

IMPLEMENTASI RUMUS-RUMUS FISIKA SMA SEBAGAI WEB SERVICE

TUGAS AKHIR

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



Oleh:

Nama : Rayyan Muchamad Maretan

No. Mahasiswa : 05523222

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

2011

HALAMAN JUDUL

IMPLEMENTASI RUMUS-RUMUS FISIKA SMA SEBAGAI WEB SERVICE

TUGAS AKHIR

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



Oleh:

Nama : Rayyan Muchamad Maretan

No. Mahasiswa : 05523222

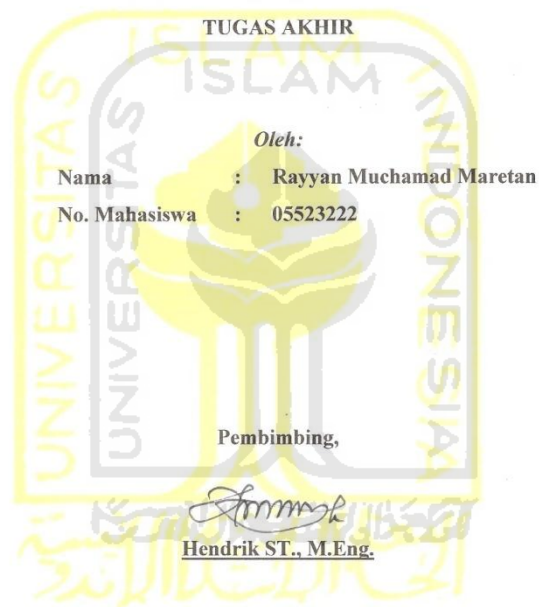
**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

LEMBAR PENGESAHAN PEMBIMBING

LEMBAR PENGESAHAN PEMBIMBING

IMPLEMENTASI RUMUS-RUMUS FISIKA SMA
SEBAGAI WEB SERVICE



LEMBAR PENGESAHAN PENGUJI

LEMBAR PENGESAHAN PENGUJI

IMPLEMENTASI RUMUS-RUMUS FISIKA SMA SEBAGAI WEB SERVICE

TUGAS AKHIR

Oleh:

Nama : Rayyan Muchamad Maretan

No. Mahasiswa : 05523222

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 25 Maret 2011

Tim Penguji

Tanda Tangan

Hendrik ST., M.Eng.
Ketua

Syarif Hidayat S.Kom., MIT.
Anggota I

R. Teduh Dirgahayu ST., M.Sc., Ph.D.
Anggota II

Mengetahui,
Ketua Jurusan Teknik Informatika
Universitas Islam Indonesia



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

KATA PENGANTAR



Segala puji dan syukur bagi Allah Subhanahu Wa Ta'ala, tiada Ilah selain-Nya, yang seandainya pohon-pohon di bumi menjadi pena dan lautan menjadi tinta dan ditambahkan tujuh laut lagi niscaya tidaklah cukup untuk menuliskan ilmu-Nya.

Diantara kasih sayang Allah kepada hamba-Nya adalah dengan memberi sedikit ilmu-Nya yang dengan ilmu itu manusia belajar dan memanfaatkannya serta menuliskannya. Dan karena kasih sayang dan karunia-Nya, laporan penelitian dengan judul “IMPLEMENTASI RUMUS-RUMUS FISIKA SEBAGAI WEB SERVICE” ini dapat diselesaikan. Shalawat serta salam semoga tercurah kepada junjungan kita Rasulullah Muhammad Shallallahu ‘Alaihi Wa Sallam yang dengan bimbingannya, manusia khususnya umat islam menuntut ilmu lalu beramal dengan ilmu.

Laporan Tugas Akhir ini disusun sebagai salah satu syarat guna memperoleh gelar sarjana Teknik Informatika pada Universitas Islam Indonesia. Dan juga sebagai sarana untuk mempraktekan secara langsung ilmu dan teori yang telah diperoleh selama menjalani masa studi di Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.

Dan untuk merealisasikan sabda Rasulullah Shallallahu ‘Alaihi Wa Sallam yang artinya: *”Seorang belum merealisasikan rasa syukur kepada Allah selama ia tidak mampu bersyukur (berterimakasih) atas kebaikan orang lain terhadap dirinya.”* (HR. Ahmad, Abu Daud, Tirmidzi) maka penulis menyampaikan ucapan terima kasih dan penghargaan setinggi-tingginya atas bantuan, bimbingan, dukungan, dan do’a dari berbagai pihak yang ikut membantu demi kelancaran pelaksanaan Tugas Akhir ini. Untuk itu dengan segala kerendahan hati penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Ayahanda dan ibunda tercinta atas segala kebaikan, do'a dan pengorbanan untuk kehidupan kami. Ya Allah muliakanlah keduanya, ampunilah dosa-dosa mereka dan sayangilah mereka sebagaimana mereka menyayangi kami sewaktu kami masih kecil.
2. Kedua adikku, Rabie' dan Raami yang telah banyak memberi motivasi untuk cepat lulus. Dan seorang yang telah kuanggap kakak, Muhammad Saleh Badhel.
3. Bapak Prof. Dr. Edy Suandi Hamid M.Ec., selaku Rektor Universitas Islam Indonesia dan seluruh jajaran Rektorat Universitas Islam Indonesia.
4. Bapak Yudi Prayudi S.Si., M.Kom, selaku Ketua Jurusan Teknik Informatika. Terima kasih atas kemudahan dan dukungan yang telah diberikan.
5. Bapak Fathul Wahid ST., M.Sc, selaku Dosen Pembimbing pertama dan pemberi ide judul Tugas Akhir ini. Terima kasih atas masukan dan motivasi selama ini.
6. Bapak Hendrik ST., M.Eng selaku Dosen Pembimbing pengganti, saya belajar banyak dari semangat Bapak, terimakasih atas pengarahan, bimbingan, motivasi serta masukan selama pelaksanaan Tugas Akhir dan penulisan laporan.
7. Seluruh Dosen dan Staf Pengajar Jurusan Teknik Informatika, terimakasih atas ilmu dan pengetahuan yang telah diberikan semoga menjadi amal jariyah.
8. Para penghuni Asrama Putra Amanah, Abdullah Sulaiman Maretan –Tetep Teges!, Saleh Bafadal –Ana gak tahu daahh kalo gak ada nt, Umar Sami Bahaj –Jiron kamil, Jamal Kelib –Hi Jams Badegul! Always cool man!, Hafid Kelib –Fiid, njaluk gulone ho, Ikrimach Abbad –semoga bisulnya cepet sembuh. Untuk dua orang terakhir, terima kasih banyak atas *printer*-nya. Dan kepada shohibul amanah yang lain, Puja, Bayu, Faisal, Ryan –Terima kasih telah membayar listrik tepat waktu.
9. Ibu Purwanti Wakos, selaku ibu kos yang telah memberi banyak kemudahan dalam urusan pembayaran dan lain-lain. Terima kasih telah memberi dukungan Ibu.

10. Shohibul tukar pikiran dan mantan “Penghuni Jalur Gaza” (jare Salim): Robiq “Naas Hasan” Shammakh –Syukran sudah ngoprak-ngoprak tapi kadang yo nggodo, Hanif “Naas Thayyib” Hatrash –madza taf’al?, Salim Shammakh – Baarakallahu laka wa baaraka ‘alayka wa jama’a baynakuma fii khair... terima kasih atas ide, bantuan, dan pengajaran tentang *Web Service*, jazakallahu khair. Muhammad Rashid Babsel –syukran wa jazakallahu khair untuk segala bantuan teknis, dan akan selalu ku ingat KISS (Keep It Simple St*pid).
11. Teman-teman Alien ‘05’ terimakasih atas kerja sama, kekompakan, kekeluargaan dan persahabatan yang selama ini kalian berikan, serta seluruh teman-teman Informatika UII, jadilah ilmuwan yang berakhlaq Al-Qur’an.
12. Semua pihak yang tidak dapat penulis sebutkan satu persatu dalam membantu penyusunan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan, oleh sebab itu penulis sangat mengharapkan kritik serta saran yang bersifat membangun untuk perbaikan dimasa mendatang. Semoga Tugas Akhir ini dapat berguna dan memberikan manfaat bagi penulis serta semua pembaca.

Yogyakarta, 4 April 2011

Penulis

SARI

Web Service sebagai paradigma baru dalam pengimplementasian fleksibilitas akses dan pertukaran data memudahkan proses integrasi dan kolaborasi pada lingkungan *platform* yang *heterogen* menggunakan XML dan *web* melalui jaringan.

Tugas Akhir ini bertujuan untuk merancang dan membangun *Web Service* dengan mengimplementasikan rumus-rumus Fisika SMA sebagai studi kasus. Sistem ini menyediakan informasi mengenai rumus-rumus Fisika SMA dan disertai perhitungannya. Keunggulan sistem ini adalah dapat dimanfaatkan oleh berbagai kalangan yang menggunakan *platform* dan implementasi teknis yang berbeda.

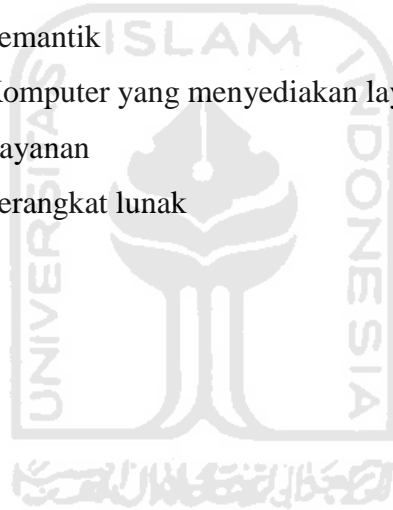
Pada akhirnya, dalam Tugas Akhir ini dibangun satu *server* dengan dua klien yang berbeda implementasi teknis sebagai contoh dan alat uji sistem.

Kata kunci: *Web Service*, Rumus Fisika



TAKARIR

<i>Client</i>	: Klien, Komputer yang menggunakan layanan dari <i>server</i>
<i>Input</i>	: Masukan
<i>Interface</i>	: Antarmuka
<i>Output</i>	: Keluaran
<i>Platform</i>	: Tempat suatu aplikasi dijalankan
<i>Provider</i>	: Penyedia layanan
<i>Requester</i>	: Peminta layanan
<i>Response</i>	: Tanggapan, jawaban, reaksi
<i>Semantic</i>	: Semantik
<i>Server</i>	: Komputer yang menyediakan layanan
<i>Service</i>	: Layanan
<i>Software</i>	: Perangkat lunak



DAFTAR ISI

HALAMAN JUDUL	ii
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI.....	iv
KATA PENGANTAR.....	v
SARI	viii
TAKARIR.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL	xiii
DAFTAR GAMBAR.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metodologi Penelitian	3
1.7 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI	6
2.1 Pengenalan <i>Web Service</i>	6
2.1.1 Definisi <i>Web Service</i>	7
2.1.2 <i>Platform Web Service</i>	9
2.1.3 Komponen-komponen utama pendukung <i>Web Service</i>	10
2.1.4 Kelebihan dan kelemahan menggunakan <i>Web Service</i>	12

2.2	NuSOAP	15
2.3	Pemodelan Berbasis Objek	15
2.3.1	Pemodelan Visual.....	15
2.3.2	UML (<i>Unified Modeling Language</i>).....	16
2.4	Fisika	18
2.5	Mata pelajaran Fisika SMA kurikulum 2004	19
2.5.1	Pengertian Kurikulum	19
2.5.2	Kurikulum Fisika	19
2.5.3	Mata Pelajaran Fisika SMA	20
BAB III METODOLOGI		21
3.1	Metode Perancangan.....	21
3.2	Analisis Kebutuhan Perancangan Sistem	21
3.2.1	Perancangan Arsitektur Integrasi Sistem	21
3.2.2	<i>Use Case Diagram</i>	22
3.2.3	<i>Class Diagram</i>	24
3.2.4	<i>Sequence Diagram</i>	25
3.3	Perancangan Basis Data.....	28
3.3.1	Rancangan Struktur Tabel.....	28
3.3.2	Relasi Tabel.....	30
BAB IV HASIL DAN PEMBAHASAN		31
4.1	Implementasi <i>Web Service</i>	31
4.1.1	Implementasi <i>Web Service</i> Rumus Fisika Berbasis PHP.....	31
4.2	Hasil Implementasi <i>Web Service</i>	35
4.2.1	Hasil Implementasi <i>Web Service</i> Pada <i>Client</i> Berbasis <i>Web</i>	35
4.2.2	Hasil Implementasi <i>Web Service</i> Pada <i>Client</i> berbasis <i>Mobile</i> ...	40

4.3	Halaman <i>Admin</i>	46
4.4	Pengujian Sistem.....	48
4.4.1	Pengujian Halaman Klien Berbasis <i>Web</i>	48
4.5	Evaluasi Sistem.....	53
BAB V KESIMPULAN DAN SARAN.....		58
DAFTAR PUSTAKA.....		59
LAMPIRAN.....		60



DAFTAR TABEL

Tabel 3.1 Struktur Tabel Kelas	29
Tabel 3.2 Struktur Tabel Bab	29
Tabel 3.3 Struktur Tabel Kategori	29
Tabel 3.4 Struktur Tabel Rumus	30
Tabel 4.1 Fungsi-fungsi yang digunakan	31
Tabel 4.2 Hasil Penilaian Responden.....	55



DAFTAR GAMBAR

Gambar 2.1 Proses umum yang terjadi di <i>Web Service</i>	9
Gambar 2.2 Arsitektur umum <i>Web Service</i>	10
Gambar 2.3 Mekanisme request-response WSDL.	12
Gambar 3.1. Arsitektur Integrasi Sistem <i>Web Service</i> Rumus Fisika.....	22
Gambar 3.2 <i>Use Case Diagram</i> Aplikasi <i>Web Service</i> Rumus Fisika SMA.....	23
Gambar 3.3 <i>Class Diagram</i> <i>Web Service</i> Rumus Fisika	24
Gambar 3.4 <i>Class Diagram</i> untuk manipulasi data kelas, bab, kategori, dan rumus pada sisi <i>admin</i>	25
Gambar 3.5 <i>Sequence Diagram</i> <i>Web Service</i> Rumus Fisika SMA.....	26
Gambar 3.6 <i>Sequence Diagram</i> manipulasi data kelas oleh <i>admin</i>	27
Gambar 3.7 <i>Sequence Diagram</i> manipulasi data bab oleh <i>admin</i>	27
Gambar 3.8 <i>Sequence Diagram</i> manipulasi data kategori oleh <i>admin</i>	28
Gambar 3.9 <i>Sequence Diagram</i> manipulasi data rumus oleh <i>admin</i>	28
Gambar 3.10 Relasi Tabel.....	30
Gambar 4.1 Kumpulan <i>Service</i> rumus Fisika	33
Gambar 4.2 Detail <i>Service</i> <code>getKelas()</code>	34
Gambar 4.3 Hasil Skema WSDL	34
Gambar 4.4 Halaman Awal <i>Web Service Client</i>	36
Gambar 4.5 Pilihan Kelas	36
Gambar 4.6 Pilihan Bab	37
Gambar 4.7 Pilihan Kategori.....	37
Gambar 4.8 Pilihan Rumus	38
Gambar 4.9 Memasukkan nilai variabel yang diketahui.....	39

Gambar 4.10 Hasil Perhitungan	40
Gambar 4.11 Aplikasi sebelum dibuka	41
Gambar 4.12 Pilihan Kelas	42
Gambar 4.13 Pilihan Bab	43
Gambar 4.14 Pilihan Kategori.....	43
Gambar 4.15 Pilihan Rumus	44
Gambar 4.16 Form variabel-variabel yang diketahui	45
Gambar 4.17 Hasil Perhitungan	45
Gambar 4.18 Halaman <i>Admin</i> Untuk Manipulasi Data Kelas	46
Gambar 4.19 Halaman <i>Admin</i> Untuk Memanipulasi Data Bab	47
Gambar 4.20 Halaman <i>Admin</i> Untuk Memanipulasi Data Kategori.....	47
Gambar 4.21 Halaman <i>Admin</i> Untuk Memanipulasi Data Rumus	48
Gambar 4.22 Pengujian dengan nilai satu digit.	49
Gambar 4.23 Pengujian dengan nilai masukan lima belas digit.	49
Gambar 4.24 Pengujian dengan nilai minus.	50
Gambar 4.25 Pengujian dengan nilai bilangan riil.....	51
Gambar 4.26 Pengujian dengan masukan karakter.	52
Gambar 4.27 Pengujian dengan nilai melebihi digit maksimal.	53
Gambar 4.28 Diagram Nilai Rerata Responden.....	55
Gambar 4.29 Diagram Nilai Rerata Setiap Pernyataan.....	56
Gambar 4.30 Diagram Hasil Evaluasi.....	57

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi, khususnya dibidang teknologi informasi merupakan salah satu pemicu terjadinya perubahan pola pikir manusia untuk dapat memperoleh informasi secara cepat, akurat, dan dapat dipercaya. *Internet* sebagai salah satu media teknologi informasi sangat besar manfaatnya bagi masyarakat. *Internet* telah menyatukan jarak dan berbagai perbedaan yang sebelumnya menjadi penghambat dalam komunikasi. Seiring dengan maraknya penggunaan *Internet* tersebut maka semakin banyak pula aplikasi-aplikasi yang dibutuhkan oleh pengguna, seperti aplikasi: *e-commerce*, *e-learning*, *e-government*, *e-travel*, *e-procurement*, *digital library* dan lain-lain. Secara khusus hal ini sangat nyata terlihat dalam kegiatan bisnis dan usaha serta pendidikan.

Dalam dunia pendidikan misalnya, sudah mulai dikembangkan berbagai aplikasi untuk membantu proses pembelajaran seperti *digital library*, *e-learning* dan sebagainya. Banyak institusi pendidikan yang telah menyadari pentingnya aplikasi-aplikasi pendukung pembelajaran sebagai salah satu media untuk meningkatkan pengetahuan. Tetapi informasi pada aplikasi yang ada saat ini belum terintegrasi atau berdiri sendiri-sendiri. Ini menyebabkan pengembangan yang dilakukan pada satu aplikasi hanya akan mempengaruhi pengguna aplikasi itu sendiri serta tidak meratanya informasi dalam hal ini pengetahuan kepada para pengguna aplikasi yang lain. Bayangkan apabila dibuat suatu modul aplikasi yang dapat dipasang atau ditambahkan pada aplikasi-aplikasi yang ada saat ini dengan tidak dipengaruhi oleh perbedaan *platform*.

Munculnya teknologi *Web Service* dalam dunia *Internet* memberikan paradigma baru dalam pengimplementasian fleksibilitas akses dan pertukaran data yang terdapat pada aplikasi memungkinkan pengguna mendapatkan informasi yang dibutuhkan tidak hanya dari satu sistem informasi tapi dari berbagai sistem informasi yang sama-sama menyediakan *Web Service*.

Web Service menjadi populer di *enterprise* karena kemampuannya dalam mengintegrasikan aplikasi-aplikasi yang berbeda *platform* (Manes, 2003). *Web Service* mendukung komunikasi antaraplikasi dan integrasi aplikasi dengan menggunakan XML dan *web*.

Penerapan *Web Service* akan memudahkan proses integrasi dan kolaborasi antaraplikasi pada lingkungan *platform* yang *heterogen* baik melalui jaringan *Intranet* maupun *Internet*. Karakteristik *Web Service* bersifat terbuka (*non-proprietary*), terangkai secara bebas (*loosely coupled*), dan *modular*.

Salah satu yang membutuhkan sistem aplikasi sebagai alat bantu belajar bagi siswa adalah Fisika karena banyak berkaitan dengan angka dan rumus-rumus sehingga banyak siswa yang kurang berminat untuk memperdalam ilmu Fisika.

1.2 Rumusan Masalah

Bagaimana membuat suatu alat bantu dan pendukung mata pelajaran Fisika untuk tingkat SMA yang dapat dimanfaatkan oleh perorangan atau institusi yang menggunakan *platform* maupun implementasi teknis yang berbeda dengan memanfaatkan *Web Service*.

1.3 Batasan Masalah

Permasalahan yang dibahas pada tugas akhir ini dibatasi pada perancangan dan implementasi kinerja *Web Service* sebagai pendukung dan alat bantu pendidikan serta pembelajaran pada tingkat SMA. Mata pelajaran yang digunakan sebagai objek penelitian adalah Fisika SMA kurikulum tahun 2004 yang mencakup rumus-rumus sebagai berikut:

- a) Materi pokok kelas X: Besaran, pengukuran dan vektor; karakteristik gerak; penerapan hukum Newton; tata surya; suhu dan kalor; cahaya; hakekat gelombang elektromagnetik; listrik dinamis.
- b) Materi pokok kelas XI: Gerak dengan analisis vektor; energi, usaha, dan daya; impuls dan momentum; momentum sudut dan rotasi benda tegar; fluida; teori kinetik gas; termodinamika.

- c) Materi pokok kelas XII: Gaya listrik dan medan listrik; medan magnet, gaya Lorentz dan induksi elektromagnetik; gelombang dan bunyi, radiasi benda hitam, teori atom, relativitas, zat padat/semikonduktor; radioaktivitas; jagat raya.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah memahami karakteristik teknologi *Web Service* dan kemudian memanfaatkannya dalam pembuatan suatu *service* sebagai alat bantu dan pendukung mata pelajaran Fisika pada tingkat SMA.

1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian adalah:

- a) Ketersediaan sistem berbasis *Web Service* yang mampu mempermudah institusi-institusi pendidikan atau perorangan untuk menambahkan pada sistem mereka *service* alat bantu ajar mata pelajaran Fisika SMA tanpa memperhatikan seluk-beluk pemrograman dari *Web Service* itu sendiri.
- b) Mempermudah para pengajar untuk mendapat referensi rumus-rumus Fisika sebagai alat bantu pendidikan.
- c) Mempermudah para siswa SMA untuk belajar dan menguasai ilmu Fisika.

1.6 Metodologi Penelitian

Metode yang dikembangkan adalah sebagai berikut:

1. Studi Pustaka

Studi pustaka ditujukan untuk mengetahui secara rinci mengenai teori-teori *Web Service*, informasi mengenai rumus-rumus Fisika SMA, dan kompetensi kurikulum SMA. Data diperoleh dari buku-buku dan literatur yang didapat melalui *Internet*.

2. Perancangan Sistem

Metode perancangan yang digunakan dalam perancangan sistem ini adalah menggunakan metode perancangan berorientasi objek dengan menggunakan UML (*Unified Modelling Language*) sebagai alat bantu

perancangan. Untuk diagram UML yang dipakai dalam perancangan sistem ini adalah *use case diagram*, *sequence diagram*, *class diagram*.

3. Implementasi

Implementasi merupakan tahapan untuk membuat atau menerapkan hasil dari perancangan yang telah dibuat. Dimana didalamnya termasuk tahap pengkodean program atau dengan kata lain menerjemahkan data atau memecahkan permasalahan yang telah dirancang ke dalam bahasa pemrograman yang telah ditentukan sebelumnya.

4. Pengujian

Pengujian yaitu mencoba dan menguji kinerja *Web Service* yang telah dibuat serta mencari kelemahan yang masih ada pada aplikasi sistem kemudian memperbaiki kelemahan yang ada, sehingga *Web Service* dapat bekerja dengan baik.

5. Evaluasi

Evaluasi hasil keseluruhan kegiatan perancangan aplikasi sistem serta kesimpulan dan saran yang bisa diambil dari kegiatan perancangan aplikasi sistem.

1.7 Sistematika Penulisan

Adapun sistematika penulisan karya ilmiah ini terdiri dari 5 bab yang secara garis besar sebagai berikut:

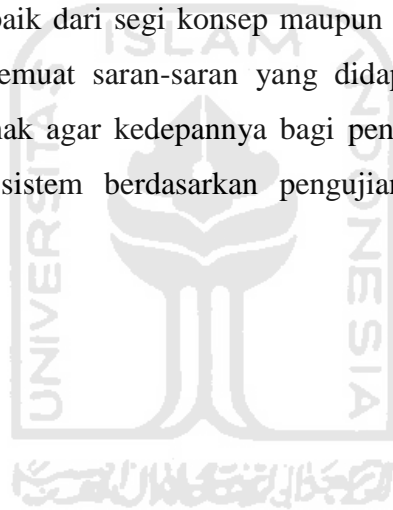
Bab I Pendahuluan, berisi pendahuluan yang menjelaskan latar belakang dari tugas akhir, gambaran umum permasalahan beserta batasan masalah yang menjadi tolak ukur penulisan dalam melakukan penelitian, tujuan penelitian, manfaat penelitian, metodologi penelitian serta sistematika penulisan yang digunakan.

Bab II Landasan Teori, berisi tentang landasan teori, yang merupakan pembahasan tentang teori-teori yang digunakan dan relevan dengan topik tugas akhir, dimulai dari konsep *Web Service* yang bersifat umum sampai dengan teori tentang perangkat lunak yang digunakan untuk mendukung perancangan *Web Service*.

Bab III Metodologi, dalam bab ini akan dibahas tentang perancangan dari sistem yang akan dibuat dan aplikasi apa saja yang akan digunakan dalam pembuatan sistem tersebut.

Bab IV Hasil dan Pembahasan, memuat hasil dan pembahasan perangkat lunak, yaitu membahas tentang implementasi perangkat lunak, pengujian dan analisis perangkat lunak baik pengujian yang sifatnya valid maupun non valid yang keseluruhan pengujian ini bertujuan untuk melihat sejauh mana validitas informasi yang diberikan oleh perangkat lunak dilengkapi dengan pembahasan hasil informasi sistem serta evaluasi penggunaan sistem.

Bab V Kesimpulan dan Saran, memuat rangkuman yang didapat dari penjabaran secara detail baik dari segi konsep maupun dari sisi perangkat lunak yang dibuat dan juga memuat saran-saran yang didapatkan dari kekurangan-kekurangan perangkat lunak agar kedepannya bagi pengembangan sistem dapat memaksimalkan kinerja sistem berdasarkan pengujian yang telah dilakukan sebelumnya.



BAB II

LANDASAN TEORI

2.1 Pengenalan *Web Service*

Web Service adalah aplikasi perangkat lunak yang dapat diakses melalui jarak jauh menggunakan berbagai bahasa berbasis XML (*eXtensible Markup Language*). Biasanya, *Web Service* diidentifikasi menggunakan URL, seperti situs *web* lainnya. Yang membuat *Web Service* berbeda dari situs *web* biasa adalah jenis interaksi yang dapat diberikan.

Sebagian besar situs *web* dirancang untuk memberikan tanggapan atas permintaan dari seseorang. Orang akan mengetik URL dari situs atau mengklik pada *hyperlink* untuk melakukan permintaan. Permintaan ini mengambil bentuk sebuah dokumen teks yang berisi beberapa instruksi yang cukup sederhana untuk *server*. Instruksi-instruksi ini terbatas pada nama dokumen yang harus dikembalikan atau panggilan ke sisi *server* program, bersama dengan beberapa parameter (Potts dan Kopack, 2003).

Teknologi *Web Service* menawarkan kemudahan dalam menjembatani pulau-pulau informasi tanpa mempermasalahkan perbedaan teknologi yang digunakan oleh masing-masing sumber. Misalkan sebuah situs informasi dibangun menggunakan basis data Oracle sedangkan situs lainnya menggunakan MySQL, kemudian situs lainnya menggunakan perangkat lunak *open source* dalam membangun situs. *Web Service* akan mengatasi perbedaan ini.

Tidak hanya informasi, *Web Service* dapat dikembangkan ke arah penggunaan aplikasi, ilustrasinya misalnya, jika sebuah situs memerlukan aplikasi untuk menghitung menggunakan rumus-rumus Fisika, maka situs tersebut dapat mencari *web* yang menyediakan layanan aplikasi rumus-rumus Fisika, tanpa perlu membuat sendiri aplikasi yang diperlukan. Situs tersebut hanya mengirimkan variabel-variabel yang akan dihitung selanjutnya akan menerima hasil perhitungan tersebut.

Web Service adalah turunan berikutnya dari aplikasi berbasis *web* dan berupa aplikasi modular yang dapat dipublikasikan, diletakkan, dan dibangkitkan antar *web*, atau bahkan antaraplikasi. *Web Service* dapat diibaratkan sebagai pustaka yang menyediakan berbagai macam informasi dan layanan ke aplikasi lain antar *web*. Ada 4 aplikasi utama untuk *Web Service*, yaitu (Siswoutomo, 2004:7):

- a) **Integrasi aplikasi-aplikasi**, juga sering disebut EAI (*Enterprise Application Integration*). Misalkan, manajemen *stock* langsung dapat membuat *order* dengan manajemen *order* walaupun dibangun dengan sistem yang sama sekali berbeda.
- b) **Integrasi bisnis-bisnis**, misalkan sebuah toko menggunakan sistem/aplikasi untuk melakukan permintaan secara otomatis ke aplikasi yang berada pada *vendor* pemasok.
- c) **Distribusi informasi**, sebuah situs dapat menawarkan berbagai informasi yang dapat diambil oleh situs lain.
- d) **Fungsionalitas aplikasi**, misalnya sebuah situs memerlukan sebuah aplikasi *program* penghitung rumus Fisika, situs tersebut tidak perlu membuat aplikasi tersebut, hanya tinggal memanfaatkan *program* yang sudah jadi dari situs lain yang menyediakan aplikasi *program* tersebut.

2.1.1 Definisi *Web Service*

Definisi *Web Service* (W3C) adalah "Sebuah sistem perangkat lunak yang didesain untuk mendukung interaksi antara sebuah mesin ke mesin lainnya melalui sebuah jaringan, yang mempunyai *interface* untuk mendeskripsikan data melalui sebuah *format* yang umum (khususnya WSDL), dan sistem lainnya yang berinteraksi dengan *Web Service* berbicara melalui pesan dengan SOAP, biasanya disampaikan menggunakan protokol standar yaitu HTTP, dengan menggunakan XML sebagai bahasa standar yang dapat dibaca antar *platform* dan dapat melakukan pertukaran data antar *platform*" (Siswoutomo, 2004:12).

Dari definisi di atas ada beberapa elemen penting yang mendukung *Web Service* yaitu WSDL (*Web Services Description Language*), SOAP (*Simple Object*

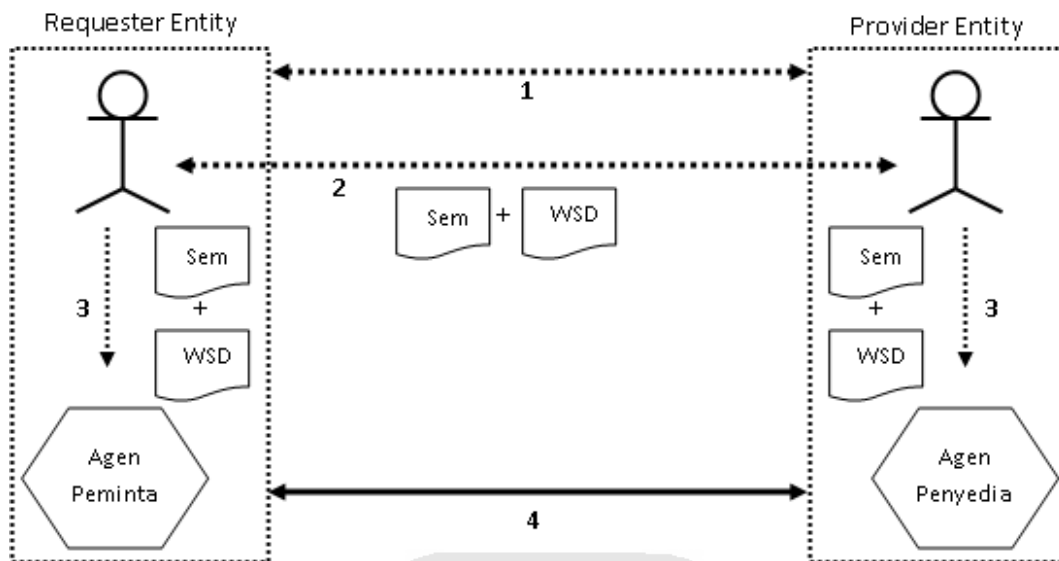
Access Protocol), dan XML (*Extensible Markup Language*). Kata kunci yang tidak kalah pentingnya adalah interaksi. Dengan adanya *Web Service*, dapat mengkomunikasikan berbagai *software* dari berbagai mesin dengan *platform* yang berbeda. Untuk dapat menjalankan fungsi tersebut, sebuah *Web Service* memerlukan agen, agen adalah potongan perangkat lunak yang mengirim dan menerima pesan, agen tersebut dapat ditulis dengan berbagai bahasa pemrograman. Komunikasi di dalam *Web Service* membutuhkan *providers entity* (penyedia) dan *requesters entity* (peminta). *Provider Entity* adalah seseorang atau organisasi yang menyediakan agen untuk layanan atau fungsi tertentu. Sedangkan *requester entity* adalah seseorang atau organisasi yang menggunakan layanan yang disediakan oleh *provider*. Agar dapat berkomunikasi dengan lancar, kedua entitas ini harus sepakat menggunakan *semantic* dan mekanisme yang sama dalam pertukaran pesan.

Mekanisme pertukaran pesan didokumentasikan di *Web Service Description* (WSD). WSD adalah spesifikasi mesin yang dapat berproses pada *interface Web Service*. WSD mendefinisikan *format* pesan, tipe data, *transport protocol*, dan format serialisasi yang digunakan antara agen di *requester* dan agen di *provider*. WSD juga menentukan satu atau lebih *network* tempat agen *provider* akan dipanggil.

Semantic yang disepakati di dalam berkomunikasi pada sebuah *Web Service* adalah ekspektasi dari tingkah laku *service*. Bisa disebut semacam kontrak antara entitas peminta dan penyedia mengenai kegunaan dan konsekuensi dari interaksi tersebut. Secara umum langkah-langkah yang diperlukan adalah:

- a) Entitas peminta dan penyedia saling mengetahui satu sama lain. Atau setidaknya salah satu mengetahui yang lain.
- b) Entitas peminta dan penyedia sama-sama setuju deskripsi layanan dan *semantic* yang akan mengatur interaksi antarpeminta dan penyedia.
- c) Deskripsi layanan dan *semantic* berdasarkan agen peminta dan penyedia.
- d) Agen peminta dan penyedia bertukar pesan.

Langkah-langkah di atas ditunjukkan pada Gambar 2.1



Gambar 2.1 Proses umum yang terjadi di *Web Service*.

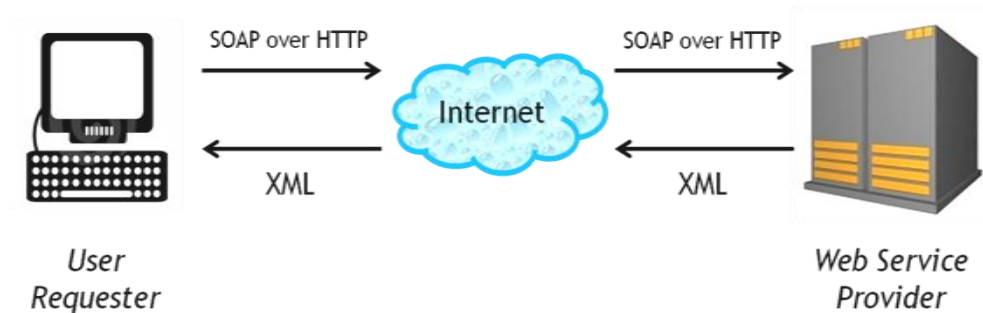
Sumber: Siswoutomo (2004:14)

Keterangan Gambar 2.1:

- 1) Entitas peminta dan penyedia saling mengetahui satu sama lain.
- 2) Entitas peminta dan penyedia menyetujui deskripsi layanan dan *semantic* yang akan mengatur interaksi.
- 3) Deskripsi layanan yang sudah disetujui.
- 4) Entitas peminta dan penyedia saling bertukar pesan (berinteraksi).

2.1.2 Platform *Web Service*

Kemudian untuk *platform* dasar dari *Web Service* adalah XML ditambah HTTP. HTTP adalah protokol yang dapat berjalan dimana saja di *Internet*, sedangkan XML menyediakan bahasa meta (*meta language*), jadi pada dasarnya aplikasi yang dibangun dengan teknologi *Web Service* dengan bahasa pemrograman apa saja, bisa berkomunikasi dengan aplikasi lain yang dibangun dengan bahasa yang bisa saja sama sekali berbeda. Aplikasi tersebut dapat mengirim pesan dalam bentuk XML ke aplikasi lain dan mendapatkan respon XML dari aplikasi tersebut. Arsitektur umum dari *Web Service* ditunjukkan pada Gambar 2.2



Gambar 2.2 Arsitektur umum *Web Service*.

Dari ilustrasi pada gambar 2.2, SOAP adalah *format* standar pengiriman pesan berbentuk XML yang digunakan untuk melakukan proses *request* dan *response* antara *Web Service (provider)* dengan aplikasi yang memanggilnya (*requester*). Dokumen SOAP yang digunakan untuk melakukan permintaan disebut *SOAP request* sedangkan dokumen SOAP yang diperoleh dari *Web Service* disebut dengan *SOAP response* (Lucky, 2008). XML sendiri merupakan elemen penting bagi pengembang yang ingin mengembangkan *Web Service*. XML juga memiliki kemampuan untuk integrasi data di samping pertukaran data antar *platform*, dengan adanya XML, *programmer* tidak perlu bingung menyatukan (mengintegrasikan) berbagai data yang dibangun dengan berbagai aplikasi pada *platform* yang berbeda.

2.1.3 Komponen-komponen utama pendukung *Web Service*

Dalam membangun aplikasi berbasis *Web Service* dibutuhkan beberapa komponen aplikasi untuk mendukung pengembangannya. Komponen-komponen tersebut di antaranya adalah: XML, SOAP, WSDL, UDDI.

a) XML (*eXtensible Markup Language*)

XML merupakan dasar yang penting bagi terbentuknya sistem *Web Service*, karena sebuah *Web Service* berkomunikasi dengan aplikasi yang memanggilnya menggunakan XML. XML berbentuk teks sehingga mudah

untuk ditransportasikan menggunakan *protocol* HTTP. Beberapa kelebihan XML (Siswoutomo, 2004:27):

1. *Intelligence*. XML dapat menangani berbagai level kompleksitas.
2. *Adaptation*. Tag dapat diberi nama sesuai keinginan dan kebutuhan.
3. *Maintenance*. Mudah dalam pemeliharaan karena hanya berisi data dan *markup*. *Stylesheet* dan *link* terpisah dari dokumen.
4. *Linking*. XML dapat melakukan *link* lebih dari HTML. XML dapat *me-link* satu atau lebih poin di dalam maupun di luar data.
5. *Simplicity*. XML lebih sederhana apabila dibandingkan dengan bahasa lain.
6. *Portability*. XML mempunyai sifat portabilitas yang bagus karena memisahkan antara data dan pengaturan tampilan.

b) SOAP

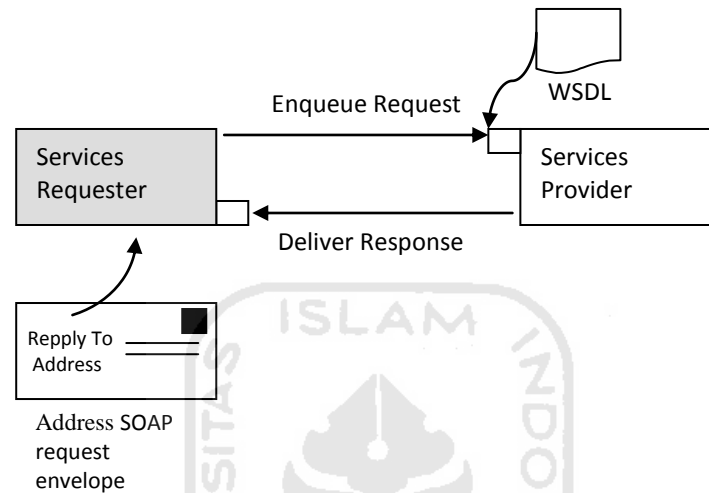
SOAP pada awalnya adalah singkatan dari *Simple Object Access Protocol*, tetapi sekarang SOAP adalah sebuah nama dan bukan singkatan dari apapun. SOAP adalah protokol untuk pertukaran informasi dengan desentralisasi dan terdistribusi.

SOAP adalah protokol pengiriman pesan yang digunakan untuk berkomunikasi antar aplikasi melalui *Internet* berbasis XML. SOAP bersifat independen baik dari sisi *platform* maupun bahasa, oleh karenanya SOAP sangat sederhana dan mudah dikembangkan (Siswoutomo, 2004:187).

c) *Web Service Description Language* (WSDL)

WSDL merupakan suatu dokumen XML yang menjelaskan metode-metode apa saja yang tersedia dalam suatu *Web Service*, parameter apa saja yang diperlukan untuk memanggil suatu metode dan apa hasil dari metode yang akan dipanggil. WSDL dapat dikatakan representasi kontrak antara *requester* dan *provider*-nya. Secara lebih teknis adalah representasi kontrak antara *client* dan kode di *server*. Dengan menggunakan WSDL, *client* dapat memanfaatkan fungsi-fungsi publik yang disediakan oleh *server*.

WSDL adalah dokumen XML yang *machine-readable* dan bukan *human-readable*. Oleh karena itu, dokumen tersebut dapat dimanfaatkan untuk melakukan otomatisasi proses pengintegrasian layanan ke aplikasi *requestor*. Gambar 2.3 adalah gambar detail dari *request-response* yang terjadi di WSDL.



Gambar 2.3 Mekanisme *request-response* WSDL.

Sumber: Siswoutomo (2004:209)

d) *Universal Description, Discovery, and Integration* (UDDI)

UDDI merupakan suatu *directory service* yang digunakan untuk meregistrasikan dan mencari *Web Service*, dimana di dalamnya terdapat kumpulan *Web Services* berdasarkan kata kunci dan kategori tertentu. Dengan menggunakan UDDI, *Web Service* dapat didaftarkan agar dapat dicari dan ditemukan oleh orang lain. Informasi standar dan implementasi UDDI ini dapat dilihat di <http://www.uddi.org>.

2.1.4 Kelebihan dan kelemahan menggunakan *Web Service*

a) Kelebihan menggunakan *Web Service*

Ada banyak kelebihan menggunakan *Web Service*. Faktanya, banyak aplikasi *Web Service* akan sangat mahal atau tidak mungkin untuk diduplikasi menggunakan teknologi yang lain. Salah satu tantangan bagi para *technical leader* adalah untuk menemukan peluang dimana *Web*

Service dapat digunakan untuk memecahkan masalah-masalah sekarang ini (Potts dan Kopack, 2003).

Berikut adalah beberapa kelebihan *Web Service* (Lucky, 2008):

1. *Lintas Platform*

Penggunaan *Web Service* memungkinkan komputer-komputer yang berbeda sistem operasi dapat saling bertukar data. Sebagai contoh, jika anda ingin mengakses sebuah *Web Service*, sistem operasi yang anda gunakan tidak perlu sama dengan sistem operasi yang digunakan oleh *Web Service* tersebut. Yang perlu diketahui hanyalah metode apa saja yang disediakan oleh *Web Service* tersebut.

2. *Language Independent*

Sebuah *Web Service* dapat diakses menggunakan bahasa pemrograman apa saja. *Web Service* yang dibuat dengan PHP bisa diakses oleh berbagai bahasa pemrograman *web* seperti PHP itu sendiri, JSP hingga bahasa pemrograman lainnya seperti Delphi, Java, dan VB.NET. Dan yang lebih menyenangkan lagi, *Web Service* juga memungkinkan untuk diakses oleh *mobile device* seperti *handphone* dengan memanfaatkan aplikasi berbasis Java untuk *mobile device* (*Java 2 Micro Edition* atau lebih dikenal dengan singkatan J2ME).

3. *Jembatan Penghubung dengan Basis data*

Pada umumnya sebuah aplikasi memerlukan *driver* basis data agar bisa melakukan koneksi ke sebuah basis data. *Web Service* dapat dijadikan sebagai “jembatan” penghubung antara aplikasi dengan basis data. Dengan memanfaatkan *Web Service*, sebuah aplikasi tidak lagi memerlukan *driver* basis data dan tidak perlu mengetahui basis data apa yang digunakan oleh *server* serta bagaimana struktur basis data tersebut jika ingin mengaksesnya. Aplikasi tersebut cukup mengetahui metode atau fungsi apa saja yang disediakan *Web Service* untuk memanfaatkan fasilitasnya.

4. Mempermudah Proses Pertukaran Data

Penggunaan *Web Service* dapat mempermudah dan mempercepat pertukaran data antara dua atau lebih *server* yang berbeda sistem operasi dan basis data serta jarak yang terpisah jauh.

5. Penggunaan Kembali Komponen Aplikasi

Beberapa aplikasi yang berbeda bisa saja memerlukan sebuah fungsi yang sama. Misalnya saja, ada aplikasi *web* dan *desktop* yang memerlukan sebuah fungsi untuk melakukan konversi kurs mata uang yang berbeda. Daripada membuat fungsi yang sama dua kali, lebih baik membuat fungsi kurs tersebut dalam sebuah *Web Service*.

b) Kelemahan *Web Service*

Di antara kelebihan atau keunggulan dari sebuah sistem yang dibangun dengan teknologi *Web Service* ada beberapa kelemahan yang juga perlu dipertimbangkan ketika akan membangun sebuah sistem berbasis *Web Service*. Di antara kelemahannya adalah sebagai berikut:

1. *Web Service* Tidak Mendukung Transaksi

Dalam basis data dikenal dengan istilah *transaction*, dimana transaksi-transaksi yang dilakukan terhadap basis data dapat di-*commit* atau di-*rollback* berdasarkan kondisi tertentu. Protokol SOAP saat ini belum mendukung transaksi.

2. Mekanisme *Callback*

Web Service tidak dapat digunakan untuk melakukan *callback* terhadap *client*, artinya *Web Service* tidak dapat mengaktifkan fungsi-fungsi yang berada di *client*.

3. *Performance*

Web Service membutuhkan suatu infrastruktur jaringan yang cukup baik, dalam hal ini tidak dapat begitu saja mengimplentasikan *Web Service* dalam sistem yang tidak didukung dengan infrastruktur jaringan yang memadai. Karena kinerja *Web Service* kedinamisannya sulit diprediksi. Oleh karena itu, jaringan *Internet/Intranet* dengan

bandwidth yang kecil akan tidak cocok menggunakan pendekatan *Web Service*.

2.2 NuSOAP

NuSOAP adalah salah satu dari sekian *toolkit* yang tersedia untuk *programmer* PHP yang ingin bekerja dengan layanan SOAP. Keunggulan yang ditawarkan NuSOAP adalah karena kesederhanaan, sebab justru karena sederhana dan mudah sehingga kecepatannya menjadi lebih baik. Adanya NuSOAP menyebabkan keuntungan menggunakan bahasa PHP sebagai bahasa pemrograman.

NuSOAP ditulis oleh Dietrich Ayala dengan proyek awal bernama SOAPx4. selanjutnya ia disewa oleh NuSphere untuk terus mengembangkan proyek tersebut dan diberi nama NuSOAP. NuSOAP termasuk *class* PHP dan tidak membutuhkan ekstensi apa-apa. *Developer* dapat menggunakan NuSOAP ini untuk berbagai keperluan.

2.3 Pemodelan Berbasis Objek

2.3.1 Pemodelan Visual

Pemodelan visual adalah proses penggambaran informasi-informasi secara grafis dengan notasi-notasi baku yang telah disepakati sebelumnya. Notasi-notasi baku sangat penting demi suatu alasan: komunikasi. Dengan notasi-notasi pemodelan yang bersifat baku, komunikasi yang baik akan terjalin dengan mudah antar anggota tim pengembang sistem/perangkat lunak. Sebagai contoh, jika kita akan membangun rumah dengan bantuan seorang arsitek, maka kita akan mendapatkan rancangan rumah, biasanya dalam bentuk denah-denang atau maket yang akan memandu kontraktor/pembangun dalam membangun rumah kita tersebut. Cara yang sama dilakukan oleh para pengembang sistem informasi/perangka lunak. Untuk mendapatkan spesifikasi perangkat lunak yang (diharapkan) sesuai dengan keinginan dan kebutuhan pengguna (*user's needed and expectations*), para pengembang melakukan pemodelan-pemodelan secara visual (Nugroho, 2005).

2.3.2 UML (*Unified Modeling Language*)

UML adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berorientasi objek. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema basis data, dan komponen-komponen yang diperlukan dalam sistem *software*.

UML merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek. UML adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem menjadi piranti lunak. UML menawarkan sebuah standar untuk merancang model sistem.

Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut yang berbeda-beda sehingga kita bisa mendapatkan pemahaman secara menyeluruh. Untuk upaya tersebut UML menyediakan delapan jenis diagram yang dapat dikelompokkan berdasarkan sifatnya, statis atau dinamis. Kedelapan jenis diagram dalam UML itu adalah (Nugroho, 2005):

1. *Class Diagram*

Class Diagram menggambarkan struktur statis dari sebuah sistem. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.

2. *Use-Case Diagram*

Use-Case Diagram bersifat statis. Diagram ini memperlihatkan himpunan *use case* dan akto-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

3. *Sequence Diagram*

Sequence Diagram bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

4. *Collaboration Diagram*

Collaboration Diagram bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan (*message*).

5. *Statechart Diagram*

Statechart Diagram bersifat dinamis. *Diagram state* ini memperlihatkan *states* pada sistem; memuat *state*, transisi, *event*, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.

6. *Activity Diagram*

Activity Diagram bersifat dinamis. Diagram aktifitas ini adalah tipe khusus dari diagram *state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antarobjek.

7. *Component Diagram*

Component Diagram bersifat statis. Diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka (*interfaces*), serta kolaborasi-kolaborasi.

8. *Deployment Diagram*

Deployment Diagram bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run-time*). Diagram ini memuat

simpul-simpul (*node*) beserta komponen-komponen yang ada di dalamnya. *Deployment diagram* berhubungan erat dengan diagram komponen dimana *deployment diagram* memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kedelapan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak; semuanya dibuat sesuai dengan kebutuhan. Juga pada pemodelan dengan UML dimungkinkan menggunakan diagram-diagram lain (misalnya DFD [*Data Flow Diagram*], ERD [*Entity Relationship Diagram*], dan sebagainya) sejauh itu memang diperlukan untuk mendapatkan pemahaman mendalam tentang suatu sistem/perangkat lunak. Dalam perancangan *Web Service* mata pelajaran Fisika SMA, hanya akan dibuat tiga diagram yaitu *use case diagram*, *class diagram*, dan *sequence diagram*.

2.4 Fisika

Fisika (Bahasa Yunani: φυσικός (*physikos*), "alamiah", dan φύσις (*physis*), "Alam") adalah sains atau ilmu tentang alam dalam makna yang terluas. Fisika mempelajari gejala alam yang tidak hidup atau materi dalam lingkup ruang dan waktu. Beberapa sifat yang dipelajari dalam Fisika merupakan sifat yang ada dalam semua sistem materi yang ada, seperti hukum kekekalan energi. Sifat semacam ini sering disebut sebagai hukum Fisika. Fisika sering disebut sebagai "ilmu paling mendasar", karena setiap ilmu alam lainnya (biologi, kimia, geologi, dan lain-lain) mempelajari jenis sistem materi tertentu yang mematuhi hukum Fisika.

Fisika juga berkaitan erat dengan matematika. Teori Fisika banyak dinyatakan dalam notasi matematis, dan matematika yang digunakan biasanya lebih rumit daripada matematika yang digunakan dalam bidang sains lainnya. Perbedaan antara Fisika dan Matematika adalah: Fisika berkaitan dengan pemerian dunia material, sedangkan matematika berkaitan dengan pola-pola abstrak yang tak selalu berhubungan dengan dunia material.

2.5 Mata pelajaran Fisika SMA kurikulum 2004

2.5.1 Pengertian Kurikulum

“Kurikulum adalah perangkat mata pelajaran yang diberikan oleh suatu lembaga penyelenggara pendidikan yang berisi rancangan pelajaran yang akan diberikan kepada peserta pelajaran dalam satu periode jenjang pendidikan. Penyusunan perangkat mata pelajaran ini disesuaikan dengan keadaan dan kemampuan setiap jenjang pendidikan dalam penyelenggaraan pendidikan tersebut. Lama waktu dalam satu kurikulum biasanya disesuaikan dengan maksud dan tujuan dari sistem pendidikan yang dilaksanakan. Kurikulum ini dimaksudkan untuk dapat mengarahkan pendidikan menuju arah dan tujuan yang dimaksudkan dalam kegiatan pembelajaran secara menyeluruh” (<http://id.wikipedia.org/wiki/Kurikulum>).

2.5.2 Kurikulum Fisika

Abad XXI dikenal sebagai abad globalisasi dan abad teknologi informasi. Perubahan yang sangat cepat dan dramatis dalam bidang ini merupakan fakta dalam kehidupan siswa. Pengembangan kemampuan siswa dalam bidang sains, khususnya bidang Fisika merupakan salah satu kunci keberhasilan peningkatan kemampuan dalam menyesuaikan diri dengan perubahan dan memasuki dunia teknologi, termasuk teknologi informasi. Untuk kepentingan pribadi, sosial, ekonomi dan lingkungan, siswa perlu dibekali dengan kompetensi yang memadai agar menjadi peserta aktif dalam masyarakat. Kurikulum Fisika menyediakan berbagai pengalaman belajar untuk memahami konsep dan proses sains. Pemahaman ini bermanfaat bagi siswa agar dapat (DPN, 2003):

- a) Menanggapi isu lokal, nasional, kawasan dunia, sosial, ekonomi, lingkungan dan etika.
- b) Menilai secara kritis perkembangan dalam bidang sains dan teknologi serta dampaknya.
- c) Memberi sumbangan terhadap kelangsungan perkembangan sains dan teknologi.

- d) Memilih karir yang tepat. Oleh karena itu, kurikulum ini lebih menekankan agar siswa menjadi pelajar yang aktif dan luwes.

2.5.3 Mata Pelajaran Fisika SMA

Sains berkaitan dengan cara mencari tahu tentang alam secara sistematis, sehingga sains bukan hanya penguasaan kumpulan pengetahuan yang berupa fakta-fakta, konsep-konsep, atau prinsip-prinsip saja tetapi juga merupakan suatu proses penemuan. Pendidikan sains di sekolah menengah diharapkan dapat menjadi wahana bagi siswa untuk mempelajari diri sendiri dan alam sekitar, serta prospek pengembangan lebih lanjut dalam menerapkannya di kehidupan sehari-hari.

Pendidikan sains menekankan pada pemberian pengalaman langsung untuk mengembangkan kompetensi agar siswa mampu menjelajahi dan memahami alam sekitar secara ilmiah. Pendidikan sains diarahkan untuk “mencari tahu” dan “berbuat” sehingga dapat membantu siswa untuk memperoleh pemahaman yang lebih mendalam tentang alam sekitar.

Mata Pelajaran Fisika adalah salah satu mata pelajaran dalam rumpun sains yang dapat mengembangkan kemampuan berpikir analitis induktif dan deduktif dalam menyelesaikan masalah yang berkaitan dengan peristiwa alam sekitar, baik secara kualitatif maupun kuantitatif dengan menggunakan matematika, serta dapat mengembangkan pengetahuan, keterampilan, dan sikap percaya diri (DPN, 2003).

BAB III

METODOLOGI

3.1 Metode Perancangan

Metode perancangan untuk membangun aplikasi ini adalah metode perancangan berorientasi objek dengan menggunakan UML (*Unified Modelling Language*) sebagai alat bantu perancangan. Untuk diagram UML yang digunakan dalam perancangan aplikasi ini adalah *use case diagram*, *class diagram*, dan *sequence diagram*.

3.2 Analisis Kebutuhan Perancangan Sistem

Sebelum membuat perancangan, berikut adalah beberapa identifikasi *Web Service* yang akan dibangun dan diharapkan dapat memenuhi kebutuhan-kebutuhan sebagai berikut:

- a) *Web Service* ini dapat menerima *input* berupa pilihan kelas, bab, kategori, dan rumus yang akan digunakan.
- b) *Web Service* ini dapat menerima *input* berupa nilai variabel-variabel yang akan dihitung.
- c) *Web Service* ini dapat menghitung nilai variabel-variabel yang diterima menggunakan rumus-rumus Fisika SMA.
- d) *Web Service* ini dapat memberikan *output* berupa hasil perhitungan.

3.2.1 Perancangan Arsitektur Integrasi Sistem

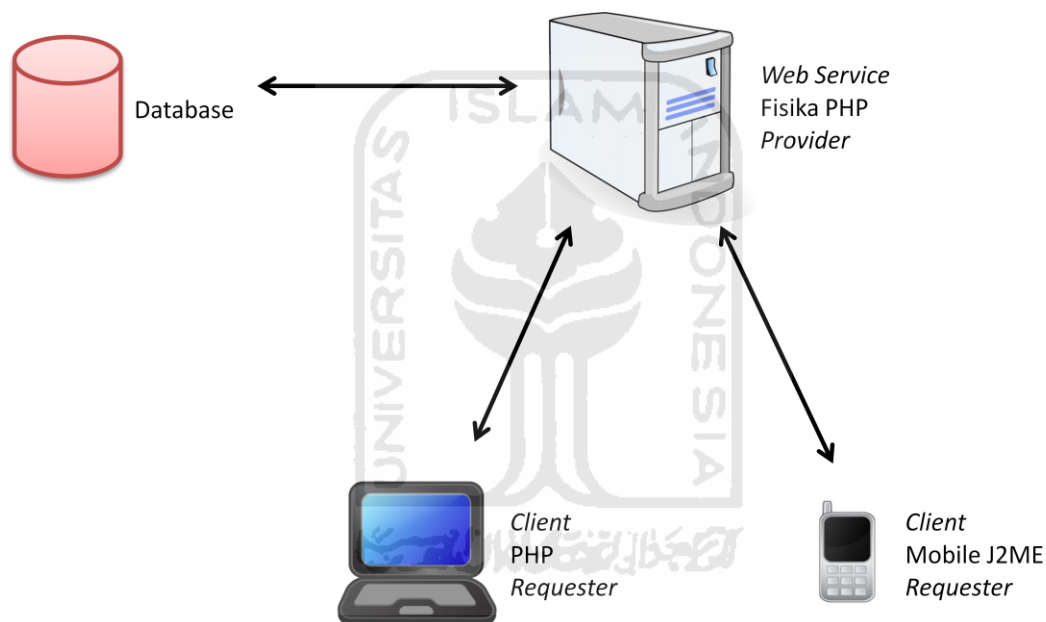
Pada tahapan ini dibuat sebuah rancangan model arsitektur sistem aplikasi *Web Service* rumus Fisika. *Web Service* sebagai media komunikasi antara *server* rumus Fisika dengan *client*. *Service* yang dibangun untuk mengakses basis data rumus Fisika dibatasi sesuai dengan fungsi aplikasi yang ada.

Dalam pengembangannya, implementasi sistem tersebut membutuhkan sebuah *Web Service* yang akan dibangun dengan bahasa pemrograman PHP dan basis data MySQL serta sebuah *client* yang mengakses *Web Service* tersebut.

Client berperan untuk menampilkan data dari hasil panggilan *service* rumus-rumus Fisika.

Kemudian untuk arsitektur aplikasi yang akan dibangun dengan *Web Service* ini mempunyai akses ke basis data rumus Fisika. Akses yang ada adalah *send and request*, komunikasi yang ada adalah dua arah dimana pada sisi *client* dapat melakukan permintaan dengan parameter tertentu dan melakukan pengiriman data dengan parameter tertentu melalui *Web Service*. Kemudian *Web Service* meneruskan ke basis data rumus Fisika.

Gambar arsitektur yang akan dibangun tersebut ada pada Gambar 3.1.



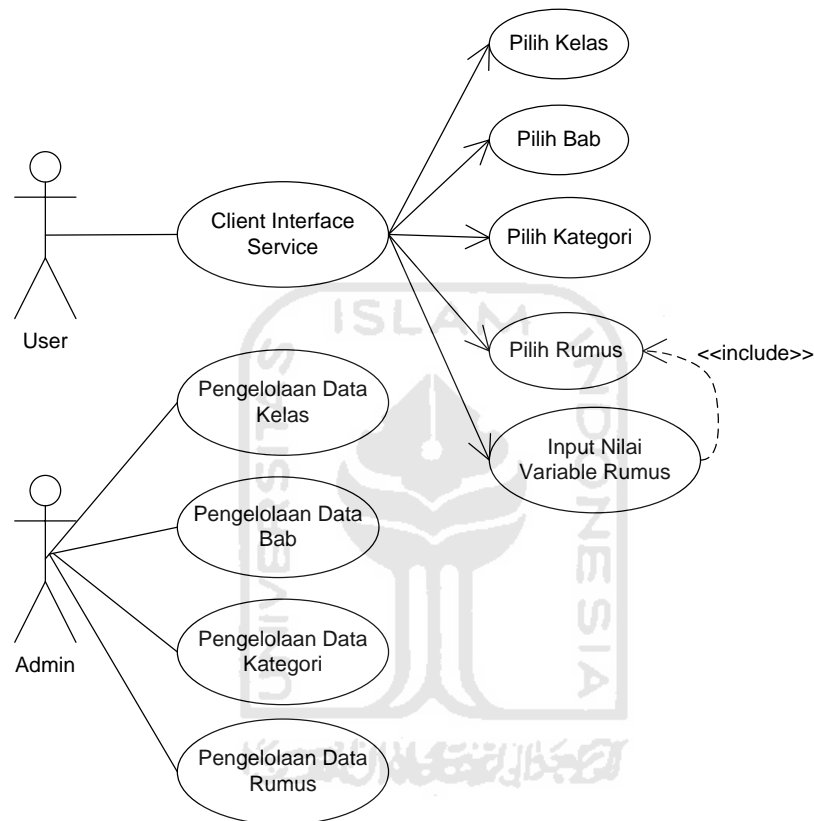
Gambar 3.1 Arsitektur Integrasi Sistem *Web Service* Rumus Fisika

3.2.2 Use Case Diagram

Use case diagram menggambarkan fungsi sistem jika dilihat menurut pandangan yang berada diluar sistem (aktor). Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana berinteraksi dengan dunia luar. Dengan diagram ini dapat diketahui cakupan dari sistem, siapa saja aktor yang berperan dalam sistem, dan interaksi antara aktor dengan sistem.

Pada aplikasi *Web Service* sebagai pendukung mata pelajaran Fisika SMA, *use case* menjelaskan hubungan antara sistem dengan aktor. Hubungan dapat

berupa *input* aktor kedalam sistem atau *output* sistem kepada aktor. Aktor dapat berupa seorang siswa, guru, pegawai, orang tua atau selainnya yang digambarkan dengan *user* dan aktor berupa *admin*. Setiap aktor mempunyai wilayah interaksi yang berbeda. Berikut ini (Gambar 3.1) adalah gambar yang menjelaskan *Web Service* pendukung mata pelajaran Fisika SMA dalam model *use case diagram*:



Gambar 3.2 *Use Case Diagram* Aplikasi *Web Service* Rumus Fisika SMA

Berikut aktor yang terlibat dalam proses *use case diagram*:

1. *User*

User adalah siapa saja yang mengakses *Web Service*, baik itu guru, siswa, pegawai, orang tua, dan sebagainya.

2. *Admin*

Admin adalah seorang yang berwenang untuk memanipulasi data rumus-rumus Fisika.

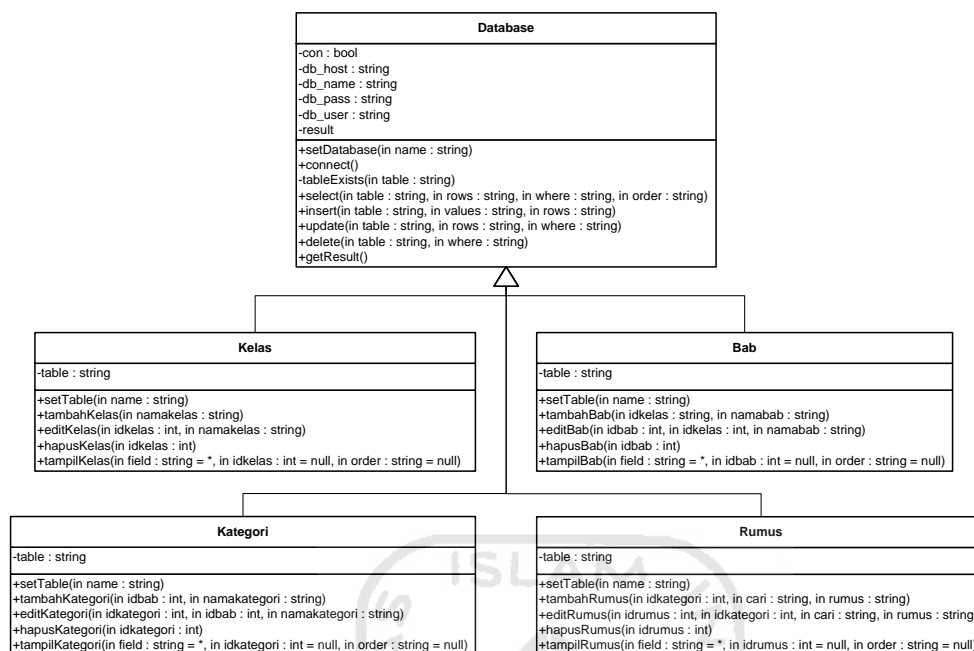
3.2.3 Class Diagram

Class diagram digunakan untuk visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak digunakan. *Class diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan *object* beserta hubungan antarkelas di dalam sistem seperti *containment*, pewarisan, asosiasi dan lain-lain. *Class diagram* adalah inti dari desain berorientasi objek. *Class diagram* menggambarkan tipe objek di dalam sistem dan relasi diantara objek-objek tersebut. Gambar 3.3 adalah *class diagram Web Service* rumus Fisika. Pada *class diagram Web Service* rumus Fisika terdapat enam metode yang digunakan untuk interaksi dengan *client*.



Gambar 3.3 *Class Diagram Web Service* Rumus Fisika

Class diagram pada gambar 3.4 berisi metode-metode yang digunakan untuk memanipulasi basis data oleh *admin*. Terdiri dari satu *parent class* dan empat *child classes*. Kelas dengan nama *Database* berisi metode-metode yang umum digunakan untuk manipulasi tabel pada basis data. Sedangkan empat kelas lainnya berisi metode-metode yang khusus digunakan untuk manipulasi setiap tabel yang berada dalam basis data rumus Fisika, yaitu tabel kelas, bab, kategori, dan rumus.

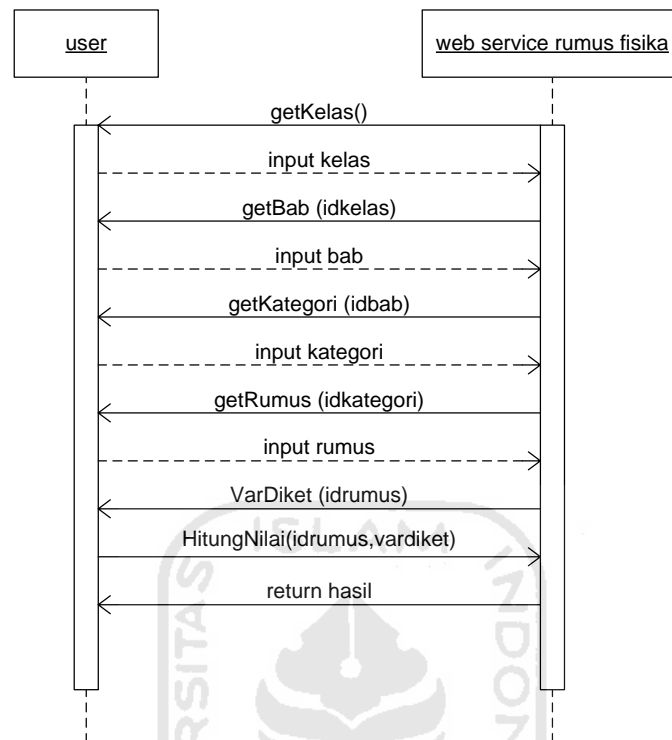


Gambar 3.4 *Class Diagram* untuk manipulasi data kelas, bab, kategori, dan rumus pada sisi *admin*

3.2.4 *Sequence Diagram*

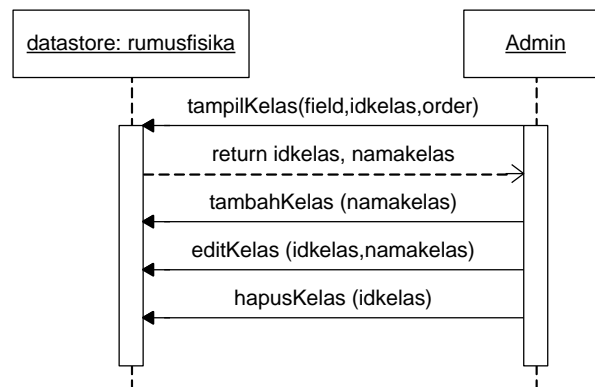
Sequence diagram digunakan untuk menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi vertical (waktu) dan dimensi horisontal (objek-objek yang terkait). Gambar 3.5 menunjukkan *sequence diagram* aplikasi *Web Service* pendukung mata pelajaran Fisika SMA.

Pada saat *user* berinteraksi dengan *Web Service*, maka akan ditampilkan daftar kelas dengan memanggil fungsi `getKelas()`, setelah kelas dipilih maka akan ditampilkan daftar bab dengan memanggil fungsi `getBab()` sesuai dengan kelas yang dipilih. Begitu juga dengan kategori sesuai dengan bab yang dipilih serta rumus sesuai dengan kategori yang dipilih. Setelah memilih rumus yang dikehendaki maka *user* akan memasukkan nilai dari variabel-variabel yang diketahui dan *Web Service* akan menghitungnya dan menampilkan hasilnya.



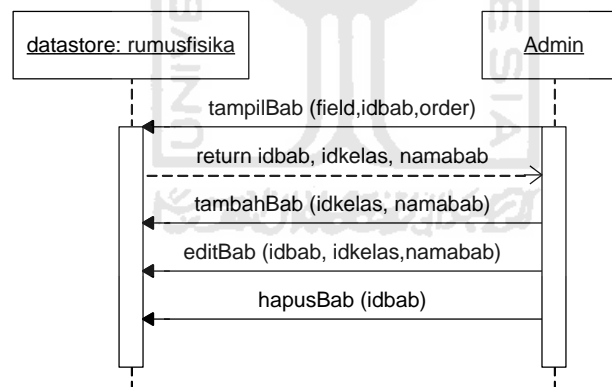
Gambar 3.5 *Sequence Diagram* Web Service Rumus Fisika SMA

Gambar 3.6 menunjukkan *sequence diagram* manipulasi data kelas oleh *admin*. Pemanggilan setiap fungsi dilakukan oleh *admin* untuk menampilkan, menambah, mengedit, dan menghapus data kelas pada basis data.



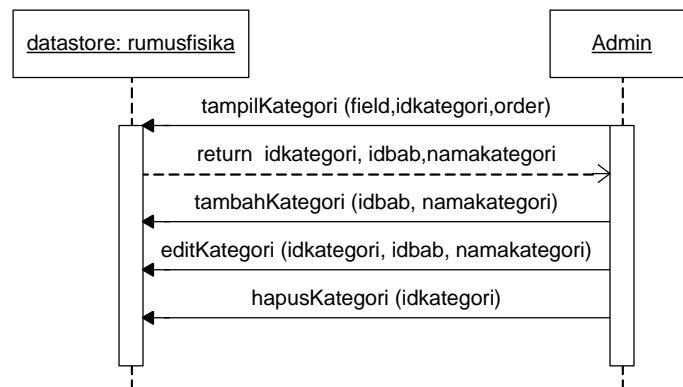
Gambar 3.6 *Sequence Diagram* manipulasi data kelas oleh *admin*

Gambar 3.7 menunjukkan *sequence diagram* manipulasi data bab oleh *admin*. Pemanggilan setiap fungsi dilakukan oleh *admin* untuk menampilkan, menambah, mengedit, dan menghapus data bab pada basis data.



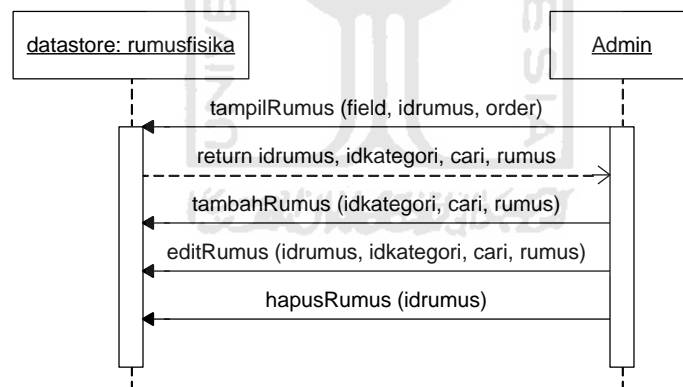
Gambar 3.7 *Sequence Diagram* manipulasi data bab oleh *admin*

Gambar 3.8 menunjukkan *sequence diagram* manipulasi data kategori oleh *admin*. Pemanggilan setiap fungsi dilakukan oleh *admin* untuk menampilkan, menambah, mengedit, dan menghapus data kategori pada basis data.



Gambar 3.8 *Sequence Diagram* manipulasi data kategori oleh *admin*

Gambar 3.9 menunjukkan *sequence diagram* manipulasi data rumus oleh *admin*. Pemanggilan setiap fungsi dilakukan oleh *admin* untuk menampilkan, menambah, mengedit, dan menghapus data rumus pada basis data.



Gambar 3.9 *Sequence Diagram* manipulasi data rumus oleh *admin*

3.3 Perancangan Basis Data

Implementasi *Web Service* rumus Fisika membutuhkan basis data sebagai tempat penyimpanan rumus-rumus Fisika. Setiap rumus diklasifikasikan sesuai dengan kelas, bab, kategori, dan variabel dari rumus yang dicari.

3.3.1 Rancangan Struktur Tabel

1) Kelas

Tabel kelas merupakan tabel yang menyimpan informasi tentang nama kelas pada tingkatan Sekolah Menengah Atas. Tabel 3.1 menunjukkan struktur tabel kelas.

Tabel 3.1 Struktur Tabel Kelas

Nama Field	Tipe Data	Ukuran	Keterangan	Jenis Field
<u>idkelas</u>	int	3	Id Kelas	Primary Key
namakelas	varchar	25	Nama Kelas	

2) Bab

Tabel bab merupakan tabel yang menyimpan informasi tentang nama-nama bab pada mata pelajaran Fisika SMA. Tabel 3.2 menunjukkan struktur tabel bab.

Tabel 3.2 Struktur Tabel Bab

Nama Field	Tipe Data	Ukuran	Keterangan	Jenis Field
<u>idbab</u>	int	5	Id Bab	Primary Key
idkelas	int	3	Id Kelas	Foreign Key
namabab	varchar	30	Nama Bab	

3) Kategori

Tabel kategori merupakan tabel yang menyimpan informasi tentang nama-nama kategori rumus Fisika SMA. Tabel 3.3 menunjukkan struktur tabel kategori.

Tabel 3.3 Struktur Tabel Kategori

Nama Field	Tipe Data	Ukuran	Keterangan	Jenis Field
<u>idkategori</u>	int	5	Id Kategori	Primary Key
idbab	int	5	Id Bab	Foreign Key
namakategori	varchar	30	Nama Bab	

4) Rumus

Tabel rumus merupakan tabel yang menyimpan rumus-rumus Fisika berdasarkan variabel yang akan dicari. Tabel 3.4 menunjukkan struktur tabel rumus.

Tabel 3.4 Struktur Tabel Rumus

Nama Field	Tipe Data	Ukuran	Keterangan	Jenis Field
<u>idrumus</u>	int	5	Id Rumus	Primary Key
idkategori	int	3	Id Kategori	Foreign Key
cari	varchar	5	Variabel Cari	
rumus	varchar	50	Rumus	

3.3.2 Relasi Tabel

Gambar 3.10 menunjukkan relasi tabel dari basis data rumus Fisika sebagai tempat penyimpanan rumus-rumus Fisika SMA.



Gambar 3.10 Relasi Tabel

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi *Web Service*

Implementasi rumus Fisika SMA berbasis *Web Service* ini berhasil dibangun dengan membangun satu *Web Service* yang menyediakan perhitungan rumus-rumus Fisika menggunakan bahasa pemrograman berbasis *web* yaitu PHP dengan basis data MySQL. Dan untuk pengujian digunakan dua jenis *client* yaitu, yang pertama dibangun berbasis *web* dan yang kedua berbasis *mobile*.

4.1.1 Implementasi *Web Service* Rumus Fisika Berbasis PHP

Dalam proses pembuatan *Web Service* rumus Fisika terdapat beberapa fungsi-fungsi yang dibuat untuk mengakses basis data MySQL. *Service* tersebut yang nantinya dipanggil dan digunakan untuk membangun sistem perhitungan rumus Fisika berbasis *Web Service*. Beberapa fungsi yang terbentuk ditunjukkan pada tabel 4.1.

Tabel 4.1 Fungsi-fungsi yang digunakan

Nama Fungsi	Parameter	Output	Keterangan
getKelas	-	Nama-nama kelas	Fungsi untuk menampilkan data dari tabel kelas.
getBab	idkelas	Nama-nama bab	Fungsi untuk menampilkan data dari tabel bab berdasarkan kelas.
getKategori	idbab	Nama-nama kategori	Fungsi untuk menampilkan data dari tabel kategori berdasarkan bab.

getRumus	idkategori	Rumus-rumus Fisika	Fungsi untuk menampilkan rumus-rumus Fisika berdasarkan kategori.
VarDiket	idrumus	Variabel-variabel yang diketahui nilainya.	Fungsi untuk menampilkan variabel yang diketahui atau variabel yang akan dimasukkan nilainya oleh <i>client</i> dari rumus yang dipilih.
HitungRumus	idrumus, vardiket	Hasil perhitungan.	Fungsi untuk menampilkan hasil perhitungan dari rumus yang dipilih setelah memasukkan nilai variabel yang diketahui.

Fungsi-fungsi pada tabel 4.1 lalu didaftarkan pada dokumen WSDL. Tujuan didaftarkannya fungsi adalah agar *Web Service* mengenali fungsi-fungsi apa saja yang tersedia dan parameter-parameternya. Berikut adalah cara pendaftaran fungsi pada WSDL.

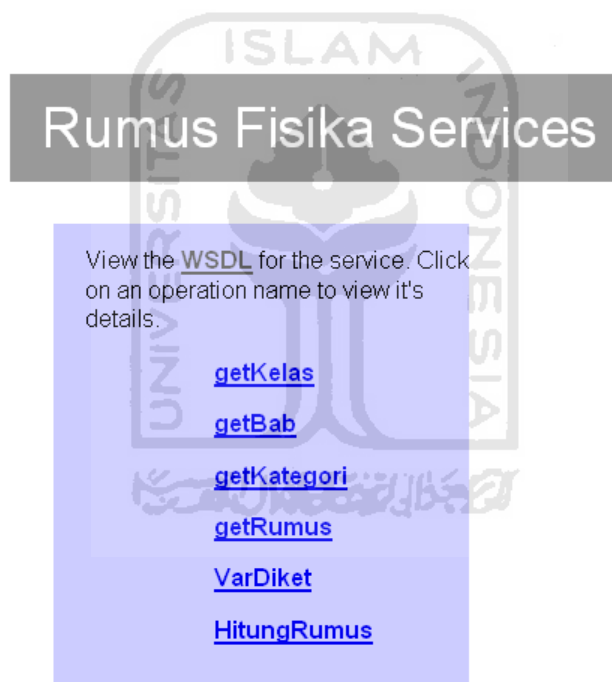
```
register (nama_fungsi string, parameter_input array,
parameter_output array, namespace string, soapaction
string)
```

Seperti yang dilihat, fungsi *register* mempunyai lima parameter. Parameter pertama diisi dengan nama fungsi yang ingin didaftarkan, misal: `getBab`. Parameter kedua adalah parameter *input* fungsi dengan tipe data array, dimana nama parameternya akan menjadi *key* dan tipe datanya akan menjadi *value*/isinya, misal: `array("namakelas"=>"xsd:string")`. Parameter ketiga adalah

parameter *output* dengan tipe data array, cara pendefisiannya sama dengan parameter *input*. Parameter keempat dan kelima adalah *name space* dan *soap action*. Parameter *name space* diisi dengan nama yang telah ditentukan sebagai *service*, misal: ***RumusFisikaService***. Parameter *soap action* umumnya diisi dengan *name space* diikuti tanda pagar (#) lalu diikuti nama fungsi yang didaftarkan, misal: ***RumusFisikaService#getBab***.

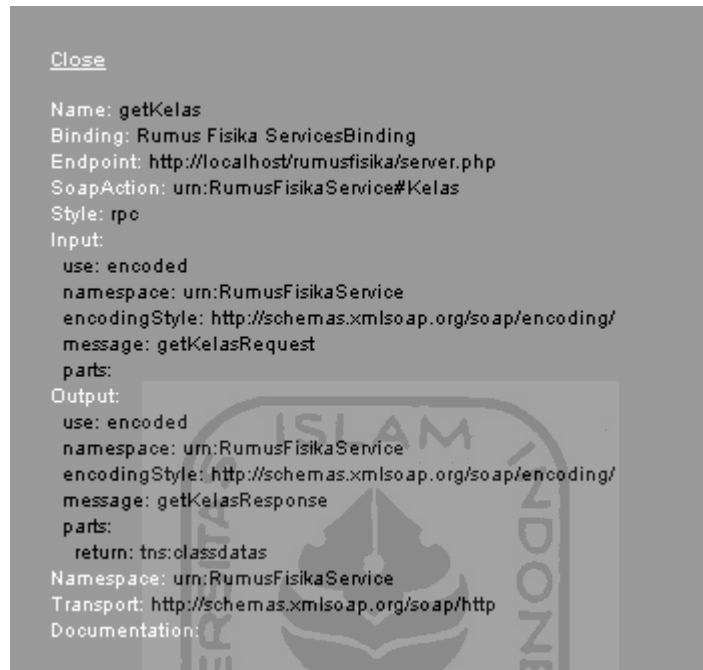
Proses integrasi sistem dengan *Web Service* ini dapat dilakukan dengan penggunaan dokumen WSDL yang dapat diakses pada alamat WSDL-nya yaitu: **<http://localhost:80/rumusfisika/server.php?wsdl>**.

Gambar 4.1 menunjukkan daftar fungsi atau *method* yang terdaftar pada WSDL.



Gambar 4.1 Kumpulan *Service* rumus Fisika

Untuk mengetahui detail *request* dan *response* dari masing-masing *service*, bisa dilakukan dengan memilih salah satu *service* yang tersedia. Setelah memilih salah satu *service* yang tersedia maka akan ditampilkan detail *request* dan *response* dari *service* tersebut seperti pada gambar 4.2 menampilkan detail *service* dari `getKelas()`.



Gambar 4.2 Detail Service getKelas()

Dan hasil dari skema WSDL-nya ditunjukkan pada gambar 4.3.

```

- <definitions targetNamespace="urn:RumusFisikaService">
+ <types></types>
  <message name="getKelasRequest"/>
+ <message name="getKelasResponse"/></message>
+ <message name="getBabRequest"/></message>
+ <message name="getBabResponse"/></message>
+ <message name="getKategoriRequest"/></message>
+ <message name="getKategoriResponse"/></message>
+ <message name="getRumusRequest"/></message>
+ <message name="getRumusResponse"/></message>
+ <message name="VarDiketRequest"/></message>
+ <message name="VarDiketResponse"/></message>
+ <message name="HitungRumusRequest"/></message>
+ <message name="HitungRumusResponse"/></message>
+ <portType name="Rumus Fisika ServicesPortType"/></portType>
+ <binding name="Rumus Fisika ServicesBinding" type="tns:Rumus Fisika ServicesPortType"/></binding>
+ <service name="Rumus Fisika Services"/></service>
</definitions>

```

Gambar 4.3 Hasil Skema WSDL

4.2 Hasil Implementasi *Web Service*

Sistem yang telah dikembangkan adalah metode perhitungan rumus-rumus Fisika SMA yang didukung dengan teknologi *Web Service* dan XML. *Server* dibangun dengan PHP dan basis data MySQL serta dua *client*, yang pertama berbasis *web* dan dibangun dengan PHP, yang kedua berbasis *mobile* dan dibangun dengan J2ME. Untuk mengakses *service* yang ada, *client* tidak perlu berurusan dengan basis data yang digunakan. Ketika ada perubahan basis data pada masing-masing *service*, *client* yang menggunakan *service* tersebut tetap akan berjalan dan tidak perlu mengetahui tentang detail basis data yang digunakan.

4.2.1 Hasil Implementasi *Web Service* Pada *Client* Berbasis *Web*

Client berbasis *web* yang digunakan dalam penelitian ini dibangun dengan bahasa pemrograman PHP dan diakses menggunakan *web browser*. *Client* akan memanggil dan memanfaatkan fungsi-fungsi yang telah disediakan oleh *web service provider*.

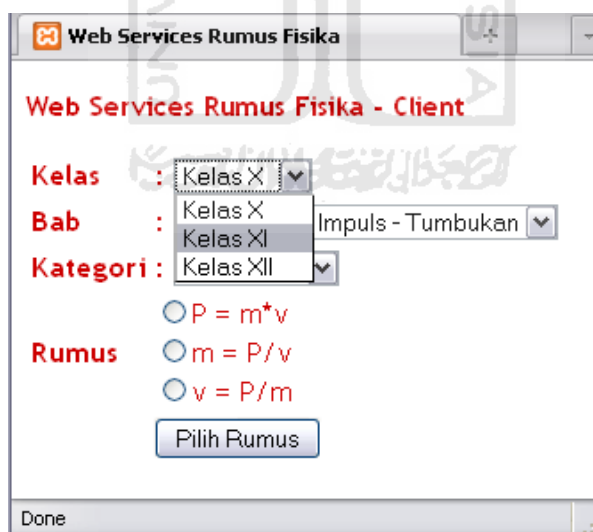
Halaman *client* dapat dimodifikasi dengan *interface* yang dikehendaki oleh pihak *client* sendiri. Didalam penelitian ini penulis memilih untuk menampilkan halaman *client* secara sederhana agar lebih mudah dipahami.

Gambar 4.4 menunjukkan halaman awal *client* yang menampilkan kelas, bab, kategori, dan rumus yang akan dipilih. Rumus-rumus Fisika yang ada didalam sistem diklasifikasikan berdasarkan kelas, bab, kategori rumus, dan rumus berdasarkan variabel yang dicari.



Gambar 4.4 Halaman Awal *Web Service Client*

Gambar 4.5 menunjukkan bahwa *client* mulai mengakses *Web Service* rumus Fisika dengan memanggil fungsi `getKelas()` dan menampilkannya pada halaman berupa *drop down list*. Perlu dipahami bahwa kelas yang dimaksud disini adalah kelas atau tingkatan pada Sekolah Menengah Atas (SMA).



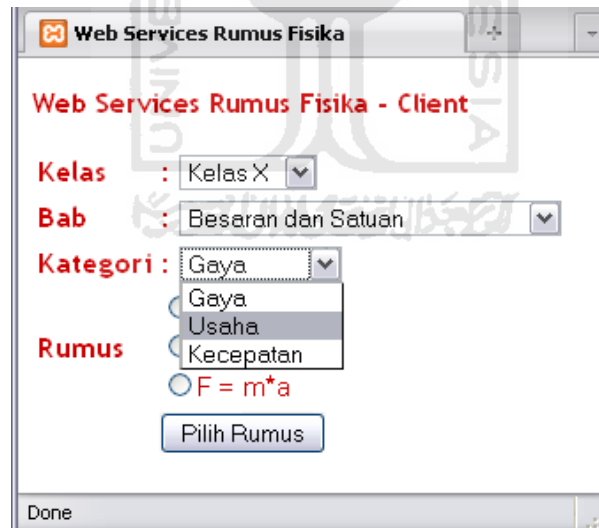
Gambar 4.5 Pilihan Kelas

Ketika kelas dipilih, *client* akan memanggil fungsi `getBab()` dengan parameter `idkelas` yang dipilih lalu menampilkan bab dari kelas yang dipilih dalam bentuk *drop down list*. Hal ini dapat dilihat pada gambar 4.6.



Gambar 4.6 Pilihan Bab

Sama seperti sebelumnya, gambar 4.7 menunjukkan pilihan kategori setelah bab ditentukan yaitu dengan cara memanggil fungsi `getKategori()` dengan parameter `idbab` yang dipilih.



Gambar 4.7 Pilihan Kategori

Setelah menentukan kategori rumus, maka akan muncul rumus-rumus yang dikehendaki berdasarkan variabel yang dicari. Hal ini dapat dilihat pada gambar 4.8.



Web Services Rumus Fisika - Client

Kelas : Kelas X

Bab : Besaran dan Satuan

Kategori : Kecepatan

Rumus

$v = s/t$

$t = s/v$

$s = v \cdot t$

Pilih Rumus

Done

Gambar 4.8 Pilihan Rumus

Setelah memilih rumus yang nilainya dicari, maka muncul *form* untuk memasukkan nilai dari variabel-variabel yang diketahui. Dalam hal ini, *client* memanggil fungsi `VarDiket()` dengan parameter `idrumus` untuk menentukan variabel-variabel yang diketahui nilainya serta menampilkan dalam bentuk *form input*. Hal ini ditunjukkan pada gambar 4.9.

Web Services Rumus Fisika

Web Services Rumus Fisika - Client

Kelas : Kelas X

Bab : Besaran dan Satuan

Kategori : Kecepatan

Rumus

$v = s/t$

$t = s/v$

$s = v \cdot t$

Pilih Rumus

Masukkan Nilai Variabel Yang Diketahui

Nilai s : 34

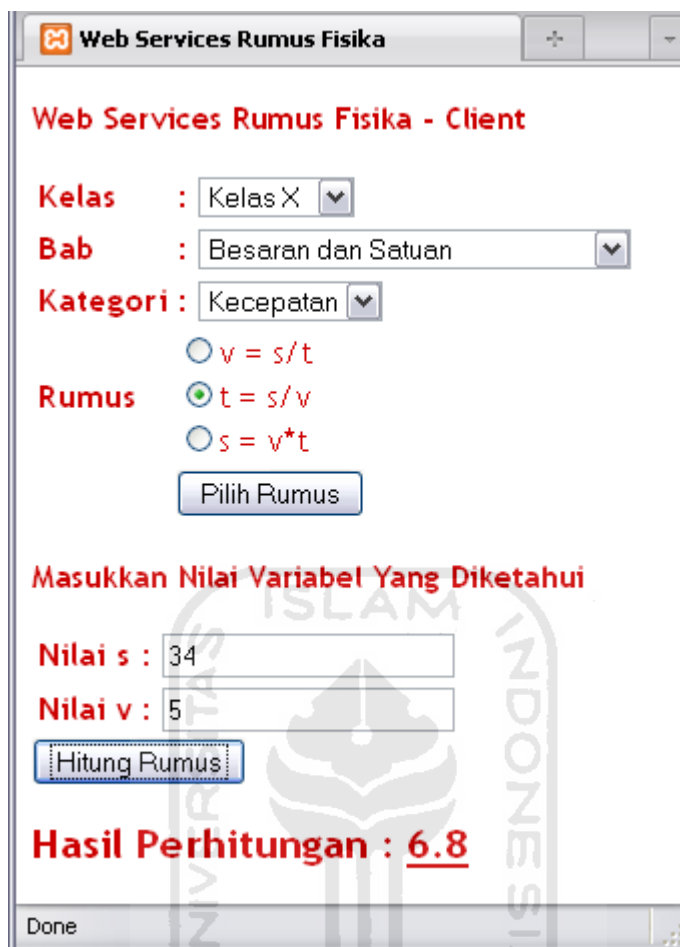
Nilai v : 5

Hitung Rumus

Done

Gambar 4.9 Memasukkan nilai variabel yang diketahui

Pada akhirnya *client* menampilkan hasil perhitungan dengan cara memanggil fungsi `HitungRumus()` dengan parameter nilai dari variabel-variabel yang diketahui. Hal ini dapat dilihat pada gambar 4.10.



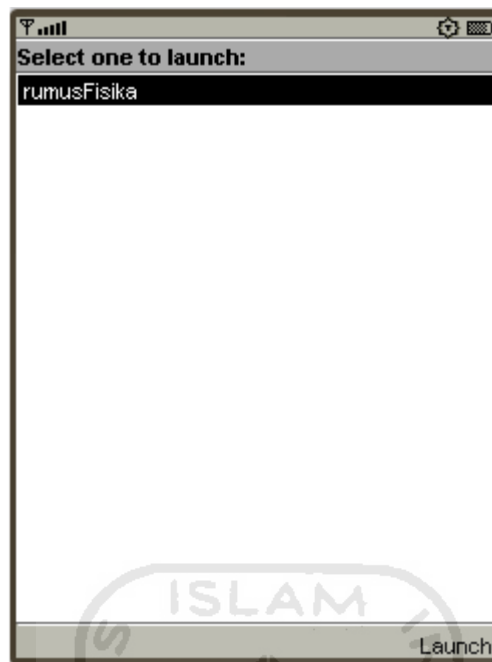
Gambar 4.10 Hasil Perhitungan

4.2.2 Hasil Implementasi Web Service Pada Client berbasis Mobile

Dalam penelitian ini, *client* berbasis *mobile* dibangun menggunakan J2ME (*Java 2 Micro Edition*). J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya (Shalahuddin dan Rosa, 2010).

Web Service client menggunakan J2ME memiliki konsep yang sama dengan *client* pada PHP dari sisi pemanggilan fungsi dan parameter. Yang membedakan adalah *library* yang digunakan untuk berkomunikasi melalui *Web Service*, jika di PHP menggunakan NuSOAP, maka di J2ME menggunakan WSOAP atau *Wingfoot SOAP*. WSOAP adalah salah satu *library* yang dapat digunakan untuk komunikasi melalui *Web Service* pada aplikasi *handphone*.

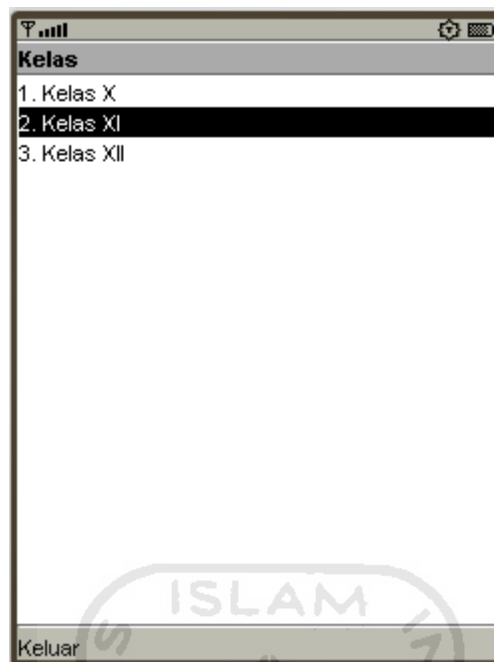
Gambar 4.11 menunjukkan halaman aplikasi sebelum dibuka.



Gambar 4.11 Aplikasi sebelum dibuka

Ketika aplikasi rumusFisika dibuka, maka *client* akan mulai mengakses *Web Service* dan memanggil fungsi `getKelas()` serta menampilkannya pada aplikasi berupa *list*.

Gambar 4.12 menunjukkan tampilan dari hasil pemanggilan fungsi `getKelas()`. Hasilnya adalah menampilkan semua nama kelas Sekolah Menengah Atas yang berada pada basis data tabel kelas.



Gambar 4.12 Pilihan Kelas

Setelah memilih kelas maka *client* akan memanggil fungsi `getBab()` dengan parameter `idkelas` dan meminta data bab berdasarkan nama kelas yang dipilih. Gambar 4.13 menunjukkan hasil tampilan semua nama bab yang berada pada basis data tabel bab berdasarkan nama kelas yang dipilih.



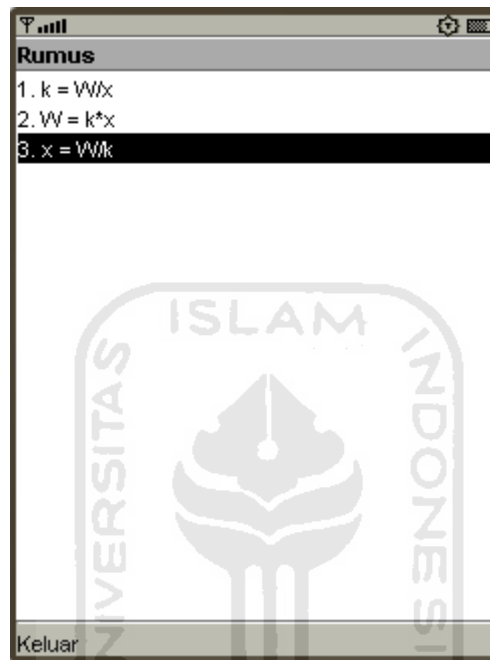
Gambar 4.13 Pilihan Bab

Tidak berbeda dengan sebelumnya, *client* hanya tinggal memanggil fungsi yang ada di *server* berdasarkan pilihan bab yang dilakukan oleh *user*. Pilihan kategori muncul dengan memanggil fungsi `getKategori()` dengan parameter `idbab` yang dipilih seperti pada gambar 4.14.



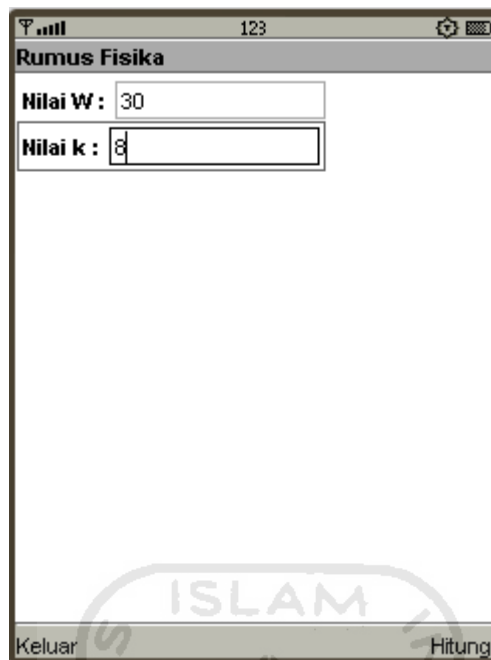
Gambar 4.14 Pilihan Kategori

Pilihan rumus muncul dengan dipanggilnya fungsi `getRumus()` oleh *client* dengan parameter `idkategori`. Halaman pada gambar 4.15 adalah halaman terakhir dalam memilih rumus, selanjutnya akan ditampilkan variabel-variabel yang diketahui dan ditampilkan dalam bentuk *form*.



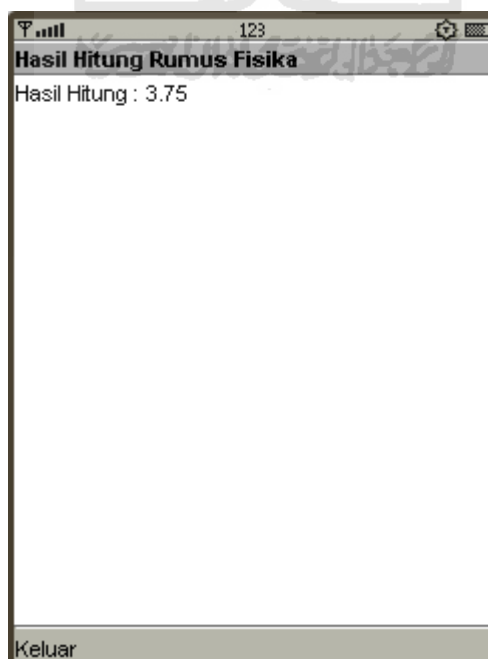
Gambar 4.15 Pilihan Rumus

Gambar 4.16 menunjukkan *form* yang muncul setelah memilih rumus. *Form* ditampilkan berdasarkan variabel yang diketahui nilainya oleh *client*. *Client* akan mengisi nilainya lalu menekan tombol "Hitung".



Gambar 4.16 Form variabel-variabel yang diketahui

Setelah memasukkan nilai dari variabel-variabel yang diketahui, ditampilkanlah hasil dari perhitungan rumus Fisika. Prosesnya yaitu memanggil fungsi `HitungRumus()` dan memasukkan parameter nilai variabel-variabel yang diketahui lalu menampilkan hasilnya pada gambar 4.17.



Gambar 4.17 Hasil Perhitungan

4.3 Halaman Admin

Halaman *admin* adalah halaman yang digunakan untuk memanipulasi data rumus Fisika. Halaman ini yang berinteraksi dengan basis data rumus Fisika. Dengan halaman ini, *admin* dapat memanipulasi data rumus-rumus Fisika seperti menambah data, meng-*edit* data, dan menghapus data.

Gambar 4.18 menunjukkan halaman *admin* untuk memanipulasi data kelas, serta melihat kelas-kelas yang telah di-*input*. Gambar 4.19 menunjukkan halaman *admin* untuk memanipulasi data bab, serta melihat bab-bab yang telah di-*input*. Gambar 4.20 menunjukkan halaman *admin* untuk memanipulasi data kategori dan gambar 4.21 menunjukkan halaman *admin* untuk memanipulasi data rumus.



Gambar 4.18 Halaman Admin Untuk Manipulasi Data Kelas



Gambar 4.19 Halaman *Admin* Untuk Memanipulasi Data Bab



Gambar 4.20 Halaman *Admin* Untuk Memanipulasi Data Kategori

Formula	Unit	Unit	Nama	Kategori	Kelas
F	$W \cdot s$	Usaha	Besaran dan Satuan	Kelas X	
W	F/s	Usaha	Besaran dan Satuan	Kelas X	
a	F/m	Gaya	Besaran dan Satuan	Kelas X	
m	F/a	Gaya	Besaran dan Satuan	Kelas X	
F	$m \cdot a$	Gaya	Besaran dan Satuan	Kelas X	
t	s/v	Kecepatan	Besaran dan Satuan	Kelas X	
s	$v \cdot t$	Kecepatan	Besaran dan Satuan	Kelas X	
p	$m \cdot v$	Momentum	Momentum - Impuls - Tumbukan	Kelas X	
m	p/v	Momentum	Momentum - Impuls - Tumbukan	Kelas X	
v	p/m	Momentum	Momentum - Impuls - Tumbukan	Kelas X	
k	W/x	Konstanta Pegas	Gelombang Bunyi	Kelas XI	
W	$k \cdot x$	Konstanta Pegas	Gelombang Bunyi	Kelas XI	
x	W/k	Konstanta Pegas	Gelombang Bunyi	Kelas XI	
F	$-k \cdot y$	Gaya Pegas	Gelombang Bunyi	Kelas XI	
E	$k \cdot Q/(r^2)$	Kuat Medan Listrik	Listrik Statis	Kelas XI	
W	$1/2 \cdot C \cdot V^2$	Energi Kapasitor	Listrik Statis	Kelas XI	
m'	$m \cdot (\sqrt{1 - (v^2/c^2)})$	Massa dan Energi	Relativitas	Kelas XII	
L'	$L \cdot (\sqrt{1 - (v^2/c^2)})$	Kontraksi Lorentz	Relativitas	Kelas XII	

Kategori :

Cari :

Rumus :

Gambar 4.21 Halaman Admin Untuk Memanipulasi Data Rumus

4.4 Pengujian Sistem

Pengujian sistem adalah suatu kegiatan yang dilakukan untuk mencoba dan menguji kinerja aplikasi yang telah dibuat serta mencari kelemahan yang masih ada pada aplikasi sistem kemudian memperbaiki kelemahan yang ada sehingga aplikasi bekerja dengan baik. Pengujian bertujuan untuk mendapatkan kinerja yang optimal sehingga dapat digunakan untuk menyelesaikan masalah. Setiap pengujian menggunakan variabel masukan yang disesuaikan.

4.4.1 Pengujian Halaman Klien Berbasis Web

Halaman klien adalah halaman yang digunakan oleh pengguna untuk mengakses *Web Service* rumus Fisika. Pengujian akan dilakukan dengan memberi masukan (*input*) berupa nilai numerik, alphabet, karakter serta memberi masukan yang melebihi digit dan pengujian yang lainnya.

Pengujian pertama yaitu dengan memasukkan nilai berupa bilangan bulat minimal satu digit dan maksimal lima belas digit. Pengujian dengan satu digit

dapat dilihat pada gambar 4.22 dan pengujian dengan 15 digit dapat dilihat pada gambar 4.23. Kedua pengujian ini berhasil berjalan dengan sempurna.

Web Services Rumus Fisika - Client

Kelas : Kelas X

Bab : Besaran dan Satuan

Kategori : Kecepatan

Rumus

$v = s/t$

$t = s/v$

$s = v*t$

Pilih Rumus

Masukkan Nilai Variabel Yang Diketahui

Nilai v : 2

Nilai t : 5

Hitung Rumus

Hasil Perhitungan : 10

Gambar 4.22 Pengujian dengan nilai satu digit.

Web Services Rumus Fisika - Client

Kelas : Kelas X

Bab : Besaran dan Satuan

Kategori : Kecepatan

Rumus

$v = s/t$

$t = s/v$

$s = v*t$

Pilih Rumus

Masukkan Nilai Variabel Yang Diketahui

Nilai v : 123456789012345

Nilai t : 2

Hitung Rumus

Hasil Perhitungan : 246913578025000

Gambar 4.23 Pengujian dengan nilai masukan lima belas digit.

Pengujian kedua adalah dengan memasukkan nilai minus kedalam perhitungan. Hal ini dapat dilihat pada gambar 4.24.

Web Services Rumus Fisika - Client

Kelas : Kelas X

Bab : Besaran dan Satuan

Kategori : Kecepatan

Rumus

- $v = s/t$
- $t = s/v$
- $s = v*t$

Pilih Rumus

Masukkan Nilai Variabel Yang Diketahui

Nilai v : -92

Nilai t : 87

Hitung Rumus

Hasil Perhitungan : -8004

Gambar 4.24 Pengujian dengan nilai minus.

Pengujian ketiga adalah dengan memasukkan nilai berupa bilangan riil. Ini juga untuk mengetahui bahwa masukan dapat berupa karakter minus dan titik. Hal ini dapat dilihat pada gambar 4.25.

Web Services Rumus Fisika - Client

Kelas : Kelas X

Bab : Besaran dan Satuan

Kategori : Kecepatan

Rumus

$v = s/t$

$t = s/v$

$s = v \cdot t$

Pilih Rumus

Masukkan Nilai Variabel Yang Diketahui

Nilai v : -5

Nilai t : 3.5

Hitung Rumus

Hasil Perhitungan : -17.5

Gambar 4.25 Pengujian dengan nilai bilangan riil.

Pengujian keempat adalah dengan memasukkan karakter selain titik dan minus. Hasilnya akan dimunculkan *error* dan saran untuk memeriksa nilai yang dimasukkan. Gambar 4.26 menunjukkan hasil dari pengujian tersebut.

Web Services Rumus Fisika - Client

Kelas : Kelas X ▾

Bab : Besaran dan Satuan ▾

Kategori : Kecepatan ▾

Rumus

$v = s/t$

$t = s/v$

$s = v*t$

Masukkan Nilai Variabel Yang Diketahui

Nilai v :

Nilai t :

Hasil Perhitungan :

ERROR!!
Silahkan periksa nilai yang anda masukkan...

Gambar 4.26 Pengujian dengan masukan karakter.

Pengujian kelima adalah dengan memasukkan nilai dengan digit melebihi yang diperbolehkan yaitu lebih dari lima belas digit tanpa titik atau lebih dari sepuluh digit dengan titik dan lebih dari lima digit setelah titik. Pengujian ini dapat dilihat pada gambar 4.27.

Gambar 4.27 Pengujian dengan nilai melebihi digit maksimal.

Dari hasil pengujian dapat disimpulkan bahwa program telah dapat menghitung nilai yang dimasukkan oleh *user* dan mengatasi kemungkinan *error* yang disebabkan oleh *user* dalam memasukkan nilai. *Error handling* ini bertujuan untuk meningkatkan validitas perhitungan data/nilai yang dimasukkan oleh *user*.

4.5 Evaluasi Sistem

Evaluasi sistem adalah suatu kegiatan yang dilakukan untuk mengevaluasi keunggulan dan kelemahan sistem serta untuk menilai apakah sistem tersebut dirancang sesuai dengan yang diharapkan. Kegiatan evaluasi melibatkan *user* atau pengguna aplikasi sebagai sumber evaluasi.

Evaluasi pada penelitian ini menggunakan metode kuisisioner. Metode kuisisioner dipilih karena memiliki kemampuan untuk melakukan generalisasi hasil penelitian. Penggunaan metode kuisisioner bertujuan untuk mengetahui respon dari

responden mengenai penggunaan halaman *client Web Service* rumus Fisika. Responden terdiri dari 10 orang dengan berbagai macam profesi yang rata-rata adalah mahasiswa. Setiap responden diminta untuk merespon setiap pernyataan dengan: Sangat Setuju (SS), Setuju (S), Netral (N), Tidak Setuju (TS), Sangat Tidak Setuju (STS).

Berikut adalah pernyataan-pernyataan yang digunakan pada kuisioner:

1. Program ini mudah untuk digunakan.
2. Program ini interaktif dan *simple*.
3. Saya merasa senang belajar dengan program ini.
4. Perpaduan pemilihan dan komposisi warna tampilan dalam program ini sesuai.
5. Adanya program ini membantu saya memecahkan masalah yang saya jumpai, yaitu kesulitan memahami pelajaran rumus Fisika.
6. Setelah saya menggunakan program ini, saya menjadi termotivasi untuk lebih tanggap terhadap teknologi.
7. Setelah saya menggunakan program ini, saya termotivasi untuk lebih giat belajar.

Masing-masing pernyataan kuisioner yang diberikan bertujuan untuk mendapatkan informasi dari responden dan tujuannya adalah sebagai berikut:

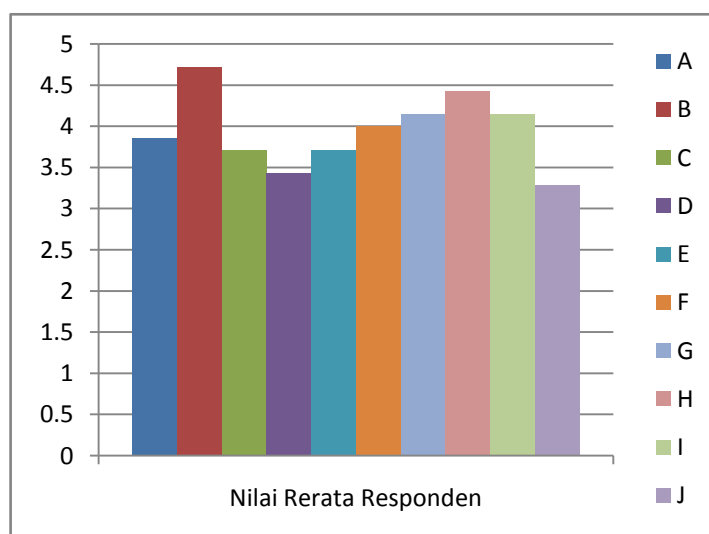
1. Pernyataan pertama, kedua, ketiga, dan keempat bertujuan untuk mengetahui kemudahan interaksi dengan program ini.
2. Pernyataan kelima bertujuan untuk mengetahui apakah program ini dapat menjadi solusi bagi seorang yang sulit memahami rumus Fisika.
3. Pernyataan keenam bertujuan untuk mengetahui motivasi tanggap teknologi yang didapat setelah menggunakan program ini.
4. Pernyataan ketujuh bertujuan untuk mengetahui apakah program ini dapat membantu meningkatkan semangat belajar.

Tabel 4.2 menunjukkan perhitungan hasil penilaian responden.

Tabel 4.2 Hasil Penilaian Responden

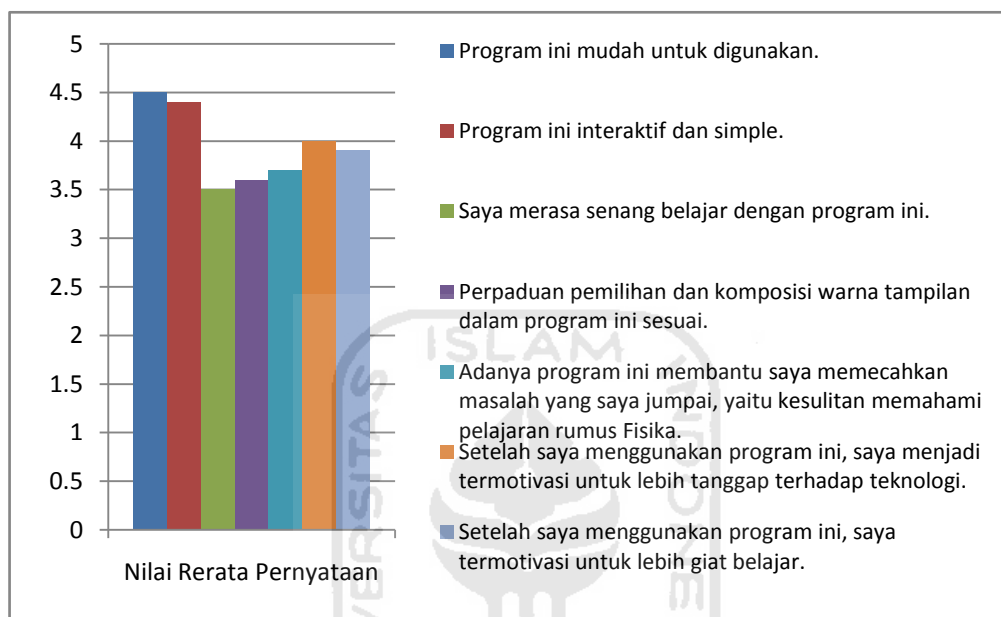
Pernyataan	Responden										Rerata
	A	B	C	D	E	F	G	H	I	J	
Program ini mudah untuk digunakan.	4	5	5	5	5	5	5	5	4	2	4.5
Program ini interaktif dan <i>simple</i> .	4	5	5	5	4	5	5	4	4	3	4.4
Saya merasa senang belajar dengan program ini.	4	4	3	3	3	3	4	5	4	2	3.5
Perpaduan pemilihan dan komposisi warna tampilan dalam program ini sesuai.	3	4	4	3	2	3	3	5	4	5	3.6
Adanya program ini membantu saya memecahkan masalah yang saya jumpai, yaitu kesulitan memahami pelajaran rumus Fisika.	4	5	3	3	3	3	4	5	4	3	3.7
Setelah saya menggunakan program ini, saya menjadi termotivasi untuk lebih tanggap terhadap teknologi.	4	5	4	3	4	4	4	3	4	5	4
Setelah saya menggunakan program ini, saya termotivasi untuk lebih giat belajar.	4	5	2	2	5	5	4	4	5	3	3.9
Nilai Rerata Responden	3.9	4.7	3.7	3.4	3.7	4	4.1	4.4	4.1	3.3	

Dari hasil penilaian responden dari tabel 4.2 dapat dibuat diagram untuk mengetahui nilai rata-rata dari setiap responden, yaitu pada gambar 4.28.



Gambar 4.28 Diagram Nilai Rerata Responden

Dan pada gambar 4.29 ditunjukkan diagram nilai rata-rata setiap pernyataan. Setiap pernyataan yang dijawab responden, dihitung nilai rata-ratanya lalu ditampilkan dalam bentuk diagram.



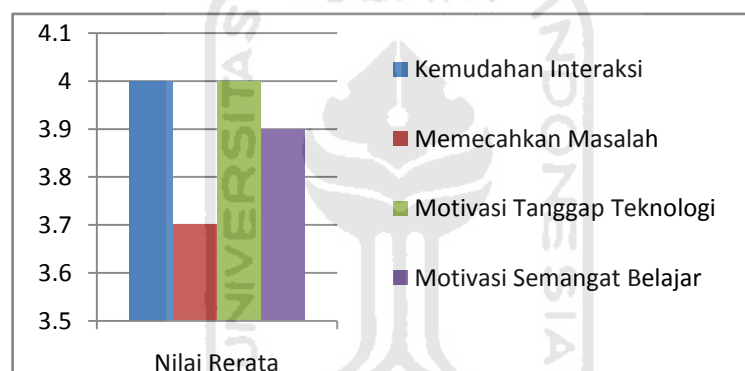
Gambar 4.29 Diagram Nilai Rerata Setiap Pernyataan

Penjelasan hasil penilaian responden berdasarkan setiap pernyataan dari tabel 4.2 adalah sebagai berikut:

1. Pernyataan satu mendapatkan nilai rata-rata 4.5 yang berarti sangat setuju. Dan ini menunjukkan bahwa secara umum program ini mudah sekali untuk digunakan.
2. Pernyataan dua mendapatkan nilai rata-rata 4.4 yang berarti sangat setuju. Ini menunjukkan bahwa program ini interaktif dan *simple*.
3. Pernyataan tiga mendapatkan nilai rata-rata 3.5 yang berarti setuju. Ini menunjukkan bahwa secara umum orang senang belajar menggunakan program ini.
4. Pernyataan empat mendapatkan nilai rata-rata 3.6 yang berarti setuju. Ini menunjukkan bahwa pemilihan warna dan tampilan dalam program ini sesuai.

5. Pernyataan lima mendapatkan nilai rata-rata 3.7 yang berarti setuju. Ini menunjukkan bahwa program ini membantu sebagai solusi memahami mata pelajaran rumus Fisika.
6. Pernyataan enam mendapatkan nilai rata-rata 4 yang berarti setuju. Ini menunjukkan bahwa program ini membantu memotivasi untuk lebih tanggap teknologi.
7. Pernyataan tujuh mendapatkan nilai rata-rata 3.9 yang berarti setuju. Ini menunjukkan bahwa program ini membantu untuk meningkatkan semangat belajar.

Pada gambar 4.30 ditunjukkan diagram hasil evaluasi dari informasi yang diberikan oleh responden.



Gambar 4.30 Diagram Hasil Evaluasi

Dari pembahasan hasil evaluasi dan pengujian dapat disimpulkan bahwa program ini sudah berjalan sesuai yang diharapkan, yaitu dapat membantu meningkatkan motivasi belajar rumus-rumus Fisika SMA, serta memberi kemudahan dalam mempelajarinya.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pembahasan yang telah diuraikan pada bab-bab sebelumnya maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Sistem yang berhasil dibangun berupa *Web Service* sebagai pendukung mata pelajaran Fisika SMA yang dibuat menggunakan bahasa pemrograman PHP dan basis data MySQL.
2. Sistem dikembangkan dengan dibangunnya dua contoh *client* yang dapat memanggil sistem *server*, *client* pertama berbasis *web* dan dibangun dengan PHP, *client* kedua berbasis *mobile* dan dibangun dengan J2ME.
3. Sistem yang dikembangkan dapat memberikan informasi mengenai rumus-rumus Fisika SMA dan perhitungannya berbasis *Web Service*.
4. *Web Service* rumus Fisika menyediakan fungsi-fungsi yang dapat dipanggil oleh *client* dengan *platform* berbeda.

5.2 Saran

Berdasarkan pembahasan dan pengujian yang telah dilakukan, masih banyak terdapat kekurangan dan kelemahan sistem sehingga perlu dikembangkan lagi agar kinerjanya lebih baik. Beberapa hal yang masih perlu dikembangkan adalah:

1. Sistem belum mencakup semua metode perhitungan yang ada dalam rumus-rumus Fisika.
2. Sistem belum dapat menampilkan tahapan-tahapan perhitungan rumus, tetapi hanya hasil akhir dari perhitungan saja yang ditampilkan.

Dengan beberapa kekurangan dan kelemahan tersebut diharapkan dapat dikembangkan sistem yang lebih baik dan lebih menyenangkan.

DAFTAR PUSTAKA

- [DPN03] Departemen Pendidikan Nasional. 2003. *Standar Kompetensi Mata Pelajaran Fisika Sekolah Menengah Atas dan Madrasah Aliyah*. Jakarta: Author.
- [HAR06] Haryanto, R.L. 2006. *Aplikasi Zodiak Menggunakan PHP Web Service*. ilmukomputer.com.
- [IKR10] Ikrimach. 2010. *Implementasi Tutorial Pemrograman Web dengan Metode Ajax dan Web Service*. Tesis, tidak diterbitkan. Yogyakarta: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
- [LUC08] Lucky. 2008. *XML Web Service Aplikasi Desktop, Internet dan Handphone*. Jakarta:Jasakom.
- [PER06] Peranginangin, Kasiman. 2006. *Aplikasi WEB dengan PHP dan MySQL*. Yogyakarta:Penerbit ANDI.
- [POT03] Potts, Stephen. & Kopack, Mike. 2003. *Sams Teach Yourself Web Services in 24 Hours*. Indianapolis,USA:Sams Publishing.
- [RAH07] Raharjo, Budi., et all. 2007. *Mudah Belajar Java*. Bandung: Penerbit INFORMATIKA.
- [SAL09] Salim. 2009. *Integrasi Aplikasi Pemesanan Tiket Pesawat Online Berbasis Web Services*. Tesis, tidak diterbitkan. Yogyakarta: Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Gadjah Mada.
- [SHA10] Shalahuddin, M. & S, Rosa A. 2010. *Pemrograman J2ME (Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile)*. Bandung:Penerbit INFORMATIKA.
- [SIM02] Simpson, John E. 2002. *Just XML*. Yogyakarta:Penerbit ANDI.
- [SIS04] Siswoutomo, W. 2004. *Membangun Web Service Open Source Menggunakan PHP*. Jakarta:PT Gramedia.

LAMPIRAN

Kuisisioner

Petunjuk: Berilah tanda V pada kolom yang paling sesuai dengan jawaban anda.

Pernyataan	SS	S	N	TS	STS
Program ini mudah untuk digunakan.					
Program ini interaktif dan <i>simple</i> .					
Saya merasa senang belajar dengan program ini.					
Perpaduan pemilihan dan komposisi warna tampilan dalam program ini sesuai.					
Adanya program ini membantu saya memecahkan masalah yang saya jumpai, yaitu kesulitan memahami pelajaran rumus Fisika.					
Setelah saya menggunakan program ini, saya menjadi termotivasi untuk lebih tanggap terhadap teknologi.					
Setelah saya menggunakan program ini, saya termotivasi untuk lebih giat belajar.					

SS = Sangat Setuju

S = Setuju

N = Netral

TS = Tidak Setuju

STS = Sangat Tidak Setuju