



ARSITEKTUR MOBILE LEARNING MANAGEMENT SYSTEM BERORIENTASI SERVIS

Reny Medikawati Taufiq¹⁾

¹⁾Fakultas Ilmu Komputer, Universitas Muhammadiyah Riau

Jl. KH. Ahmad Dahlan No. 88, Sukajadi, Pekanbaru, Telp : (0761) – 35008, Fax : (0761) – 20497

e-mail: renymedikawati@umri.ac.id¹⁾

Pada tesis ini dihasilkan usulan Arsitektur Mobile Learning Management System (MLMS) Berorientasi Servis sebagai model untuk membangun Mobile Learning Application yang lintas platform. Arsitektur MLMS Berorientasi Servis dirancang sebagai solusi mengenai permasalahan keberagaman platform dan keterbatasan (bandwidth, memori, processor, konektivitas, keamanan, display) pada perangkat mobile. Adanya beberapa platform perangkat mobile (Blackberry, Symbian, Windows Mobile, Apple's iOS, Android, dll) menyulitkan ketika ingin membuat sebuah aplikasi yang dapat bekerja secara lintas platform. Untuk platform yang berbeda, aplikasi harus dibuat ulang, yang tentunya memakan waktu, biaya, serta usaha dan sumber daya yang tidak sedikit. Arsitektur MLMS Berorientasi Servis dihasilkan dengan melakukan analisis terhadap solusi multiplatform yang sudah ada, analisis Service Oriented Architecture (SOA) sebagai solusi multiplatform, analisis keterbatasan perangkat mobile. Analisis pemanfaatan Learning Management System (LMS) pada sisi server. Dari hasil analisis, metode SOA dengan thin native client merupakan solusi yang dapat mengatasi masalah multiplatform. LMS yang sudah ada dapat digunakan sebagai back-end. Dengan diimplementasikannya konsep SOA, maka dibangun servis-servis sebagai perantara logika bisnis dan logika aplikasi. Servis membungkus fitur-fitur LMS untuk menyediakan layanan bagi logika bisnis. Analisis dan perancangan servis dilakukan dengan metode Mainstream SOA Methodology (MSOAM) yang menghasilkan 15 servis. Pengujian dan evaluasi terhadap arsitektur yang telah dirancang dilakukan dengan mengimplementasikannya menjadi sebuah aplikasi MLMS berbasis Android yang memanfaatkan Moodle di sisi server. 15 servis yang ada melayani fungsionalitas MLMS antara lain (1) otentikasi peserta, (2) pendaftaran kursus, (3) mengikuti kursus, (4) melihat informasi nilai, (5) melihat informasi tugas. Hasil pengujian menunjukkan bahwa arsitektur yang diusulkan dapat diimplementasikan dengan baik..

Kata Kunci : LMS, MLMS, MSOAM, multiplatform, SOA

ABSTRACT

In this thesis a Service Oriented Mobile Learning Management System (MLMS) architecture was proposed as a model for building cross platform Mobile Learning Applications. Service-oriented MLMS architecture is designed as a solution to the problems of platform diversity and limitations (bandwidth, memory, processor, connectivity, security, display) on mobile devices. The existence of several mobile device platforms (Blackberry, Symbian, Windows Mobile, Apple's iOS, Android, etc.) makes it difficult to create an application that can work across platforms. For different platforms, the application must be regenerated, which certainly takes time, costs, and effort and resources that are not small. Service-oriented MLMS architecture is produced by analyzing existing multiplatform solutions, analyzing Service Oriented Architecture (SOA) as a multiplatform solution, analyzing the limitations of mobile devices. Analysis of the utilization of the Learning Management System (LMS) on the server side. From the results of the analysis, the SOA method with thin native client is a solution that can overcome multiplatform problems. Existing LMS can be used as a back-end. With the implementation of the SOA concept, services are built as intermediaries for business logic and application logic. The service wraps the LMS features to provide services for business logic. Analysis and design of services is carried out using the Mainstream SOA Methodology (MSOAM) method which produces 15 services. Testing and evaluation of the architecture that has been designed is done by implementing it into an Android-based MLMS application that uses Moodle on the server side. 15 services that serve MLMS functionality include (1) authenticating participants, (2) enrolling courses, (3) taking courses, (4) viewing value information, (5) viewing task information. The test results show that the proposed architecture can be implemented properly.

Keyword : LMS, MLMS, MSOAM, multiplatform, SOA

I. PENDAHULUAN

TEKNOLOGI *mobile* merupakan bagian yang sudah sangat sering digunakan dalam kehidupan sehari-hari. Dengan adanya teknologi *mobile*, komunikasi dapat dilakukan kapanpun dan dimanapun. Banyak hal yang dapat dilakukan dengan menggunakan satu perangkat *mobile* seperti mengakses berbagai jenis informasi, mengambil foto, merekam gambar bergerak, suara dan teks. Teknologi *mobile* terus berkembang, produsen perangkat *mobile* menawarkan teknologi dengan kemampuan multimedia yang semakin canggih. Pada tahun 2013 diprediksikan bahwa jumlah perangkat *mobile* yang aktif digunakan- seperti *smartphone* dan *mobile phone* yang dilengkapi dengan browser akan lebih banyak dibandingkan dengan jumlah *personal computer* yang aktif digunakan “[5]”. Hal ini merupakan tantangan bagi pendidik dan perancang aplikasi untuk melakukan eksplorasi mengenai sejauh mana teknologi *mobile* dapat digunakan untuk mendukung proses belajar yang kaya, yang dapat dilakukan baik di kelas, di rumah, maupun ketika duduk di taman, atau ketika sedang makan di kafetaria.

Teknologi dan pembelajaran dapat dipadukan, sehingga menghasilkan proses belajar yang interaktif dan dapat dilakukan di mana saja, kapan saja. Salah satu contohnya adalah *mobile learning management system (MLMS)* yang digunakan untuk mengelola aktivitas *mobile learning*. *Mobile learning* membawa manfaat yang signifikan yaitu ketersediaan materi ajar yang dapat diakses setiap saat dan visualisasi materi yang menarik. Namun ada beberapa kendala yaitu, jika dibandingkan dengan *personal computer*, perangkat *mobile* memiliki keterbatasan seperti bandwidth komunikasi, sumber daya dan metode presentasi serta interaksi. Selain itu dengan adanya berbagai jenis *platform* sistem operasi seperti Android, Blackberry, Windows Mobile, Symbian, serta iPhone yang memiliki kemampuan yang berbeda, maka akan sulit untuk membangun sebuah solusi *mobile learning* yang cocok untuk setiap jenis perangkat *mobile* “[10]”.

“[8]” solusi untuk masalah teknis ini salah satunya adalah menggunakan format dan standar yang reusable. Pendekatan yang digunakan antara lain adalah dengan

menggunakan teknik pembangunan software yang memungkinkan developer membangun sebuah aplikasi yang dapat dijalankan pada platform perangkat *mobile* yang berbeda (*cross-platform development*). Arsitektur yang dapat digunakan untuk membangun sistem pembelajaran ini adalah dengan menggunakan *service oriented architecture (SOA)*. Beberapa karakteristik SOA yaitu menggunakan standar terbuka dan mendukung keberagaman vendor “[3]”.

Learning Management System yang sudah ada dapat digunakan untuk menangani pengelolaan aktivitas belajar, karena sistem yang mengatur konten e-learning telah banyak beredar dan mapan, dikembangkan oleh edukator dan pakar teknologi dengan melakukan penelitian “[9]”. LMS dapat digunakan sebagai back-end pada MLMS dengan melakukan teknik integrasi tertentu.

Berdasarkan latar belakang yang telah dipaparkan diatas, rumusan masalah yang menjadi dasar penelitian ini adalah bagaimana merancang suatu arsitektur yang tepat untuk MLMS berorientasi servis. Selanjutnya arsitektur ini diharapkan dapat digunakan oleh para pengembang MLMS, sehingga tidak tergantung pada platform perangkat *mobile* yang digunakan.

II. METODOLOGI

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Studi Literatur, mengenai solusi multiplatform yang telah ada, referensi mengenai konsep dan arsitektur berorientasi servis, metode membangun sistem berbasis SOA, kemungkinan penggunaan LMS sebagai pengelola aktivitas belajar pada MLMS.
2. Analisis hal yang akan dianalisis adalah : (1) Solusi *multiplatform* yang ada dan solusi apa saja yang *feasible* untuk diterapkan pada MLMS, (2) SOA sebagai solusi *multiplatform*, (3) Penggunaan LMS sebagai pengelola aktivitas belajar pada MLMS, (3) Kebutuhan dan keterbatasan perangkat *mobile* sebagai klien pada MLMS.
3. Perancangan Arsitektur MLMS Berorientasi Servis: setelah dilakukan analisis mengenai solusi multiplatform, SOA, pemanfaatan LMS, kebutuhan dan keterbatasan perangkat *mobile*, maka

dirancanglah sebuah arsitektur MLMS yang mengimplementasikan SOA.

4. Pembangunan MLMS berorientasi service : untuk melakukan pengujian terhadap Arsitektur *Mobile Learning Management System* Berorientasi Servis maka dibangunlah MLMS yang mengimplementasikan arsitektur tersebut. Pembangunan MLMS berorientasi servis meliputi dua kegiatan yaitu pembangunan *service provider* yang akan membungkus fungsionalitas LMS dan pembangunan aplikasi *client* yang akan memanfaatkan fungsionalitas LMS via *service provider*. Metode yang digunakan untuk membangun MLMS berbasis SOA ini menggunakan *Mainstream SOA Methodology (MSOAM)* “[4]” yang mendukung protokol REST.
5. Evaluasi : evaluasi dilakukan untuk menilai keberhasilan penerapan usulan arsitektur pada MLMS dan aplikasi *mobile learning* yang dibangun.

III. STUDI LITERATUR

Mobile berasal dari kata latin “*mobilis*” yang artinya “bergerak”. “[7]” Jika dikaitkan dengan teknologi, maka *mobile* secara umum berarti teknologi yang portable (mudah dibawa-bawa) dan personal.

Konvergensi kemampuan perangkat *mobile* , yaitu kombinasi teknologi yang diterapkan pada satu perangkat *mobile*, adalah hal yang memicu terjadinya revolusi *mobile*. Perangkat terbaru memiliki kemampuan yang kaya misalnya input melalui *touch screen* dengan *on screen* atau *built in* keyboard dan *noise-cancelling microphones*; output melalui *vibrant screen* dan *audio* yang berkualitas; sensor seperti GPS, kamera, kompas dan sensor arah; dan konektivitas melalui beberapa metode networking seperti Wi-Fi, Bluetooth, data via phone service, dan kabel. Konvergensi ini memperkaya channel-channel yang menunjang aktivitas belajar, namun tantangannya adalah bagaimana caranya merancang interaktivitas. Channel-channel ini menyediakan cara-cara baru yang mungkin saja dapat menunjang proses belajar “[8]”.

Teknologi *mobile* mejadi semakin konvergen dan *embedded*, satu perangkat

memiliki banyak kemampuan. Teknologi seperti ini dapat memberikan dampak yang baik jika digunakan pada aktivitas belajar. Kegiatan belajar semakin dapat dilaksanakan di luar ruang kelas, baik secara virtual maupun nyata, menjadi lebih situated, personal, kolaboratif dan lifelong.

Mobile platform merupakan lingkungan perangkat *mobile* yang meliputi perangkat keras, perangkat lunak dan komponen lain yang berhubungan dengan lingkungan-yang menyediakan dan memfasilitasi akses terhadap aplikasi “[2]”. *Mobile platform* merupakan hal yang penting dan menentukan dalam mengembangkan aplikasi baru. Tidak hanya berfungsi sebagai *software stack* yang menangani interaksi aplikasi, tetapi juga menyediakan *toolset* untuk pembangunan aplikasi, mengikat *developer* dengan syarat-syarat yang substantif, dan menentukan *user interface* yang dapat digunakan untuk mengontrol bagaimana aplikasi mengakses data dan lokasi fisik “[2]”. Saat ini ada beberapa *mobile software platform* yaitu Blackberry, Symbian, Windows Mobile, Apple’s iOS dan Android, yang paling populer adalah Apple’s iOS dan the Google-Android “[2]”.

Pengembangan aplikasi *mobile* dilakukan berdasarkan *platform* dan lingkungan pengembangan yang spesifik terhadap *platform* perangkat *mobile* yang bersangkutan. Masing-masing *platform* memiliki *source development toolkit (SDK)* tersendiri yang digunakan untuk mengembangkan aplikasi “[1]”. Adanya keberagaman *mobile platform* yang memiliki kemampuan berbeda, menyulitkan ketika ingin membangun sebuah solusi *mobile learning* yang cocok untuk setiap jenis perangkat *mobile* “[10]” yang dapat bekerja secara *multiplatform/cross-platform*. Untuk platform yang berbeda, aplikasi harus dibuat ulang, yang tentunya memakan waktu, biaya, serta usaha dan sumber daya yang tidak sedikit “[5]”.

Permasalahan *multiplatform* ini coba diatasi oleh kaum akademisi maupun kalangan pengembang perangkat lunak serta vendor perangkat *mobile*, solusi yang ditawarkan adalah menggunakan paradigma *aspect oriented, model driven-development, seperation of concern, middleware* dan *service oriented architecture (SOA)*. “[8]” solusi untuk masalah

teknis ini salah satunya adalah menggunakan format dan standar yang *reusable*. Pendekatan yang digunakan antara lain adalah dengan menggunakan teknik pembangunan software yang memungkinkan developer membangun sebuah aplikasi yang dapat dijalankan pada platform perangkat *mobile* yang berbeda (*cross-platform development*).

Dari seluruh solusi ini, solusi yang *feasible* dan dapat diimplementasikan (tidak berbentuk konsep) adalah *middleware* dan SOA. Solusi yang berbasis *middleware* menggunakan konsep sebuah *layer* baru yang berfungsi sebagai jembatan antara aplikasi dan perangkat *mobile*, sedangkan solusi yang berbasis SOA menggunakan konsep semua fitur yang disediakan oleh suatu aplikasi akan dipandang sebagai *services* dan tidak lagi dilihat bagaimana implementasinya “[6]”.

Metode pengembangan sistem berbasis SOA sekilas hampir sama dengan metode untuk membangun sistem terdistribusi, kegiatan yang dilakukan antara lain: perancangan web servis, pengembangan, dan *deployment* sesuai dengan teknologi *front-end* dan *back-end* yang digunakan. Namun untuk mengembangkan sistem berbasis SOA, terdapat fase-fase yang harus disesuaikan dengan konsep berorientasi servis.

REST memiliki aturan dan syarat yang spesifik yang harus diperhatikan ketika mengembangkan REST *service* pada solusi *service oriented*. Untuk itu perlu dipertimbangkan metode yang tepat untuk menggabungkan karakteristik REST dengan metodologi SOA. Salah satu metode yang relevan adalah MSOAM (*Mainstream SOA Methodology*) dengan beberapa penambahan yang signifikan “[4]”. Istilah *mainstream* digunakan karena pada MSOAM dihasilkan dari penelitian yang mendalam pada semua platform SOA yang umum digunakan, sehingga bersifat generik dan tidak eksklusif untuk platform tertentu saja.

Berikut ini merupakan tahapan-tahapan pengembangan SOA dengan menggunakan REST “[4]”:

1. SOA Adoption Planning
2. Service Inventory Analysis
3. Service-Oriented Analysis (Service Modeling)

4. Service-Oriented Design (Service Contract)
5. Service Logic Design
6. Service Development
7. Service Testing
8. Service Deployment and Maintenance
9. Service Usage and Monitoring
10. Service Discovery
11. Service Versioning and Retirement

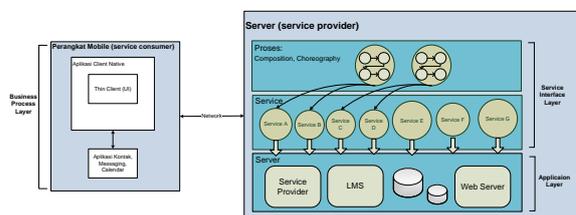
Tahapan-tahapan yang akan dilakukan pada penelitian ini adalah tahapan *service inventory analysis* hingga *service testing*.

IV. ANALISIS DAN PERANCANGAN MODEL

SOA mengusulkan konsep layering pada arsitektur aplikasi, layer pada SOA terdiri dari layer application, service interface, dan business process. Layering dilakukan dengan maksud memisahkan (logic grouping) antara user interface dengan proses bisnis maupun bagian teknis aplikasi, sehingga komunikasi antara pengguna servis (service consumer/service requester) dan penyedia servis (service provider) berjalan secara loosely couple, tidak terikat satu sama lain. Ketika aplikasi membutuhkan service, maka aplikasi tersebut akan mencarinya ke penyedia service. Setelah kebutuhan service terpenuhi, penyedia dan pengguna tidak saling berkomunikasi lagi.

Arsitektur SOA pada mobile platform (biasa disebut dengan mobile SOA) menggunakan arsitektur client-server dengan perangkat mobile client bertindak sebagai service consumer, dan server bertindak sebagai service provider. Untuk mendukung konsep ini, arsitektur yang cocok adalah arsitektur client-server-thin client/browser dan client-Server-thin native client. Namun arsitektur client-server-thin client/browser tidak mampu memberikan akses terhadap sumber daya perangkat mobile secara leluasa seperti calendar, camera, gps dan fitur yang lainnya yang dapat digunakan sebagai alat input/sensing pada aktivitas mobile learning. Padahal sebenarnya sumber daya perangkat mobile tersebut dapat memperkaya aktivitas belajar pada mobile learning, seperti yang telah dijelaskan pada Studi Literatur. Oleh karena itu

arsitektur yang lebih cocok adalah Client-Server dengan thin native client. Usulan arsitektur layering pada sistem mobile learning berbasis SOA dengan memanfaatkan LMS dapat dilihat pada Gambar 1.



Gambar 1. Usulan Arsitektur MLMS Berorientasi Servis

Dengan menerapkan arsitektur Client-Server dengan thin native client, aplikasi client pada perangkat mobile hanya berperan sebagai service requester/consumer saja, yaitu aplikasi yang hanya menyediakan user interface saja yang menjadi jembatan untuk mengakses servis pada server. Aplikasi client berupa user interface ini akan dipasang pada perangkat mobile. Aplikasi ini berupa aplikasi native yang sesuai dengan platform sistem operasi perangkat mobile. Untuk platform sistem operasi yang berlainan, maka dibutuhkan pengembangan aplikasi client yang sesuai dengan platform sistem operasi perangkat mobile tersebut. Namun karena client hanya berupa aplikasi yang hanya menerima input dan menampilkan output, maka aplikasi yang dibuat hanya aplikasi ringan, business logic dan application logic tidak berada pada aplikasi client, sehingga ketika ada perubahan pada business logic, aplikasi client tidak harus diubah. Selain itu arsitektur ini memungkinkan akses penuh terhadap fitur-fitur perangkat mobile, karena client dikembangkan dalam bahasa native sehingga fitur-fitur perangkat mobile dapat diakses secara langsung menggunakan API yang tersedia.

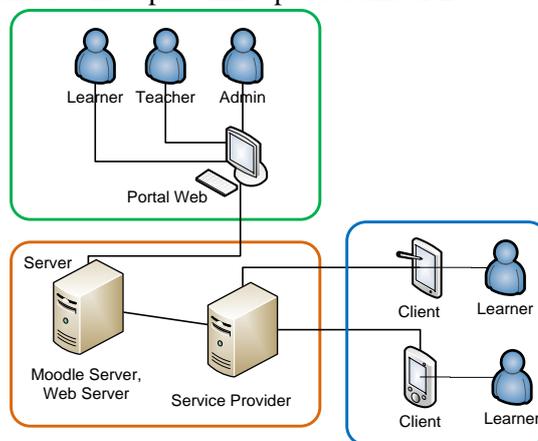
Tugas server adalah menyediakan service-service yang dibutuhkan oleh aplikasi client seperti distribusi materi, akses informasi, mendapatkan pengumuman, login dan logout, pendaftaran kursus dan lain-lain. Seluruh proses pada setiap service tersebut selanjutnya akan ditangani oleh LMS. Jika terdapat request dari

client, service provider akan menyalurkan permintaan tersebut untuk diproses oleh LMS. Pada layer SOA, posisi LMS, database dan web server termasuk pada application layer.

Service layer akan menampung service-service yang mengimplementasikan logik bisnis yang ada pada business process layer. Service yang ada pada layer ini bisa mencakup logik bisnis yang paling kecil (atomic service), atau dengan cakupan logik bisnis yang lebih besar, yaitu dengan melakukan komposisi beberapa service lain.

V. IMPLEMENTASI

Pembangunan MLMS dilakukan dalam rangka melakukan pengujian terhadap arsitektur MLMS berorientasi servis yang telah dirancang. Terdapat dua bagian yang akan dibangun pada MLMS, yaitu bagian *server* dan bagian *client*. Pada *server*, *application layer* terdiri dari *Learning Management System (LMS)*, *web server* dan *database server*. Sebuah *Learning Management System (LMS)* yang sudah ada akan digunakan pada sisi server sebagai back-end *Mobile Learning Management System (MLMS)* yang akan dibangun. LMS ini bertugas untuk menyediakan fungsionalitas utama dan melakukan pengelolaan kegiatan belajar. Fungsionalitas tersebut dapat diakses oleh instruktur/*teacher* dan *learner* secara langsung melalui portal web yang disediakan oleh LMS. Gambaran umum sistem dapat dilihat pada Gambar 2.



Gambar 2. Gambaran Umum Sistem

Sebuah *Learning Management System* (LMS) yang sudah ada akan digunakan pada sisi *server* sebagai *back-end Mobile Learning Management System* (MLMS). LMS yang digunakan adalah Moodle. Moodle dipilih karena merupakan salah satu LMS *open source* yang terkemuka dan digunakan dengan oleh banyak institusi pendidikan. Untuk integrasi dengan aplikasi pada *client*, Moodle telah menyediakan *web service* dan mendukung beberapa protokol komunikasi antar servis..

Webservis moodle menyediakan layanan, yaitu berupa data tertentu. Misalnya servis `core_enrol_get_users_courses`, servis ini menyediakan data berupa daftar kursus apa saja yang diikuti oleh pengguna tertentu. Servis yang disediakan oleh Moodle ini dapat dibungkus kembali oleh *entity service* yang dihasilkan pada tahap analisis berorientasi servis, sehingga *entity service* tidak perlu melakukan *query* langsung ke database terkait kebutuhan data tertentu. Hal ini sangat membantu pengembang servis karena untuk mengambil data daftar kursus tersebut, query harus mengakses minimal 3 tabel pada database moodle, tetapi dengan webservis moodle hanya perlu memanggil satu fungsi saja pada servis tertentu.

Fitur webservis Moodle ini merupakan ekstensi yang diberikan oleh LMS Moodle kepada klien atau sistem lain agar dapat memanfaatkan data yang ada pada basisdata Moodle tanpa harus menggunakan *query*. Webservis Moodle bersifat spesifik untuk suatu pekerjaan tertentu, sehingga harus dibungkus lagi oleh servis yang nanti dihasilkan pada tahap analisis berorientasi servis, sehingga servis Moodle memenuhi spesifikasi yang telah ditetapkan pada MSOAM. Moodle menyediakan 85 web service, pemetaan web service yang dibutuhkan untuk menjalankan fungsionalitas arsitektur dapat dilihat pada Tabel 1.

TABEL I
PEMETAAN FUNGSIONALITAS MLMS DAN FUNGSI WEB SERVICE MOODLE

No	Fitur	Deskripsi	Fungsi Web Service Moodle
----	-------	-----------	---------------------------

1	Otentikasi peserta kursus	Autentikasi user	-
2	Pendaftaran peserta kursus	Pendaftaran learner pada kursus tertentu	-
3	Penyelenggaraan kursus	Melihat kursus, mendownload file yang tersedia pada kursus	<code>core_course_get_courses()</code> <code>core_enrol_get_users_courses()</code> <code>core_course_get_content()</code> <code>webservice/pluginfile.php</code>
4	Informasi tugas	Melihat informasi tugas berdasarkan kursus tertentu	<code>core_calendar_get_calendar_events()</code>
5	Informasi nilai	Melihat nilai berdasarkan kursus yang diikuti/kursus tertentu	<code>core_grades_get_grades()</code>

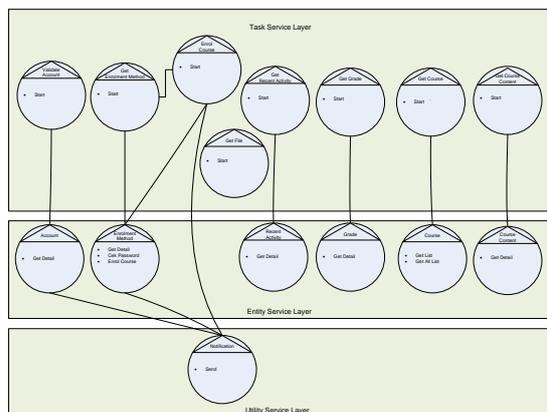
Berikut ini adalah *service inventory blueprint*, daftar seluruh kandidat servis yang dihasilkan pada tahap analisis berorientasi servis lengkap dengan *service capability* yang dimiliki oleh masing-masing servis

TABEL II
DAFTAR SERVICE CANDIDATE BESERTA CAPABILITY CANDIDATE-NYA

No	Service Candidate	Jenis Service	Service Capability Candidate
1	Account	Entity	Get Detail
2	Course	Entity	Get List, Get All List
3	Course Content	Entity	Get Detail
4	Enrolment Method	Entity	Get Detail

5	Notificat ion	Util ity	Send
6	Recent Activity	Enti ty	Get Detail
7	Grade	Enti ty	Get Detail
8	Get Course Content	Tas k	Start
9	Enrol Course	Tas k	Start
0	1 Validate Account	Tas k	Start
1	1 Get Course	Tas k	Start
2	1 Get Recent Activity	Tas k	Start
3	1 Get Grade	Tas k	Start
4	1 Get Enrolment Mthod	Tas k	Start
5	1 Get File	Tas k	Start

Untuk memenuhi kebutuhan bisnis dalam cakupan tertentu, servis harus mengkomposisi servis lain (Orkestrasi Task Service). Gambar 4 berikut ini merupakan logika solusi untuk memenuhi kebutuhan proses bisnis MLMS, yang dikelompokkan berdasarkan layer service modelnya, yaitu (1) *task service layer*, (2) *entity service layer*, (3) *utility service layer*.



Gambar 4. Orkestrasi Task Service

VI. EVALUASI

Pada bab ini dibahas mengenai evaluasi yang dilakukan terhadap arsitektur MLMS yang telah dirancang. Evaluasi dilakukan untuk menilai keberhasilan penerapan arsitektur sebagai usulan solusi dari permasalahan *multiplatform* dan keterbatasan perangkat *mobile*.

Arsitektur yang dirancang diterapkan pada pembangunan MLMS berorientasi servis. Pembangunan MLMS berorientasi servis meliputi dua kegiatan yaitu pembangunan *service provider* yang membungkus fungsionalitas LMS dan pembangunan aplikasi klien yang memanfaatkan fungsionalitas LMS via *service provider*.

Solusi *multiplatform* yang disediakan oleh usulan arsitektur terletak pada *service interface layer* dimana pada *layer* ini secara logika terdapat 15 servis yang telah dihasilkan berdasarkan fungsionalitas yang ditetapkan. Pembangunan servis dilakukan dengan menggunakan MSOAM. Servis-servis yang telah dihasilkan kemudian dipanggil oleh aplikasi klien ketika mengeksekusi proses bisnis yang ada dengan menggunakan protokol HTTP. Pada studi ini, hanya dibangun aplikasi klien berbasis Android. Untuk pengujian yang lebih lengkap sebaiknya servis diuji coba dengan beberapa *platform*. Aplikasi klien yang dibangun bersifat *native thin client* untuk mengatasi keterbatasan yang ada pada perangkat *mobile*.

Kemudian MLMS yang telah dibangun dimanfaatkan pada aplikasi *mobile learning*. MLMS dilengkapi dengan konten-konten, seperti akun pengguna, kursus, konten kursus, metode pendaftaran dan nilai. Servis-servis yang ada telah berhasil memberikan respon berupa data yang sesuai dengan permintaan, servis telah berhasil menyediakan layanan sesuai dengan spesifikasinya.

VII. PENUTUP

Pada pengerjaan thesis ini telah berhasil dirancang Arsitektur Mobile Learning Management System Berbasis Servis dengan

melakukan serangkaian analisis yaitu analisis terhadap metode solusi *multiplatform* yang sudah ada, analisis *Service Oriented Architecture* (SOA) sebagai solusi *multiplatform*, analisis kebutuhan dan keterbatasan perangkat *mobile*. Analisis pemanfaatan *Learning Management System* (LMS) pada sisi *server*.

Solusi *multiplatform* yang disediakan oleh usulan arsitektur terletak pada *service interface layer* yang memisahkan logika bisnis dan logika aplikasi. *Service interface layer* berfungsi sebagai perantara, sehingga logika bisnis tidak langsung diimplementasikan secara teknis, tetapi diimplementasikan sebagai servis-servis yang penggunaannya terikat pada platform dan bahasa pemrograman tertentu.

Dalam proses implementasi servis, digunakan MSOAM untuk analisis dan perancangan, dihasilkan 15 kandidat *service* yang terbagi menjadi 3 layer yaitu *utility service layer*, *entity service layer*, dan *task service layer*. Pembagian ini dimaksudkan untuk memberikan abstraksi terhadap sistem. Service consumer hanya dapat berinteraksi dengan *task service layer* saja, sehingga kompleksitas proses yang ada tidak perlu diketahui oleh *service consumer*.

Pada pengerjaan thesis ini, kandidat servis diimplementasikan sebagai servis REST, sehingga tidak menggunakan deskripsi service (wsdl) seperti halnya pada servis SOAP. Oleh karena itu *service consumer* harus berpedoman pada *service contract* untuk bisa menggunakan servis yang ada. Hal ini memiliki *trade off* yaitu overhead service bisa dihindari, namun pada saat melakukan komposisi, pengembang melakukan secara manual sesuai dengan *service contract* yang ada.

Untuk keperluan data tertentu, *entity service* yang dikembangkan tidak mengakses web service moodle, karena belum tersedia. Untuk pengembangan selanjutnya, untuk menghindari akses langsung ke beberapa tabel pada database, sebaiknya menggunakan external service sendiri yang menangani keperluan data tertentu.

DAFTAR PUSTAKA

- [1] Alamri, H.S. & Mustafa, B.A. (University M.P., Software Engineering Challenges in Multi Platform Mobile Application Development. Available at: http://www.academia.edu/9406502/Software_Engineering_Challenges_in_Multi_Platform_Mobile_Application_Development.
- [2] Anon, 2012. *MOBILE PLATFORMS AS INTERMEDIARIES : LIABILITY PROTECTIONS IN THE UNITED STATES , THE EUROPEAN UNION , AND CANADA*, Available at: <https://cdt.org/files/pdfs/Mobile-Platforms-As-Intermediaries.pdf>.
- [3] Erl, T., 2005. *Service-Oriented Architecture: Concepts, Technology, and Design*, Upper Saddle River, NJ: Prentice Hall PTR.
- [4] Erl, T., Pautasso, C. & Balasubramanian, R., 2012. *SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST*, Prentice Hall PTR.
- [5] Gartner, 2010. Gartner Highlights Key Predictions for IT Organizations and Users in 2010 and Beyond. Tersedia di: <http://www.gartner.com/it/page.jsp?id=1278413> [Accessed August 15, 2013].
- [6] Ismiraldi, Y., 2010. *Studi Penerapan SOA untuk Aplikasi Mobile*. Institut Teknologi Bandung.
- [7] Naismith, L. et al., 2004. Literature Review in Mobile Technologies and Learning. Tersedia di: https://ira.le.ac.uk/bitstream/2381/8132/4/%5b08%5dMobile_Review%5b1%5d.pdf.

- [8] Quinn, C.N.P., 2011. *Mobile Learning : Landscape and Trends*, Santa Rosa. Tersedia di: <https://commons.lbl.gov/download/attachments/77828943/mobile2011report-f2.pdf>.
- [9] Rodin, L., 2004. E-learning to m-learning: an investigation into the potential for content conversion. *Mobile Learning Anytime Everywhere from MLEARN*, p.171.
- [10] Walczak, K. et al., 2011. Adaptable Mobile User Interfaces for E-learning Repositories. , 01. Tersedia di: <http://web20erc.eu/sites/default/files/Adaptabl e-Mobile-Interfaces-CR-04-final.pdf>.