

# Implementasi Web service NuSOAP pada Pengembangan Akses SIM Akademik

Falahah\*, Rizky Dwi Saputra

Program Studi Teknik Informatika, Fakultas Teknik

Universitas Widyatama

Jl. Cikutra no.204A, Bandung

falahah@widyatama.ac.id\*, saputra.dwi.rizky@gmail.com

**Abstrak**— *Web services* merupakan salah satu mekanisme layanan di lingkungan aplikasi web yang memudahkan untuk menyediakan akses tambahan terhadap aplikasi yang sudah berjalan, tanpa harus memodifikasi aplikasi secara keseluruhan. Melalui *web service*, fitur akses tambahan dan kebutuhan integrasi data dapat dengan mudah dilayani sesuai dengan kebutuhan organisasi. Pada penelitian ini, *web service* diterapkan sebagai satu pendekatan pengembangan akses sistem informasi (SIM) Akademik untuk pemenuhan kebutuhan akses melalui perangkat *smart phone* berbasis android. Inisiatif ini dibangun atas dasar keinginan akses data dalam ruang lingkup yang lebih terbatas (hanya data tertentu), cepat dan mudah. NuSOAP dipilih karena *web service* ini dapat diterapkan pada berbagai lingkungan versi PHP tanpa memerlukan pustaka tambahan. Pengembangan akses melalui android dibatasi pada akses mahasiswa terhadap informasi-informasi penting seperti jadwal kuliah, kehadiran dan nilai.

**Kata kunci**—*Web service; NuSOAP; PHP; SIM akademik; akses; Android.*

## I. PENDAHULUAN

Pengelolaan data melalui sistem informasi di suatu organisasi merupakan suatu kebutuhan mutlak di organisasi modern saat ini. Namun, inisiatif pengembangan dan implementasi sistem informasi pada masa-masa awal biasanya tidak terencana dengan baik dan cenderung hanya berusaha menyelesaikan persoalan di satu proses bisnis atau unit tertentu. Hal ini berakibat pada kurang terintegrasinya data dari satu sistem dengan sistem lainnya dikarenakan data disimpan pada lokasi dan format yang berbeda. Salah satu cara yang dapat diterapkan untuk menyelesaikan masalah integrasi ini, tanpa harus mengganggu sistem yang sudah berjalan, yaitu dengan menggunakan *web service*.

Penggunaan *web service* memungkinkan pengelola sistem untuk membuat modul-modul integrator kecil yang dapat mencari layanan, mengirimkan pesan, dan menerima/mengirimkan data lintas aplikasi tanpa harus memodifikasi ataupun mengganggu sistem yang sudah berjalan. Penggunaan *web service* dapat juga dipandang sebagai peluang perluasan akses dari aplikasi yang sudah berjalan, tanpa harus melakukan pengembangan aplikasi skala besar atau memodifikasi aplikasi yang sudah berjalan. Melalui *web service*, kita dapat menambahkan titik akses baru, memperluas penggunaan informasi dan menghubungkan satu aplikasi dengan aplikasi lain tanpa mengganggu aplikasi itu sendiri.

Peluang ini kemudian dicoba diterapkan untuk pengembangan sistem informasi akademik di Universitas Widyatama agar dapat diakses oleh mahasiswa melalui perangkat *smart phone* dan penambahan beberapa fitur yang belum tersedia, tanpa harus melakukan modifikasi pada sistem itu sendiri. Inisiatif pengembangan aplikasi ini didasari oleh kondisi bahwa tampilan sistem informasi akademik yang tersedia saat ini akan terpotong jika ditayangkan melalui layar *smart phone*, dan waktu akses data yang *relative* lama. Di lain pihak, kebutuhan mahasiswa untuk mengakses sistem tersebut sangat tinggi.

Hasil penelitian ini adalah tersedianya aplikasi berbasis android yang dapat dijalankan oleh mahasiswa pada *smart phone*, yang menyediakan akses terhadap data-data tertentu di portal akademik mahasiswa. Melalui aplikasi ini, mahasiswa dapat melihat langsung data akademik dan perkuliahan seperti jadwal kuliah, perubahan jadwal, nilai dan absensi. Fitur yang ditambahkan adalah simulasi perhitungan indeks prestasi (IP). Aplikasi ini akan berperan sebagai *Add on* atau akses *alternative* dari sistem informasi akademik yang sudah ada.

## II. KAJIAN PUSTAKA

### A. *Web services* dan SOAP

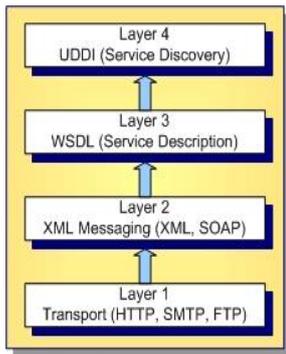
*Web services* didefinisikan sebagai aplikasi *software* yang diidentifikasi dengan sebuah URI, yang memiliki *interface* dan *binding* yang dapat didefinisikan, dideskripsikan dan ditemukan sebagai sebuah artefak XML. *Web services* mendukung langsung interaksi dengan *software agent* lain melalui pesan berbasis XML yang dipertukarkan melalui *protocol internet*[1].

*Web services* bersifat *independent* terhadap bahasa pemrograman, sistem operasi, dan *hardware* sehingga dapat berjalan dengan mulus antara *service consumer* dan *provider*. Teknologi *web service* didasari oleh beberapa *open-technology* seperti:

- *Extensible Markup Language (XML)*
- *Simple object access protocol (SOAP)*
- *Universal Description, Discovery and Integration (UDDI)*
- *Web services Description Language (WSDL)*

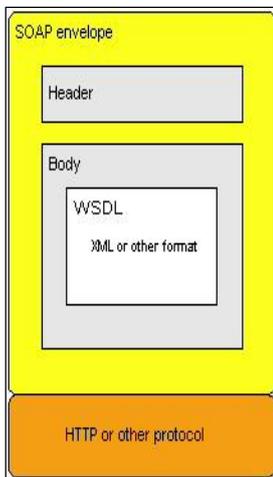
Peranan masing-masing komponen pada *web service* dapat dinyatakan sebagai diagram lapisan tumpukan *protocol web services* seperti pada Gambar 1. Pada Gambar 1 terlihat bahwa

layer 1 (transport), merupakan layer yang berperan sebagai interfaces antara services (layer 3 dan layer 4), dengan pihak yang menggunakan services[2].



Gambar 1. Lapisan protokol web services[2]

SOAP merupakan layanan web service yang berbasis pada RPC (Remote Procedure Call) yang terlihat seperti Gambar 2. SOAP dapat digunakan untuk memanggil method atau fungsi yang berada di komputer lain melalui internet. Agar client bisa mengetahui method tersedia, port, format data input output, atau keterangan lain maka dideklarasikan oleh standar WSDL[3].



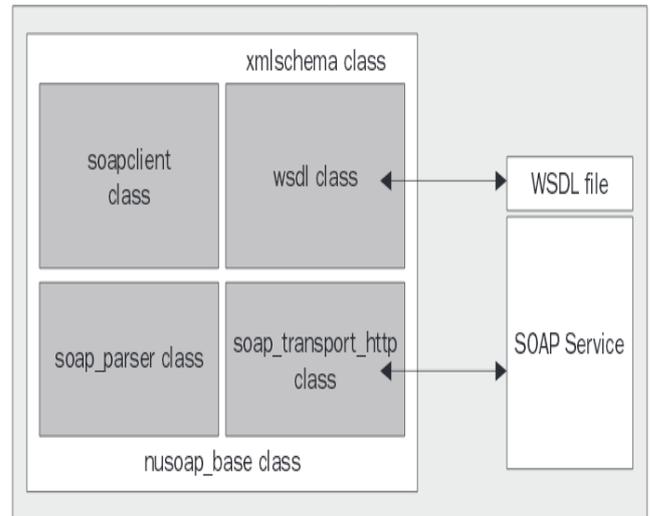
Gambar 2. Struktur komponen SOAP [4]

**B. NuSOAP**

NuSOAP adalah bentuk lain dari SOAPx4, yang disediakan dari NuSphere dan Dietrich Ayala, yang terdiri dari sekumpulan class PHP. NuSOAP dapat digunakan langsung tanpa memerlukan ekstensi PHP apapun. NuSOAP dapat digunakan untuk mengakses web service yang dibuat berdasarkan SOAP 1.1, WSDL 1.1., dan http 1.0/1.1. [5] Keuntungan penggunaan NuSOAP dibandingkan dengan soapServer dan soapClient yang sudah disediakan oleh PHP5 adalah :

- a. Mudah diimplementasikan, tidak perlu membuat dokumen WSDL sendiri.
- b. Kompatibel, tidak perlu ekstensi khusus dan dapat digunakan pada PHP versi lama misalnya versi 4.0.

NuSOAP berupa toolkit yang bersifat component-based, yang terdiri dari sebuah class dasar yang menyediakan method dan envelope. Interaksi web service diperoleh melalui class client yang disebut sebagai soapclient. Operasi SOAP dieksekusi melalui method call(), class soapclient akan memanggil URL dari file WSDL sebagai argument. Skema cara kerja NuSOAP dapat dilihat pada Gambar 3.



Gambar 3. Skema Arsitektur NuSOAP[5]

Cara implementasi NuSOAP[6]:

- a. Unduh NuSOAP dari situs: <http://sourceforge.net/projects/nusoap/>
- b. Siapkan SOAP Server, misalnya seperti contoh berikut, yaitu sebuah SOAP server yang akan menerima masukan satu parameter dan mengeluarkan hasil berupa string.

```
require_once('lib/nusoap.php');
$server = new nusoap_server;
$server->configureWSDL('server', 'urn:server');
$server->wsdl->schemaTargetNamespace = 'urn:server';
$server->register('hello',
array('username' => 'xsd:string'),
array('return' => 'xsd:string'), //output
'urn:server', //namespace
'urn:server#helloServer', //soapaction
'rpc', // style
'encoded', // use
'Just say hello'); //description
function hello($username) {
return 'Howdy, '.$username.'!';}
$HTTP_RAW_POST_DATA = isset($HTTP_RAW_POST_DATA) ?
$HTTP_RAW_POST_DATA : '';
$server->service($HTTP_RAW_POST_DATA);
```

- c. Buat SOAP Client untuk memanggil web service tersebut:

```
require_once('lib/nusoap.php');
$wsdl = "http://localhost/nusoap_test/ws.php?wsdl";
$client = new nusoap_client($wsdl, 'wsdl');
$error = $client->getError();
if ($error) {echo 'Error' . $error;
exit();}
//calling our first simple entry point
$result1=$client->call('hello',
array('username'=>'Reni'));
print_r($result1);
```

### C. Metodologi Implementasi Web service

Meskipun *web service* saat ini umumnya berupa pelengkap bagi interkoneksi berbagai aplikasi di lingkungan organisasi, namun proses pengembangan dan implementasinya secara umum tidak jauh berbeda dengan proses pengembangan dan implementasi perangkat lunak pada umumnya.

*Oasis-open.org* mendefinisikan metodologi implementasi *web service* dalam 6 fase utama, yaitu *requirement, analysis, design, coding, test* dan *deployment*[7]. Setiap fase memuat sekumpulan aktivitas dan tugas serta artifak-artifak tertentu. Tabel 1 menampilkan daftar aktivitas di setiap fase implementasi *web service*.

TABEL 1. FASE DAN AKTIVITAS PADA IMPLEMENTASI WEB SERVICE[7]

Fase	Aktivitas
Requirement	Determine needs Elicit requirements Manage requirements Model usage scenario Prepare test case for UAT and sistem test
Analysis	Select technology platform as implementation framework Define candidate architecture Decide granularity Identify reusable WS Identify service interface Prepare test case
Design	Transform signature of reusable WS Refine service interface of new WS Design WS Refine test cases
Coding	Construct WS code Construct WS client code Unit Test
Testing	Functionality test Integration test Sistem test User acceptance test
Deployment	Prepare deployment environment Deploy WS Test deployment Create end user support material Publish WS

### III. HASIL DAN DISKUSI

Pada penelitian ini, fase-fase pada metodologi implementasi *web service* tidak dapat diterapkan sepenuhnya, dikarenakan keterbatasan waktu dan sumber daya. Bagian yang belum dicoba diterapkan antara lain adalah semua aktivitas yang terkait pengujian, kecuali pengujian fungsionalitas.

#### A. Fase Requirement

Pada fase *requirement*, aktivitas diprioritaskan pada pendefinisian kebutuhan dan skenario penggunaan *web service*. Permasalahan yang ada yaitu sudah tersedia sistem informasi akademik namun banyak mahasiswa mengalami kesulitan membaca informasi pada sistem tersebut jika dibuka melalui *smartphone*, dikarenakan tata letak informasi banyak yang terpotong dan waktu tayang informasi relatif lama. Untuk mengatasinya, akan dibuat sebuah aplikasi yang memenuhi kebutuhan tersebut:

- Dapat dibuka melalui *smartphone*
- Dapat menayangkan informasi tanpa terpotong-potong
- Waktu panggil informasi *relative* cepat
- Dapat menayangkan informasi yang paling sering dibutuhkan mahasiswa yaitu jadwal kuliah, perubahan jadwal kuliah atau kuliah pengganti, rekap kehadiran dan nilai.
- Data terintegrasi dengan sistem informasi akademik, namun proses pengembangan aplikasi tidak mengganggu sistem informasi yang sudah berjalan.

#### B. Fase Analysis

Berdasarkan identifikasi kebutuhan di atas, kemudian dilakukan analisis atas lingkungan implementasi, termasuk lingkungan tempat sistem informasi akademik yang sudah berjalan, kebutuhan fungsional dan arsitektur aplikasi yang akan dikembangkan.

Sistem informasi akademik yang sudah berjalan, dikembangkan menggunakan *database* MySQL dan Bahasa pemrograman PHP, dapat diakses oleh setiap mahasiswa melalui internet ataupun intranet. Sistem ini memuat semua informasi terkait kegiatan akademik mulai dari perwalian, perkuliahan, ujian hingga nilai akhir.

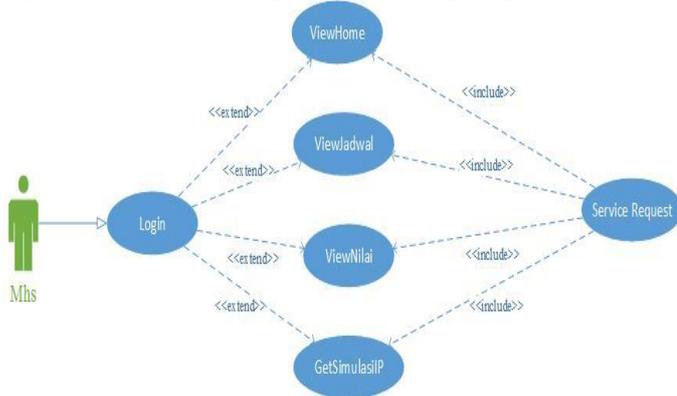
Sistem ini berupa satu portal dengan menggunakan *single sign-on* untuk mengakses semua fitur, termasuk terintegrasi dengan sistem keuangan (pembayaran uang kuliah), dan *email* mahasiswa. Atas dasar kondisi sistem eksisting, maka pengembangan sistem yang baru harus terintegrasi dengan lingkungan sistem yang lama, sehingga dipilih model *web service* SOAP dengan menggunakan *NuSOAP* yang merupakan *toolkit* SOAP untuk PHP.

Berdasarkan identifikasi kebutuhan sistem pada fase *requirement*, selanjutnya dapat diidentifikasi kebutuhan fungsional seperti pada Tabel 2.

TABEL 2. KEBUTUHAN FUNGSIONAL SISTEM

No	Deskripsi Kebutuhan	Keterangan
1	Jadwal Kuliah (Reguler, Uts, Uas, SP)	Daftar jadwal kuliah mahasiswa berdasarkan semester dan tahun ajaran
2	Jadwal Pengganti	Melihat jadwal pengganti mata kuliah
3	Hasil Studi (IP,IPK)	Melihat nilai per semester dan transkrip
4	Kehadiran Mahasiswa	Kehadiran mahasiswa
5	Simulasi IP	Memprediksi nilai mahasiswa per semester yang sedang di jalani

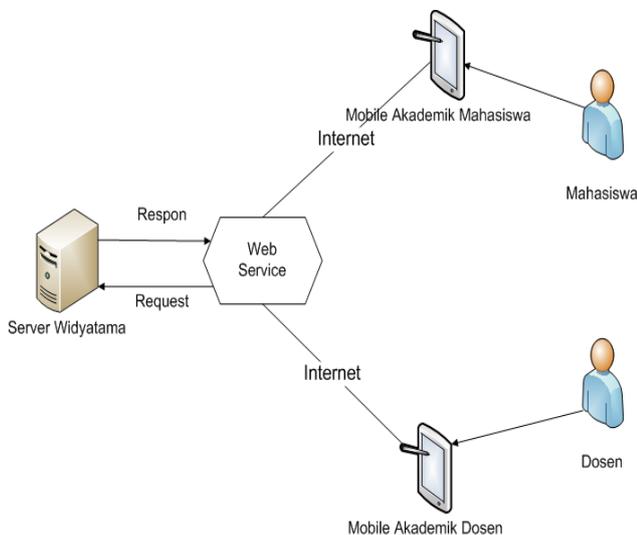
Kebutuhan fungsional sistem ini secara skematis dapat digambarkan melalui diagram *use case* seperti pada Gambar 4.



Gambar 4. Diagram *use case*

User dapat memanfaatkan fungsi-fungsi *web service* yang telah disediakan oleh sistem atau aplikasi. Setiap pemanggilan fungsi yang ada di *web service* tersebut, sistem melewati proses *service request*.

Mengacu pada lingkungan sistem yang sudah berjalan, selanjutnya dikembangkan arsitektur aplikasi tempat implementasi *web service*. *Web service* ini kemudian akan diakses oleh aplikasi berbasis android dari perangkat *smartphone*. Skema arsitektur ini dapat dilihat pada Gambar 5.



Gambar 5. Arsitektur Aplikasi Mobile Akademik

*Web service*, pada arsitektur seperti yang terlihat pada Gambar 5 berperan sebagai perantara bagi *request* yang dikirimkan dari *client* berupa aplikasi android yang berjalan di *smartphone*. Setelah menerima *request* data, *server* akan menanggapi dengan mengirimkan sebuah data berupa XML menuju *web service* dan dikirimkan ke Aplikasi *Mobile akademik*.

Pemanggilan data memanfaatkan internet untuk mengakses *web service*. Pada gambar terlihat ada dua jenis aplikasi *mobile* yaitu untuk mahasiswa dan dosen, namun pada penelitian ini, yang akan dikembangkan adalah aplikasi untuk mahasiswa saja.

### C. Fase Perancangan

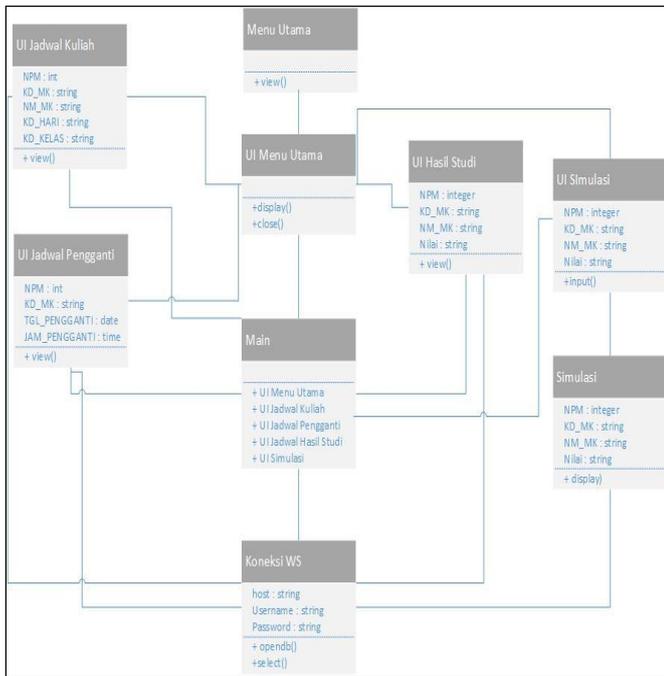
Pada fase perancangan, langkah-langkah yang dilakukan adalah:

- Merancang *class* berdasarkan fungsionalitas *system* (Gambar 6) beserta *scenario* detail untuk setiap fungsi (seperti contoh pada Gambar 7). Setiap *class* yang akan mengakses database, akan melakukan koneksi melalui *web service* (WS). Gambar 7 menampilkan contoh akses data melalui koneksi *web service*.
- Mengidentifikasi fungsi-fungsi yang dibutuhkan, tabel yang akan diakses dan merancang *query* yang akan diterapkan.

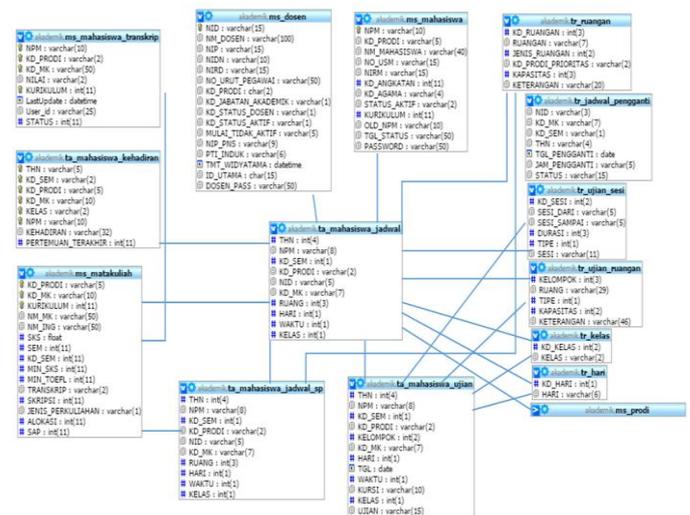
Fungsi-fungsi yang dibutuhkan adalah berupa fungsi tanpa parameter seperti yang terlihat pada Tabel 3. Sumber data berasal dari berbagai *tabel* yang terdapat di dalam sistem informasi akademik. Relasi antar *tabel* yang digunakan pada pengembangan *web service* ini dapat dilihat pada Gambar 8.

TABEL 3. FUNGSI TANPA PARAMETER

Nama Fungsi	Tipe Keluaran Data	Deskripsi
login_mhs	Array String	Fungsi untuk memvalidasi inputan <i>ID</i> dan <i>Password</i> untuk masuk ke aplikasi
jadwal_matkul_mhs	Array String	Fungsi untuk mendapatkan data jadwal kuliah mahasiswa per semester dan tahun ajaran
jadwal_ujian_mhs	Array String	Fungsi untuk mendapatkan data jadwal ujian mahasiswa per semester dan tahun ajaran
cek_jadwal_pengganti_i_mhs	Array String	Fungsi untuk mendapatkan data jadwal pengganti yang sudah dosen tentukan
jadwal_sp_mhs	Array String	Fungsi untuk mendapatkan data jadwal matakuliah yang diulang
nilai_semester_mhs	Array String	Fungsi untuk mendapatkan data nilai mahasiswa per semester
nilai_transkrip_persemester_mhs	Array String	Fungsi untuk mendapatkan data nilai mahasiswa keseluruhan semester
kehadiran_mhs	Array String	Fungsi untuk mendapatkan data jumlah mahasiswa baru regular per prodi
get_angkatan_mhs	Array String	Fungsi untuk mendapatkan data angkatan mhs
get_max_semester_mhs	Array String	Fungsi untuk mendapatkan data semester terakhir
get_tahun_sp_mhs	get_tahun_sp_mhs	Fungsi untuk mendapatkan data tahun apa saja mahasiswa terdaftar sp



Gambar 6. Class diagram aplikasi akses SIM akademik



Gambar 8. Relasi antar tabel

Selanjutnya dibuat rancangan *query* untuk menampilkan data yang dibutuhkan, seperti pada contoh berikut:

**Fungsi *jadwal\_matkul\_mhs***

```
SELECT ms_dosen.NM_DOSEN, tr_hari.HARI, ms_matakuliah.SKS,
tr_ruangan.RUANGAN, tr_ujian_sesi.SESI, tr_kelas.KELAS,
ta_mahasiswa_jadwal.KD_MK, ms_matakuliah.NM_MK
FROM ta_mahasiswa_jadwal, ms_matakuliah, tr_hari, tr_ruangan,
tr_ujian_sesi, tr_kelas, ms_dosen
WHERE ta_mahasiswa_jadwal.KD_MK = ms_matakuliah.KD_MK AND
ta_mahasiswa_jadwal.NPM = '$npm'
AND ta_mahasiswa_jadwal.HARI=tr_hari.KD_HARI
AND ta_mahasiswa_jadwal.KELAS=tr_kelas.KD_KELAS
AND ta_mahasiswa_jadwal.RUANG=tr_ruangan.KD_RUANGAN
AND ta_mahasiswa_jadwal.WAKTU=tr_ujian_sesi.KD_SESI
AND ta_mahasiswa_jadwal.NID=ms_dosen.NID
AND ta_mahasiswa_jadwal.THN = '$tahun'
AND ta_mahasiswa_jadwal.KD_SEM = '$semester'
GROUP BY ta_mahasiswa_jadwal.KD_MK
```

**Fungsi *nilai\_transkrip\_persemester\_mhs***

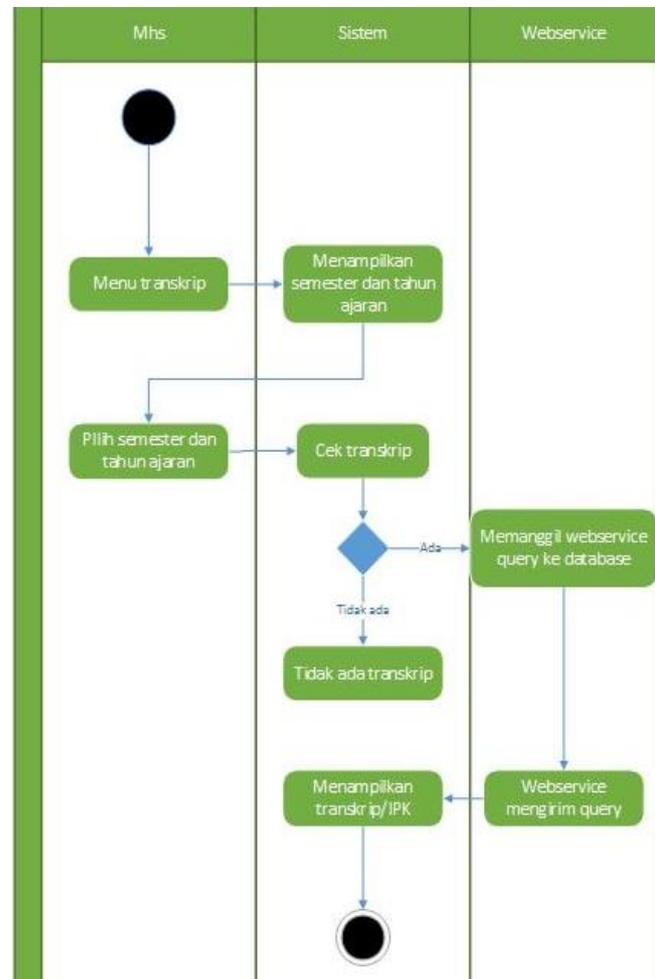
```
SELECT ms_matakuliah.KD_MK, ms_matakuliah.NM_MK,
ms_matakuliah.SKS, ms_mahasiswa_transaksi.NILAI
FROM ms_matakuliah, ms_mahasiswa_transaksi
WHERE ms_mahasiswa_transaksi.KD_MK = ms_matakuliah.KD_MK
AND ms_mahasiswa_transaksi.NPM = '$npm'
AND ms_matakuliah.SEM = '$semester'
```

**Fungsi *kehadiran\_mhs***

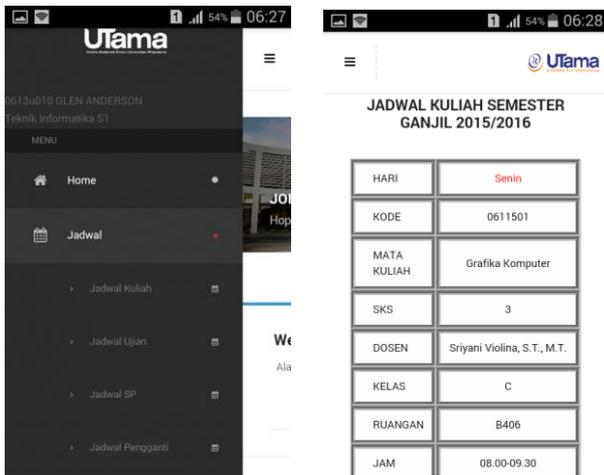
```
SELECT ta_mahasiswa_kehadiran.KD_MK, ms_matakuliah.NM_MK,
ta_mahasiswa_kehadiran.PERTEMUAN_TERAKHIR
FROM ta_mahasiswa_kehadiran, ms_matakuliah
WHERE ta_mahasiswa_kehadiran.KD_MK = ms_matakuliah.KD_MK
AND ta_mahasiswa_kehadiran.NPM = '$npm'
AND ta_mahasiswa_kehadiran.THN = '$tahun'
AND ta_mahasiswa_kehadiran.KD_SEM = '$semester'
```

**D. Implementasi Perancangan**

Berdasarkan hasil rancangan sebelumnya, hasil implementasi akses sistem informasi akademik menggunakan *web service* melalui android dapat dilihat pada Gambar 9, 10 dan 11. Gambar 9 menampilkan menu untuk melihat jadwal kuliah.

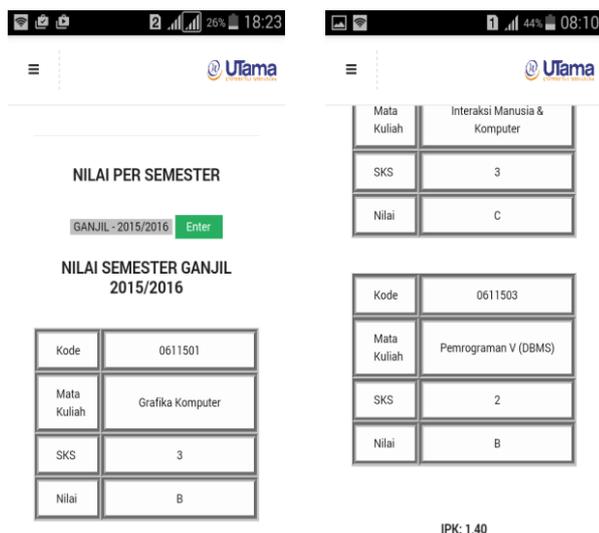


Gambar 7. Activity diagram untuk fungsi “viewnilai”



Gambar 9. Tampilan jadwal kuliah

Tampilan menu untuk melihat nilai dapat dilihat pada Gambar 10.



Gambar 10. Tampilan Nilai Per Semester

Tampilan menu untuk melihat rekap kehadiran mahasiswa dapat dilihat pada Gambar 11.

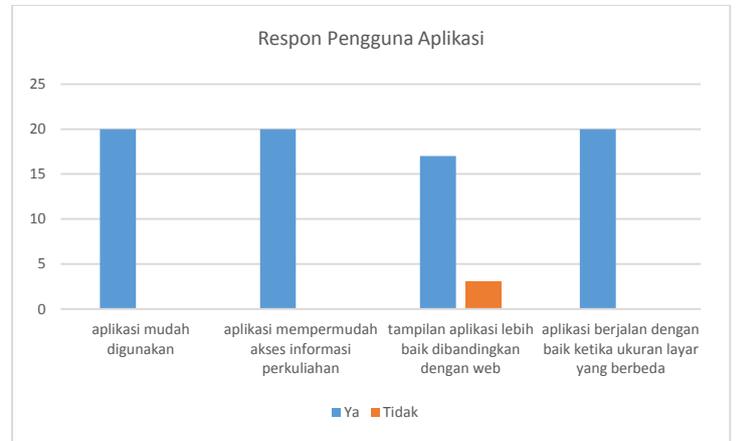


Gambar 11. Tampilan rekap kehadiran per matakuliah

#### E. Pengujian Kemanfaatan

Untuk melihat sejauh mana tanggapan pengguna terhadap aplikasi akses SIM Akademik berbasis android ini, maka dilakukan uji langsung kepada sekitar 20 mahasiswa sebagai *sample* pengguna.

Tujuan pengujian ini untuk melihat apakah aplikasi yang sudah dikembangkan dapat memudahkan mahasiswa untuk mengakses portal akademik melalui aplikasi *mobile* berbasis android, dibandingkan dengan akses melalui *web browser* langsung. Gambar 12 menampilkan hasil persentase pertanyaan-pertanyaan terkait uji pemanfaatan aplikasi.



Gambar 12. Tanggapan Pengguna Aplikasi

Berdasarkan hasil pengisian kuesioner yang telah dilakukan oleh mahasiswa diperoleh sebanyak 100% aplikasi mudah digunakan, 100% aplikasi mempermudah akses informasi, 97% tampilan aplikasi lebih baik dibandingkan dengan *web*, 100% aplikasi berjalan dengan baik ketika ukuran layar yang berbeda.

#### IV. KESIMPULAN

Berdasarkan hasil penelitian pengembangan aplikasi akses SIM Akademik menggunakan *Web Service NuSOAP* ini dapat disimpulkan beberapa hal:

1. Penggunaan *Web Service NuSOAP* dapat diterapkan pada berbagai lingkungan implementasi PHP tanpa memerlukan pustaka tambahan, sehingga memudahkan pada fase implementasi.
2. Penggunaan *Web Service* memungkinkan kita untuk menambahkan fitur pada aplikasi yang sudah berjalan tanpa mengganggu aplikasi itu sendiri. Fitur tambahan ini dapat memperluas fungsionalitas aplikasi yang sudah ada dan dapat dimanfaatkan untuk mendukung integrasi data.
3. Pengembangan akses SIM Akademik berupa aplikasi berbasis android yang dapat menampilkan data-data yang paling sering diakses mahasiswa, dapat membantu mahasiswa mengakses data tersebut secara lebih mudah dan cepat.
4. Aplikasi akses SIM Akademik berbasis android ini sudah diujicobakan kepada pengguna dan mendapat tanggapan yang positif dari segi kemudahan dan kecepatan akses data.

#### DAFTAR PUSTAKA

- [1] D, Aremu, R., O, Adesina, O., 2011, Web services : A Solution to Interoperability Problem in Sharing Grid Resources, ARPN Journal of System and Software, Vol.1. No.4, July 2011, pp. 141-148
- [2] \_\_\_\_\_, Layer of Web Services Protocol Stack, diakses dari : [http://docs.embarcadero.com/products/rad\\_studio/radstudio2007/RS2007\\_helpupdates/HUpdate3/EN/html/devnet/webservicesprotocol\\_xml.html#](http://docs.embarcadero.com/products/rad_studio/radstudio2007/RS2007_helpupdates/HUpdate3/EN/html/devnet/webservicesprotocol_xml.html#)
- [3] W, Moore., C, Allen., S, Chua., D, Koch, D, Marrazo., 2005, Managing Information Access to an Enterprise Information System using J2EE and Services Oriented Architecture, IBM Redbook, diakses dari : <http://www.redbooks.ibm.com/redbooks/pdfs/sg246371.pdf>
- [4] \_\_\_\_\_, Service-Oriented Architecture (SOA) Definition, diakses dari : [http://www.service-architecture.com/articles/web-services/service-oriented\\_architecture\\_soa\\_definition.html](http://www.service-architecture.com/articles/web-services/service-oriented_architecture_soa_definition.html)
- [5] D, Ayala, C, Browne, V, Chopra., P, Sarang., K, Apshankar., T, McAllister., Professional Open Source Web Service, Wrox Programmer to Programmer, Diakses dari : <http://www.actt.co/data/Product/34006.pdf>
- [6] -----, 2012, Creating webservice server and client using NuSOAP, diakses dari : <http://www.ahowto.net/php/creating-webservice-server-and-client-using-nusoap/>
- [7] W, Lee, E., 2005, Web service Implementation Methodology, diakses dari : [https://www.oasis-open.org/committees/download.php/13420/fwsi-im-1.0-guidelines-doc-wd-publicReviewDraft.htm#\\_Toc105472595](https://www.oasis-open.org/committees/download.php/13420/fwsi-im-1.0-guidelines-doc-wd-publicReviewDraft.htm#_Toc105472595)