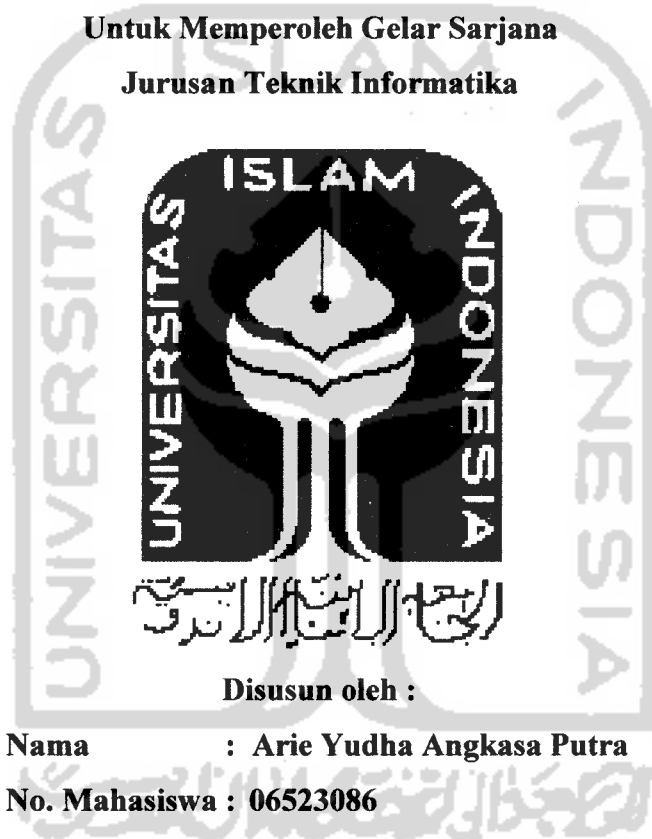


**SISTEM INFORMASI PRESENSI KESISWAAN
SMP / SMA MENGGUNAKAN WEB SERVICE**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika**



Disusun oleh :

Nama : Arie Yudha Angkasa Putra

No. Mahasiswa : 06523086

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

YOGYAKARTA

2010

LEMBAR PENGESAHAN PEMBIMBING

**SISTEM INFORMASI PRESENSI KESISWAAN
SMP / SMA MENGGUNAKAN WEB SERVICE**

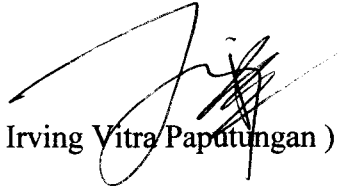
TUGAS AKHIR



Yogyakarta, Juni 2010

Menyetujui,

Pembimbing Tugas Akhir


(Irving Vitra Paputungan) XVIII/10

LEMBAR PENGESAHAN PENGUJI
SISTEM INFORMASI PRESENSI KESISWAAN
SMP / SMA MENGGUNAKAN WEB SERVICE

TUGAS AKHIR

Disusun Oleh :

Nama : Arie Yudha Angkasa Putra

NIM : 06523086

Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelara Sarjana Teknik Informatika Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Tim Penguji

Yogyakarta, Juni 2010

Tanda Tangan

1. Irving Vitra Paputungan
2. R.Teduh Dirgahayu , S.T.,M.Sc.
3. Arwan Ahmad Khoiruddin, S.Kom., M.Cs.

()

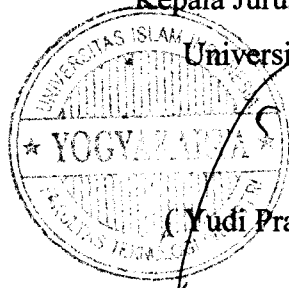
()

()

Mengetahui ,

Kepala Jurusan Teknik Informatika

Universitas Islam Indonesia



(Yudi Prayudi, S.Si, M.Kom)

PERSEMBAHAN

Terima kasih Allah SWT, karena hanya dengan rahmat dan hidayah-Mu semua dapat terjadi.

Kupersembahkan Untuk :

Kedua Orang Tua yang telah membantu untuk berjalan sejauh ini.

Lien yang telah sabar dan setia menunggu

Teman - teman komuri yang telah membantu dalam sharing ilmu.

Teman - teman KPC yang telah membantu dalam mengingat java

Teman - teman Fire 06 dan yang lain- lain yang tidak dapat disebutkan namanya satu persatu untuk dukungan, semangat dan bantuannya.



MOTTO

Terus Bermimpi dan Berusaha

*Hidup Ini Indah Tinggal Bagaimana Kita
Menikmatinya*

*Semua Keberhasilan Terdapat Jalan Dan Proses,
Tinggal Bagaimana Kita Dalam Mensiasatinya*

"Life is free like we used to be"



KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillah penulis panjatkan puji syukur kehadiran Allah SWT karena dengan rahmat, taufik serta hidayah-Nya penulis dapat menyelesaikan laporan Tugas Akhir ini dengan judul ” **SISTEM INFORMASI PRESENSI KESISWAAN SMP / SMA MENGGUNAKAN WEB SERVICE**” dengan baik. Shalawat dan salam semoga tercurahkan atas junjungan kita Nabi Muhammad SAW, sahabat serta para pengikutnya.

Laporan tugas akhir ini disusun guna melengkapi persyaratan dalam menyelesaikan program studi Strata-1 jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia, Yogyakarta.

Dengan terselesaikannya laporan tugas akhir ini, penulis mengucapkan terima kasih kepada :

1. Bapak Ir. Gumbolo Hadi Susanto.,M.Sc. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Yudi Prayudi, S.Si., M.Kom. selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak Irving Vitra Papatungan, selaku Dosen Pembimbing yang telah membimbing dan memberi motivasi kepada penulis untuk menyelesaikan tugas akhir ini.

4. Kedua Orang Tua dan Saudara, yang telah memberikan doa, dukungan dan motivasi untuk menyelesaikan laporan Tugas Akhir ini..
5. Teman-teman Informatika 2006 dan teman-teman yang tidak dapat disebutkan satu-persatu, terima kasih atas dorongan, bantuan dan motivasinya.

Penulis menyadari bahwa penyusunan laporan tugas akhir ini masih terdapat kekurangan karena keterbatasan ilmu dan pengetahuan yang penulis miliki. Oleh karena itu, penulis sangat mengharapkan saran dan kritik yang membangun demi kesempurnaan tugas selanjutnya.

Akhir kata, penulis berharap semoga laporan tugas akhir ini dapat mendatangkan manfaat bagi penulis dan dapat menjadi referensi dikemudian hari bagi para pembaca serta mendatangkan ridha dari Allah SWT. Amin.

Wassalamu 'alaikum Wr. Wb.

Yogyakarta, Juni 2010

Penulis

SARI

Pesatnya perkembangan sistem informasi menuntut akan kemampuan sistem dalam melayani kebutuhan klien akan informasi. Kebutuhan klien yang beragam menuntut akan keamanan data pada server dan kemampuan sistem yang tidak hanya terbatas pada satu bahasa pemrograman atau sistem operasi. Hal tersebut yang sering menjadi permasalahan dalam melayani kebutuhan akan informasi. Untuk memenuhi kebutuhan informasi tersebut dibuat sebuah teknologi yaitu *Web Service*.

Web Service adalah sebuah teknologi yang dijadikan pen jembatan antara aplikasi client dan server. Teknologi ini juga menyediakan keamanan dalam proses pertukaran informasinya. Teknologi RESTful web service merupakan teknologi yang baru dan pengembangan dari teknologi *web service* pendahulunya(SOAP). Teknik arsitektur RESTful *web service* ini telah digunakan pada sistem informasi berskala besar pada tingkat pertukaran informasinya dan keberagaman aplikasi client. Hal ini membuktikan lebih ramah dan mudah digunakan dalam berbagai bahasa pemrograman dan sistem informasi.

Dalam perancangan, program aplikasi ini meminta suatu input yang digunakan sebagai penentu format pertukaran data dan validasi user sebagai penentu hak akses dari web service. Output yang dihasilkan oleh sistem web service berupa XML atau JSON yang berisikan data presensi.

Dari pengujian yang telah dilakukan maka diperoleh suatu kesimpulan bahwa aplikasi web service yang dibuat ini mampu untuk mengamankan akses web service dan mempermudah client dalam melakukan akses data dikarenakan aturan format pertukaran data yang tidak terlalu kompleks.

Kata Kunci :

RESTful Web Service, SOAP, Sistem Informasi, XML, JSON.

TAKARIR

<i>Array</i>	Susunan, aturan.
<i>Client</i>	User.
<i>Decoder</i>	Penerjemah kode.
<i>Encoder</i>	Pengkripsi kode.
<i>Grammar</i>	Tata bahasa.
<i>HTTP Response Code</i>	Kode respon HTTP.
<i>HTTP Request</i>	Permintaan HTTP dari client.
<i>Independent</i>	Berdiri sendiri
<i>Mark-up language</i>	Bahasa penanda
<i>Mobile device</i>	Perangkat telepon
<i>Native</i>	Asli
<i>Private</i>	Pribadi
<i>Reliability</i>	Keandalan suatu sistem.
<i>Request</i>	Permintaan.
<i>Resources</i>	Sumber.
<i>Response</i>	Tanggapan.
<i>Requirements</i>	Kebutuhan.

DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN LEMBAR PENGESAHAN KEASLIAN TUGAS AKHIR.....	ii
HALAMAN LEMBAR PENGESAHAN PEMBIMBING	iii
HALAMAN LEMBAR PENGESAHAN PENGUJI.....	iv
HALAMAN PERSEMBAHAN.....	v
HALAMAN MOTTO	vi
KATA PENGANTAR.....	vii
SARI	ix
TAKARIR.....	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xv
DAFTAR GAMBAR.....	xvii
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3

1.6	Metode Penelitian.....	4
1.7	Sistematika Penulisan.....	4

BAB II DASAR TEORI

2.1	Web Service.....	7
2.1.1	Pengertian REST (REpresentational State Transfer).....	9
2.1.2	REST HTTP Request Method.....	9
2.1.3	REST HTTP Response Code.....	10
2.1.4	Prinsip REST.....	10
2.1.5	XML.....	12
2.1.6	JSON.....	14
2.2	JavaME.....	17
2.3	ZEND Framework.....	18

BAB III METODOLOGI

3.1	Definisi masalah.....	20
3.2	Kebutuhan proses sistem.....	21
3.3	Kebutuhan masukan sistem.....	22
3.4	Keluaran Sistem.....	24
3.5	Batasan Perancangan.....	25
3.6	Analisis Sistem.....	25
3.6.1	Model Proses Sistem.....	26
3.6.2	Entity Relasional Diagram (ERD).....	43
3.6.3	Perancangan Basis Data.....	46
3.6.4	Kinerja yang harus dipenuhi.....	48

3.6.5	Keluaran Aplikasi (Output).....	49
-------	---------------------------------	----

BAB IV HASIL DAN PEMBAHASAN

4.1	Batasan implementasi.....	50
4.1.1	Perangkat Lunak yang dipakai.....	50
4.2	Implementasi Sistem Web Service.....	51
4.2.1	Implementasi fungsi ubah password admin / guru.....	51
4.2.2	Implementasi fungsi lihat presensi semester admin / guru.....	53
4.2.3	Implementasi fungsi lihat presensi detail admin / guru.....	56
4.2.4	Implementasi fungsi verifikasi admin / guru.....	59
4.2.5	Implementasi fungsi lihat presensi siswa.....	61
4.3	Implementasi Sistem Web Service Client (Mobile Device).....	63
4.3.1	Implementasi fungsi verifikasi user pada client.....	63
4.3.2	Implementasi fungsi lihat presensi semester pada client.....	64
4.3.3	Implementasi fungsi lihat presensi detail pada client.....	64
4.3.4	Implementasi fungsi ubah password pada client.....	65
4.3.5	Implementasi fungsi lihat presensi pada client.....	66
4.4	Implementasi Sistem Input Data Presensi.....	67
4.4.1	Implementasi penginputan data presensi pada sistem informasi presensi kesiswaan SMP / SMA.....	67

BAB V PENUTUP

5.1	Kesimpulan.....	69
5.2	Saran.....	70

DAFTAR GAMBAR

Gambar 2.1 Format JSON dalam RFC 4627.....	16
Gambar 2.2 Zend Foldering.....	18
Gambar 3.1 Gambaran Web Service Sistem Informasi Presensi Kesiswaan.....	21
Gambar 3.2 Use Case Diagram Admin / Guru.....	28
Gambar 3.3 Use Case Diagram Siswa.....	29
Gambar 3.4 Class Diagram Web Service Sistem Informasi Presensi Kesiswaan SMP / SMA.....	31
Gambar 3.5 Sequence Diagram Fungsi Ubah Password Admin / Guru.....	32
Gambar 3.6 Sequence Diagram Fungsi Lihat Presensi Semester Admin / Guru	33
Gambar 3.7 Sequence Diagram Fungsi Lihat Presensi Detail Admin / Guru	34
Gambar 3.8 Sequence Diagram Fungsi Verifikasi Admin / Guru.....	35
Gambar 3.9 Sequence Diagram Fungsi Lihat Data Presensi Siswa	36
Gambar 3.10 Activity Diagram Fungsi Ubah Password Admin / Guru.....	38
Gambar 3.11 Activity Diagram Fungsi Lihat Presensi Semester Admin / Guru	39
Gambar 3.12 Activity Diagram Fungsi Lihat Presensi Detail Admin / Guru	40
Gambar 3.13 Activity Diagram Fungsi Verifikasi Admin / Guru.....	41
Gambar 3.14 Activity Diagram Fungsi Lihat Presensi Siswa	42
Gambar 3.15 ERD Web Service Sistem Informasi Presensi Kesiswaan SMP / SMA	45
Gambar 4.1 Implementasi Fungsi Verifikasi User Pada Aplikasi Client Java ME..	63
Gambar 4.2 Implementasi Fungsi Lihat Presensi Semester Pada Aplikasi Client Java ME.....	64
Gambar 4.3 Implementasi Fungsi Lihat Presensi Detail Pada Aplikasi Client Java ME.....	65
Gambar 4.4 Implementasi Fungsi Ubah Password Pada Aplikasi Client Java ME .	66
Gambar 4.5 Implementasi Fungsi Lihat Presensi Pada Aplikasi Client Java ME....	67

Gambar 4.6 implementasi Penambahan Data Presensi Pada Sistem Informasi

Presensi Kesiswaan SMP / SMA68



DAFTAR TABEL

Tabel 2.1 Perbedaan Web Service dan Website.....	8
Tabel 2.2 Kolerasi Method dengan CRUD	9
Tabel 3.1 Tabel Guru.....	46
Tabel 3.2 Tabel Presensi.....	47
Tabel 3.3 Tabel Presensi Aktif.....	47
Tabel 3.4 Tabel Presensi Backup	48



BAB I

PENDAHULUAN

1.1 Latar Belakang

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi dan bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan informasi kepada pihak luar tertentu untuk pengambilan keputusan (Magaline, 2009). Sistem informasi presensi kesiswaan contohnya, sebagai penyedia informasi presensi kesiswaan sekarang ini dalam pengembangannya seringkali hanya terbatas pada aplikasi berbasis web.

Aplikasi berbasis web kurang mampu memenuhi kebutuhan *client* yang beragam seiring dengan perkembangan pesat di bidang teknologi informasi. Hal tersebut diakibatkan oleh perangkat keras, sistem operasi dan bahasa pemrograman yang beredar menjadi semakin banyak. Kebutuhan yang beragam dalam kemampuan mengakses data membuat kemudahan dan keamanan data tidak terjamin. Dampak positif dari keberagaman teknologi dan bahasa adalah banyaknya pilihan untuk menggunakan sistem operasi dan bahasa pemrograman. Namun terdapat dampak negatif yang muncul yaitu pada penggunaan sistem operasi dan bahasa pemrograman yang berbeda-beda untuk setiap komputer akan mempersulit proses pertukaran data dan informasi diantara komputer - komputer tersebut. Belum lagi jika perangkat lain seperti telepon genggam atau *mobile device* lainnya juga perlu melakukan pertukaran data dengan komputer - komputer tersebut.

Pertukaran data pada sistem informasi membutuhkan *aksesibilitas* yang beragam dan menuntut keamanan dalam proses pertukaran datanya. Kemampuan akses suatu sistem secara universal dapat diwujudkan dengan adanya *web service* (Short, 2003). Layanan web atau yang lebih dikenal dengan *web service* merupakan istilah serupa payung yang menerangkan sekumpulan layanan dan protokol standar

- c. Siswa dan guru dapat mengetahui sejarah presensi secara online dengan menggunakan *handphone*.
- d. Memberikan sumbangan bagi studi dan penelitian selanjutnya.

1.6 Metode Penelitian

Untuk mengumpulkan data penyusunan laporan ini, maka penyusun menggunakan metode :

- a. Metode Pengumpulan Data.

Pengumpulan data dalam penelitian ini menggunakan metode studi pustaka. Metode ini digunakan untuk mendapatkan informasi tambahan seperti gaya arsitektur *web service*, XML, JSON dan javaME yang digunakan sebagai acuan dalam membangun aplikasi, dalam hal ini aplikasi *web service* sistem informasi presensi kesiswaan dan aplikasi *mobile* sebagai *client web service*.

- b. Metode Pengembangan Sistem.

Metode pengembangan sistem yang digunakan meliputi analisis *web service* sistem informasi presensi kesiswaan seperti penentuan fungsi - fungsi aplikasi *web service* yang dibutuhkan oleh *client* dalam memperoleh informasi, perancangan aplikasi seperti penentuan format pertukaran data maupun teknik pengamanan dalam membatasi hak akses client, implementasi aplikasi dan analisis kinerja aplikasi dalam bahasa pemrograman yang berbeda.

1.7 Sistematika Penulisan

Untuk mempermudah pembacaan yang lebih akurat dan memberikan gambaran secara menyeluruh masalah yang akan dibahas, maka sistematika laporan dibagi dalam lima bab dan garis besar isinya yaitu :

Bab I Pendahuluan

Bab ini membahas tentang latar belakang masalah pembuatan Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan Web Service, rumusan

BAB II

DASAR TEORI

2.1 Web Service

World Wide Web Consortium (W3C) , organisasi yang mengembangkan standar dalam dunia web mendefinisikan *web service* sebagai sebuah *software* aplikasi yang teridentifikasi oleh URL dan memiliki *interface* yang didefinisikan, dideskripsikan, dan dimengerti oleh xml dan juga mendukung interaksi langsung dengan software aplikasi yang lain dengan menggunakan message berbasis xml melalui protokol internet (Booth,2004).

W3C mendefinisikan *web service* sebagai sebuah sistem perangkat lunak yang dirancang untuk mendukung komunikasi dan interaksi antar mesin ke mesin (*Machine to Machine*) melalui sebuah *network* (jaringan). *Web Service* juga termasuk Web APIs yang dapat diakses melalui jaringan seperti misalnya internet, dan dieksekusi melalui sebuah sistem jarak jauh sesuai dengan layanan yang diminta.

Web service merupakan software aplikasi yang tidak terpengaruh oleh *platform*, ia akan menyediakan method yang akan dapat diakses oleh network. *Web service* juga akan menggunakan XML untuk pertukaran data, khususnya pada dua entitas bisnis yang berbeda. Beberapa bahasa standar dalam pertukaran data yang ada didalam *web service* adalah : extensible markup language (XML) dan JavaScript Object Notation(JSON). Sedangkan dalam pembuatan *web service* itu sendiri digunakan konsep arsitektur seperti Simple Object Access Protocol (SOAP), REpresentational State Transfer(REST), dan XML-RPC(XML-Remote Procedure Calls).

Kemunculan teknologi *web service* yang saat ini sedang populer memberikan paradigma baru dalam mengimplementasikan akses dan aplikasi informasi. Perkembangan *web service* diawali dengan kehadiran XML untuk pertukaran data berbasis web secara fleksibel.

Web service merupakan salah satu bentuk implementasi dari arsitektur N-tier yang diletakkan pada tier business services yang artinya didalam web service hanya tersedia fungsi – fungsi yang nantinya dapat digunakan oleh suatu aplikasi (Hadiwinata, 2003).

Web service dapat diartikan sebagai *software resource* yang melakukan suatu fungsi dan menyediakan jawabannya tetapi tidak memiliki tampilan yang menarik seperti website. *Web service* dikembangkan dengan mengambil satu set fungsi dan membungkusnya agar layanan yang dilakukan terlihat dan dapat diakses oleh aplikasi software lainnya.

Pemanfaatan teknologi *web service* sekarang ini dapat kita lihat pada situs jejaring sosial seperti facebook, twitter, plurk dan lain - lain. Situs jejaring tersebut telah menggunakan web service sebagai alat komunikasi antara aplikasi *client* yang berupa aplikasi *mobile* yang dapat mengubah status profil, mengirimkan pesan kepada teman, login pada aplikasi lain dengan menggunakan akun jejaring sosial dan lain - lain. Dengan adanya teknologi tersebut maka akan diciptakan sebuah jembatan atau portal antara *client* dan *server* untuk saling berhubungan dan dapat menciptakan keamanan dalam proses komunikasi karena aplikasi *client* tidak berhubungan langsung dengan database.

Ada beberapa yang membedakan antara *web service* dengan web site (Daniel, 2003) antara lain :

Tabel 2.1 Perbedaan Web Service dan Website

WEB SERVICE	WEBSITE
a. Tidak memiliki <i>interface</i> yang bagus	a. Memiliki web <i>interface</i>
b. Dibuat untuk berinteraksi langsung dengan aplikasi yang lain walaupun berbeda OS / Konsep sekalipun.	b. Dibuat untuk berinteraksi langsung dengan user
c. Dibuat untuk bekerja pada semua tipe aplikasi <i>client</i>	c. Dibuat untuk bekerja pada <i>web browser</i> .

- Sudah dibatasi untuk jenis konten, sering kali dibuat *code on demand* / tipe data yang dikeluarkan disesuaikan dengan permintaan dari *client*.

d. Protokol yang digunakan adalah :

- *Client/Server*

Antarmuka *client* dan *server* tidak berhubungan, di tujukan untuk memberikan kebebasan pada *client* akan apa saja data yang ditampilkan dan proses apa saja yang dilakukan oleh *client*.

- *Stateless*

Sistem *web service* ini bersifat mandiri atau berdiri sendiri. Berdiri sendiri diartikan bahwa *server* tidak mengurus aplikasi *client* seperti penyimpanan *session client*, data yang ingin disimpan dalam aplikasi *client*, dan sebagainya. Jadi *server* hanya memberikan respon yang *client* minta berdasarkan *request method* dan parameter yang dikirimkan oleh *client*.

- *Cacheable*

Sistem dari *web service* REST ini mendukung penyimpanan data sementara. Ini dimaksudkan untuk mempercepat respon dari *server* kepada *client* karena data yang dikirim oleh *server* selalu sama dalam jangka waktu tertentu sehingga data tersebut disimpan dalam *memory cache* / penyimpanan data sementara pada *server* untuk selanjutnya respon pada *client* akan dikirimkan secara langsung dari penyimpanan data sementara pada *server*. Data yang disimpan sementara ini akan diubah saat *database* pada *server* dilakukan perubahan.

- *Layered*

Web service bekerja sebagai perantara antara *server* dan *client*. Dan aplikasi *client* tidak selalu terhubung pada *server*. Sistem ini dapat meningkatkan keamanan pengaksesan data.

2.1.5 XML

XML merupakan kepanjangan dari *Extensible Markup Language*. XML merupakan suatu Teknologi yang digunakan untuk membawa suatu data dan bisa digunakan untuk mengirim suatu data atau bisa didefinisikan sebuah *meta language* untuk mendeskripsikan sebuah data. Teknologi XML mempunyai fungsi sebagai *markup language* atau sering disebut sebagai bahasa penandaan dan merupakan salah satu dari standarisasi web yang diberikan oleh W3C (W3C,2000).

XML (eXtensible Markup Language) merupakan bahasa web turunan dari SGML (Standart Generalized Markup Language) yang ada sebelumnya. XML hampir sama dengan HTML, dimana keduanya sama-sama turunan dari SGML. Teknologi XML dikembangkan mulai tahun 1966 dan mendapatkan pengakuan dari World Wide Web Consortium (W3C) pada bulan Februari 1998. Sedangkan SGML sendiri telah dikembangkan pada awal tahun 1980-an. Pada saat HTML dikembangkan pada tahun 1990, para pengagas XML mengadopsi bagian paling penting SGML dan dengan berpedoman pada pengembangan HTML menghasilkan bahasa penanda yang tidak kalah hebatnya dengan SGML.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa *meta-markup* yang menyediakan format tertentu untuk dokumen – dokumen yang mempunyai data terstruktur. Bahasa penanda adalah mekanisme untuk mengenal secara terstruktur di dokumen. XML adalah suatu aplikasi profil dari SGML. Seperti yang didefinisikan dari ISO 8879, SGML adalah cara standart dan *vendor-independent* (Bintang, 2009).

XML tidak mempunyai definisi secara tepat karena ada yang berpendapat bahwa XML bukanlah suatu bahasa pemrograman, melainkan XML merupakan sintaks yang digunakan untuk menjelaskan bahasa penanda lain (Dournaee,2002), sehingga dinamakan *meta-language*. Meskipun demikian pendapat yang XML bukan merupakan bahasa penanda, didasarkan bahwa XML merupakan bahasa penanda

terpisah untuk tujuan terpisah. Selain itu XML bukanlah solusi semua hal untuk tujuan semua user. Sedangkan peran dari *markup* itu sendiri berupa (Potts, 2003):

- a. *Markup* dapat menambah maksud arti (semantik) suatu data.
- b. Dapat memisahkan data.
- c. Dapat mendefinisikan peran data.
- d. Dapat mendefinisikan batasan data.
- e. Dapat menfenisikan keterhubungan data.

XML merupakan sebuah himpunan bagian dari SGML yang bertujuan agar SGML secara generik dapat melayani, menerima, dan memproses di dalam web dengan cara seperti yang dimungkinkan HTML saat ini. XML didesain untuk kemudahan implementasi dan interoperabilitas dengan SGML maupun HTML. XML adalah bahasa penanda yang dirancang untuk penyampaian informasi melalui World Wide Web (www) atau sering disebut web (W3C,2000).

XML merubah cara berpikir untuk mengembangkan suatu *software* terutama aplikasi web. Masalah yang sering dihadapi sekarang adalah bagaimana caranya untuk bertukar informasi antar satu aplikasi dengan aplikasi lain. Kadang kolaborasi antara satu aplikasi dengan aplikasi yang lain masih harus ditentukan dengan spesifikasi aplikasi tersebut. Padahal seharusnya hanya perlu mendapatkan informasi data bukan mengerti cara kerja aplikasi lain itu, disinilah visi internet yang belum tercapai. Visi ini adalah dunia internet dimana PC, *server*, *smart devices* dan *internet-based device* dapat berkolaborasi tanpa ada halangan. Bisnis – bisnis akan dapat bertukar data menyediakan *customized* dan *comprehensive* solusi kepada customer. Hal yang paling utama adalah informasi yang dibutuhkan dapat diakses dari mana saja dan dengan perangkat komputer, *platform*, atau aplikasi yang digunakan.

XML dapat memungkinkan pertukaran informasi atau data antar perangkat (*server*, PCs, smart device, aplikasi, dan situs web). Data ini akan menjadi *independent (unlocked)*, memudahkannya untuk diorganisir, diprogram, dan dirubah, dan ditukar antar situs web atau aplikasi apa saja. Karena kebutuhan ini, maka makin banyak teknologi berbasis XML yang keluar (Richardson, 2007).

Keuntungan penggunaan XML adalah:

- a. *Platform Independent*.
- b. Bisa dibuat dan dibuka disembarang editor karena XML disimpan dalam format text murni.
- c. sintaks jelas dan mudah dipahami.
- d. Bisa diakses dan di olah oleh berbagai macam bahasa pemrograman.
- e. Bebas Untuk dipelajari dan dipakai tanpa harus membayar lisensi.

Pada REST *web service* penggunaan XML pada proses pertukaran data antara *server* dan *client* dapat dibuat format tersendiri. Format ini tidak terikat dengan WSDL (Web Service Definition Language) dan UDDI (Universal Description Discovery and Integration). Contoh *response* XML pada REST *web service* adalah sebagai berikut :

```
<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <response>
    <message>you have not specified any method!</message>
  </response>
  <status>failed</status>
</rest>
```

Format penulisan sintaks yang tidak tersebut akan sangat memudahkan aplikasi *client* dalam melakukan penguraian isi dari file XML dan mengurangi tingkat konsumsi *bandwith*.

2.1.6 JSON

JSON merupakan format pertukaran data yang lebih ringan daripada XML, lebih ringkas dimana lebih mudah dibaca dan untuk diuraikan maupun dibentuk. JSON juga merupakan bahasa pemrograman yang berdiri sendiri dan ramah dengan bahasa pemrograman yang lain (Sang, 2007).

JSON, singkatan dari *JavaScript Object Notation* (notasi objek JavaScript), adalah suatu format ringkas pertukaran data komputer. Formatnya berbasis teks dan

terbaca – manusia serta digunakan untuk merepresentasikan struktur data sederhana dan larik asosiatif (disebut objek). Format JSON sering digunakan untuk mentransmisikan data terstruktur melalui suatu koneksi jaringan pada suatu proses yang disebut serialisasi. Walaupun JSON didasarkan pada subset bahasa pemrograman JavaScript (secara spesifik, edisi ketiga standar ECMA-262, Desember 1999) dan umumnya digunakan dengan bahasa tersebut, JSON dianggap sebagai format data yang tak tergantung pada suatu bahasa. Kode untuk pengolahan dan pembuatan data JSON telah tersedia untuk banyak jenis bahasa pemrograman. Situs www.json.org menyediakan daftar komprehensif pengikatan JSON yang tersedia, disusun menurut bahasa. Format JSON dispesifikasikan di RFC 4627 oleh Douglas Crockford. Tipe media Internet resmi JSON adalah *application/json* sedangkan ekstensi berkasnya adalah *.json*.

Berbeda dengan XML yang bersifat universal, JSON-RPC menggunakan format JSON untuk mentransmisikan data terstruktur. JSON-RPC memiliki kesamaan tipe sistem seperti halnya JSON untuk menspesifikasikan data terstruktur dalam sebuah *call*. Dalam bentuk abstrak (sintak dan *grammar*), JSON mampu merepresentasikan empat tipe primitif: *String*, *Number*, *Boolean* dan *Null*

- a. *String* Sekuensial karakter *unicode* yang terdiri dari nol atau lebih.
- b. *Number* Representasi *integer*, bisa sebagai *sign* atau *unsign*.
- c. *Boolean* Komponen yang secara literal menyatakan *true* atau *false*.
- d. *Null* Komponen yang secara literal menyatakan *null*.

Serta dua tipe struktur: *Object* dan *Array*

- a. *Array* Tipe primitif terurut.
- b. *Object* Pasangan tipe primitif tidak terurut.

Pada beberapa implementasi *native*, elemen ini bisa terurut seperti halnya *Array*. Bahasa pemrograman yang mengimplementasikan *encoder* atau *decoder* format JSON, akan melakukan pemetaan atau konversi tipe-tipe primitif JSON menjadi tipe-tipe primitif atau turunan bahasa itu sendiri. Hampir semua bahasa pemrograman yang ada saat ini seperti: C/C++, Java, PHP, PERL, Python, Delphi,

Flex, Scheme, memiliki implementasi encoder dan decoder format JSON. Proses pemetaan struktur data dari sebuah bahasa *native* ke dalam format JSON disebut serialisasi, sedangkan proses pemetaan dari format JSON ke dalam struktur data bahasa *native* disebut deserialisasi. Pada umumnya, tipe *primitive* yang dimiliki oleh JSON ada pada hampir semua bahasa pemrograman sehingga hampir tidak ada masalah dalam proses serialisasi maupun deserialisasi. (admin, 2008)

Format JSON dispesifikasikan dalam RFC 4627 oleh Douglas Crocford. Dalam AJAX, JSON merupakan alternatif dari format data yang umumnya digunakan yaitu XML.

JSON dengan 2 element	JSON dengan 2 kecabangan
<pre>{ "data": { "element": { "item1": "aaaaaaaa", "item2": "bbbbbbbb" } } }</pre>	<pre>{ "data" { "element1" { "element": { "item1" ["aaaaaaaa", "bbbbbbbb"] } } } }</pre>

Gambar 2.1 Format JSON dalam RFC 4627

Pada gambar diatas merupakan contoh data JSON yang digunakan dalam pengujian. Data tersebut sama dengan data XML seperti pada gambar diatas. Untuk JSON tidak terdapat atribut sehingga untuk perbandingan XML dengan Atribut adalah JSON dengan elemen (Honeys, 2008).

Dalam percobaan pengaksesan data XML dan JSON yang mempunyai data yang sama membuktikan bahwa JSON mempunyai keunggulan dibanding xml yaitu dari segi ukuran *file response* dan kecepatan pengaksesan data (Masykur, 2008).

2.2 JavaME

J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. Pada J2ME, jika perangkat lunak berfungsi baik pada perangkat maka belum tentu juga berfungsi baik pada perangkat yang lainnya. J2ME membawa Java ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer desktop yang biasanya lebih kecil dibandingkan dengan perangkat komputer tersebut.

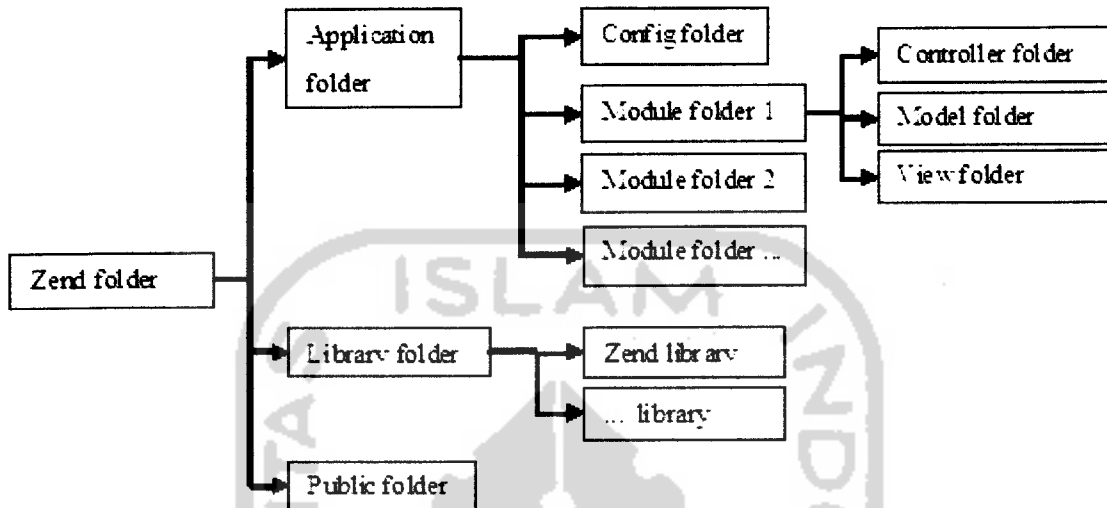
Program J2ME, seperti semua program JAVA adalah diterjemahkan oleh VM. Program – program tersebut dieksekusi ke dalam bytecode dan diterjemahkan dengan Java Virtual Machine(JVM). Ini berarti bahwa program-program tersebut tidak berhubungan langsung dengan perangkat. J2ME menyediakan suatu interface yang sesuai dengan perangkat. Aplikasi-aplikasi tersebut tidak harus dieksekusi ulang supaya mampu dijalankan pada mesin yang berbeda.

Inti dari J2ME terletak pada konfigurasi dan profil – profil. Suatu konfigurasi menggambarkan lingkungan runtime dasar dari suatu sistem Java ME. Ia menggambarkan *core library*, *virtual machine*, fitur keamanan dan jaringan. Sebuah profil memberikan *library* tambahan untuk suatu kelas tertentu pada sebuah perangkat. Profil – profil menyediakan *user interface*(UI) API, *persistence*, *messaging library*, dan sebagainya.

Satu set *library* tambahan atau *package* tambahan menyediakan kemampuan program tambahan. Pemasukan *package* ini ke dalam perangkat Java ME dapat berubah – ubah karena tergantung pada kemampuan sebuah perangkat. Hal ini dapat dicontohkan dalam penggunaan *package* pengolahan data JSON. Data JSON jarang digunakan oleh aplikasi *mobile* sehingga *package* ini tidak disediakan oleh perangkat *mobile*.

2.3 ZEND Framework

Zend Framework adalah framework yang telah memiliki Model – View – Controller (MVC). Dalam direktori folder zend dapat digambarkan sebagai berikut :



Gambar 2.2 Zend Foldering

Pada gambar diatas terdapat 3 main folder utama yaitu *application folder*, *library folder* dan *public folder*. *Application folder* berisikan konfigurasi framework dan logika sistem yang dipisahkan menjadi beberapa modul menurut kebutuhan. Pada modul – modul tersebut dikategorikan kembali kedalam 3 folder yaitu *controller*, *model* dan *view*. Pada *library folder* terdapat *zend library* sebagai *library* standar dan apabila kita akan menambahkan *library*, akan ditambahkan pada *folder* tersebut. Sedangkan pada *public library* hanya berisikan indeks dari web dan layout atau css bagi yang membutuhkan.

Zend framework juga telah mendukung penggunaan *REST service* secara menyeluruh. Zend sendiri merupakan framework dengan bahasa PHP. Fungsi – fungsi yang digunakan dalam pembuatan *service* presensi ini yaitu:

a. *Zend_Rest_Route*

Ini digunakan untuk memetakan *URI service* yang akan dibuat.

b. `Zend_Module_Autoloader`

Ini digunakan untuk memetakan model yang berisi *class database* untuk melakukan *query* pada *database*.

c. `setAutoloaderNamespaces`

Ini digunakan untuk mengenali penamaan *file library* yang dibuat secara manual sehingga *library* yang kita buat dapat digunakan dalam Zend MVC.

d. `Zend_Db_Table_Abstract`

Ini digunakan untuk membuat fungsi *query database* dalam Zend.



BAB III

METODOLOGI

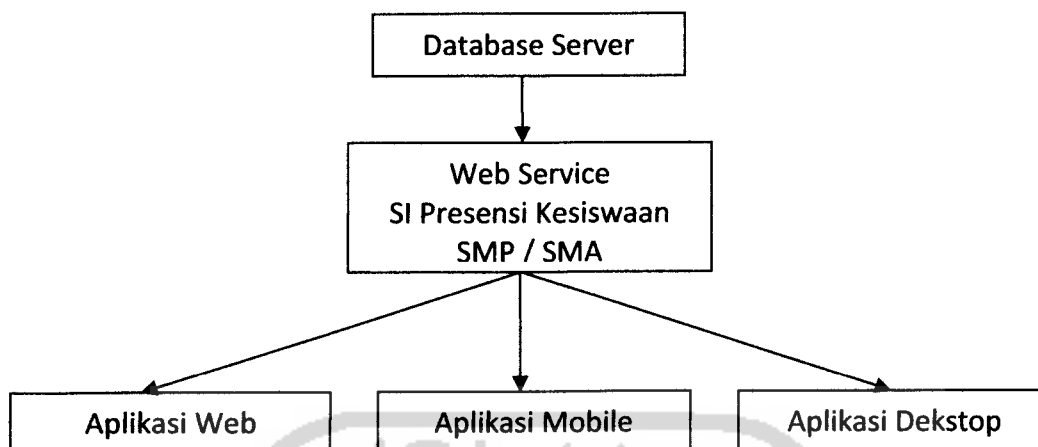
3.1 Definisi masalah

Web service sebagai sebuah sistem perangkat lunak yang dirancang untuk mendukung komunikasi dan interaksi antar mesin ke mesin (*Machine to Machine*) melalui sebuah *network* / jaringan (W3C,2000). Dalam membuat *web service* hal diperlukan autentikasi atau pengamanan dalam mengakses modul *web service* dan penentuan format pertukaran data (W3C,2000). Hal tersebut merupakan hal yang harus diperhatikan dalam *web service* dikarenakan keamanan data pada *server* dan kemudahan *client* dalam mengakses data.

Autentikasi dalam mengakses dimaksudkan untuk mengatur hak akses *client* dalam menggunakan layanan. Selain pada masalah keamanan akses terdapat juga masalah dalam hal format pertukaran data dalam *web service*.

Format pertukaran data yang biasa didukung oleh *web service* adalah XML. Pada SOAP *web service* proses pertukaran data digunakan XML dengan aturan WSDL dan UDDI. Penggunaan aturan WSDL pada *file* XML mengakibatkan besarnya *file* XML dan kecepatan dalam pengolahan *file* XML tersebut menjadi semakin lama. Selain format XML terdapat juga format JSON yang cenderung lebih ramah dengan bahasa pemrograman. JSON ini belum didukung oleh SOAP *web service*. Kecepatan akses *file* JSON cenderung lebih cepat daripada XML. Hal ini dikarenakan secara struktural, JSON lebih mengarah ke *object* dan *array*.

Dari permasalahan diatas dapat digambarkan sebagai berikut :



Gambar 3.1 Gambaran Web Service Sistem Informasi Presensi Kesiswaan

Pada gambar diatas dijelaskan pengelompokan web sistem informasi presensi kesiswaan berdasar jenisnya. Pada website sistem informasi presensi client melakukan akses langsung pada aplikasi website sistem informasi presensi kesiswaan. Aplikasi *client* pada bagian ini tidak berdiri sendiri melainkan pada *server*. Berdiri sendiri tersebut dimaksudkan apabila aplikasi *client* terletak pada *server* yang berbeda. Sedangkan pada web service sistem informasi presensi kesiswaan ditujukan untuk *client* yang berdiri sendiri. Aplikasi *client* ini menggunakan *web service* sebagai perantara antara aplikasi dan *database server*.

3.2 Kebutuhan proses sistem

Kebutuhan proses sistem yang diperlukan bagi *web service* sistem informasi presensi yaitu :

1. Fungsi verifikasi guru

Ini digunakan untuk melakukan pengecekan akun admin atau guru dengan data admin atau guru yang ada pada *database*.

2. Fungsi verifikasi siswa

Ini digunakan untuk melakukan pengecekan akun siswa dengan data admin atau guru yang ada pada *database*.

3. Fungsi lihat berita

Ini digunakan untuk melihat berita terbaru sistem informasi presensi kesiswaan.

4. Fungsi lihat kelas

Ini digunakan untuk mendukung fungsi lihat presensi semester, lihat presensi detail dan lihat presensi dalam memperoleh "id_kelas".

5. Fungsi lihat semester tahun ajaran

Ini digunakan untuk mendukung fungsi lihat presensi semester, lihat presensi detail dan lihat presensi dalam memperoleh "sem_thn_ajar".

6. Fungsi lihat presensi semester

Ini digunakan untuk melihat data presensi siswa secara total oleh admin / guru.

7. Fungsi lihat presensi detail

Ini digunakan untuk melihat data presensi siswa secara mendetail oleh admin / guru.

8. Fungsi lihat presensi

Ini digunakan untuk melihat presensi pribadi siswa secara total.

9. Fungsi lihat data pribadi admin / guru

Ini digunakan untuk melihat data pribadi oleh admin / guru.

10. Fungsi lihat data pribadi siswa

Ini digunakan untuk melihat data pribadi oleh siswa.

11. Fungsi ubah password admin / guru

Ini digunakan untuk mengubah *password* admin / guru.

12. Fungsi ubah password siswa

Ini digunakan untuk mengubah *password* siswa.

3.3 Kebutuhan masukan sistem

Kebutuhan masukan dari pengguna *web service* sistem informasi presensi kesiswaan diperlukan untuk beberapa keperluan yang berbeda, yaitu :

1. Kebutuhan masukan untuk pengguna *web service* sistem informasi presensi kesiswaan dengan hak akses sebagai guru / admin yaitu :
 - a. Proses validasi user, kebutuhan masukannya adalah
 - NIP
 - Secret (password guru / admin)
 - *HTTP request method* : GET
 - b. Proses lihat berita, kebutuhan masukannya adalah
 - *HTTP request method* : GET
 - c. Proses ubah password, kebutuhan masukannya adalah
 - NIP
 - Password lama
 - Password baru
 - *HTTP request method* : PUT
 - d. Proses lihat data pribadi, kebutuhan masukannya adalah
 - username (otentikasi dari URI)
 - password (otentikasi dari URI)
 - *HTTP request method* : GET
 - e. Proses lihat presensi semesteran siswa, kebutuhan masukannya adalah
 - username (otentikasi dari URI)
 - password (otentikasi dari URI)
 - *HTTP request method* : GET
 - f. Proses lihat detail presensi siswa, kebutuhan masukannya adalah
 - username (otentikasi dari URI)
 - password (otentikasi dari URI)
 - nomor induk siswa
 - status presensi
 - *HTTP request method* : GET

2. Kebutuhan masukan untuk pengguna web service sistem informasi presensi kesiswaan dengan hak akses sebagai *siswa* pada web service sistem informasi presensi yaitu :
 - a. Proses validasi user, kebutuhan masukannya adalah
 - *Nomor Induk*
 - *Secret (password siswa)*
 - *HTTP request method: GET*
 - b. Proses lihat berita, kebutuhan masukannya adalah
 - *HTTP request method: GET*
 - c. Proses ubah password, kebutuhan masukannya adalah
 - *Nomor Induk*
 - *Password lama*
 - *Password baru*
 - *HTTP request method: PUT*
 - d. Proses lihat data pribadi, kebutuhan masukannya adalah
 - *username (autentikasi dari URI)*
 - *password (autentikasi dari URI)*
 - *HTTP request method: GET*
 - e. Proses lihat presensi pribadi siswa, kebutuhan masukannya adalah
 - *username (autentikasi dari URI)*
 - *password (autentikasi dari URI)*
 - *HTTP request method: GET*

3.4 Keluaran Sistem

Adapun keluaran yang dihasilkan merupakan informasi dari hasil respon *web service* sistem informasi presensi. Data – data tersebut antara lain adalah sebagai berikut.

- a. Data berita terbaru.

Berisi tentang pengumuman sekolah.

b. Data diri pengguna.

Berisi data pribadi seperti nama, foto, alamat, dan lain – lain.

c. Data presensi

Pada user siswa hanya ditampilkan data presensi dari siswa yang mengakses. Sedangkan pada user guru / admin akan ditampilkan list presensi siswa berdasarkan tahun ajaran, semester, dan kelas siswa.

3.5 Batasan Perancangan

Dalam perancangan aplikasi *web service* sistem informasi presensi kesiswaan terdapat beberapa batasan , yaitu sebagai berikut :

1. Fungsi untuk menampilkan informasi presensi kesiswaan
Pada user guru / admin informasi presensi ditampilkan secara terperinci sedangkan pada user siswa hanya ditampilkan jumlah total presensi.
2. Fungsi untuk menampilkan data pribadi
3. Fungsi untuk mengubah password
4. Fungsi untuk menampilkan data berita
5. Fungsi untuk menampilkan data kelas
6. Fungsi untuk menampilkan data tahun ajaran
7. Fungsi untuk menampilkan data semester tahun ajaran

3.6 Analisis Sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui semua permasalahan serta kebutuhan yang diperlukan dalam pengembangan aplikasi. Analisis dilakukan dengan mencari dan menentukan permasalahan yang dihadapi serta semua kebutuhan seperti analisis masalah, analisis sistem, masukan dan keluaran sistem, antarmuka sistem, dan fungsi – fungsi yang dibutuhkan.

Metode analisis yang digunakan dalam mengembangkan perangkat lunak ini adalah metode analisis berorientasi objek dengan menggunakan standar *Unified Modeling Language* (UML) sebagai alat bantu.

Pada sistem aplikasi sistem ini akan dibuat dua buah aplikasi yang berbeda, yaitu aplikasi *Web Service Sistem Informasi Presensi* pada *server*. Secara garis besar, *Web Service Sistem Informasi Presensi* adalah program utama yang menyediakan layanan pengaksesan informasi presensi siswa. Sedangkan pada *client* digunakan aplikasi *mobile* sederhana yang berfungsi untuk mengakses layanan pada *web service server*.

3.6.1 Model Proses Sistem

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti, 2003).

UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax*/semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson OOSE (Object-Oriented Software Engineering).

Berikut ini akan dijelaskan beberapa yang akan digunakan perancangan aplikasi web service sistem informasi presensi kesiswaan SMP / SMA.

a. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah *entitas* manusia atau mesin yang berinteraksi dengan system untuk melakukan pekerjaan-pekerjaan tertentu (Dharwiyanti, 2003).

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan *client*, dan merancang *test case* untuk semua feature yang ada pada sistem.

Sebuah *use case* dapat memasukkan fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang dimasukkan akan dipanggil setiap kali *use case* yang memasukkan dieksekusi secara normal.

Sebuah *use case* dapat dimasuki lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas.

Sebuah *use case* juga dapat memperluas *use case* lain dengan kebiasaannya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Diagram *Use Case* berguna dalam tiga hal (Dharwiyanti, 2003):

- Menjelaskan fasilitas yang ada (*requirements*)

Use Case baru selalu menghasilkan fasilitas baru ketika sistem di analisa, dan design menjadi lebih jelas.

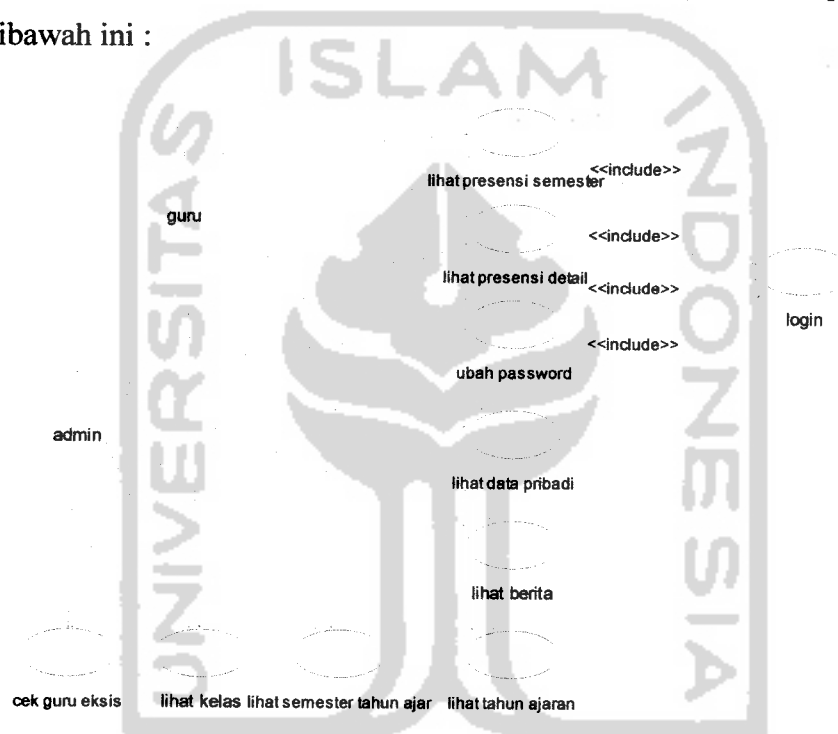
- Komunikasi dengan *client*

Penggunaan notasi dan simbol dalam diagram *Use Case* membuat pengembang lebih mudah berkomunikasi dengan klien – kliennya.

- Membuat test dari kasus-kasus secara umum

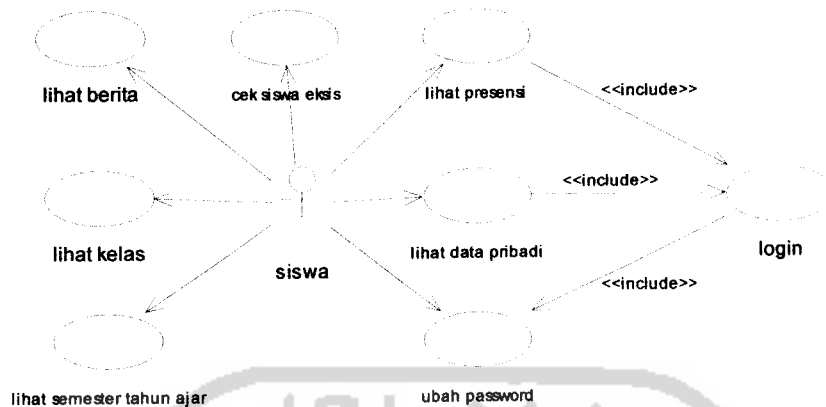
Kumpulan dari kejadian-kejadian untuk *Use Case* bisa dilakukan test kasus layak untuk kejadian-kejadian tersebut.

Use case Diagram aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA pada sisi admin atau guru, dapat dilihat pada gambar dibawah ini :



Gambar 3.2 Use Case Diagram Admin / Guru

Sedangkan pada sisi siswa dapat dilihat pada gambar 3.2 .



Gambar 3.3 Use Case Diagram Siswa

b. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi) (Dharwiyanti, 2003).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan *stereotype*)
2. Atribut
3. *Metoda*

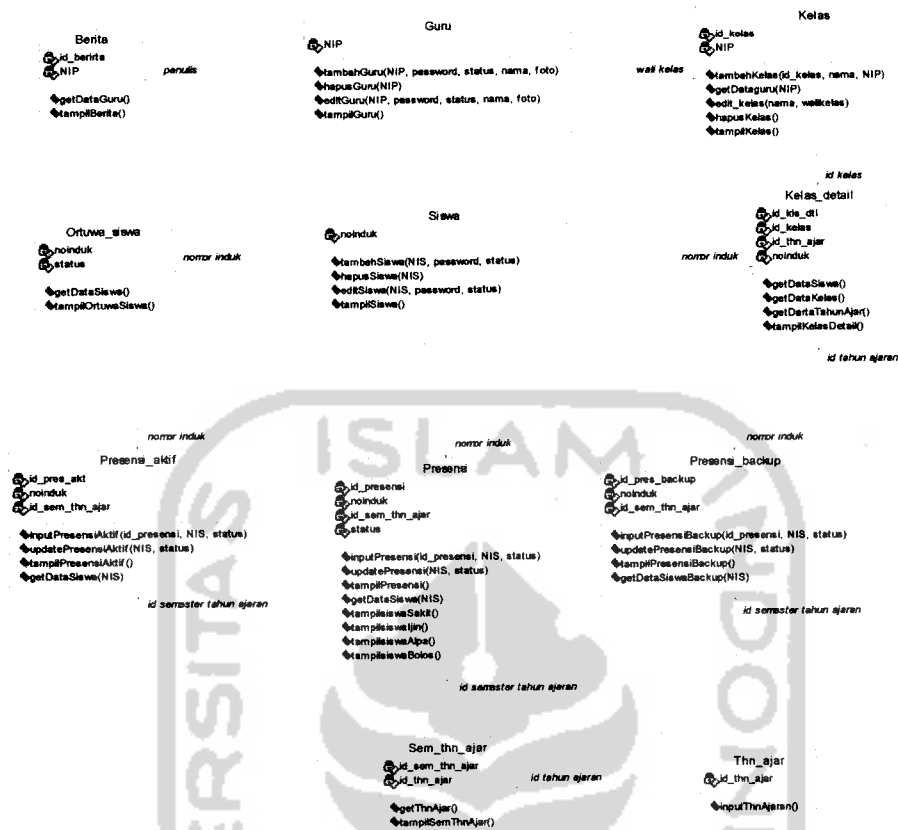
Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak- anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja

Pada *Class Diagram* Hubungan Antar *Class* dapat dikategorikan sebagai berikut :

1. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
3. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

Class Diagram aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA pada sisi admin atau guru, dapat dilihat pada gambar dibawah ini :



Gambar 3.4 Class Diagram Web Service Sistem Informasi Presensi Kesiswaan SMP / SMA

c. Sequence Diagram

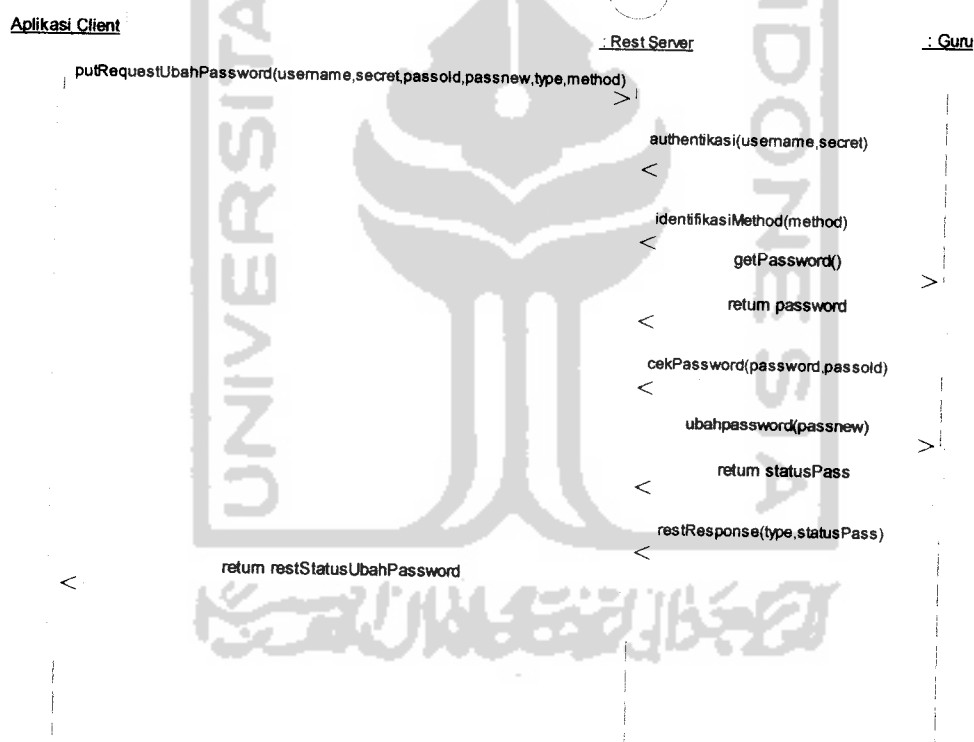
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi *horizontal* (objek – objek yang terkait) (Dharwiyanti, 2003).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal

dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Sequence Diagram fungsi ubah password pada user admin atau guru pada aplikasi web service sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



Gambar 3.5 Sequence Diagram Fungsi Ubah Password Admin / Guru

Pada *sequence diagram* ubah password admin / guru ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *passold* (password

lama), *passnew*(password baru), *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga *REST server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “PUT” atau “POST”(apabila dilakukan dengan *method* post maka pada parameter *request* ditambahkan parameter “_method” yang berisi “PUT”) maka proses akan dilanjutkan dengan mengambil data pribadi. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan *type* yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Sequence Diagram fungsi lihat data presensi semester pada user admin atau guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :

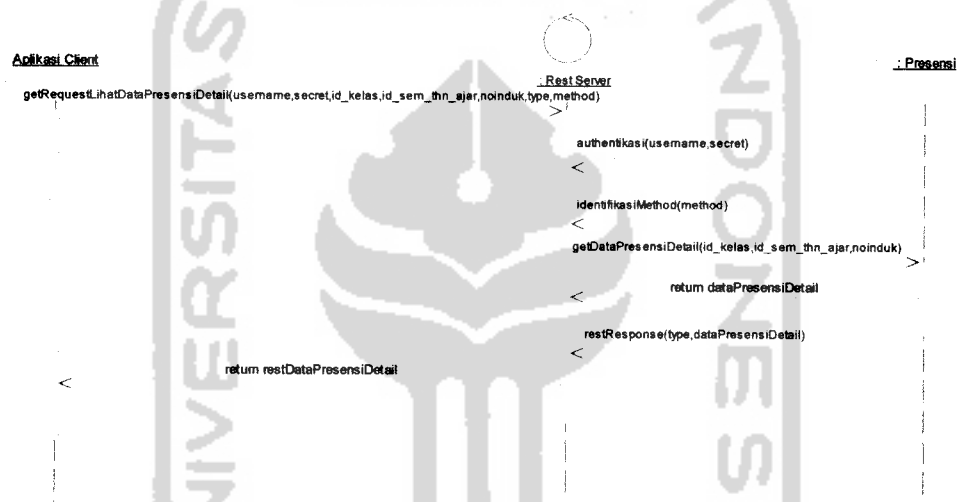


Gambar 3.6 Sequence Diagram Fungsi Lihat Presensi Semester Admin / Guru

Pada *sequence diagram* lihat presensi semester admin / guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *id_kelas*, *id_sem_thn_ajar* (id semester tahun ajaran), *type* dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga *REST*

server dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “GET” maka proses akan dilanjutkan dengan mengambil data presensi semester. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan type yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Sequence Diagram fungsi lihat data presensi detail pada user admin atau guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :

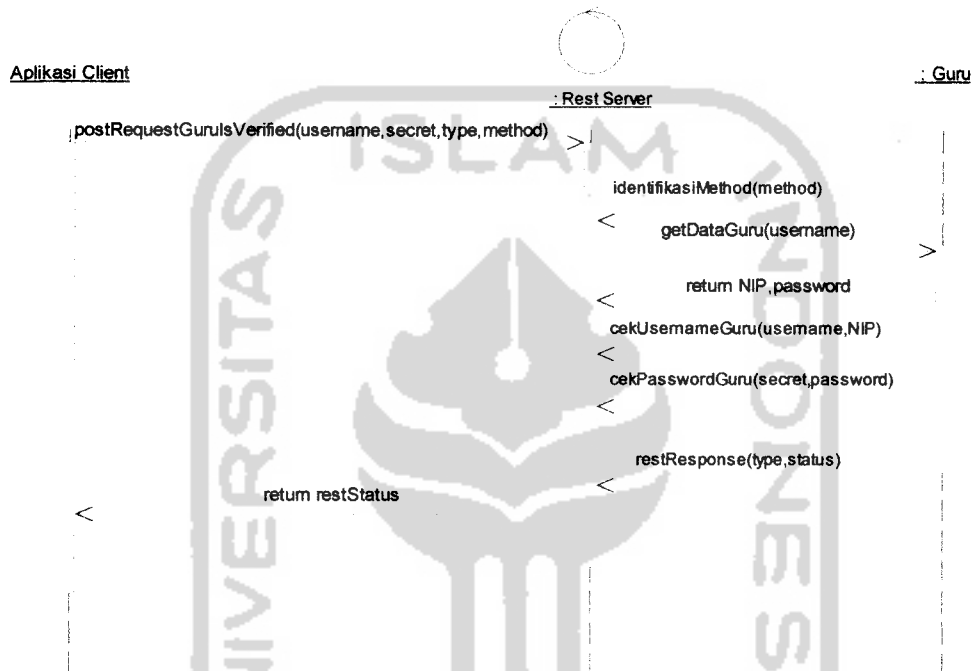


Gambar 3.7 Sequence Diagram Fungsi Lihat Presensi Detail Admin / Guru

Pada *sequence diagram* lihat presensi detail admin / guru ini aplikasi client mengirimkan *request* dengan menyertakan parameter *id_kelas*, *id_sem_thn_ajar* (id semester tahun ajaran), *noinduk*, *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga REST *server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “GET” maka proses akan dilanjutkan dengan mengambil data presensi detail. Setelah itu proses yang dilakukan selanjutnya

adalah membuat respon berdasarkan type yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Sequence Diagram fungsi verifikasi guru pada user guru atau admin pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



Gambar 3.8 Sequence Diagram Fungsi Verifikasi Admin / Guru

Pada *sequence diagram* verifikasi admin / guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter username, secret (password), type, dan *method* pengiriman *request* sehingga REST server dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “POST” maka proses akan dilanjutkan dengan mengambil data username dan password kemudian dilakukan pencocokan dengan parameter yang dikirimkan *client* (username, secret). Setelah itu proses yang dilakukan selanjutnya adalah

membuat respon berdasarkan type yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Sequence Diagram fungsi lihat data presensi pada user siswa pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



Gambar 3.9 Sequence Diagram Fungsi Lihat Data Presensi Siswa

Pada *sequence diagram* lihat presensi siswa pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *id_kelas*, *id_sem_thn_ajar* (id semester tahun ajaran), *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga REST *server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “GET” maka proses akan dilanjutkan dengan mengambil data presensi. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan *type* yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

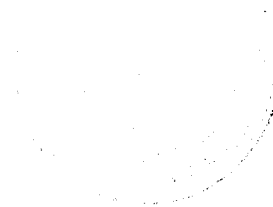
d. Activity Diagram

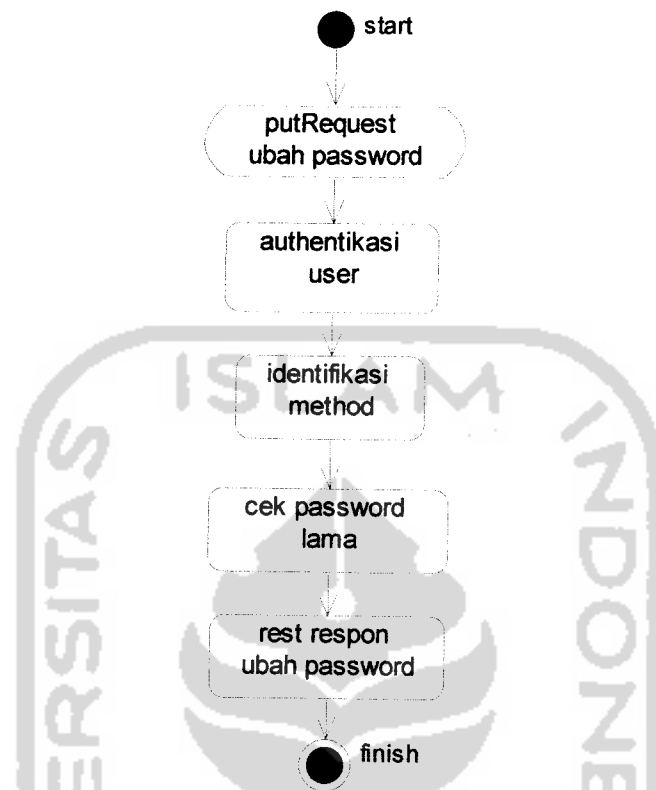
Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi dipicu oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih (Dharwiyanti, 2003).

Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan kebiasaan pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity Diagram fungsi ubah password pada user guru / admin pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



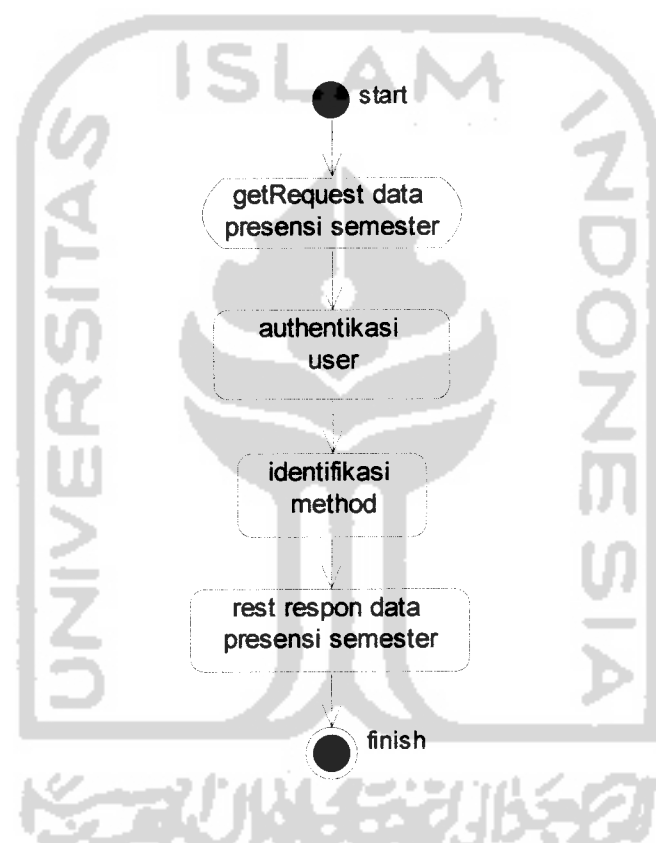


Gambar 3.10 Activity Diagram Fungsi Ubah Password Admin / Guru

Pada activity diagram fungsi ubah password admin / guru pada aplikasi web service sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *password* lama), *password* baru), *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai *authentikasi user* sehingga *REST server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan *identifikasi method* yang digunakan. Apabila *method* yang digunakan adalah “PUT” atau “POST” (apabila *method* yang digunakan adalah “POST” maka diperlukan parameter tambahan “_method” yang berisi “PUT”) maka proses akan dilanjutkan dengan mengambil data *NIP* dan *password* yang selanjutnya akan dicocokkan dengan data yang dikirimkan

client (passold). Apabila password lama cocok maka akan dilakukan perubahan password. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan type yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Activity Diagram fungsi lihat presensi semester pada user guru / admin pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



Gambar 3.11 Activity Diagram Fungsi Lihat Presensi Semester Admin / Guru

Pada *activity diagram* lihat data presensi semester admin / guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *id_kelas*, *id_sem_thn_ajar* (id semester tahun ajaran), *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga REST

server dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “GET” maka proses akan dilanjutkan dengan mengambil data presensi semester admin / guru berdasarkan *id_kelas* dan *id_sem_thn_ajar*. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan type yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Activity Diagram fungsi lihat presensi detail pada user guru / admin pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :

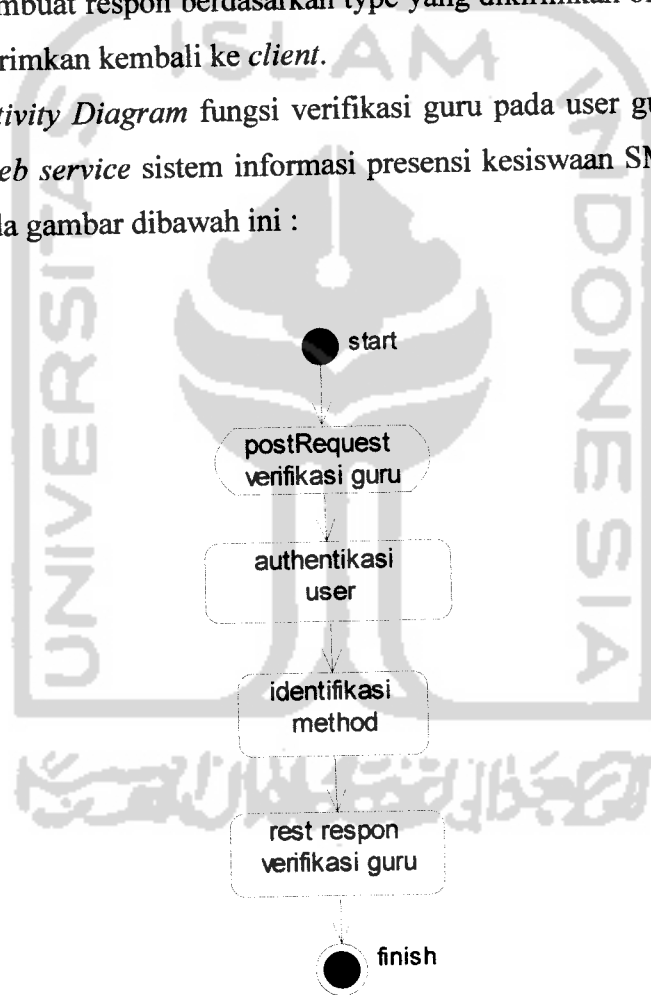


Gambar 3.12 Activity Diagram Fungsi Lihat Presensi Detail Admin / Guru

Pada *activity diagram* lihat data presensi detail admin / guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *id_kelas*,

id_sem_thn_ajar (id semester tahun ajaran), *noinduk*, *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga REST *server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “GET” maka proses akan dilanjutkan dengan mengambil data presensi semester admin / guru berdasarkan *noinduk*, *id_kelas*, dan *id_sem_thn_ajar*. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan *type* yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

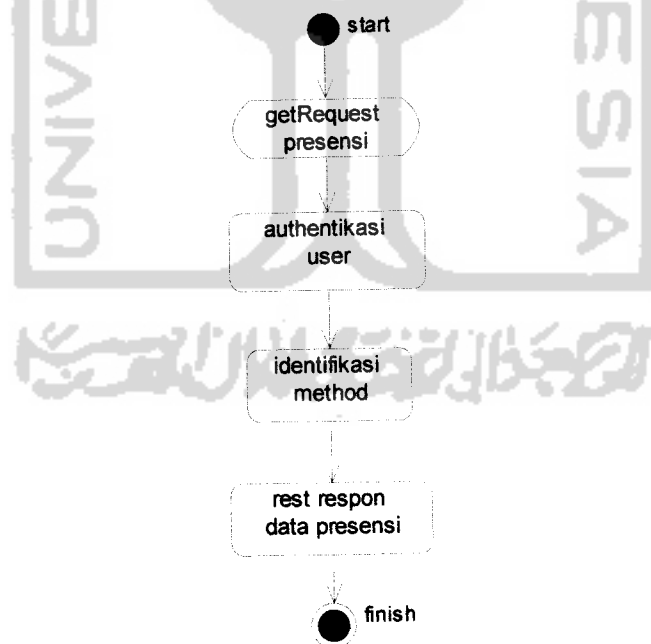
Activity Diagram fungsi verifikasi guru pada user guru / admin pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



Gambar 3.13 Activity Diagram Fungsi Verifikasi Admin / Guru

Pada *activity diagram* verifikasi admin / guru pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *username*, *secret* (*password*), *type*, dan *method* pengiriman sehingga *REST server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “POST” maka proses akan dilanjutkan dengan mengambil data *NIP* dan *password* admin / guru yang kemudian dicocokkan dengan parameter inputan *client* (*username* dan *secret*). Setelah itu proses yang selanjutnya adalah membuat respon berdasarkan *type* yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

Activity Diagram fungsi lihat presensi pada user siswa pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA, dapat dilihat pada gambar dibawah ini :



Gambar 3.14 Activity Diagram Fungsi Lihat Presensi Siswa

Pada *activity diagram* lihat data presensi siswa pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA ini aplikasi *client* mengirimkan *request* dengan menyertakan parameter *id_kelas*, *id_sem_thn_ajar* (id semester tahun ajaran), *type*, dan *method* pengiriman *request* beserta *username* dan *secret* sebagai autentikasi user sehingga *REST server* dapat mengenali *request*. Setelah *request* diterima maka akan dilakukan identifikasi *method* yang digunakan. Apabila *method* yang digunakan adalah “GET” maka proses akan dilanjutkan dengan mengambil data presensi siswa. Setelah itu proses yang dilakukan selanjutnya adalah membuat respon berdasarkan *type* yang dikirimkan oleh aplikasi *client* dan mengirimkan kembali ke *client*.

3.6.2 Entity Relasional Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam **basis data** berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu (rahmad, 2009) :

a. Entiti

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999: 30). Simbol dari entiti ini biasanya digambarkan dengan persegi panjang.

b. Atribut

Setiap entitas pasti mempunyai elemen yang disebut *atribut* yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar *atribut* diwakili oleh simbol elips.

c. **Hubungan / Relasi**

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Relasi dapat digambarkan sebagai berikut :

Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu **basis data** yaitu satu ke satu (*one to one*), satu ke banyak (*one to many*), dan banyak ke banyak (*many to many*).

Dalam pembuatan aplikasi web service sistem informasi presensi kesiswaan SMP / SMA ini dapat dibuat ERD sebagai berikut :



3.6.3 Perancangan Basis Data

Pada perancangan basis data / database dalam website Sistem Informasi Presensi Kesiswaan ini menggunakan MySQL database. Pada Website ini terdapat 12 tabel yang akan digunakan pada aplikasi *web service* sistem informasi presensi kesiswaan SMP / SMA antara lain :

3.6.3.1 Tabel Guru

Tabel ini digunakan untuk menyimpan semua data guru yang ada di sekolah.

Tabel 3.1 Tabel Guru

<i>Nama Kolom</i>	<i>Tipe Data</i>	<i>Keterangan</i>
nip	varchar(25)	primary key
password	varchar(50)	
status	enum('1','2','3')	
nama	varchar(255)	
gambar	varchar(25)	
tpt_lahir	varchar(255)	
tgl_lahir	date	
alamat	varchar(255)	
kodepos	varchar(10)	
pangjab	varchar(50)	
pendidikan	varchar(50)	
propinsi	varchar(50)	
kota	varchar(50)	
agama	varchar(25)	
kelamin	enum('L','P')	

3.6.3.2 Tabel Presensi

Tabel ini digunakan untuk menyimpan data presensi siswa yang ada di sekolah.

Tabel 3.2 Tabel Presensi

<i>Nama Kolom</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id_presensi	int(11)	primary key
noinduk	varchar(10)	foreign key
id_sem_thn_ajar	int(5)	foreign key
status	enum('1','2','3','4')	
tgl_update	date	

3.6.3.3 Tabel Presensi Aktif

Tabel ini digunakan untuk menyimpan data presensi total siswa yang aktif di sekolah.

Tabel 3.3 Tabel Presensi Aktif

<i>Nama Kolom</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id_pres_akt	int(11)	primary key
noinduk	varchar(10)	foreign key
id_sem_thn_ajar	int(5)	foreign key
hadir	int(5)	
sakit	int(5)	
ijin	int(5)	
alfa	int(5)	
tgl_update	date	

3.6.3.4 Tabel Presensi Backup

Tabel ini digunakan untuk menyimpan data presensi siswa yang telah lulus dari sekolah.

Tabel 3.4 Tabel Presensi Backup

<i>Nama Kolom</i>	<i>Tipe Data</i>	<i>Keterangan</i>
id_pres_akt	int(11)	primary key
noinduk	varchar(10)	foreign key
id_sem_thn_ajar	int(5)	foreign key
hadir	int(5)	
sakit	int(5)	
ijin	int(5)	
alfa	int(5)	
tgl_update	date	

3.6.4 Kinerja yang harus dipenuhi

Aplikasi *web service* sistem informasi presensi yang dibangun, diharapkan dapat memenuhi kebutuhan-kebutuhan antara lain :

- a. Mampu memberikan layanan kepada admin, guru dan siswa untuk mempermudah dalam mengakses data presensi siswa.
- b. Mampu menjembatani aplikasi *server* dengan aplikasi *client* yang beragam dari segi bahasa pemrograman maupun *platform*.

3.6.5 Keluaran Aplikasi (Output)

Berdasarkan proses yang dilakukan dalam Aplikasi *Web Service* Sistem Informasi Presensi Kesiswaan SMP / SMA ini, maka akan diperoleh berupa informasi sebagai berikut :

1. Informasi presensi siswa baik secara umum maupun mendetail.
2. Informasi berita dari sekolah.
3. Informasi data pribadi pengguna.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Batasan implementasi

4.1.1 Perangkat Lunak yang dipakai

Perangkat lunak yang dibutuhkan untuk membangun *web service* sistem informasi presensi kesiswaan ini adalah :

- *NetBeans IDE 6.7.1*
NetBeans IDE 6.7.1 menyediakan fasilitas yang memungkinkan untuk menyusun sebuah program dengan memasang objek – objek grafis dalam sebuah *form*.
- *MySQL*
MySQL merupakan salah satu aplikasi yang mengolah *database*. *MySQL* banyak digunakan untuk mendukung *database* yang bersifat *client/server*. *MySQL* juga dapat mengolah *database* dalam jumlah besar.
- *PHP*
PHP merupakan salah satu bahasa pemrograman berbasis web. *PHP* banyak digunakan dalam pembuatan web karena simpel dan mampu bersaing dengan bahasa pemrograman berbasis web yang lain. Selain itu dukungan framework dalam bahasa ini sudah sangat baik.
- *Zend Framework 1.9.5*
Zend merupakan framework dengan bahasa pemrograman *PHP* yang cukup memenuhi dari fungsi – fungsi yang dimilikinya. *Zend framework* telah mendukung pembuatan *REST web service* dan telah teruji dalam framework *PHP*.

4.2 Implementasi Sistem Web Service

Pada implementasi API ini akan diperlihatkan *response header* dan *response body* dari *web service* presensi yang diakses melalui protokol HTTP.

4.2.1 Implementasi fungsi ubah password admin / guru

Fungsi ubah password user admin / guru digunakan untuk mengubah password admin / guru yang ada pada *web service* presensi kesiswaan. Untuk keterangan lebih jelasnya dapat dilihat dibawah ini.

URL

`http://sipres.ac.id/guru/ubahPassword`

Formats

xml, json

HTTP Method

PUT / POST

Apabila menggunakan *method* "POST" maka dibutuhkan parameter input tambahan "_method" dengan isi "PUT".

Dibutuhkan autentikasi

Ya (username & secret)

Dibutuhkan username dan secret / *password* sebagai autentikasi *web service*.

Parameter

type

"type" merupakan parameter yang tidak selalu dibutuhkan oleh server. Parameter ini berfungsi untuk mengatur respon dari web service apakah akan dalam format XML atau JSON. Pada web service informasi presensi ini telah ditetapkan secara default data yang dikembalikan oleh admin adalah XML apabila parameter type kosong atau salah.

passold

"passold" merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan *password* lama dari user. Ini akan digunakan sebagai validasi penggantian *password*.

passnew

“passnew” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan *password* baru dari user. Ini akan digunakan sebagai pengganti *password* yang lama.

RESPON

Header respon XML

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 17:58:41 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.81 mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 175
Connection: close
Content-type: text/xml
```

Header respon JSON

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 18:02:20 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.81 mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 94
Connection: close
Content-type: application/json
```

XML respon

Password berhasil dirubah

```
<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>password berhasil diubah!</response>
  <status>success</status>
</rest>
```

Password gagal dirubah

```
<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>password gagal diubah!</response>
  <status>failed</status>
</rest>
```

Password lama tidak sama

```
<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>password lama tidak sama!</response>
  <status>failed</status>
</rest>
```

JSON respon**Password berhasil dirubah**

```
{
  "rest"      : "arieformatic",
  "app"       : "GURU",
  "response"  : "password berhasil diubah!",
  "status"    : "success"
}
```

Password gagal dirubah

```
{
  "rest"      : "arieformatic",
  "app"       : "GURU",
  "response"  : "password gagal diubah!",
  "status"    : "failed"
}
```

Password lama tidak sama

```
{
  "rest"      : "arieformatic",
  "app"       : "GURU",
  "response"  : "password lama tidak sama!",
  "status"    : "failed"
}
```

4.2.2 Implementasi fungsi lihat presensi semester admin / guru

Fungsi lihat presensi semester user siswa digunakan untuk mengakses data presensi semester yang ada pada *web service* presensi kesiswaan. Untuk keterangan lebih jelasnya dapat dilihat dibawah ini.

URL

`http://sipres.ac.id/guru/presensiSemester`

Formats

xml, json

HTTP Method

GET

Dibutuhkan autentikasi

Ya (username & secret)

Dibutuhkan username dan secret / *password* sebagai autentikasi *web service*.

Parameter

type

“type” merupakan parameter yang tidak selalu dibutuhkan oleh *server*. Parameter ini berfungsi untuk mengatur respon dari *web service* apakah akan dalam format XML atau JSON. Pada *web service* informasi presensi ini telah ditetapkan secara default data yang dikembalikan oleh admin adalah XML apabila parameter type kosong atau salah.

kelas

“kelas” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan id kelas. Id kelas ini diperoleh dari fungsi lihat kelas. Id kelas ini digunakan untuk menentukan presensi semester yang akan ditampilkan.

sem_thn_ajar

“sem_thn_ajar” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan id semester tahun ajaran. Id semester tahun ajaran ini diperoleh dari fungsi lihat semester tahun ajaran. Id semester tahun ajaran ini digunakan untuk menentukan presensi semester yang akan ditampilkan.

RESPON

Header respon XML

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 22:01:53 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 3270
Connection: close
Content-type: text/xml
```

Header respon JSON

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 22:02:29 GMT
```

```

Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.81 mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 2308
Connection: close
Content-type: application/json

```

XML respon

```

<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>
    <siswa noinduk="1001">
      <nama>siswa 1001</nama>
      <hadir>4</hadir>
      <sakit>0</sakit>
      <ijin>3</ijin>
      <alfa>2</alfa>
      <tglupdate>04 April 2010, 15:39</tglupdate>
    </siswa>
    <siswa noinduk="1002">
      <nama>siswa 1002</nama>
      <hadir>1</hadir>
      <sakit>3</sakit>
      <ijin>1</ijin>
      <alfa>4</alfa>
      <tglupdate>04 April 2010, 15:39</tglupdate>
    </siswa>
    <siswa noinduk="1003">
      <nama>siswa 1003</nama>
      <hadir>2</hadir>
      <sakit>2</sakit>
      <ijin>3</ijin>
      <alfa>2</alfa>
      <tglupdate>04 April 2010, 15:39</tglupdate>
    </siswa>
  </response>
  <status>success</status>
</rest>

```

JSON respon

```

{
  "rest"      : "arieformatic",
  "app"       : "GURU",
  "response"  :
    [ {
      "noinduk" : "1001",
      "nama"    : "siswa 1001",
      "hadir"   : 4,
      "sakit"   : 0,

```



```

        "ijin"      : 3,
        "alfa"     : 2,
        "tglupdate" : "04 April 2010, 15:39"
    }, {
        "noinduk"   : "1002",
        "nama"      : "siswa 1002",
        "hadir"     : 1,
        "sakit"     : 3,
        "ijin"      : 1,
        "alfa"     : 4,
        "tglupdate" : "04 April 2010, 15:39"
    } ],
    "status" : "success"
}

```

4.2.3 Implementasi fungsi lihat presensi detail admin / guru

Fungsi lihat presensi detail user siswa digunakan untuk mengakses data presensi detail yang ada pada *web service* presensi kesiswaan. Untuk keterangan lebih jelasnya dapat dilihat dibawah ini.

URL

`http://sipres.ac.id/guru/presensiDetail`

Formats

xml, json

HTTP Method

GET

Dibutuhkan autentikasi

Ya (username & secret)

Dibutuhkan username dan secret / *password* sebagai autentikasi *web service*.

Parameter

type

“type” merupakan parameter yang tidak selalu dibutuhkan oleh *server*. Parameter ini berfungsi untuk mengatur respon dari *web service* apakah akan dalam format XML atau JSON. Pada *web service* informasi presensi ini telah ditetapkan secara default data yang dikembalikan oleh admin adalah XML apabila parameter type kosong atau salah.

noinduk

“noinduk” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan nomor induk siswa.

status

“status” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan status dari presensi yang ingin ditampilkan. Keterangan dari isi parameter ini yaitu :

- 1 : Hadir
- 2 : Sakit
- 3 : Ijin
- 4 : Alfa

sem_thn_ajar

“sem_thn_ajar” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan id semester tahun ajaran. Id semester tahun ajaran ini diperoleh dari fungsi lihat semester tahun ajaran. Id semester tahun ajaran ini digunakan untuk menentukan presensi semester yang akan ditampilkan.

RESPON

Header respon XML

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 22:14:09 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 442
Connection: close
Content-type: text/xml
```

Header respon JSON

```
HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 22:14:46 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 284
Connection: close
```

Content-type: application/json

XML respon

```
<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>
    <siswa noinduk="1001">
      <tglupdate>Tuesday, 04 May 2010</tglupdate>
    </siswa>
    <siswa noinduk="1001">
      <tglupdate>Wednesday,           05           May
2010</tglupdate>
    </siswa>
    <siswa noinduk="1001">
      <tglupdate>Friday, 07 May 2010</tglupdate>
    </siswa>
    <siswa noinduk="1001">
      <tglupdate>Friday, 28 May 2010</tglupdate>
    </siswa>
  </response>
  <status>success</status>
</rest>
```

JSON respon

```
{
  "rest"      : "arieformatic",
  "app"       : "GURU",
  "response"  : [ {
    "noinduk" : "1001",
    "tglupdate" : "Tuesday, 04 May 2010"
  }, {
    "noinduk" : "1001",
    "tglupdate" : "Wednesday, 05 May 2010"
  }, {
    "noinduk" : "1001",
    "tglupdate" : "Friday, 07 May 2010"
  }, {
    "noinduk" : "1001",
    "tglupdate" : "Friday, 28 May 2010"
  } ],
  "status"    : "success"
}
```

4.2.4 Implementasi fungsi verifikasi admin / guru

Fungsi verifikasi admin / guru digunakan untuk membuktikan bahwa *user* admin / guru memang benar – benar merupakan *user* dari *web service* presensi kesiswaan. Untuk keterangan lebih jelasnya dapat dilihat dibawah ini.

URL

`http://sipres.ac.id/auth/guruIsVerified`

Formats

`xml, json`

HTTP Method

`POST`

Dibutuhkan autentikasi

Tidak

Parameter

`type`

“`type`” merupakan parameter yang tidak selalu dibutuhkan oleh *server*. Parameter ini berfungsi untuk mengatur respon dari *web service* apakah akan dalam format XML atau JSON. Pada *web service* informasi presensi ini telah ditetapkan secara default data yang dikembalikan oleh admin adalah XML apabila parameter `type` kosong atau salah.

`username`

“`username`” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan nomor induk dari user admin / guru.

`secret`

“`secret`” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan password dari user guru.

RESPON

Header respon XML

`HTTP/1.1 200 OK`

`Date: Wed, 02 Jun 2010 23:27:29 GMT`

```

Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.81 mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 156
Connection: close
Content-type: text/xml

```

Header respon JSON

```

HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 23:28:11 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.81 mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 75
Connection: close
Content-type: application/json

```

XML respon

Valid

```

<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>valid</response>
  <status>success</status>
</rest>

```

Tidak valid

```

<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>GURU</app>
  <response>password salah</response>
  <status>failed</status>
</rest>

```

JSON respon

Valid

```

{
  "rest"      : "arieformatic",
  "app"       : "GURU",
  "response"  : "valid",
  "status"    : "success"
}

```

Tidak Valid

```

{

```

```

    "rest"      : "arieformatic",
    "app"       : "GURU",
    "response"  : "password salah",
    "status"    : "failed"
  }

```

4.2.5 Implementasi fungsi lihat presensi siswa

Fungsi lihat presensi user siswa digunakan untuk mengakses data presensi semester yang ada pada *web service* presensi kesiswaan. Untuk keterangan lebih jelasnya dapat dilihat dibawah ini.

URL

```
http://sipres.ac.id/siswa/presensi
```

Formats

xml, json

HTTP Method

GET

Dibutuhkan autentikasi

Ya (username & secret)

Dibutuhkan username dan secret / *password* sebagai autentikasi *web service*.

Parameter

type

“type” merupakan parameter yang tidak selalu dibutuhkan oleh *server*. Parameter ini berfungsi untuk mengatur respon dari *web service* apakah akan dalam format XML atau JSON. Pada *web service* informasi presensi ini telah ditetapkan secara default data yang dikembalikan oleh admin adalah XML apabila parameter type kosong atau salah.

sem_thn_ajar

“sem_thn_ajar” merupakan parameter yang dibutuhkan oleh *server*. Parameter ini berisikan id semester tahun ajaran. Id semester tahun ajaran ini diperoleh dari fungsi lihat semester tahun ajaran. Id semester tahun ajaran ini digunakan untuk menentukan presensi semester yang akan ditampilkan.

RESPON**Header respon XML**

```

HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 22:58:13 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 307
Connection: close
Content-type: text/xml

```

Header respon JSON

```

HTTP/1.1 200 OK
Date: Wed, 02 Jun 2010 23:00:09 GMT
Server: Apache/2.2.14 (Win32) DAV/2 mod_ssl/2.2.14
OpenSSL/0.9.8l mod_autoindex_color PHP/5.3.1 mod_apreq2-
20090110/2.7.1 mod_perl/2.0.4 Perl/v5.10.1
X-powered-by: PHP/5.3.1
Content-length: 181
Connection: close
Content-type: application/json

```

XML respon

```

<?xml version="1.0" encoding="UTF-8"?>
<rest generator="arieformatic" version="1.0">
  <app>SISWA</app>
  <response>
    <siswa noinduk="1001">
      <nama>siswa 1001</nama>
      <hadir>4</hadir>
      <sakit>0</sakit>
      <ijin>3</ijin>
      <alfa>2</alfa>
      <tglupdate>04 April 2010, 15:39</tglupdate>
    </siswa>
  </response>
  <status>success</status>
</rest>

```

JSON respon

```

{
  "rest"      : "arieformatic",
  "app"       : "SISWA",
  "response"  :
    [ {
      "noinduk" : "1001",
      "nama"    : "siswa 1001",
      "hadir"   : 4,

```

```

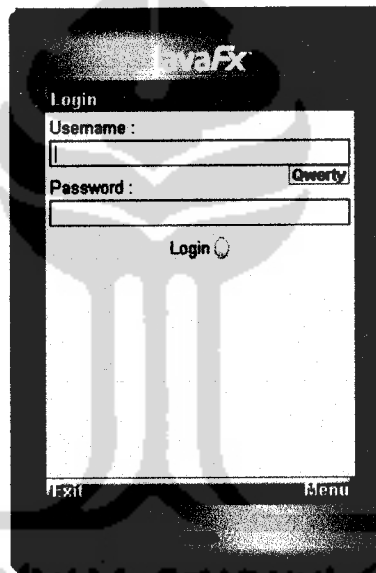
        "sakit"      : 0,
        "ijin"      : 3,
        "alfa"      : 2,
        "tglupdate" : "04 April 2010, 15:39"
    } ],
    "status"      : "success"
}

```

4.3 Implementasi Sistem Web Service Client (Mobile Device)

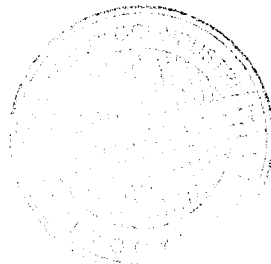
4.3.1 Implementasi fungsi verifikasi user pada client

Halaman berikut merupakan implementasi dari pengaksesan fungsi verifikasi user.



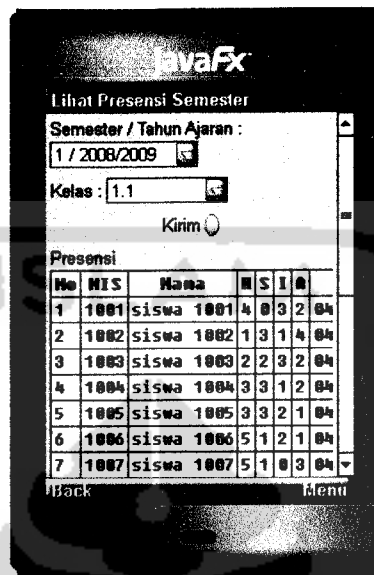
Gambar 4.1 Implementasi Fungsi Verifikasi User Pada Aplikasi Client Java ME

Pada aplikasi *client* berbasis java ME ini digunakan fungsi verifikasi user. Fungsi ini digunakan untuk fungsi login pada aplikasi client. Pada aplikasi client ini digunakan JSON sebagai format pertukaran data.



4.3.2 Implementasi fungsi lihat presensi semester pada client

Halaman berikut merupakan implementasi dari pengaksesan fungsi lihat presensi semester.

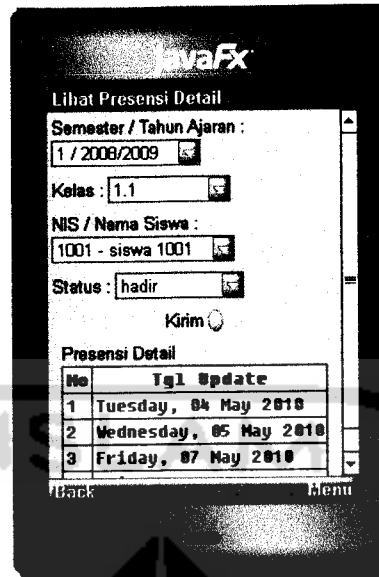


Gambar 4.2 Implementasi Fungsi Lihat Presensi Semester Pada Aplikasi Client Java ME

Pada aplikasi *client* berbasis java ME ini digunakan fungsi lihat presensi semester sebagai fungsi utama dan fungsi lihat semester tahun ajaran, lihat kelas sebagai fungsi pendukung. fungsi ini digunakan untuk fungsi lihat presensi semester secara keseluruhan berdasarkan semester dan tahun ajaran serta kelas siswa. Pada aplikasi *client* ini digunakan JSON sebagai format pertukaran data.

4.3.3 Implementasi fungsi lihat presensi detail pada client

Halaman berikut merupakan implementasi dari pengaksesan fungsi lihat presensi detail.

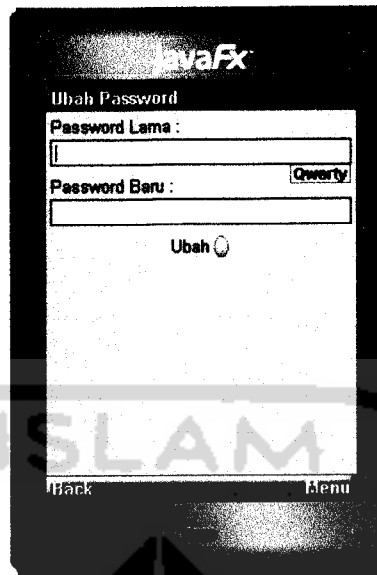


Gambar 4.3 Implementasi Fungsi Lihat Presensi Detail Pada Aplikasi Client Java ME

Pada aplikasi *client* berbasis java ME ini digunakan fungsi lihat presensi detail sebagai fungsi utama dan *API* lihat semester tahun ajaran, lihat kelas sebagai fungsi pendukung. fungsi ini digunakan untuk fungsi lihat presensi secara mendetail berdasarkan NIS / nomor induk siswa semester dan tahun ajaran serta kelas siswa. Pada aplikasi *client* ini digunakan JSON sebagai format pertukaran data.

4.3.4 Implementasi fungsi ubah password pada client

Halaman berikut merupakan implementasi dari pengaksesan fungsi ubah password pada *client*.

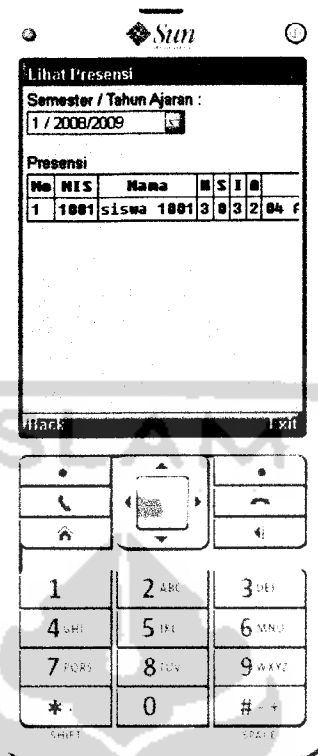


Gambar 4.4 Implementasi Fungsi Ubah Password Pada Aplikasi Client Java ME

Pada aplikasi *client* berbasis java ME ini digunakan fungsi ubah password . Fungsi ini digunakan untuk fungsi ubah password. *User* diharuskan untuk menginputkan username dan password sebagai parameter input ke web service. Pada aplikasi *client* ini digunakan JSON sebagai format pertukaran data.

4.3.5 Implementasi fungsi lihat presensi pada client

Halaman berikut merupakan implementasi dari pengaksesan fungsi ubah password pada *client*.



Gambar 4.5 Implementasi Fungsi Lihat Presensi Pada Aplikasi Client Java ME

Pada aplikasi *client* berbasis java ME ini digunakan fungsi lihat presensi sebagai fungsi utama dan *API* lihat semester tahun ajaran sebagai fungsi pendukung. Fungsi ini digunakan untuk fungsi lihat presensi secara mendetail berdasarkan NIS / nomor induk siswa yang login, semester dan tahun ajaran. Pada aplikasi *client* ini digunakan JSON sebagai format pertukaran data.

4.4 Implementasi Sistem Input Data Presensi

4.4.1 Implementasi penginputan data presensi pada sistem informasi presensi kesiswaan SMP / SMA

Halaman berikut merupakan implementasi penambahan data presensi melalui sistem informasi presensi kesiswaan SMP / SMA yang nantinya akan digunakan pada *web service* pada fungsi lihat presensi semester, lihat presensi detail maupun lihat presensi.

Tambah Data Presensi [manual]

Semester / Tahun Ajaran : 1 / (2008/2009) (aktif) ▾
 Kelas : 1.1 ▾
 Tanggal : 11 Juni 2010 ▾ 12:00:00 ▾

Tampil

No	NIS	Nama	H	S	I	A
1	1001	siswa 1001	•			
2	1002	siswa 1002	•			
3	1003	siswa 1003	•			
4	1004	siswa 1004	•			
5	1005	siswa 1005	•			
6	1006	siswa 1006	•			
7	1007	siswa 1007	•			
8	1008	siswa 1008	•			
9	1009	siswa 1009	•			
10	1010	siswa 1010	•			
11	1011	siswa 1011	•			
12	1012	siswa 1012	•			
13	1013	siswa 1013	•			
14	1014	siswa 1014	•			
15	1015	siswa 1015	•			
16	1016	siswa 1016	•			
17	1017	siswa 1017	•			
18	1018	siswa 1018	•			
19	1019	siswa 1019	•			
20	1020	siswa 1020	•			

input

Gambar 4.6 implementasi Penambahan Data Presensi Pada Sistem Informasi Presensi Kesiswaan SMP / SMA

BAB V

PENUTUP

5.1 Kesimpulan

Dari perancangan dan implementasi Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan Web Service ada beberapa kesimpulan yang dapat dikemukakan sebagai hasil dari penelitian, yaitu :

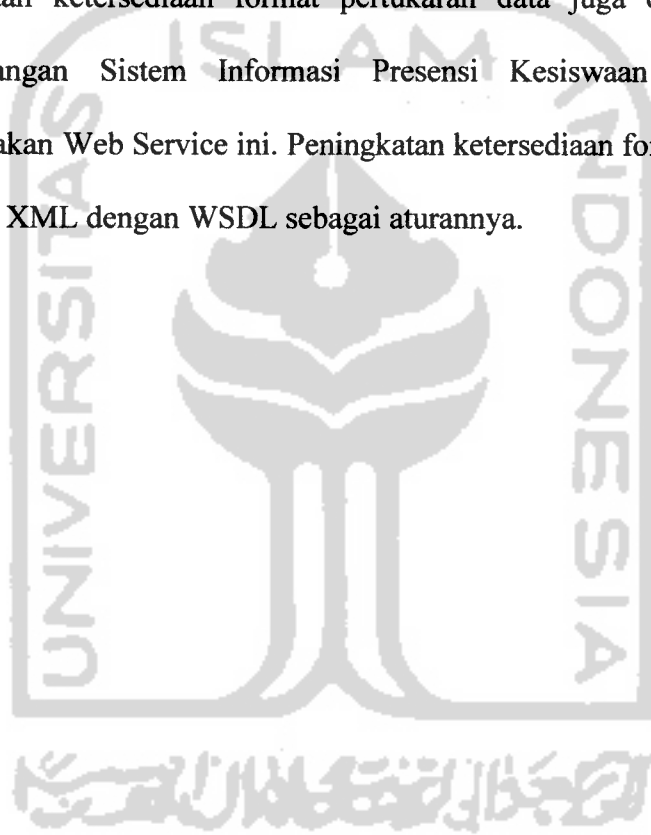
1. Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan *Web Service* telah mengimplementasikan keamanan akses modul – modul yang ada sehingga dapat menjamin keamanan data yang ada pada *server*.
2. Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan Web Service telah menyediakan 2 format pertukaran data yaitu JSON dan XML yang sederhana dalam aturannya / ringkas sehingga akan mengurangi besar *bandwith* dalam pengaksesan dan memudahkan pengembang dalam pengolahan datanya.

5.2 Saran

Saran yang diberikan untuk mendukung kelancaran Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan Web Service sebagai berikut :

1. Perlu adanya peningkatan dan pengembangan lebih lanjut terhadap Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan Web Service, tidak hanya sistem informasi presensi kesiswaan sekolah tetapi pada proses – proses yang lain seperti pembayaran dan nilai.

2. Peningkatan keamanan data secara berlapis atau penggunaan metode keamanan yang berbeda dapat digunakan dalam pengembangan web service ini. Penggunaan metode keamanan melalui enkripsi data kiriman client seperti pada aplikasi web service jejaring sosial (facebook) dapat dilakukan dalam membuat keragaman proses pengamanan web service.
3. Peningkatan ketersediaan format pertukaran data juga diperlukan dalam pengembangan Sistem Informasi Presensi Kesiswaan SMP / SMA menggunakan Web Service ini. Peningkatan ketersediaan format seperti XML atom atau XML dengan WSDL sebagai aturannya.



DAFTAR PUSTAKA

- Admin. (den 17 November 2008). *JSON Javascript Object Notation*. Hämtat från IT telkom:
http://www.ittelkom.ac.id/library/index.php?view=article&catid=15%3Apemrosesan-sinyal&id=304%3Ajson&option=com_content&Itemid=15 den 9 April 2010
- Costello, R. L. (2010). *REST Web Service*. Hämtat från Xfront:
<http://www.xfront.com/REST-Web-Services.html> den 2 Juny 2010
- Cowan, J. (2005). an introduction to building Web Service. *Restful Web Service* , 25.
- Dharwiyanti, S. (2003). UML. *Pengantar Unified Modeling Language (UML)* .
- Fielding, R. (2010). HTTP/1.1. *URIs, Connections, and Message Parsing* .
- Honeys, M. (den 25 September 2008). *JSON Javascript Object Notation*. Hämtat från IT telkom:
http://www.ittelkom.ac.id/library/index.php?view=article&catid=20%3Ainformatika&id=176%3Ajson&option=com_content&Itemid=15 den 9 April 2010
- Magaline, F. (2009). Konsep Dasar Sistem Informasi. *Sistem Informasi* , 1.
- Masykur, A. (den 14 Februari 2008). *JSON vs XML*. Hämtat från Ahmad Masykur:
<http://www.masykur.web.id/post/JSON-vs-XML.aspx> den 9 April 2010
- Potts, S. (2003). *Teach Yourself Web services in 24 hours*. Sams.
- Richardson, L. (2007). *RESTful Web Service*. Safari books online.

Sang, S. (den 3 10 2007). *Introduction to JSON (JavaScript Object Notation)*. Hämtat från javapassion: <http://www.javapassion.com/ajax/JSON.pdf> den 24 5 2010

Short, S. (2003). *Building XML Web Service for the Microsoft .net Platform*. Jakarta: PT. Elex Media Computindo.

