

Membangun Web API dengan menggunakan JSON sebagai format serialisasi data

Muhammad Ghazali
Program Studi Teknik Informatika
Fakultas Teknik
Universitas Widyatama
<muhammad.ghazali@widyatama.ac.id>

25 Januari 2013

Daftar Isi

Latar Belakang dan Masalah	1
Rumusan Masalah	2
Batasan Masalah	3
Hipotesa	3
Landasan Teori	3
Prosedur Penelitian	5
Jadwal Kerja	6
Usulan Pembimbing	6

Ringkasan

LayangLayang Mobile merupakan salah perusahaan yang bergerak di bidang *mobile application development*. Saat ini LayangLayang Mobile sedang mengembangkan sebuah produk bernama CampusLife. Produk yang dikembangkan tersebut merupakan aplikasi *mobile* yang bertujuan untuk membantu civitas kampus mengakses informasi relevan tentang kampus mereka.

Saat ini CampusLife membutuhkan Web API yang digunakan untuk mendistribusikan data - data yang tersimpan di *data store* untuk selanjutnya dikonsumsi oleh aplikasi *mobile*. Aplikasi *mobile* CampusLife akan didistribusikan ke pengguna yang menggunakan Android *smartphone*. *Smartphone* merupakan salah satu *mobile computing devices* yang memiliki masa hidup baterai dan ketersediaan *bandwidth* yang terbatas. Dengan kedua keterbatasan tersebut data yang dikirimkan harus diformat secara efektif untuk menghasilkan ukuran data yang optimal untuk dikirimkan dari Web API ke aplikasi *mobile*.

Kata kunci: Web API, JSON, Format Serialisasi Data

Latar Belakang dan Masalah

CampusLife adalah *mobile information directory application* yang dikembangkan oleh LayangLayang Mobile¹ untuk menyediakan informasi yang relevan kepada civitas kampus. Salah satu fitur utama yang akan dirilis dalam waktu dekat adalah menyediakan informasi *event-event* terbaru kepada civitas kampus.

Setiap informasi yang ditampilkan melalui aplikasi *mobile* CampusLife merupakan data yang sudah diolah dan diambil dari Web API² CampusLife. Saat ini LLM belum memiliki Web API tersebut. Berdasarkan kondisi tersebut, penulis bekerjasama dengan LLM untuk membangun Web API CampusLife. Web API yang akan dibangun bertujuan untuk membuka akses secara tidak langsung ke *data store*³ yang tersimpan di salah satu layanan *Database as a Service*⁴ yang digunakan oleh LLM di AppFog⁵. Seluruh data-data *event* yang tersimpan di *data store* akan diolah oleh Web API menjadi data dengan format yang dapat dikonsumsi dengan mudah oleh aplikasi *mobile* CampusLife. Proses pengolahan tersebut dinamakan serialisasi data⁶.

Sampai awal tahun 2013 ini, sudah ada lebih dari 10 format serialisasi data⁷:

1. ASN.1
2. Bencode
3. *Candle Markup*
4. *Comma-separated values (CSV)*
5. BSON
6. *D-Bus Message Protocol*
7. JSON
8. MessagePack
9. Netstrings
10. OGDLE

¹yang selanjutnya dalam proposal ini LayangLayang Mobile disebut sebagai LLM

²http://en.wikipedia.org/wiki/Web_API

³http://en.wikipedia.org/wiki/Data_store

⁴http://en.wikipedia.org/wiki/Cloud_database

⁵<http://www.appfog.com/>

⁶Lihat bagian Landasan teori: Serialiasi Data

⁷http://en.wikipedia.org/wiki/Comparison_of_data_serialization_formats

11. *Property list*
12. *Protocol Buffers*
13. S-expressions
14. Sereal
15. *Structured Data eXchange Formats*
16. Thrift
17. *eXternal Data Representation*
18. XML
19. XML-RPC
20. YAML

Di antara format serialisasi data yang sudah disebutkan di atas, XML⁸ dan JSON⁹ merupakan format serialisasi data yang paling terkenal saat ini[2]. Dalam penelitian ini penulis akan memilih format serialisasi data JSON untuk digunakan merepresentasikan setiap data-data *event* dalam format yang dapat dikonsumsi oleh aplikasi *mobile* CampusLife. Penulis memilih format serialisasi data JSON karena JSON lebih mudah dibaca ditulis dan dibaca oleh mesin (komputer) dan manusia. Selain itu JSON lebih mudah untuk diproses karena memiliki struktur yang lebih sederhana dibandingkan XML[6][3].

Dalam penelitian ini penulis akan mengkaji penerapan format serialisasi data JSON yang efektif untuk menghasilkan ukuran data yang optimal saat pengiriman data berlangsung dari Web API ke aplikasi *mobile* CampusLife. Hasil akhir yang diharapkan adalah format serialisasi data JSON mampu menghasilkan ukuran data yang optimal untuk dikonsumsi oleh aplikasi *mobile*.

Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka masalah yang akan diteliti dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana JSON dapat diterapkan secara efektif agar data yang dihasilkan dapat memiliki ukuran data yang optimal untuk dikonsumsi oleh aplikasi *mobile* CampusLife?

⁸<http://www.w3.org/XML/>

⁹<http://json.org/>

Batasan Masalah

Untuk memperjelas ruang lingkup pelaksanaan penelitian, penulis memiliki batasan masalah meliputi:

- Pembangunan Web API hanya akan sampai pada tahap purwa-rupa.
- Pembangunan Web API hanya akan meliputi API untuk mengambil data-data *event*.
- JSON hanya mampu merepresentasikan data dalam bentuk teks, oleh karena itu data yang akan digunakan hanya terbatas pada data yang berbasis teks.
- Skema data *event* akan disediakan oleh pihak LLM.
- Penulis akan melakukan demo untuk mengakses Web API melalui Android¹⁰ *smartphone* yang sudah terpasang aplikasi mobile CampusLife. Aktivitas demo akan difokuskan pada pengambilan data-data *event* dari purwa-rupa Web API yang dibuat oleh penulis.
- Tidak membahas mengenai keamanan perangkat lunak, data dan jaringan.
- Pengembangan perangkat lunak menggunakan sebagian praktek dari *Agile* dan tidak akan membahas *Agile* secara komprehensif.

Hipotesa

Diharapkan penerapan format serialisasi data JSON secara efektif dapat menghasilkan ukuran data yang optimal untuk dikonsumsi oleh aplikasi mobile CampusLife.

Landasan Teori

Web API

Web API (*Application Programming Interface*) merupakan seperangkat *HTTP request message*¹¹ yang telah ditetapkan beserta definisi dari struktur *HTTP response message*¹². Sementara Web API kadang - kadang dianggap sinonim untuk layanan

¹⁰<http://www.android.com/>

¹¹<http://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html>

¹²<http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html>

web (*web service*), Web 2.0 aplikasi biasanya sudah pindah dari layanan web berbasis SOAP ke arah layanan web berbasis REST[7][10].

Layanan Web Berbasis REST

Representational State Transfer (REST) mendefinisikan seperangkat prinsip arsitektur dimana penulis dapat merancang layanan Web yang berfokus pada sistem *resource*, termasuk bagaimana keadaan *resource* dipanggil dan ditransfer melalui HTTP oleh berbagai macam klien yang ditulis dalam bahasa (pemrograman) yang berbeda. Jika diukur dengan jumlah layanan Web yang menggunakannya, REST telah muncul dalam beberapa tahun terakhir sebagai model desain layanan Web yang dominan. Bahkan, REST memiliki dampak besar di Web yang telah sebagian besar menggantikan desain antarmuka berbasis SOAP-WSDL dan karena REST memiliki gaya yang jauh lebih sederhana untuk digunakan[8].

Serialisasi data

Dalam ilmu komputer, dalam konteks penyimpanan data dan transmisi, serialisasi adalah proses menerjemahkan struktur basis data atau kondisi objek ke dalam format yang dapat disimpan lebih lanjut dalam basis data (misalnya, dalam sebuah berkas atau *buffer* memori, atau ditransmisikan melalui koneksi jaringan) dan dilakukan proses deserialisasi komputer lain[4].

JSON

JavaScript Object Notation (JSON) adalah format *data interchange* ringan. Sangat mudah bagi manusia untuk membaca dan menulis JSON. Sangat mudah untuk mesin (komputer) untuk mengurai dan menghasilkan JSON. JSON berbasiskan pada subset dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi 3 - Desember 1999. JSON merupakan format teks yang benar - benar bahasa independen tetapi menggunakan konvensi yang akrab bagi *programmer* dari keluarga bahasa C, termasuk C, C + +, C#, Java, JavaScript, Perl, Python, dan banyak lainnya. Properti ini membuat JSON sebagai bahasa *data interchange* yang ideal[9].

Metodologi Pengembangan Perangkat Lunak Agile

Pengembangan perangkat lunak *Agile* adalah sekelompok metode pengembangan perangkat lunak berdasarkan metode pengembangan iteratif dan incremental¹³, di

¹³http://en.wikipedia.org/wiki/Iterative_and_incremental_development

mana persyaratan dan solusi berkembang melalui kolaborasi. Ini mempromosikan perencanaan adaptif, perkembangan dan pengiriman yang evolusioner, pendekatan time-boxed¹⁴ berulang, dan mendorong respon cepat dan fleksibel terhadap perubahan. Ini adalah kerangka kerja konseptual yang mempromosikan interaksi diramalkan sepanjang siklus pengembangan perangkat lunak[5].

Prosedur Penelitian

Tahapan Penelitian

1. Identifikasi masalah
2. Perumusan hipotesis
3. Pengujian hipotesis
4. Kesimpulan

Lingkungan Pengembangan

Penelitian akan dibantu dengan menggunakan beberapa kaskas dan teknologi berikut:

- NodeJS¹⁵, akan digunakan untuk membangun Web API.
- MongoDB¹⁶, akan digunakan sebagai basis data.
- Git¹⁷ dan Github¹⁸, akan digunakan untuk mengelola riwayat *source code* dan dokumen laporan serta dokumen teknis.
- AppFog¹⁹, akan digunakan sebagai infrastruktur untuk pengujian Web API.
- NetBeans IDE²⁰, akan digunakan untuk penyuntingan *source code*.
- LaTeX²¹, akan digunakan untuk penyuntingan dokumen laporan dan dokumen teknis.

¹⁴<http://en.wikipedia.org/wiki/Timeboxing>

¹⁵<http://nodejs.org/>

¹⁶<http://www.mongodb.org/>

¹⁷<http://git-scm.com/>

¹⁸<http://github.com/>

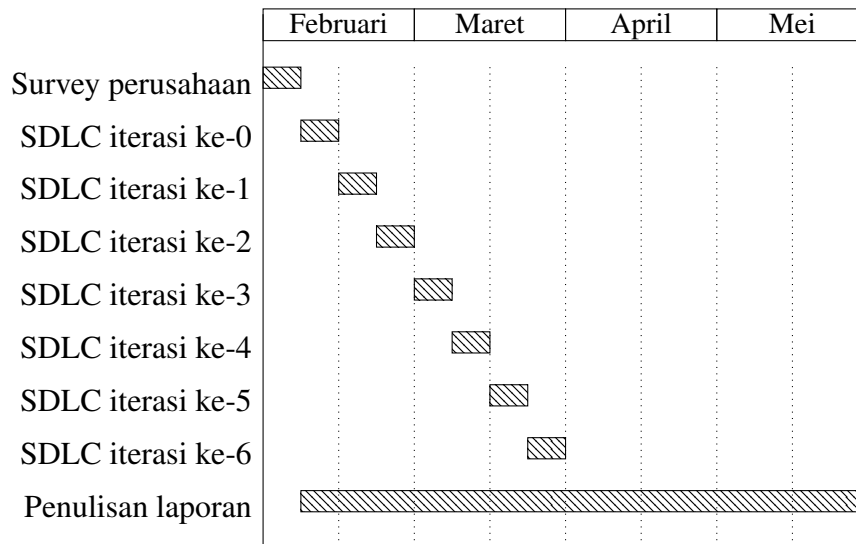
¹⁹<http://www.appfog.com/>

²⁰<http://netbeans.org/>

²¹<http://www.latex-project.org/>

- Laptop Asus Zenbook UX 32VD²² untuk pembangunan web API dan pembuatan dokumen laporan penelitian serta dokumen teknis.

Jadwal Kerja



SDLC (*Software Development Life Cycle*²³) terdiri dari tahap berikut:

1. Analisis Kebutuhan
2. Design
3. Implementasi
4. Pengujian

Penulis menggunakan metodologi pengembangan perangkat lunak *Agile*²⁴ dengan *time-box* selama 1 minggu di setiap iterasi.

Usulan Pembimbing

Penulis mengharapkan proses penelitian ini dibimbing oleh ibu Sriyani Violina, S.T., M.T..

²²http://www.asus.com/Notebooks_Ultrabooks/ASUS_ZENBOOK_UX32VD/

²³Analisis, Desain, Implementasi, Testing

²⁴http://en.wikipedia.org/wiki/Agile_software_development

Bibliografi

- [1] Deepak, G., and Dr. Pradeep B S. "Challenging Issues and Limitations of Mobile Computing." *International Journal of Computer Technology and Applications* 3.1 (2012): Academic Journals Database. Web. 8 Jan. 2013.
- [2] Audie Sumaray dan S. Kami Makki. "A comparison of data serialization formats for optimal efficiency on a mobile platform". *6th International Conference on Ubiquitous Information Management and Communication* (2012): Artikel No. 48. ACM Digital Library. Web. 24 Jan 2013.
- [3] *Debate: JSON vs. XML as a data interchange format* <http://www.infoq.com/news/2006/12/json-vs-xml-debate> diakses pada 20 Januari 2012.
- [4] *Serialization* <http://en.wikipedia.org/wiki/Serialization> diakses pada 24 Januari 2012.
- [5] *Agile software development* http://en.wikipedia.org/wiki/Agile_software_development diakses pada 24 Januari 2012.
- [6] *JSON: The Fat-Free Alternative to XML* <http://www.json.org/xml.html> diakses pada 20 Januari 2012.
- [7] *Web API* http://en.wikipedia.org/wiki/Web_API diakses pada 20 Januari 2012.
- [8] *RESTful Web services: The basics* <http://www.ibm.com/developerworks/webservices/library/ws-restful/> diakses pada 14 September 2012.
- [9] *Introducing JSON* <http://www.json.org/> diakses pada 20 Januari 2012.
- [10] *How REST replaced SOAP on the Web: What it means to you* <http://www.infoq.com/articles/rest-soap> diakses pada 14 September 2012.