

## IMPLEMENTASI METODE SINKRONISASI INFORMASI AKADEMIK UNIVERSITAS AMIKOM YOGYAKARTA ANTARA WEB SERVICE DAN APLIKASI MOBILE AMIKOM SOCIAL

Rizqi Sukma Kharisma<sup>1)</sup>, Arif Dwi Laksito<sup>2)</sup>

<sup>1),2)</sup> Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta  
Jl. Ring Road Utar, Condong Catur, Depok, Sleman, Yogyakarta 55281  
Email : sukma@amikom.ac.id<sup>1)</sup>, arif.laksito@amikom.ac.id<sup>2)</sup>

### Abstrak

Informasi Akademik Universitas Amikom Yogyakarta sangat dibutuhkan bagi segenap civitas akademik di dalamnya. Informasi ini bisa didapatkan melalui webamikom.ac.id atau dengan menggunakan aplikasi Amikom Social. Untuk aplikasi Amikom Social saat ini sudah diunduh lebih dari 5000 pengguna melalui Google Play. Dalam perkembangan penggunaan aplikasi Amikom Social saat ini sering mendapatkan masukan dan keluhan di mana proses sinkronisasi berjalan lama dan berat. Dari hasil pengamatan penulis, hal ini dikarenakan metode sinkronisasi aplikasi Amikom Social saat ini kurang efektif. Proses yang terjadi saat ini transfer data yang besar sehingga membutuhkan waktu yang lama dan ukuran data yang besar pada setiap kali melakukan sinkronisasi. Dari masalah tersebut maka penulis melakukan penelitian untuk mengembangkan metode sinkronisasi yang lebih efektif, sehingga mempersingkat waktu dan meminimalisir ukuran data yang disinkronkan.

**Kata kunci:** sinkronisasi, data, aplikasi

### 1. Pendahuluan

Saat ini aplikasi Amikom Social telah diunduh kurang lebih 5000 pengguna. Dalam perkembangannya banyak keluhan dari para pengguna Amikom Social khususnya pada metode sinkronisasi Informasi Akademik dari web service Universitas Amikom Yogyakarta dengan aplikasi Amikom Social. Hal ini dikarenakan metode yang digunakan saat ini membutuhkan waktu yang lama dalam sinkronisasi Informasi Akademik. Setiap kali aplikasi Amikom Social melakukan sinkronisasi, informasi yang ada di database Amikom Social akan dihapus secara keseluruhan dan selanjutnya diperbaharui dengan mengambil seluruh data yang ada di web service. Bisa jadi informasi yang diperbaharui masih sama dengan informasi yang telah dihapus pada database Amikom Social masih sama. Hal ini dinilai kurang efektif karena tidak semua informasi perlu diperbaharui.

Dari permasalahan tersebut di atas dibutuhkan sebuah metode yang lebih efektif untuk mempercepat proses sinkronisasi data antara web service Universitas Amikom Yogyakarta dengan aplikasi Amikom Social. Sehingga informasi yang diperbaharui hanya informasi-

informasi terbaru dari web service Universitas Amikom Yogyakarta bukan keseluruhan informasi. Dalam penelitian ini penulis melakukan penelitian dengan judul "Pengembangan Metode Sinkronisasi Informasi Akademik Universitas Amikom Yogyakarta Antara Web Service dan Aplikasi Mobile Amikom Social"

### 1.1. Tinjauan Pustaka

Penelitian sejenis pernah dilakukan oleh beberapa peneliti, untuk memastikan bahwa penelitian yang penulis akan ajukan adalah original maka penulis menyampaikan tinjauan pustaka sebagai berikut :

Penelitian yang dilakukan oleh Arif Dwi Laksito (2013) dengan judul Sinkronisasi Jadwal Perkuliahan pada Aplikasi Android menggunakan Teknologi XML-RPC (Studi Kasus di STMIK AMIKOM Yogyakarta). Dalam penelitian ini Arif Dwi Laksito melakukan penelitian tentang metode sinkronisasi menggunakan teknologi XML-RPC untuk jadwal perkuliahan mahasiswa dan dosen STMIK AMIKOM Yogyakarta. Data yang digunakan hanya meliputi jadwal perkuliahan yang disinkronisasikan dari Sistem Informasi Akademik dan Google Calendar melalui aplikasi mobile Amikom Social. [1]

Persamaan dari penelitian Arif Dwi Laksito dan penulis adalah melakukan proses sinkronisasi antara data di server dan di client. Persamaan selanjutnya Arif Dwi Laksito dan penulis adalah melakukan pengembangan aplikasi mobile Amikom Social. Akan tetapi penelitian Arif Dwi Laksito dan penulis memiliki perbedaan yaitu, data yang digunakan oleh Arif Dwi Laksito untuk sinkronisasi adalah data Jadwal Perkuliahan, sedangkan penulis menggunakan data Informasi Akademik. Selanjutnya penelitian yang dilakukan oleh Arif Dwi Laksito menggunakan teknologi XML-RPC, sedangkan penulis membuat metode baru dengan pembaharuan versi data.

Penelitian sejenis selanjutnya dilakukan oleh Rachel Kurniawati (2016) dengan judul Pengembangan Sistem Informasi Kependudukan Berbasis Mobile dan Restful Web service. Dalam penelitian ini Rachel Kurniawati melakukan akses data pada menggunakan aplikasi mobile ke web server menggunakan teknologi Restful Web service untuk kegiatan sensus Keuskupan Ketapang dalam pelaporan atau pembuatan statistik. [2]

Persamaan penelitian yang dilakukan oleh Rachel Kurniawati dan penulis adalah menggunakan teknologi Restful Web service dalam pertukaran data. Akan tetapi perbedaan penelitian yang dilakukan Rachel Kurniawati dan penulis adalah, Rachel Kurniawati proses sinkronisasi data dilakukan dari mobile android yang kemudian dikirim ke server, sedangkan penulis proses sinkronisasi berasal dari web server yang dikirim ke aplikasi mobile.

Penelitian sejenis selanjutnya dilakukan oleh Muhammad Nasir (2012) dengan judul Sinkronisasi Data User Antara Sistem Informasi Perpustakaan Dengan Sistem Informasi Akademik . Dalam penelitian ini Muhammad Nasir menggunakan teknologi XML-RPC untuk pertukaran data antara Istem Informasi Perpustakaan dengan Sistem Informasi Akademik di Universitas Bina Darma. [3]

Persamaan penelitian yang dilakukan oleh Muhammad Nasir dengan penulis adalah menggunakan sinkronisasi untuk memastikan kesamaan data pada client dan server. Akan tetapi penelitian yang dilakukan oleh Muhammad Nasir dan penulis memiliki perbedaan yaitu, data yang digunakan Muhammad Nasir menggunakan data user sedangkan penulis menggunakan data Informasi Akademik. Perbedaan selanjutnya, penelitian Muhammad Nasir melakukan sinkronisasi dua sistem informasi sedangkan penelitian penulis melakukan sinkronisasi data dari web service dan aplikasi mobile.

## 1.2. Landasan Teori

### 1.2.1. Web service

Menurut Kreger *web service* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet, dalam bentuk pesan XML [4]. Sedangkan menurut Manes (2003), *web service* diartikan sebagai sepotong atau sebagian informasi atau proses yang dapat diakses oleh siapa saja, kapan saja dengan menggunakan piranti apa saja, tidak terikat dengan sistem operasi atau bahasa pemrograman yang digunakan.

*Web service* merupakan *middleware* dalam suatu jaringan internet yang memungkinkan suatu sistem dapat saling berkomunikasi pada *platform* yang berbeda. *Web service* menggunakan antarmuka berupa XML yang dapat digunakan untuk membawa operasi-operasi dan data untuk komunikasi antar sistem. [5]

### 1.2.2. Representational State Transfer (REST)

Istilah REST yang merupakan singkatan dari *Representational State Transfer* pertama kali digunakan oleh Roy Thomas Fielding, salah seorang pelopor proyek *web server Apache*, pada disertasi doktronya yang berjudul *Architectural Styles and the Design of Network-based Software Architectures* di University of California

pada tahun 2000. Fielding (2000) mengidentifikasi empat prinsip (constraints) dalam REST, yaitu:

#### 1.2.2.1. Resource Identification

Semua resource (serta statenya) yang berhubungan dengan aplikasi diberikan *identifier* yang unik dan *identifier* tersebut harus bersifat global. Konsep resource disini bukan hanya hal statis yang langsung berhubungan dengan aplikasi namun juga termasuk informasi yang dibutuhkan seperti dokumen transaksi. REST resource adalah semua hal yang bisa diakses dan ditransfer melalui *web* antara *client* dan *server*. Dan karena protokol yang digunakan untuk berkomunikasi adalah HTTP, berbagai macam tipe file bisa ditransfer, *teks file*, *flash movie*, gambar dll. Sehingga dalam REST system representasi dari *resource* tergantung dari tipe yang diminta *client* (MIME type) yang didefinisikan didalam protokol request.

#### 1.2.2.2. Uniform Resource

Semua interaksi dibangun dengan antarmuka yang seragam. REST *web service* menampilkan semua *resource* dan interaksinya dengan *interface* yang seragam, tidak seperti RPC yang menampilkan fungsi yang ada melalui *method* yang bisa dipanggil secara *remote*. Dalam REST *web service* untuk uniform interface ini menggunakan *Uniform Resource Identifier* (URI). URI pada REST *web service* berupa *hyperlink* terhadap resource meskipun REST constraint tidak menyatakan URI harus berupa *hyperlink*, namun karena teknologi yang digunakan pada *web service* adalah web sehingga URI berupa *hyperlink*. Jika menggunakan teknologi lain, REST URI tentu akan berupa hal yang berbeda, namun tetap berupa address terhadap sebuah *resource*.

#### 1.2.2.3. Self-Describing Message

Untuk setiap interaksi dengan *resource* melalui antarmuka yang seragam, REST membutuhkan representasi dari *resource* yang menggambarkan semua aspek penting yang dimiliki oleh *resource* tersebut. Representasi dari *resource* sendiri adalah semua hal yang dikirim antara *client* dan *server*. Representasi merupakan state sementara dari data sebenarnya yang terletak di suatu tempat penyimpanan. Dengan kata lain representasi merupakan stream biner beserta *metadata* yang menjelaskan bagaimana stream tersebut digunakan baik untuk *client* maupun untuk *server*. Bisa terdapat banyak jenis client yang *me-request resource* yang ada, oleh karena itu representasi setiap *client* pun dapat berbeda. Representasinya dapat berupa gambar, *text file*, *stream XML* atau *stream JSON*, tapi kesemua representasi tersebut harus tersedia melalui URI yang sama. Untuk kasus *request* yang dilakukan oleh manusia (*human user*) biasanya representasi berupa laman *web* sehingga menjadi bentuk representasi yang dapat dibaca.

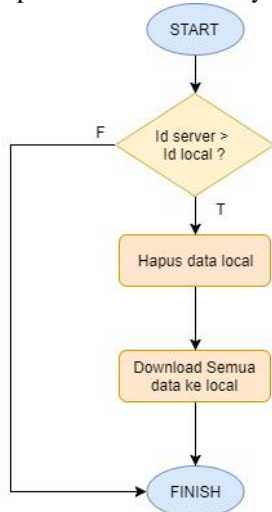
**1.2.2.4. Stateless Interaction**

Setiap interaksi antara *client* dan *server* harus memiliki state sendiri (atau dengan kata lain tidak dipengaruhi *session client*). Jadi server hanya akan memantau *resource state* bukan *client session*. [6]

**2. Pembahasan**

**2.1. Analisis Sistem**

Permasalahan dari penelitian ini adalah bagaimana men-sinkron-kan data di aplikasi yang ada di beberapa device pengguna dengan data di server web service. Perubahan data dapat terjadi di masing-masing pengguna seperti proses tambah, hapus, dan edit. Dengan semakin banyaknya data yang ada di server tidak memungkinkan jika sinkronisasi dilakukan dengan cara sebelumnya, karena proses sinkronisasi membutuhkan waktu yang lama. Gambar 1 menunjukkan diagram flowchart untuk sinkronisasi data pada sistem sebelumnya:

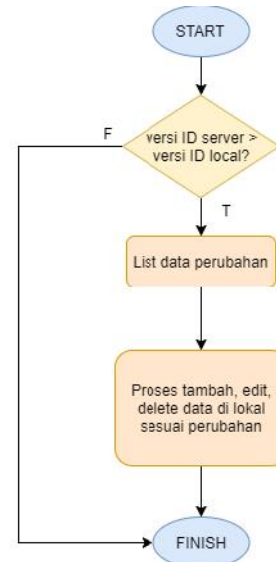


**Gambar 1.** Flowchart sinkronisasi sistem sebelumnya

**2.2. Desain Sistem**

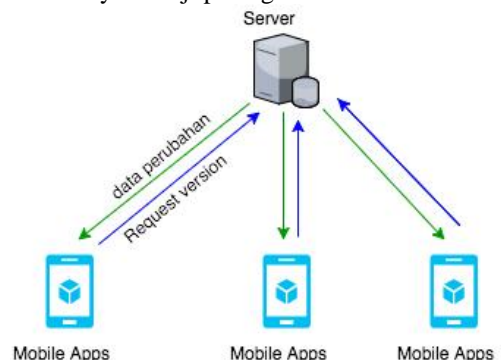
**2.2.1. Perancangan Proses**

Pada sistem yang akan dirancang ini akan dilakukan sinkronisasi data-data yang telah dilakukan perubahan oleh pengguna saja. Pada proses sinkronisasi tidak dilakukan penghapusan data di local, tetapi akan ditambahkan, dirubah atau dihapus dimana data yang telah terjadi perubahan. Metode ini dijalankan pada tiap-tiap aplikasi yang terinstal di device, jadi semua data yang ada di pengguna akan sama saat melakukan sinkronisasi. Gambar 2 menunjukkan diagram flowchart untuk sinkronisasi data pada sistem yang diusulkan:



**Gambar 2.** Flowchart sinkronisasi sistem yang diusulkan

Id dari tiap-tiap data yang berada di server dan local adalah sama. Id ini yang akan menjadi version untuk dibandingkan antara data local dan data di server. Setelah didapatkan hasil perbandingan, maka server akan mengirim sejumlah data beserta atribut perubahannya sesuai hasil perbandingan antara data di server dan di local. Masing-masing aplikasi di device pengguna bisa berbeda versinya-nya tergantung seberapa sering pengguna tersebut melakukan sinkronisasi data. Diagram komunikasinya tersaji pada gambar 3 dibawah ini.



**Gambar 3.** Diagram komunikasi server dan aplikasi mobile

**2.3. Tahapan Pengujian**

Pengujian perbandingan Amikom Sosial dengan Metode Sinkronisasi Lama (V.Old) dan Amikom Sosial dengan Metode Versioning (V.Ver). Pengujian ini dilakukan dengan kondisi :

1. Jaringan lokal
2. Tool pengujian menggunakan Postman untuk mengetahui waktu dan besaran data yang dibutuhkan saat sinkronisasi.
3. Data awal yang akan disinkronkan sebanyak 20 data
4. Perbandingan meliputi lama waktu yang dibutuhkan untuk sinkronisasi dan ukuran data yang disinkronisasi

Hasil pengujian ditunjukkan pada Tabel 1 di bawah ini :

**Tabel 1. Pengujian Perbandingan V.Old dan V.Ver**

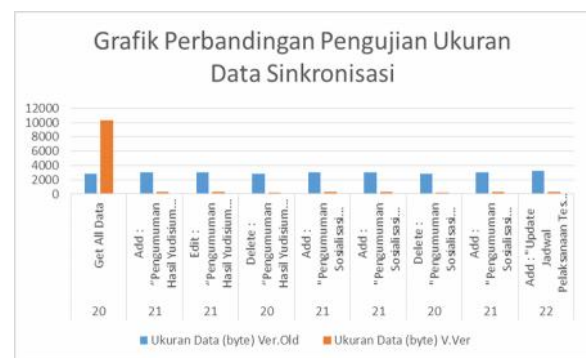
No	Banyak Data	Jenis Pembaharuan Data	Waktu (ms)		Ukuran Data (byte)	
			V.Old	V.Ver	Ver.Old	V.Ver
1	20	Get All Data	124	162	2862	10308
2	21	Add : "Pengumuman Hasil Yudisium Tahap 1 Bulan Oktober 2017"	192	146	3024	257
3	21	Edit : "Pengumuman Hasil Yudisium Tahap 1 Bulan Oktober 2017 Update"	137	84	3031	267
4	20	Delete : "Pengumuman Hasil Yudisium Tahap 1 Bulan Oktober 2017 Update"	186	98	2862	179
5	21	Add : "Pengumuman Sosialisasi Program Kunjungan Industri Mahasiswa D3 MI"	150	67	3037	270
6	21	Add : "Pengumuman Sosialisasi Program Kunjungan Industri Mahasiswa D3 MI Update"	116	90	3044	280
7	20	Delete : "Pengumuman Sosialisasi Program Kunjungan Industri Mahasiswa D3 MI Update"	123	84	2862	179
8	21	Add : "Pengumuman Sosialisasi Kunjungan Industri D3 MI Angkatan 2016/2017"	121	67	3039	272
9	22	Add : "Update Jadwal Pelaksanaan Tes Isest Amikom Language Center (ALC) Semester Ganjil Tahun Akademik 2017/2018"	108	79	3255	311

Berikut garafik hasil pengujian perbandingan Amikom Sosial dengan Metode Sinkronisasi Lama (V.Old) dan Amikom Sosial dengan Metode Versioning (V.Ver). Gambar 4 menunjukkan perbandingan pengujian

waktu sinkronisasi dan Gambar 5 menunjukkan perbandingan ukuran data saat sinkronisasi.



**Gambar 4. Grafik Perbandingan Pengujian Waktu Sinkronisasi**



**Gambar 5. Grafik Perbandingan Pengujian Ukuran Data Sinkronisasi**

### 3. Kesimpulan

Dari hasil penelitian ini dapat disimpulkan sebagai berikut :

1. Sinkronisasi Amikom Social dengan metode versioning memiliki kelemahan yaitu hanya saat pertamkali melakukan *Get All Data*, karena metode ini akan melakukan sinkronisasi seluruh data yang ada pada server termasuk setiap versi yang dilakukan baik *Add*, *Edit* dan *Delete*. Sehingga ukuran data yang di sinkronisasi sangat besar, dan membutuhkan waktu lebih lama. Sedangkan Sinkronisasi Amikom Social dengan metode lama, memiliki keunggulan saat pertama kali *Get All Data*, dikarenakan hanya menyinkronkan data yang ada pada *server*.
2. Sinkronisasi Amikom Social dengan metode versioning memiliki keunggulan saat terdapat update pada *server*, karena hanya mengambil data yang diupdate. Sehingga waktu dan ukuran data yang disinkronkan sangat kecil. Sedangkan Sinkronisasi Amikom Social dengan metode lama mengambil seluruh data yang ada pada *server*, sehingga membutuhkan waktu yang lama dan ukuran data yang besar.

## Daftar Pustaka

- [1] A. D. Laksito, "Sinkronisasi Jadwal Perkuliahan pada Aplikasi Android menggunakan Teknologi XML-RPC Studi Kasus Di Stmik Amikom Yogyakarta," in *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, Yogyakarta, 2013.
- [2] R. Kurniawati , "Pengembangan Sistem Informasi Kependudukan Berbasis Mobile dan Restful Web Service," in *Seminar Nasional Teknologi Informasi dan Komunikasi 2016 (SENTIKA 2016)*, Yogyakarta, 2016.
- [3] M. Nasir, "Sinkronisasi Data User Antara Sistem Informasi Perpustakaan Dengan Sistem Informasi Akademik," in *Seminar Nasional Informatika 2012 (semnasIF 2012)*, Yogyakarta, 2012.
- [4] H. Kreger, *Web-services Conceptual Architecture (WSCA 1.0)*, USA: IBM Software Group, 2001.
- [5] A. T. Manes, *Web Services: A Manager's Guide*, Bostron: Addison Wesley, 2003.
- [6] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," University of California, Irvine , 2000.

## Biodata Penulis

**Rizqi Sukma Kharisma**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2009. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2012. Saat ini menjadi Dosen di Universitas Amikom Yogyakarta.

**Arif Dwi Laksito**, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2006. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2013. Saat ini menjadi Dosen di Universitas Amikom Yogyakarta.

