

PENGEMBANGAN LAYANAN AKSES NILAI AKADEMIK BERBASIS *WEB SERVICES*

M. Miftakul Amin ^{*1}

¹ Jurusan Teknik Komputer Politeknik; Negeri Sriwijaya Palembang; Jalan Sriwijaya Negara, Palembang 30139; Telp. 0711 – 353414 Fax. 0711 – 355918; website : <http://polsri.ac.id>
e-mail : miftakul_a@polsri.ac.id

ABSTRAK

Penelitian ini melakukan rancang bangun sistem pengeksekusan nilai akademik berbasis dekstop, dengan memanfaatkan web services sebagai penyedia data. Aplikasi yang dikembangkan nantinya dapat digunakan oleh mahasiswa sebagai aplikasi student information center (SIC) sehingga memudahkan dalam memperoleh data nilai akademik. Aplikasi yang dikembangkan dibangun menggunakan library nusoap dengan bahasa pemrograman PHP dan basis data MySQL, sedangkan pada sisi client yang mengkonsumsi web services menggunakan bahasa pemrograman microsoft visual basic .Net. Dari penelitian yang telah dilakukan terlihat bahwa web services dapat menjembatani pertukaran data dan informasi di lingkungan yang heterogen, baik di level sistem operasi computer, sistem operasi jaringan, bahasa pemrograman dan basis data. Dengan adanya aplikasi ini, proses pengaksesan nilai akademik dapat dilakukan dengan lebih efektif.

KataKunci : nilai akademik, web services

1. PENDAHULUAN

Penggunaan Teknologi Informasi dan Komunikasi (TIK) telah merambah hampir di semua lini kehidupan. Keberadaan infrastruktur TIK telah memberikan manfaat yang cukup besar terutama dalam penyediaan informasi dan pertukaran data. Dewasa ini tuntutan penyediaan informasi lintas *platform* menjadi isu utama dan penting untuk dapat diwujudkan, dimana terdapat keberagaman infrastruktur TIK pada level sistem operasi, bahasa pemrograman dan basis data.

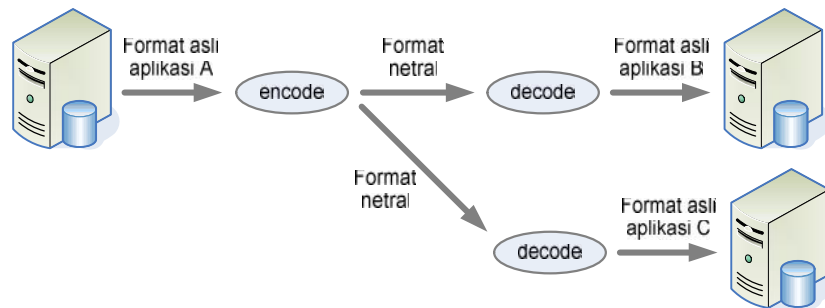
Di beberapa perguruan tinggi saat ini telah menyediakan sebuah informasi untuk layanan akademik dimana di dalamnya terdapat layanan akses nilai bagi berbagai pihak yang membutuhkan. Sistem informasi akademik yang dikembangkan biasanya dioperasikan pada lingkungan berbasis jaringan komputer baik *local area network* (LAN) maupun *wide area network* (WAN) sehingga dapat diakses dimana saja dan kapan saja. Lazimnya sistem informasi dikembangkan pada lingkungan yang homogen, sehingga sistem informasi yang dikembangkan tersebut juga terbatas hanya dapat diakses pada lingkungan yang homogen.

Saat ini terdapat salah satu teknologi yang dapat menjembatani aplikasi sehingga dapat dijalankan pada lingkungan yang heterogen yaitu teknologi *web services*, sehingga perbedaan sistem operasi komputer, sistem operasi jaringan komputer, bahasa pemrograman dan basis data bukanlah merupakan sebuah hambatan untuk dapat saling bertukar informasi. Teknologi *web services* menawarkan kemudahan dalam meningkatkan kolaborasi antar bahasa pemrograman yang berbeda, sehingga memungkinkan sebuah fungsi di dalam *web services* dapat diakses dan dieksekusi oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya. *Web services* diartikan sebagai sebuah antar muka (*interface*) yang menggambarkan

sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet dalam bentuk pesan XML [1].

Dengan menggunakan *web services* interoperabilitas antar sistem dapat diwujudkan. Interoperabilitas yang dalam *IEEE Standard Computer Dictionary* didefinisikan sebagai kemampuan 2 atau lebih sistem untuk saling tukar menukar data atau informasi dan saling dapat mempergunakan data atau informasi yang dipertukarkan tersebut [2]. Interoperabilitas bukanlah diartikan sebagai penentuan atau penyamaan penggunaan *platform* perangkat keras atau perangkat lunak, bukan juga berarti penentuan atau penyeragaman database, dan juga bukan penyeragaman bahasa pemrograman. Interoperabilitas harus dapat dicapai dalam keragaman penggunaan perangkat keras, perangkat lunak baik sistem operasi, basisdata, dan bahasa pemrograman.

Dalam pertukaran data antar aplikasi komputer yang berbeda, masalah utamanya terletak pada format data. Perbedaan format data menyebabkan data dari satu aplikasi tidak bisa begitu saja dikirimkan ke dan digunakan oleh aplikasi lainnya. Untuk itu diperlukan sebuah format “netral” yang disepakati oleh kedua aplikasi. Kata “netral” berarti tidak memihak ke format yang digunakan oleh salah satu aplikasi. Format netral ini kemudian digunakan sebagai format “antara” dalam pengiriman data, seperti ditunjukkan pada Gambar 1. Penggunaan format netral juga meningkatkan ekstensibilitas; aplikasi yang lain dapat pula memanfaatkannya, tanpa harus mengetahui format aslinya. Dewasa ini, format netral untuk pertukaran data banyak dijalankan oleh **XML** (*eXtensible Markup Language*). XML adalah sebuah format dokumen yang mampu menjelaskan struktur dan semantik (makna) dari data yang dikandung oleh dokumen tersebut [3].

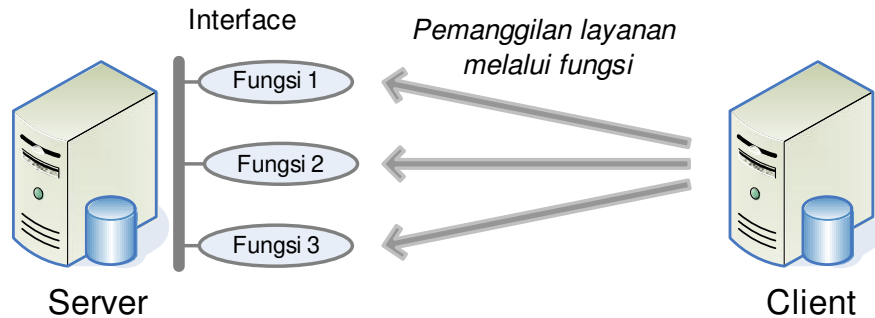


Gambar 1 Pengiriman data dengan format “netral” meningkatkan ekstensibilitas

Aplikasi-aplikasi yang berkomunikasi biasanya independen satu sama lain. Keduanya dibuat oleh pengembang yang berbeda, pada saat yang tidak sama pula. Padahal untuk bisa berkomunikasi, sebuah aplikasi harus tahu tentang cara menghubungi dan berbicara dengan aplikasi mitranya serta struktur data yang terlibat. Memberitahu semua ini berarti memberitahu detail internal dari aplikasi tersebut, dan ini yang sering menimbulkan kesulitan, karena tidak semua instansi bersedia membuka detail internal aplikasinya ke pihak lain, dengan alasan keamanan data dan sebagainya.

Untuk itu diperlukan cara lain yang lebih realistis, yang sekarang banyak dilakukan orang adalah menggunakan Service-Oriented Architecture (SOA). SOA adalah sebuah skema yang memungkinkan komunikasi antar aplikasi dilakukan secara loosely-coupled, artinya masing-masing pihak tidak perlu punya ketergantungan yang tinggi satu sama lain [4].

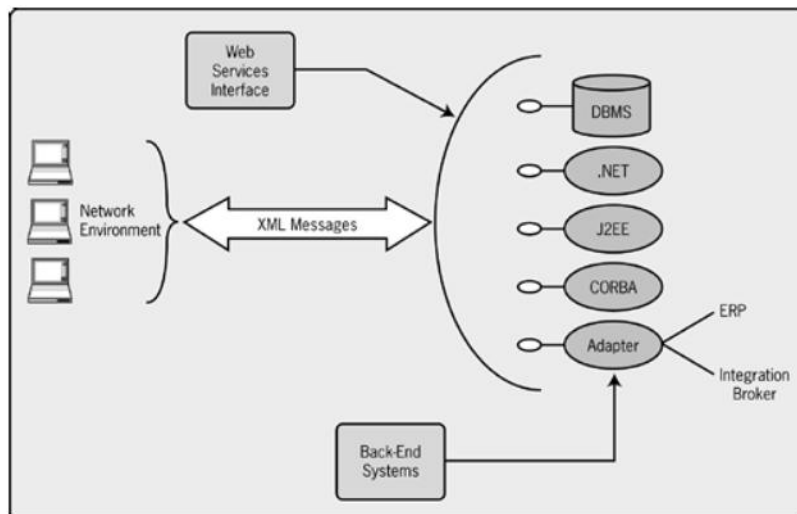
Dalam SOA, komunikasi didasarkan pada konsep layanan (*service*). Komunikasi berbasis layanan ini menggunakan prinsip *client-server* [10]. Ada aplikasi yang menyediakan layanan, dan aplikasi lain bisa meminta layanan tersebut. Permintaan terhadap layanan dilakukan dengan cara memanggil sebuah fungsi yang merepresentasikan layanan tersebut. Bila sebuah fungsi dipanggil, maka aplikasi penyedia layanan wajib memberikan layanannya ke aplikasi pemanggil. Gambar 2 menunjukkan mekanisme SOA yang bersifat *loosely-coupled*.



Gambar 2 Komunikasi *client-server* berbasis SOA

Dalam lingkungan aplikasi berbasis Web, SOA diimplementasikan dengan teknologi *web service* [5]. *Web service* menggunakan konsep seperti pada Gambar 3. *Web service* juga menyediakan abstraksi yang seragam bagi aplikasi-aplikasi *client* dan *server*. Hal ini memberikan fleksibilitas yang lebih tinggi lagi. Pada dasarnya aplikasi *server* dan *client* dapat dikembangkan dengan berbagai perangkat keras, sistem operasi, bahasa pemrograman, dan sistem basis data. Kemudian aplikasi ini dikemas sehingga bisa diakses dengan menggunakan protokol dan format standar web (*http* dan *XML*). Karena web sendiri sudah berkembang sebagai sebuah *platform* standar, maka *web service* menjadi sebuah pilihan yang menjanjikan.

Gambar 3 memperlihatkan peran *web service* yang menjembatani beragam teknologi *back-end* untuk dapat saling berkomunikasi. Sistem informasi yang menggunakan DBMS yang berbeda, dan teknologi pengembangan sistem yang berbeda seperti *.NET*, *J2EE*, *Corba* dan adapter yang lain dapat diwakili dengan *web services* sebagai interface tunggal untuk berkomunikasi dengan sistem lain melalui jaringan komputer.



Gambar 3 Hubungan *Web Services* dan *Back-end system*

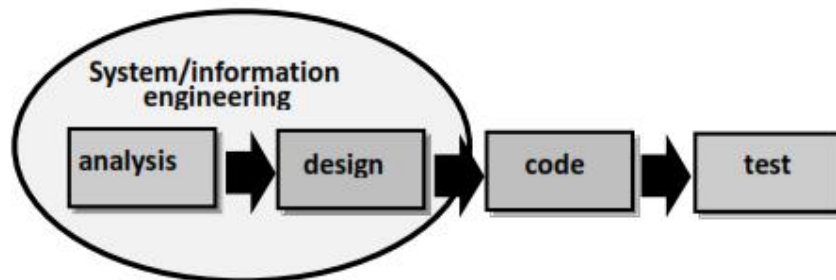
Terdapat beberapa kelebihan dalam penggunaan *web services* [6], diantaranya:

- **Lintas platform**, penggunaan *web services* memungkinkan komputer-komputer yang berbeda sistem operasi dapat saling bertukar data.
- **Language independent**, sebuah *web services* dapat diakses menggunakan bahasa pemrograman yang berbeda-beda.
- **Jembatan penghubung dengan basis data**, pada umumnya sebuah aplikasi memerlukan driver basis data supaya dapat melakukan koneksi ke sebuah basis data. *Web services* dapat dijadikan sebagai jembatan penghubung antara aplikasi dengan basis data.
- **Mempermudah pertukaran data**, pertukaran data dapat dilakukan dengan mudah antar organisasi dengan infrastruktur TIK yang berbeda.
- **Penggunaan kembali komponen aplikasi**, beberapa aplikasi yang berbeda dapat saja memerlukan sebuah fungsi yang sama. Sehingga sebuah fungsi yang telah dibuat oleh *web services* dapat langsung digunakan oleh beragam aplikasi.

Beberapa penelitian dengan topik aplikasi *web services* telah dilakukan, diantaranya oleh Laksito [7] yang mengimplementasikan *web services* untuk pengisian kartu rencana studi mahasiswa. Aplikasi pengisian KRS yang dibangun merupakan aplikasi yang sudah terintegrasi dengan *web services*. Penelitian lain juga telah dilakukan oleh Surendra [8] yang telah menggunakan *Web services* sebagai penyedia data pada aplikasi mobile yang membutuhkan data dinamis. Pengujian pada *web services* dilakukan dengan membuat file PHP secara manual menggunakan SOAP *Web services*.

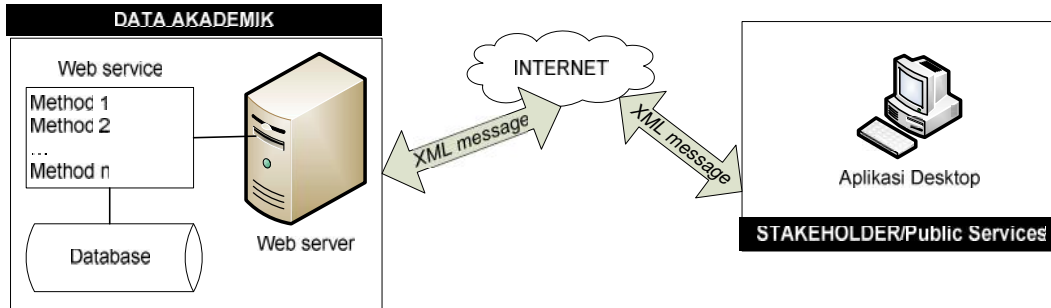
2. METODE PENELITIAN

Penelitian ini menggunakan linear model, sebagai metode pendekatan pengembangan perangkat lunak. Dalam rekayasa perangkat lunak linear model terdiri dari 4 (empat) tahapan utama, yaitu: analisa, desain, implementasi dan pengujian [9] sebagaimana dapat dilihat pada Gambar 4.



Gambar 4 Linear Model

Model desain dari arsitektur aplikasi berbasis *web services* yang dikembangkan dapat dilihat pada Gambar 5. Pada bagian *web services* terdapat sebuah *web server* yang melayani permintaan dari client. Dengan menggunakan bahasa pemrograman PHP dan library nusoap, maka pada bagian server terdapat sebuah *web services* yang terdiri dari beberapa *method* yang siap digunakan oleh aplikasi *client*. Format pertukaran data menggunakan XML melalui jaringan internet. Pada bagian client terdapat aplikasi desktop yang dibangun menggunakan microsoft visual basic .NET.



Gambar 5 Arsitektur Aplikasi

Tabel 1 memperlihatkan beberapa method *web services* yang digunakan untuk menyediakan data aplikasi client.

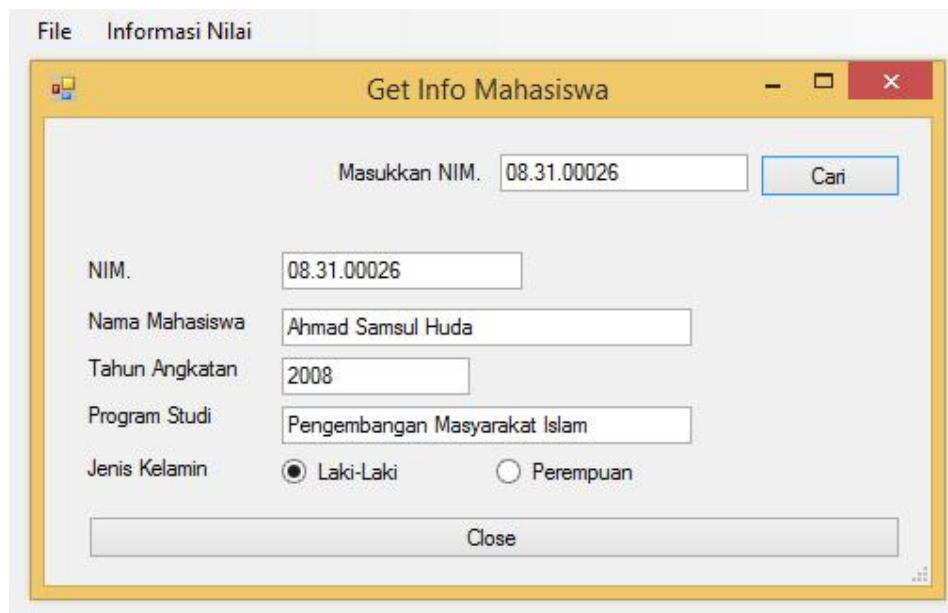
Tabel 1. Method Web Services

No.	Nama Method	Deskripsi
1.	getTA ()	Digunakan untuk mengembalikan data dengan tipe <i>array</i> berisi daftar tahun akademik yang telah terekam dalam basis data. Method ini tidak memiliki parameter input.
2.	getMhs(\$npm)	Digunakan untuk mengembalikan data detail mahasiswa. Format data yang dikembalikan berupa <i>array of struct</i> , yaitu sebuah struktur data yang berbeda-beda tipenya dalam bentuk <i>array</i> . Parameter yang digunakan adalah nomor induk/pokok mahasiswa sebagai parameter input.
3.	getKrs(\$ta,\$smt,\$npm)	Digunakan untuk mengambil data KRS/KHS setiap semester. Parameter yang diperlukan sebagai input berupa tahun akademik, semester dan npm. Data yang dikembalikan berupa <i>array of struct</i> .
4.	getTranskrip(\$npm)	Digunakan untuk mengembalikan data Rangkuman/transkrip nilai. Parameter input yang digunakan adalah npm, dan akan mengembalikan data <i>array of struct</i> .
5.	getIps(\$ta,\$smt,\$npm)	Digunakan untuk mengembalikan data indeks prestasi semester (IPS). Parameter input yang digunakan adalah tahun akademik, semester dan npm. Sedangkat method ini akan mengembalikan tipe data sederhana berupa decimal.
6.	getIpk(\$npm)	Digunakan untuk mengembalikan data indeks prestasi kumulatif (IPK). Parameter input yang digunakan adalah tahun akademik, semester dan npm. Sedangkat method ini akan mengembalikan tipe data sederhana berupa decimal.

3. HASIL DAN ANALISIS

Perangkat lunak yang digunakan dalam membangun aplikasi layanan akses nilai ini adalah *library nusoap* untuk membangun *web services*. Sedangkan aplikasi client yang digunakan untuk mengkonsumsi *web services* ini menggunakan Microsoft visual basic .Net. Aplikasi yang dibangun nantinya berfungsi sebagai *student information center* (SIC) yang khusus digunakan oleh mahasiswa untuk menyediakan layanan akses nilai.

Gambar 6 memperlihatkan form untuk mengakses detail informasi mahasiswa. Form ini mengimplementasikan method `getMhs($npm)` dengan melewati nomor induk mahasiswa sebagai parameter input. Jika data mahasiswa yang dicari terdapat dalam basis data, maka informasi mahasiswa tersebut akan ditampilkan dalam form.



Masukkan NIM.	08.31.00026	Cari
NIM.	08.31.00026	
Nama Mahasiswa	Ahmad Samsul Huda	
Tahun Angkatan	2008	
Program Studi	Pengembangan Masyarakat Islam	
Jenis Kelamin	<input checked="" type="radio"/> Laki-Laki <input type="radio"/> Perempuan	
Close		

Gambar 6 Form Get Info Mahasiswa

Selanjutnya di dalam sistem juga terdapat layanan untuk melihat daftar Kartu Rencana Studi (KRS) seperti dapat dilihat pada Gambar 7. Form ini mengimplementasikan method `web services getKrs($sta,$smt,$npm)` dengan 3 buah parameter input. Selanjutnya data-data terkait dengan KRS akan ditampilkan, berupa daftar matakuliah yang diambil pada semester tersebut, informasi kelas, dan dosen pengampu untuk setiap matakuliah serta jumlah SKS keseluruhan.

Kode MK	Nama Matakuliah	SKS	KELAS	Dosen Pengampu
STAI 0114	Teologi Aswaja	2	PMI 1	Drs. H. Abdul Kamm M.Pd.
STAI 0116	Pendidikan Multikultural	2	PMI 1	Muhamad Anco, M.Ag.
STAI 0115	Sejarah Pemikiran dan Peradaban Islam	2	PMI 1	A. Dimwati, M.Ac.
STAI 0104	Ulumul Hadits	2	PMI 1	Sri Nahain, MSI
STAI 0105	Tauhid/Ilmu Kalam	2	PMI 1	A. Tumud, SH, M.Aq.
STAI 0113	Pengantar Filsafat	2	PMI 1	Muhamad Anco, M.Ag.
STAI 0106	Pengantar Sosiologi	2	PMI 1	Arwani, M.Pd.
STAI 0102	Civic Education/Kewarganegaraan	2	PMI 1	Agus Leemana, SE, MM
STAI 0108	Bahasa Indonesia	2	PMI 1	Kholi Lur Richman, M.Aq.
STAI 0109	Bahasa Inggris I	2	PMI 1	A. Zairumeyid, SH, MA
STAI 0111	Bahasa Arab I	2	PMI 1	Ahmad Hamdan Hesaruddin, Lc. MA

Gambar 7. Form Get KRS

Method `getKrs($ta,$smt,$npm)` selain digunakan untuk menampilkan data KRS juga digunakan untuk menampilkan data Kartu Hasil Studi (KHS) seperti diperlihatkan pada Gambar 8. Perbedaan yang muncul adalah pada KHS terdapat informasi nilai dan jumlah indeks prestasi semester (IPS) dan detail data nilai untuk setiap matakuliah.

KODE MK	NAMA MATAKULIAH	SKS	KELAS	NILAI	DOEOT
STAI 0114	Teologi Aswaja	2	PMI 1	D+	3.25
STAI 0116	Pendidikan Multikultural	2	PMI 1	A-	3.5
STAI 0115	Sejarah Pemikiran dan Peradaban Islam	2	PMI 1	A	4
STAI 0104	Ulumul Hadits	2	PMI 1	B+	3.25
STAI 0105	Tauhid/Ilmu Kalam	2	PMI 1	A+	0
STAI 0113	Pengantar Filsafat	2	PMI 1	B	3
STAI 0106	Pengantar Sosiologi	2	PMI 1	B+	3.25
STAI 0102	Civic Education/Kewarganegaraan	2	PMI 1	B	3
STAI 0108	Bahasa Indonesia	2	PMI 1	B	3
STAI 0109	Bahasa Inggris I	2	PMI 1	A	3.5
STAI 0111	Bahasa Arab I	2	PMI 1	D	3

Gambar 8. Form get KHS

Method `getIps($ta,$smt,$npm)` digunakan untuk menampilkan data indeks prestasi semester (IPS). Method ini memerlukan 3 buah parameter input yaitu tahun akademik, semester dan nomor induk mahasiswa. Gambar 9 memperlihatkan tampilan dari hasil pemanggilan *web services*.

The screenshot shows a window titled "Get IP Semester". It contains the following fields and controls:

- Tahun Akademik: 2008/2009 (dropdown)
- Semester: 1 | Ganjil (dropdown)
- Masukkan NIM: 08.31.00026 (text input)
- get IPS (button)
- Nama Mahasiswa: Ahmad Samsul Huda (text input)
- Tahun Angkatan: 2008 (text input)
- Program Studi: Pengembangan Masyarakat Islam (text input)
- Jenis Kelamin: Laki-Laki Perempuan (radio buttons)
- 2.98 (result display)
- Close (button)

Gambar 9 Form getIPS

Gambar 10 memperlihatkan pemanggilan method `getIPK($npm)`. Method tersebut hanya memerlukan satu buah parameter input berupa nomor induk mahasiswa. Selanjutnya dengan mengkombinasikan dengan method `getMhs($npm)` data detail mahasiswa bersangkutan juga ditampilkan dalam form.

The screenshot shows a window titled "Get IP Kumulatif". It contains the following fields and controls:

- Masukkan NIM: 08.31.00026 (text input)
- get IPK (button)
- Nama Mahasiswa: Ahmad Samsul Huda (text input)
- Tahun Angkatan: 2008 (text input)
- Program Studi: Pengembangan Masyarakat Islam (text input)
- Jenis Kelamin: Laki-Laki Perempuan (radio buttons)
- 3.21 (result display)
- Close (button)

Gambar 10 Form Get IPK

Pada Gambar 11 memperlihatkan pemanggilan dari method `getTranskrip($npm)`. Method ini membutuhkan 1 buah parameter input berupa nomor induk mahasiswa, dan mengembalikan data berupa *array of struct* serta detail matakuliah beserta nilainya, serta IPK yang diperoleh mahasiswa tersebut.

The screenshot shows a web application window titled "RANGKUMAN NILAI" with a yellow header. Below the header, there is a form for student information. The form includes fields for "Masukkan NIM." (08.31.00026), "Nama Mahasiswa" (Ahmad Samsul Huda), "Dosen Wali/PA" (Arwani, M.Pd.), "Program Studi" (Pengembangan Masyarakat Islam), and "Jenjang" (S1). A button labeled "get Transkrip Akademik" is visible. Below the form is a table with the following data:

NO.	KODE MK	NAMA MATAKULIAH	SKS	NILAI
1	0923109	Perencanaan dan Kebijakan Pembangunan Sosial	3	B+
2	0923129	Fiqh	2	A-
3	STAI 0101	Akhlak Tasawuf	2	A-
4	STAI 0106	Pengantar Sosiologi	2	B+
5	STAI 0111	Bahasa Arab I	2	B
6	STAI 0116	Pendidikan Multikultural	2	A-
7	0923110	Psikologi Sosial	2	B+
8	0923130	Fiqh Sosial	2	B
9	STAI 0102	Civic Education/Kewarganegaraan	2	B
10	STAI 0107	Filsafat Ilmu	2	B
11	STAI 0112	Bahasa Arab II	2	A
12	STAI 0117	Dasar-dasar Manajemen	2	C+
13	0923103	Pengantar Antropologi	2	A-
14	0923111	Gender dan Pembangunan	2	B

Below the table, there are two summary fields: "Indeks Prestasi Kumulatif" with the value 3.21 and "Jumlah SKS" with the value 67. A "Close" button is located at the bottom of the window.

Gambar 11 Form Rangkuman Nilai

4. KESIMPULAN

Dari penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. *Web services* dapat digunakan untuk menjembatani pertukaran data dan informasi pada lingkungan yang heterogen.
2. Aplikasi layanan pengaksesan nilai yang telah dikembangkan telah menyajikan informasi yang sesuai dengan perancangan yang dapat mengembalikan nilai dengan tipe sederhana, *array*, *struct* dan *array of struct* untuk menyediakan informasi nilai akademik.

5. SARAN

Aplikasi yang telah dikembangkan ini dapat diintegrasikan ke dalam sistem informasi akademik yang ada di Perguruan Tinggi, sehingga memudahkan bagi pihak-pihak yang memerlukan informasi nilai akademik terutama bagi mahasiswa.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada redaksi jurnal Jupiter yang telah memberikan kesempatan kepada penulis, sehingga naskah jurnal ini dapat diterbitkan.

DAFTAR RUJUKAN

- [1] Kreger, H. 2001. *Web-services Conceptual Architecture (WSCA 1.0)*. IBM Software Group. USA
- [2] Depkopinfo. 2008. *Kerangka Acuan dan Pedoman Interoperabilitas Sistem Informasi Instansi Pemerintahan*. Jakarta: Direktorat Sistem Informasi, Perangkat Lunak dan Konten Direktorat Jenderal Aplikasi Telematika Departemen Komunikasi dan Informatika
- [3] Bray, T., Paoli, J., Sperberg-McQueen, C.M., Maler, E., dan Yergeau, F. (editor). 2008. *Extensible Markup Language 1.0 (Fourth Edition)*. Dokumen web <http://www.w3.org/TR/2006/REC-xml-20060816/> diakses pada tanggal 18 Mei 2008.
- [4] He, H. *What is Service-Oriented Architecture ?* Dokumen web <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html> diakses pada tanggal 18 Mei 2008.
- [5] Cerami, E. 2002. *Web Service Essentials*. O'Reilly.
- [6] Lucky. 2008. *XML Web Services, Aplikasi Desktop, Internet dan Handphone*. Penerbit Jasakom
- [7] Laksito, Dwi, Arif. 2010. Implementasi Web Service Pada Aplikasi Pengisian Kartu Rencana Studi Mahasiswa. *Jurnal Dasi Vol. 11 No. 1 Maret 2010*. Yogyakarta: STMIK Amikom Yogyakarta
- [8] Surendra, Martinus, Raditia, Sigit. 2014. Implementasi PHP Web Services Sebagai Penyedia Data Aplikasi Mobile. *Jurnal Ultimatics, vol. vi, no.2 Bulan Desember Tahun 2014*.
- [9] Pressman, R.S.. 2002. *Software Engineering, A Practitioner's Approach, Fifth Edition*. Inc. New York : McGraw-Hill Companies.
- [10] Nugroho, Lukito, Edi. 2017. *Interoperabilitas Data dalam Implementasi E-Government*. <https://www.slideshare.net/agungbudip/interoperabilitas-egov> diakses pada tanggal 20 Maret 2017 Pukul 13:18 WIB.