar dan mahasiswa harus memasukan NIP (pengajar) atau NIM (mahasiswa) beserta passwordnya. Aplikasi ini menggunakan *script* PHP dan *database* MySQL.



Gambar 9. Aplikasi E-Learning (<http://elearning.budiluhur.ac.id>)

Aplikasi-aplikasi tersebut terdapat pada *server* dan *database* yang berbeda, sehingga memungkinkan terjadi informasi yang tidak akurat dan tidak konsisten. Oleh karena itu dibutuhkan sebuah strategi agar hal tersebut dapat diatasi.

4.2 Strategi Pemodelan Integrasi sistem dengan Web Services

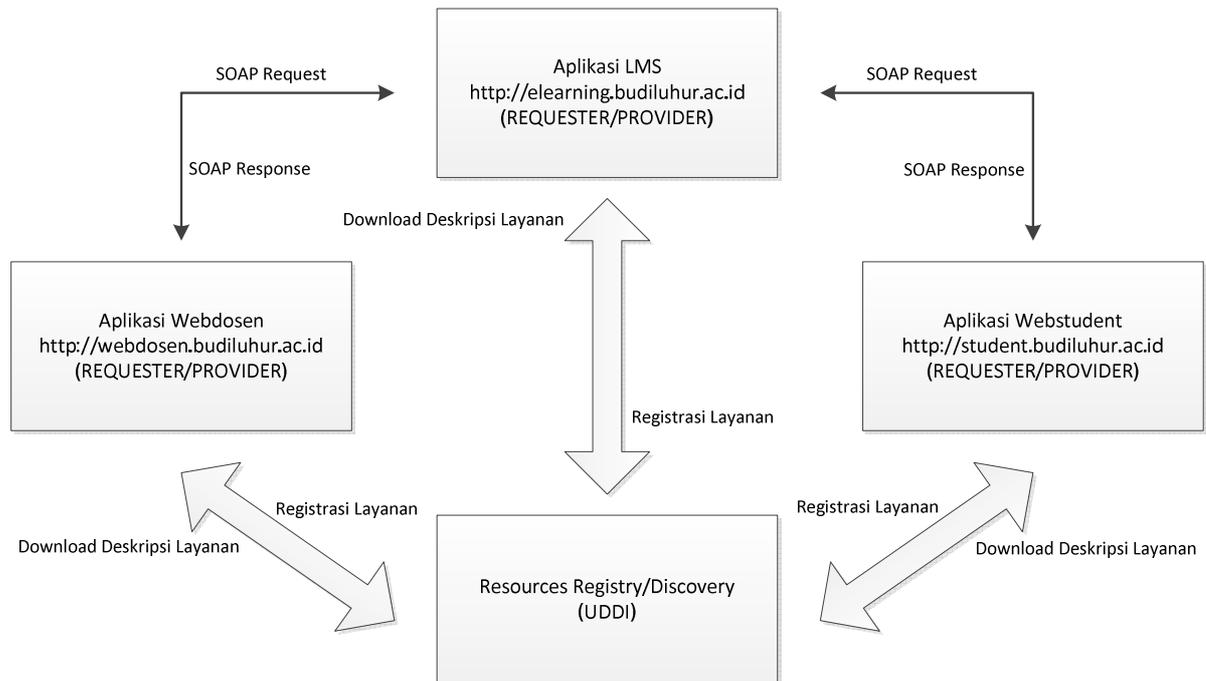
Masalah yang dihadapi ketika pengajar dan mahasiswa masuk ke dalam LMS adalah:

- authorisasi pengguna, yang disebabkan LMS memiliki *database* sendiri untuk menyimpan data *user*, sedangkan aplikasi webdosen dan webstudent juga demikian. Sehingga LMS perlu menambahkan data pengajar dan mahasiswa ke dalam *database* LMS, hal ini akan menyebabkan ketidak konsistenan data ketika ada perubahan password yang dilakukan pada masing-masing aplikasi.
- tracking* aktifitas, LMS menyediakan fasilitas untuk pengguna melihat semua aktifitas selama menggunakan *e-learning*, sedangkan aplikasi lainnya tidak dapat mengetahui apa yang

dilakukan pengguna ketika berinteraksi dengan LMS sebab informasi *tracking* disimpan pada *database* LMS. Pengajar melalui webdosen (<http://webdosen.budiluhur.ac.id>) membutuhkan informasi ini untuk melakukan penilaian terhadap mahasiswa dari aktifitas yang dilakukan (untuk absensi serta tugas). Mahasiswa melalui webstudent (<http://student.budiluhur.ac.id>) membutuhkan informasi aktifitas untuk dapat melihat aktifitasnya di LMS.

- c. data pengguna, LMS membutuhkan data pengguna (pengajar dan mahasiswa) untuk dapat ditampilkan pada *e-learning*, sementara data tersebut disimpan pada *database* yang tersebar pada aplikasi lain yaitu webdosen dan webstudent.

Dengan masalah yang dihadapi di atas, diperlukan strategi untuk mengintegrasikan antar aplikasi dengan menggunakan *web services*. Model yang diusulkan seperti pada gambar 10.



Gambar 10. Model web services pengambilan data sistem informasi akademik dengan LMS

Berdasarkan model pada gambar 10, semua aplikasi mendaftarkan layanannya ke *Resource Registry* kemudian semua aplikasi juga dapat mencari dan mendownload layanan yang disediakan oleh aplikasi lain. Proses *request* dan *response* antar aplikasi menggunakan SOAP, yang dilakukan setelah mendapatkan deskripsi yang diperlukan dari *Resource Registry*.

Ketika aplikasi mendapat *request* melalui *web services* maka akan dilakukan proses yang berhubungan dengan secara internal aplikasi, termasuk ketika ada *request* yang berhubungan dengan *database*, maka akan dilakukan proses pada *database* internal untuk kemudian dikirimkan kembali (*response*) dalam bentuk pesan SOAP. Pesan yang dikembalikan akan diterima oleh aplikasi yang mengirimkan *request* untuk kemudian diterjemahkan sesuai dengan deskripsi yang diberikan. Sebagai contoh ketika aplikasi webdosen mengirimkan *request* **getActivity()** ke aplikasi LMS, maka LMS akan memproses *request* tsb dengan menjalankan secara internal pada aplikasi LMS, kemudian akan dikembalikan (*response*) hasil dalam bentuk SOAP ke aplikasi pemanggil (webdosen).

Dengan model pada gambar 10 maka semua aplikasi dapat berkomunikasi dengan baik, sehingga terintegrasi dan menghasilkan informasi yang akurat dan konsisten.

5. Kesimpulan

Problem integrasi sistem akademik dengan LMS di Universitas Budi Luhur muncul akibat masing-masing aplikasi memiliki *script* dan *database* yang terpisah. Pengembangan dengan menggunakan *web services* yang mendukung interoperabilitas dengan SOAP akan menghilangkan ketidakakuratan informasi dan ketidak konsistenan antar aplikasi. Masing-masing aplikasi menyediakan informasi yang dibutuhkan oleh aplikasi lain, sehingga informasi menjadi akurat dan konsisten.

Referensi

- [1] Chang, C.C.; Kuo-Chan Hsiao, "An e-learning System for Information Management Education Based on Web Services," Computational Aspects of Social Networks (CASoN), 2010 International Conference on , vol., no., pp.221,224, 26-28 Sept. 2010
- [2] Hussain, N.; Khan, M.K., "Service-Oriented E-Learning Architecture Using Web Service-Based Intelligent Agents," Information and Communication Technologies, 2005. ICICT 2005. First International Conference on , vol., no., pp.137,143, 27-28 Aug. 2005
- [3] James Snell , Doug Tidwell , Pavel Kulchenko, Programming Web services with SOAP, O'Reilly & Associates, Inc., Sebastopol, CA, 2002
- [4] Maican, C.I., "Integrated University Management System Based on Open Source Tools," Internet and Web Applications and Services, 2009. ICIW '09. Fourth International Conference on , vol., no., pp.626,631, 24-28 May 2009
- [5] Mikki, M.A., "Synchronizing Moodle and University Enrollment Databases, A Case Study: Islamic University of Gaza," Information and Communication Technology (PICICT), 2013 Palestinian International Conference on , vol., no., pp.8,13, 15-16 April 2013
- [6] Moura, R.; Bernardino, J., "Integration Moodle and academics systems," Information Systems and Technologies (CISTI), 2011 6th Iberian Conference on , vol., no., pp.1,6, 15-18 June 2011
- [7] Phankokkrud, M.; Woraratpanya, K., "Web services for Learning Management Systems: Communication architecture," Communications (MICC), 2009 IEEE 9th Malaysia International Conference on , vol., no., pp.403,408, 15-17 Dec. 2009
- [8] Radenkovic, B.L.; Despotovic-Zrakic, M.S.; Barac, D.M.; Bogdanovic, Z.M.; Milic, A.R., "Web portal for adaptive e-learning," Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), 2011 10th International Conference on , vol.1, no., pp.365,368, 5-8 Oct. 2011
- [9] Reusch, P., "On the scope and the structure of e-learning platforms at universities," Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on , vol.02, no., pp.738,741, 12-14 Sept. 2013
- [10] Universitas Budi Luhur, "Rencana Strategis 2012-2020", 2012
- [11] Xu Wei; Jun Yan, "An E-learning System Architecture Based on Web Services and Intelligent Agents," Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on , vol.2, no., pp.173,177, 12-14 Aug. 2009