

Bidang Ilmu : Ilmu  
Komputer

## LAPORAN PENELITIAN



# **Implementasi Web Service untuk Mengintegrasikan Fitur Absensi Learning Management System dan Academic Information System STIMIK ESQ**

### **Tim Peneliti:**

**Ketua : Ahmad Nur Ihsan Purwanto, S.T., M.T**

**Anggota : Muhammad Fuad Zikri**

**PROGRAM STUDI ILMU KOMPUTER**

**LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT  
SEKOLAH TINGGI ILMU MANAJEMEN DAN ILMU KOMPUTER  
(STIMIK ESQ)**

**2021**

## PENGESAHAN

1. Judul Penelitian : Implementasi Web Service untuk Mengintegrasikan Fitur Absensi Learning Management System dan Academic Information System STIMIK ESQ
2. Peneliti :
  - a. Nama Lengkap : Ahmad Nur Ihsan Purwanto
  - b. Jenis Kelamin : Laki-laki
  - c. NIP/NIDN : 0310019102
  - d. Jabatan Struktural : Dosen Tetap
  - e. Jabatan fungsional : Tenaga Pengajar
  - f. Pangkat / Golongan : -
  - g. Fakultas/Program Studi : Ilmu Komputer
  - h. Pusat Penelitian : STIMIK ESQ
  - i. Alamat Institusi : Menara 165 Lt.18-19. Jl. TB Simatupang Kav 1 Cilandak
  - j. Telpon/Faks/E-mail :
3. Jangka Waktu Penelitian : 6 bulan (1 semester)
4. Pembiayaan
  - a. Jumlah biaya yang diajukan ke STIMIK ESQ : Rp. 3.000.000,00

Jakarta, 5 Januari 2021

Mengetahui,

Ketua Program Studi  
Ilmu Komputer

Ketua Peneliti

Ahlijati Nuraminah, S.Kom., M.T.I.  
NIDN: 0317128404

Ahmad Nur Ihsan Purwanto  
NIDN: 0310019102

Kepala LPPM

Danang Indrajaya, S.Si., M.Si  
NIDN: 0311118108

## IDENTITAS PENELITIAN

1. Judul Penelitian : Implementasi Web Service untuk Mengintegrasikan Fitur Absensi Learning Management System dan Academic Information System STIMIK ESQ
2. Peneliti
- a. Nama Lengkap : Ahmad Nur Ihsan Purwanto
- b. NIP/NIK : -
- c. NIDN : 0310019102
- d. Pangkat / Golongan : -
- e. Jabatan Fungsional : Tenaga Pengajar
- f. Fakultas/Prodi : Ilmu Komputer
- g. Pusat Penelitian : LP2M – Menara 165 Lt.18-19
- h. Alamat Institusi : Jl. TB Simatupang Kav.1 Cilandak Jakarta Selatan
- i. Telpon/Faks/E-mail :
3. Anggota Peneliti :

NO	NAMA	KEAHLIAN	ALOKASI WAKTU
1	Muhammad Fuad Zikri	Programing	3 bulan

4. Objek Penelitian : LMS STIMIK ESQ
5. Masa Penelitian
- Mulai : September 2020
- Berakhir : Januari 2021
6. Anggaran yang diusulkan
- Anggran yang diusulkan : Rp. 3.000.000,-
7. Lokasi Penelitian : STIMIK ESQ
8. Hasil yang ditargetkan (temuan baru/paket teknologi/hasil lain), beri penjelasan :
9. Institusi lain yang terlibat : -

## DAFTAR ISI

PENGESAHAN .....	ii
IDENTITAS PENELITIAN.....	iii
DAFTAR ISI .....	iv
DAFTAR GAMBAR .....	vi
DAFTAR TABEL.....	vii
ABSTRAK .....	viii
BAB 1 PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Identifikasi Masalah .....	4
1.3. Rumusan Masalah .....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
BAB 2 TINJAUAN LITERATUR.....	6
2.1. Landasan Teori .....	6
2.1.1. <i>Web</i> .....	6
2.1.2. <i>Web Service</i> .....	6
2.1.3. <i>Webservice RESTful</i> .....	8
2.1.4. <i>Extensible Markup Language (XML)</i> .....	8
2.1.5. <i>Hypertext Transfer Protocol (HTTP)</i> .....	9
2.1.6. Integrasi Data .....	9
2.1.7. Moodle .....	9
2.1.8. <i>Academic Information System (AIS)</i> .....	10
2.2. Penelitian Terdahulu .....	10
2.2.1. Penelitian Astika Ayuning dan Ahmad Ashari (2016).....	10
2.2.2. Penelitian Rifki Indra dan Budi Santoso (2017) .....	10
2.2.3. Penelitian Randi Rizal dan Alam Rahmatullah (2019) .....	11
2.3. Metodologi Pemecahan Masalah.....	14
2.4. Kerangka Pemikiran .....	14
BAB 3 METODE PENELITIAN.....	15
3.1. Alur Penelitian.....	15
3.2. Subyek dan Obyek Penelitian.....	16
3.2.1. Subyek Penelitian.....	16

3.2.2.	Obyek Penelitian .....	16
3.3.	Metode Pengumpulan Data .....	16
3.3.1.	Sumber Data.....	17
3.3.2.	Teknik Pengumpulan Data .....	17
3.4.	Instrumen Penelitian.....	17
3.5.	Analisis, Pengujian dan Interpretasi Data .....	19
3.5.1.	Analisis <i>Web</i> AIS dan LMS STIMIK ESQ .....	19
3.5.2.	Pembuatan Alur Rancangan <i>web service</i> .....	19
3.5.3.	Pembuatan Model <i>web service</i> .....	19
3.5.4.	Analisis Sistem.....	19
BAB 4 HASIL PENELITIAN DAN PEMBAHASAN .....		20
4.1.	Hasil Penelitian.....	20
4.1.1.	Instalasi Moodle .....	20
4.1.2.	Aktivasi Fungsi <i>Web Service</i> .....	23
4.2.	Pembahasan .....	29
4.2.1.	Implementasi <i>Web Service</i> .....	29
4.2.2.	Pengujian <i>Black Box</i> .....	32
BAB 5 PENUTUP .....		37
5.1.	Kesimpulan.....	37
5.2.	Saran.....	37
DAFTAR PUSTAKA .....		38

## DAFTAR GAMBAR

Gambar 1.1 <i>Web Service</i> .....	3
Gambar 2.1 Kerangka Pemikiran .....	14
Gambar 3.1 Alur Penelitian.....	15
Gambar 4.1 <i>Extract file Moodle</i> .....	20
Gambar 4.2 Isi <i>File Moodle</i> .....	21
Gambar 4.3 <i>XAMPP server</i> .....	21
Gambar 4.4 Memilih Bahasa pada Moodle .....	21
Gambar 4.5 <i>Database Setting</i> .....	22
Gambar 4.6 <i>Requirement Moodle</i> .....	22
Gambar 4.7 Manage Courses and Categories .....	23
Gambar 4.8 Menu Course .....	24
Gambar 4.9 Tampilan Menu Course .....	24
Gambar 4.10 Tampilan View Course Skripsi .....	25
Gambar 4.11 Add Attendance.....	25
Gambar 4.12 Enable Web Service .....	26
Gambar 4.13 Menu <i>External Service</i> .....	27
Gambar 4.14 <i>Menu Add Function Web Service</i> .....	27
Gambar 4.15 Token Admin .....	28
Gambar 4.16 Skema <i>Database</i> .....	29
Gambar 4.17 Fungsi Kode Get Course .....	29
Gambar 4.18 Halaman <i>Index.php</i> .....	30
Gambar 4.19 Data Pertemuan .....	31
Gambar 4.20 Daftar Mahasiswa.....	31
Gambar 4.21 <i>Database LMS</i> .....	32
Gambar 4.22 <i>Course Skripsi</i> .....	33
Gambar 4.23 Pertemuan Matakuliah Skripsi .....	34
Gambar 4.24 Data Siswa Pertemuan ke-1.....	35
Gambar 4.25 Data Siswa Pertemuan ke-2.....	35
Gambar 4.26 <i>Database AIS</i> .....	36

## DAFTAR TABEL

Tabel 2.1 Perbedaan SOAP dan REST .....	7
Tabel 2.2 Penelitian Terdahulu .....	12

## ABSTRAK

Judul : Implementasi Web Service untuk Mengintegrasikan Fitur Absensi Learning Management System dan Academic Information System STIMIK ESQ

Sejak virus Covid-19 menyerang ke Indonesia tepatnya awal maret 2020 semua kegiatan yang dilakukan secara *offline* seakan terhenti dan harus dilakukan secara *online*. Kegiatan belajar mengajar di kampus, sekolah, bahkan para pekerja kantoran yang harus datang dan absen pagi ke kantor hanya dilakukan dirumah saja melalui sistem yang ada. Setiap kampus memerlukan platform belajar online untuk kelangsungan kegiatan belajar mengajar selama pandemi COVID-19. Dalam mengatasi kegiatan belajar mengajar secara online STIMIK ESQ mengorganisir platform pembelajaran online yang dinamakan Moodle LMS (*Learning Management System*). STIMIK ESQ menggunakan dua *web* selama kegiatan belajar mengajar *online* yaitu Moodle LMS dan AIS. Moodle LMS digunakan sebagai kelas *online* dan absensi. AIS digunakan untuk kegiatan akademik seperti rekap nilai, rekap absensi, dan rekap kegiatan mahasiswa. Fitur absensi pada Moodle LMS dan AIS masih dimasukkan secara manual karena kedua sistem belum terdapat integrasi data. Moodle LMS dibangun menggunakan bahasa pemrograman PHP dan *database* MySQL begitu pula AIS dibangun menggunakan bahasa pemrograman PHP dan *database* MySQL yang memungkinkan kedua *web* dapat terintegrasi. Metode yang digunakan penulis untuk mengintegrasikan kedua *web* adalah penggunaan *web service* RestFull. Mengintegrasikan absensi dari Moodle LMS ke AIS agar tidak terjadi *input* absensi secara manual yang dilakukan dua kali. Pada penelitian ini penulis dapat menghasilkan sebuah *web service* metode REST dalam pengambilan data absensi. Dari hasil tersebut penelitian ini masih jauh dari kata sempurna, penulis merekomendasikan untuk menambahkan kodingan yang dapat membaca data lebih besar dalam waktu yang singkat dan *double checker* data yang sudah dimasukkan tidak dapat dimasukkan kembali.

Kata kunci: Covid-19, *Database*, *Web Service*, *REST*

# **BAB 1**

## **PENDAHULUAN**

Bab pertama dari skripsi ini menjelaskan tentang latar belakang masalah yang terjadi pada penelitian, identifikasi masalah, rumusan masalah, serta tujuan dari penelitian ini dilakukan. Pada poin terakhir dijelaskan manfaat dari dilakukannya penelitian.

### **1.1. Latar Belakang**

STIMIK ESQ merupakan kampus yang memiliki tiga jurusan yaitu manajemen bisnis, sistem informasi dan ilmu komputer. Sebagai kampus, STIMIK ESQ perlu mengelola data akademik mahasiswa yang banyak menggunakan sistem. Terciptalah *web academic information system* (AIS) yang dibangun menggunakan bahasa pemrograman PHP, *database* MySQL dan framework CI. *Web* ini *digunakan* untuk mengelola segala kegiatan dan data akademik kampus seperti data mahasiswa, data dosen, absensi kelas, kegiatan kampus, dan lainnya.

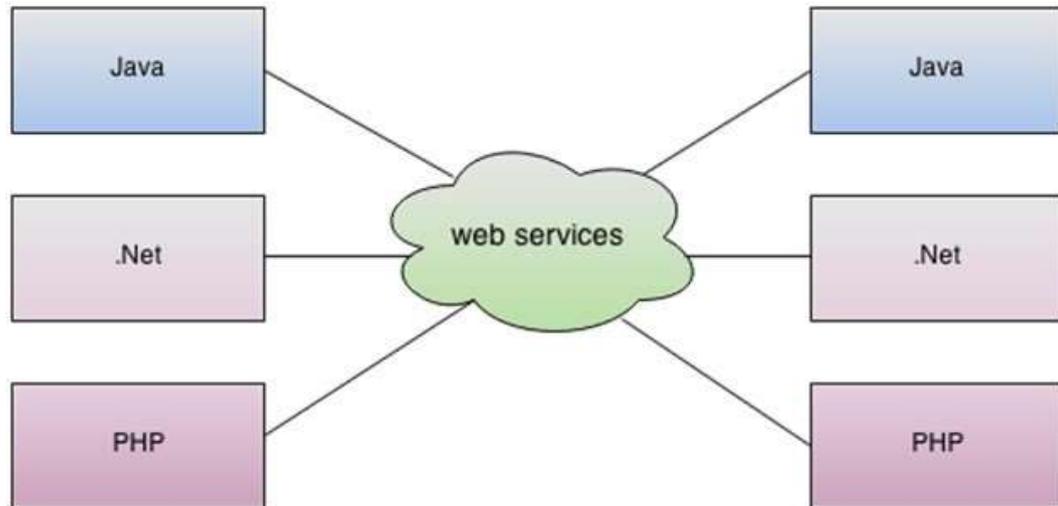
Pandemi yang masih berjalan satu tahun terakhir membuat kegiatan belajar mengajar dilakukan secara *online*. Sebagai kampus, STIMIK ESQ membangun sistem *e-learning* menggunakan moodle dengan bahasa pemrograman PHP dan *database* MySQL dengan nama *Learning Management System* (LMS). Alur penggunaan LMS yang cukup sama dengan *google classroom* dari *google* yaitu dengan *login* akun, masuk ke kelas sesuai mata kuliah dan mengikuti arahan tugas sesuai dosen pengampu. Terdapat satu perbedaan antara *google classroom* dan LMS yaitu terdapat fitur absen pada LMS.

Dosen akan membuat slot absen di LMS kemudian membuka absen 15 menit sebelum mata kuliah dimulai dan ditutup 15 menit setelah kelas dari mata kuliah tersebut dimulai. Mahasiswa akan memasukkan absen dan memulai kegiatan belajar. Setelah kegiatan belajar mengajar selesai dilakukan dosen harus memasukkan kembali data absen yang dimasukkan mahasiswa diawal perkuliahan ke AIS. Setiap mahasiswa bisa melihat rekap absen setiap sesi dan setiap mata kuliah pada fitur AIS.

Bagian akademik kampus atau yang disebut dengan BAA berfungsi untuk membuat jadwal setiap mata kuliah di AIS kemudian membuka FRS agar mahasiswa melakukan *enrollment* setiap mata kuliah. Sama halnya dengan absen, kelas mata kuliah di LMS akan dimasukkan ulang dari data *enrollment* mahasiswa dan BAA akan memasukkan secara manual mahasiswa yang mengikuti kelas di AIS ke LMS. Hal ini menyebabkan masalah absen banyak terjadi di setiap semester dengan proses memasukkan data yang dilakukan secara manual dan beberapa dosen masih ada yang bingung dengan alur penggunaan AIS dan LMS tersebut.

Berdasarkan latar belakang serta wawancara yang telah dilakukan penulis terhadap beberapa dosen dan mahasiswa di STIMIK ESQ mendapatkan beberapa masalah integrasi terhadap dua sistem pembelajaran yang digunakan yaitu AIS dan LMS yang dapat disimpulkan. Memasukkan data absen oleh dosen yang dilakukan dua kali pada sistem AIS dan LMS. Absen yang belum terintegrasi ketika mahasiswa tidak hadir, izin, maupun sakit. Dosen merasa kesulitan ketika proses absensi pada kelas umum dimana mahasiswa yang mengikuti sangat banyak. Kalkulasi nilai pada setiap mata kuliah pada sistem AIS yang tidak update. *Input* topik pembahasan per mata kuliah tiap pertemuan yang seharusnya sinkron antara AIS dan LMS.

Integrasi sistem AIS dan LMS merupakan salah satu solusi dalam memasukkan absen dan data yang tidak sinkron di kedua sistem. Aplikasi web AIS dan LMS yang dibangun dengan dasar yang sama yaitu PHP dan MySQL sehingga sangat mungkin untuk diintegrasikan. Integrasi sistem bisa dilakukan dengan menggunakan *web service* seperti pada Gambar 1.1 **Error! Reference source not found.** *Web service* merupakan aplikasi *client-service* atau komponen aplikasi untuk komunikasi antar mesin ke mesin. Terdapat dua jenis teknologi *webservice* yang bisa digunakan yaitu SOAP dan Restful.



Gambar 1.1 Web Service

SOAP, memiliki kepanjangan *Simple Object Access Protocol*, adalah sebuah spesifikasi protokol untuk pertukaran pesan atau informasi terstruktur dalam implementasi *web service* di jaringan komputer. SOAP menggunakan *Extensible Markup Language* (XML) sebagai format pesannya, dan biasanya bergantung pada protokol layer aplikasi lainnya, terutama *Hypertext Transfer Protocol* (HTTP) dan *Simple Mail Transfer Protocol* (SMTP), untuk transmisi dan negosiasi pesan. SOAP banyak digunakan untuk perusahaan atau vendor karena tidak terbuka untuk umum. Proses pengembangan yang memakan banyak biaya dan realtif sulit untuk digunakan orang awam.

*REpresentational State Transfer* (REST) adalah sebuah arsitektur software untuk sistem terdistribusi semisal *web*. REST telah berkembang sebagai model desain *web* servis yang dominan saat ini. Istilah *representational state transfer* dikenalkan dan didefinisikan pada tahun 2000 oleh Roy Fielding dalam disertasi doktoralnya. Beliau merupakan salah satu penulis utama spesifikasi HTTP versi 1.0 dan 1.1. Sesuai konstrainnya, REST biasa disebut dengan “RESTful”. RESTful yang bersifat *open source* sehingga dapat diakses dan dikembangkan siapa saja. *Tools* yang sudah tersedia dan banyak tutorial penggunaan RESTful diinternet.

RESTful memiliki beberapa kelebihan yang tidak dimiliki SOAP yaitu dapat digunakan oleh beragam jenis bahas pemrograman dan beragam platform,

lebih sederhana dan *simple* proses penggunaannya dibandingkan SOAP, lebih mudah untuk dipelajari, menggunakan HTTP disetiap bagian yang dimilikinya, dan aplikasi android yang menggunakan Rest API jauh lebih cepat dibandingkan aplikasi android berbasis *web view*.

## **1.2. Identifikasi Masalah**

Berdasarkan latar belakang serta wawancara yang telah dilakukan terhadap beberapa dosen dan mahasiswa di STIMIK ESQ dapat diidentifikasi beberapa masalah yaitu:

1. Administrasi memasukkan absen untuk kelas besar cukup merepotkan karena harus memindahkan absen secara manual dari LMS ke AIS
2. Tidak adanya integrasi data antara sistem AIS dan LMS

## **1.3. Rumusan Masalah**

Berdasarkan identifikasi yang telah diuraikan di atas, maka perumusan masalah yang akan menjadi acuan dalam penelitian ini sebagai berikut:

1. Bagaimana cara mengimplementasikan sistem integrasi absensi antara *web Academic Information System (AIS)* dan *Learning Management System (LMS)* pada STIMIK ESQ menggunakan *web service RESTful*?

## **1.4. Tujuan Penelitian**

Adapun tujuan yang ingin dicapai dalam penelitian ini berdasarkan rumusan masalah yang telah disebutkan di atas adalah:

1. Melakukan implementasi sistem integrasi absensi antara *web Academic Information System (AIS)* dan *Learning Management System (LMS)* pada STIMIK ESQ menggunakan *web service RESTful*.

## **1.5. Manfaat Penelitian**

Berdasarkan tujuan penelitian tersebut, maka manfaat yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Manfaat untuk STIMIK ESQ  
Hasi dari penelitian ini diharapkan agar dapat bermanfaat bagi pihak kampus guna meningkatkan kualitas sistem dan fitur yang dimanfaatkan para penggunanya.
2. Manfaat untuk peneliti

Penelitian ini diharapkan dapat menambah wawasan bagi peneliti dan menjadi sebuah pengalaman baru yang dapat memotivasi peneliti untuk mendalami bidang *web service*.

3. Manfaat untuk akademisi

Penelitian ini diharapkan untuk dapat bermanfaat bagi peneliti yang akan melanjutkan penelitian ini ke tingkat selanjutnya, dan menjadi acuan untuk penelitian di bidang *web service* dan integrasi sistem.

## **BAB 2**

### **TINJAUAN LITERATUR**

#### **2.1. Landasan Teori**

Bab ini akan memaparkan teori-teori pendukung yang berkaitan dengan penelitian ini secara terperinci. Adapun teori-teori yang digunakan pada penelitian ini adalah sebagai berikut:

##### **2.1.1. Web**

*Web* merupakan ruang informasi yang berisikan sumber-sumber daya yang berguna mengidentifikasi Uniform Resource Identifier (URL). Mengakses *Web* dibutuhkan browser *web* yang akan memecah bagian nama server dari URLnya ke dalam alamat Protokol Internet menggunakan basis data terdistribusi yang dikenal dengan istilah Domain Name System (DNS).

##### **2.1.2. Web Service**

*Web service* adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin ke mesin melalui jaringan. Memiliki *interface* yang dideskripsikan dalam format yang dapat dibaca mesin. Definisi ini diberikan oleh W3C (*World Wide Web Consortium*) merupakan badan yang menciptakan dan mengembangkan standar *web service*. *Web service* mempunyai alat penghubung yang direpresentasikan dalam bentuk format machine processable lebih rincinya WSDL (*Web Services Description Language*). Sistem lain saling berhubungan dengan *web service* menggunakan pesan SOAP (*Simple Object Access Protocol*) yang umumnya dikirim melalui HTTP (*Hypertext Transfer Protocol*) dalam bentuk XML (*Extensible Markup Language*). SOAP merupakan XML yang mengatur bagaimana request dan respons dari suatu *web service* akan bekerja. Tetapi secara umum *web service* tidak terbatas hanya pada standar SOAP saja.

Saat ini telah berkembang sebuah teknologi yang memungkinkan bentuk akhir dari sebuah program atau aplikasi komputer adalah sebuah *service* atau fungsi yang melakukan sebuah tugas atau proses yang spesifik dan dikenal dengan

istilah *Web Service*. Konsep ini sebenarnya bukanlah sebuah konsep yang baru, berbagai teknologi telah dikembangkan untuk mewujudkan kebutuhan akan koneksi atau hubungan antar aplikasi. Adanya *Web Service* memudahkan perpaduan fungsi dalam membangun sebuah program aplikasi tanpa bergantung lagi pada sistem operasi maupun bahasa pemrograman yang digunakan. Hal ini dimungkinkan karena *web service* berkomunikasi menggunakan sebuah standar standar format data yang universal yaitu XML dan menggunakan protokol *Simple Object Access Protocol* (SOAP). Dengan adanya *web service* menggunakan format data XML, maka *web service* juga mewariskan sifat *multi-tier* dari XML sehingga memungkinkan terjadinya integrasi antar *web service* atau aplikasi (W3C Working Group, 2004). Perbedaan *web service* SOAP dan REST tampak jelas pada Tabel 2.1 Perbedaan SOAP dan REST.

Tabel 2.1 Perbedaan SOAP dan REST

Uraian	SOAP	REST
Protokol Komunikasi	HTTP, HTTPS, SMTP, FTP	HTTP, HTTPS
Penggunaan Bandwidth	Dalam jumlah request yang banyak, relatif boros bandwidth. Hal ini karena banyaknya markup dalam penulisan format XML	Relatif hemat bandwidth, karena markup-markup ekstra seperti pada XML tidak dipakai
Tren Penggunaan	Banyak mulai beralih ke REST, meski masih tetap ada yang mempertahankan, misalnya untuk integrasi aplikasi ke sistem legasi pada sebuah perusahaan	Mulai populer, banyak dipakai oleh penyedia <i>web</i> servis terkemuka, seperti twitter, yahoo!, flickr, bloglines, technorati, google, amazon, eBay, dsb
Aturan Penulisan	Ketat, mengikuti spesifikasi XML (SOAP v1.2)	Tidak ada spesifikasi khusus
Format Respon	XML dengan spesifikasi SOAP. Agak sulit bagi kita untuk membaca langsung dan memahaminya	XML, JSON, atau format plain teks lainnya. Hal ini memudahkan penerima respon membaca dan memahaminya
Attachment File	Bisa (karena dapat mengembalikan respon dalam format binary)	Tidak bisa
Sifat <i>Web Service</i>	Tertutup, lebih ditujukan untuk vendor atau perusahaan tertentu	Terbuka, bisa diakses siapa saja
Caching <i>Web</i>	Relatif sulit	Mudah, karena menggunakan URI
Penggunaan	Standar lama (XML, HTTP) dan baru (SOAP) digunakan bersamaan	Standar yang sudah ada, seperti XML dan HTTP
Tool Pengembangan	Banyak, baik komersial	Beberapa, karena tidak begitu

	maupun opensource	dibutuhkan
Uraian	SOAP	REST
Tool Manajemen	Perlu, bahkan kadang harganya mahal	Menggunakan tool yang sudah ada pada sistem jaringan
Extensible	Bisa, banyak ekstensi termasuk standar WS-*	Relatif tidak ekstensibel
Kemudahan Implementasi	Mudah jika kita sudah memiliki lingkungan berbasis SOAP	Mudah

### 2.1.3. *Webservice RESTful*

Representational State Transfer yang disingkat REST yang merupakan gaya arsitektur untuk penerapan *web service* dalam menerapkan konsep perpindahan antar state. REST populer karena kesederhanaannya, sebagai lawan menciptakan standar baru, kerangka kerja dan teknologi. Keuntungan menggunakan REST yaitu interaksi berbasis REST menggunakan Hypertext Transfer Protocol (HTTP) internet yang merupakan hal yang sudah umum digunakan. Contoh dari pengaturan ini adalah interaksi berbasis REST semua berkomunikasi status mereka menggunakan kode status HTTP standar.

### 2.1.4. *Extensible Markup Language (XML)*

Marchal (2000) menyatakan bahwa XML kepanjangan dari Extensible Markup Language yang dikembangkan oleh W3C. Fleksibilitas XML telah mendorong penggunaannya secara luas untuk pertukaran data dalam berbagai format. XML termasuk ke dalam keluarga Markup Language dan masih satu keluarga dengan HTML (*Hypertext Markup Language*) yang digunakan untuk membuat halaman *web*. Meskipun masih satu keluarga, kedua bahasa tersebut sebenarnya memiliki fungsi yang berbeda. Jika fungsi HTML lebih diutamakan untuk mengatur tampilan, maka fungsi XML lebih diutamakan untuk menyimpan data dan informasi (Utomo, 2013). Selain itu ada satu perbedaan lagi antara HTML dan XML yaitu XML bisa didefinisikan tag milik sendiri. XML adalah bahasa universal yang dapat mempersatukan aplikasi dengan *Web Service* untuk saling berkomunikasi dan bertukar informasi.

### **2.1.5. Hypertext Transfer Protocol (HTTP)**

*Hypertext Transfer Protocol* (HTTP) protokol yang digunakan client dan server untuk bisa berkomunikasi dengan gaya request-response. HTTP menentukan bagaimana format pesan dan bagaimana cara pengirimannya, serta bagaimana *web* server dan browser beraksi dan bereaksi terhadap berbagai perintah.

### **2.1.6. Integrasi Data**

Proses integrasi sistem banyak digunakan seiring dengan perkembangan proses bisnis dan kebutuhan pengguna informasi dalam sebuah organisasi seperti sekolah, kampus, bahkan pemerintahan. Terdapat empat tipe integrasi dalam sistem informasi yaitu integrasi data, integrasi aplikasi, integrasi proses bisnis dan integrasi presentasi . Integrasi data merupakan solusi awal dalam proses integrasi sistim informasi proses ini dilakukan dengan pertukaran data antar basis data yang terdapat pada aplikasi terpisah. Permasalahan pada proses integrasi data adalah perbedaan struktur basis data yang digunakan kedua sistem diintegrasikan serta perbedaan platform.

### **2.1.7. Moodle**

MoodleNet adalah jaringan sosial yang baru untuk berbagi dan menyusun sumber daya pendidikan terbuka dengan para pendidik di seluruh dunia. Ini terintegrasi erat dengan Moodle LMS dan Moodle Workplace untuk membuat sumber daya mudah ditemukan dan digunakan, tetapi dapat berguna bahkan jika tidak menggunakan Moodle LMS. MoodleNet juga dimaksudkan sebagai cara untuk membangun profil sebagai seorang pendidik. MoodleNet dirancang untuk menjadi Teknologi Pendidikan Terbuka federasi dan fleksibel yang tidak dikendalikan oleh satu entitas, atau tunduk pada satu titik kegagalan ([moodle.org](http://moodle.org)).

Moodle menggunakan bahasa pemrograman PHP dan *database* MySQL. Fitur moodle yang dibutuhkan untuk melakukan integrasi sistem *web service* yaitu:

1. *Users*
2. *Courses*
3. *Plugins*

#### 4. Database LMS Moodle

##### **2.1.8. Academic Information System (AIS)**

*Academic Information System* merupakan sistem untuk pengelolaan data akademik dengan teknologi komputer (Sahetapy, 2019). AIS digunakan untuk kepentingan akademik seperti transkrip nilai, absensi, jadwal ujian, FRS, pembayaran, serta pengajuan skripsi. Pengajuan cuti juga dilakukan di sistem AIS

AIS menggunakan bahasa pemrograman PHP, *database* MySQL dan *framework* CI. Fitur AIS yang dibutuhkan untuk melakukan integrasi sistem *web service* yaitu:

1. Database AIS
2. Absensi

##### **2.2. Penelitian Terdahulu**

###### **2.2.1. Penelitian Astika Ayuning dan Ahmad Ashari (2016)**

Pada penelitian Astika Ayuning dan Ahmad Ashari dengan judul pemanfaatan teknologi *web service* untuk integrasi sistem layanan materi pelajaran terdistribusi sangat disinggung tentang peran sekolah dan guru dalam mendidik murid sebagai penyedia sumber materi untuk belajar. Metode pembelajaran yang baik dan materi yang relevan dengan kurikulum yang ada. Pada jurnalnya (Ayuning, 2016) menyatakan bahwa Layanan materi dapat digunakan untuk menghubungkan antar sekolah (penyedia materi) sehingga dapat saling berintegrasi data dalam satu tempat. Tentunya ini akan membawa dampak semakin banyaknya resource materi yang didapatkan. Integrasi data antar sekolah menggunakan *web service* menjawab masalah dalam hal penyediaan materi yang disajikan dalam platform berbeda beda.

###### **2.2.2. Penelitian Rifki Indra dan Budi Santoso (2017)**

Pada penelitian Implementasi *web service* pada integrasi data akademik dengan replika pangkalan data dikti yang dilakukan oleh Rifki Indra dan Budi Santoso pada tahun 2017 dengan proses pelaporan data akademik secara berkala kepada DIKTI dilakukan dua kali setiap semester. Pelaporan dilakukan di awal semester dan di akhir semester. Data akademik yang sangat banyak dan bervariasi

menjadikan sebuah kendala untuk dapat dikirim melalui mekanisme sistem teknologi. Setiap perguruan tinggi harus mengikuti sistem yang dibuat DIKTI. Untuk mengurai permasalahan yang ada, penelitian ini membuat jembatan perantara yang mempermudah proses pengiriman data dari instansi terkait ke DIKTI. Proses pengiriman data tersebut harus melalui beberapa mekanisme seperti integrasi semua data akademik dalam satu *database*, proses mapping, proses insert, update, delete, dan yang terakhir adalah sinkronisasi.

### **2.2.3. Penelitian Randi Rizal dan Alam Rahmatullah (2019)**

Pada penelitian RESTful *web service* untuk integrasi sistem akademik dan perpustakaan universitas perjuangan yang dilakukan oleh Randi Rizal dan Alam Rahmatullah pada tahun 2019 dengan kasus pada universitas perjuangan memiliki banyak sistem informasi yang telah dikembangkan untuk mendukung kegiatan administrasi akademik diantaranya sistem informasi laboratorium, sistem informasi akademik, sistem informasi perpustakaan, sistem informasi anggaran dan sistem informasi analisis dan pengembangan TIK. Banyaknya sistem yang digunakan disetiap unit, perlunya ada integrasi data antar aplikasi. (Randi, 2019) mengatakan Integrasi dapat diartikan adanya saling keterkaitan antara satu sistem dengan sistem yang lain. Pada penelitian penggunaan *web service* mampu mengatasi permasalahan interoperability dan mengintegrasikan sistem yang berbeda.

Tabel 2.2 Penelitian Terdahulu

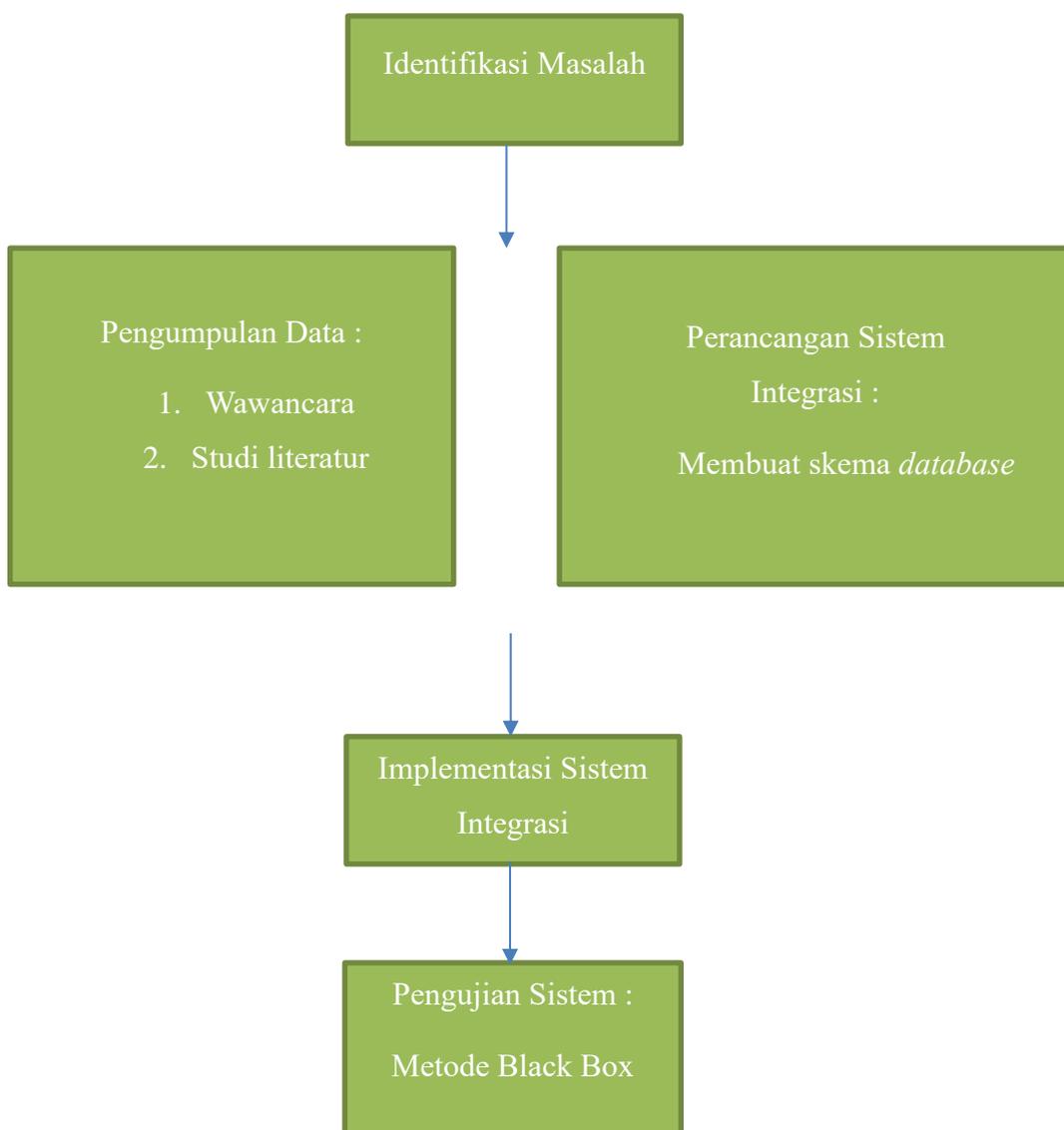
Judul	Peneliti	Tahun	Metode	Kesimpulan	Relevansi
Pemanfaatan teknologi <i>web service</i> untuk integrasi sistem layanan materi pelajaran terdistribusi	Astika Ayuning dan Ahmad Ashari	2016	Studi pustaka, observasi dan wawancara	Sistem layanan materi pelajaran yang menyediakan informasi yang beragam dan saling terintegrasi dalam satu sistem, sehingga beberapa sistem dengan struktur data dan platform yang berbeda dapat saling berkomunikasi dalam mendistribusikan informasi. Integrasi data antar layanan materi pelajaran membuat informasi dan resource materi semakin besar.	Pembuatan <i>web service</i> untuk mendukung pembelajaran guna menghubungkan data antar pelayanan materi sehingga mendapatkan materi yang luas dan lengkap. Menggunakan metode yang mirip dengan peneliti untuk pengumpulan data.
Integrasi data akademik dengan replika pangkalan data dikti	Rifki Indra dan Budi Santoso	2017	Studi literatur dan SDLC	Hasil dari penelitian ini berupa sebuah tools <i>web service</i> yang terdiri tiga modul utama yang dapat mengakomodir kebutuhan-kebutuhan pelaporan data akademik ke DIKTI yang sudah berjalan. Tools ini telah digunakan dalam dua tahun terakhir untuk keperluan pelaporan. Penelitian ini melakukan integrasi data terhadap <i>web</i> DIKTI karna memiliki banyak variable yang bervariasi dan pelaporan yang dilakukan dua kali setiap semester membuat data semakin banyak.	Pembuatan <i>web service</i> pada sistem <i>web</i> dikti dan instansi agar bisa menjembatani banyaknya data yang bervariasi setiap kali dimasukkan. Relevan dengan penelitian peneliti yang membangun model <i>web service</i> untuk menejembatani dua <i>web</i> yang digunakan untuk proses belajar.
RESTful <i>web service</i> untuk integrasi sistem akademik dan perpustakaan universitas perjuangan	Randi Rizal dan Alam Rahmatullah	2019	Observasi sistem dan analisis kebutuhan.	Penerapan teknologi <i>web service</i> dengan menggunakan arsitektur REST pada sistem informasi akademik dan sistem informasi perpustakaan mampu mengintegrasikan kedua sistem tersebut.	Penelitian yang menghubungkan beberapa sistem menjadi terintegrasi untuk memudahkan pemasukan data dan duplikasi data dengan menggunakan <i>web service</i> .

Judul	Peneliti	Tahun	Metode	Kesimpulan	Relevansi
				<p>Sehingga proses <i>input</i> dan verifikasi data hanya dilakukan satu kali, hal tersebut mengatasi terjadinya duplikasi data dan mengurangi perkerjaan <i>input</i> data. Format pertukaran data antar sistem menggunakan format JSON yang tidak bernegara (stateless), sehingga memudahkan untuk dapat diakses oleh bahasa pemrograman apapun, arsitektur maupun sistem operasi yang berbeda. Pada penelitian ini belum ada diskusi lebih mendalam mengenai keamanan data dan otentikasi antar sistem, sehingga perlu ada penelitian kembali mengenai keamanan dan otentikasi pada web <i>service</i> dengan menerapkan konsep single sign on (SSO).</p>	

### 2.3. Metodologi Pemecahan Masalah

Terdapat beberapa metodologi pemecahan masalah yang dapat diimplementasikan untuk menyelesaikan masalah terkait *web service* yang menjadi penghubung dua atau lebih sistem informasi *web*. Menggunakan *web service* SOAP yang menggunakan XML sebagai format transfer data dan *web service* RESTful yang cukup fleksibel yang dapat menggunakan XML dan Jason sebagai format transfer data.

### 2.4. Kerangka Pemikiran

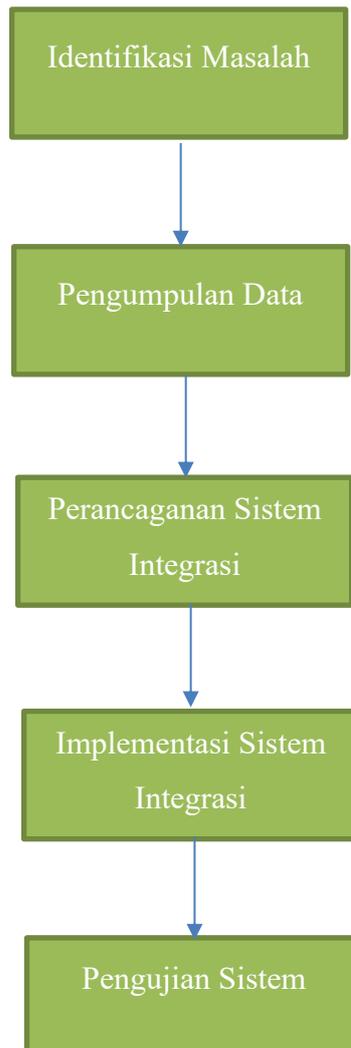


Gambar 2.1 Kerangka Pemikiran

## BAB 3

### METODE PENELITIAN

#### 3.1. Alur Penelitian



Alur penelitian merupakan tahapan – tahapan yang akan dilakukan penulis pada penelitian ini untuk mendapatkan hasil yang maksimal dan dilakukan secara runtut seperti pada . Adapun tahapan-tahapannya adalah sebagai berikut:

1. Identifikasi masalah yang terjadi pada penelitian yakni tidak adanya integrasi data antara sistem AIS dan LMS mengakibatkan pemasukan data

absen yang dilakukan secara manual dari LMS ketika absensi permatakuliah kemudian di rekap dan dipindahkan ke AIS. Beberapa dosen yang diinterview mengatakan sangat sulit untuk memindahkan absensi secara manual ketika dalam kelas umum yang mahasiswanya bisa mencapai 100 orang. Integrasi menggunakan *web service* merupakan solusi dari identifikasi masalah absensi.

2. Pengumpulan data diawali dengan mengumpulkan dokumentasi Moodle LMS dan AIS. Infrastruktur yang membangun *web* LMS dan AIS seperti *database*, bahasa pemrograman dan alur kerja sistim.
3. Perancangan sistem integrasi dilakukan dengan membuat rancangan skema *web service* dari pemasukan data absensi di LMS kemudian disimpan data absensi pada *web service* dan melakukan penembakan data ke *database* AIS sesuai *field* yang sudah diteliti. Membuat alur *database* gabungan LMS dan AIS untuk memudahkan proses implementasi.
4. Implementasi sistem integrasi dilakukan dengan membuat *web service* RESTful dengan bahasa pemrograman php. Penulis menggunakan aplikasi Visual Studio Code.
5. Pengujian sistem dilakukan menggunakan pengujian *black box*.

### **3.2. Subyek dan Obyek Penelitian**

#### **3.2.1. Subyek Penelitian**

Pada penulisan penelitian ini, penulis mengambil subyek penelitian di kampus STIMIK ESQ menara 165 di jalan TB. Simatupang Kav 1, Jakarta Selatan.

#### **3.2.2. Obyek Penelitian**

Pada penulisan penelitian ini, penulis mengambil obyek penelitian pada *Learning Management System* (LMS) dan *Academic Information System* (AIS) yang digunakan STIMIK ESQ dalam sistem akademik.

### **3.3. Metode Pengumpulan Data**

Penelitian ini menggunakan metode pendekatan kuantitatif, dimana berdasarkan studi kasus pada penelitian ini, peneliti mengobservasi sistem LMS dan AIS STIMIK ESQ agar dapat diketahui sejauh mana sistem ini bermanfaat

dan digunakan oleh para pengguna yang dituju untuk menggunakan sistem ini secara aktif. Menggunakan metode wawancara untuk mendapatkan kesimpulan dari masalah yang akan diselesaikan. *Web service* menjadi solusi untuk menghubungkan dua *web* LMS dan AIS

### **3.3.1. Sumber Data**

Data yang digunakan dalam penelitian ini bersumber dari data primer dan data sekunder.

#### 1) Data Primer

Data Primer merupakan data yang berasal langsung dari sumbernya. Data tersebut didapatkan melalui pengamatan langsung atau observasi dan melalui wawancara yang dilakukan kepada dosen Ilmu Komputer dan dosen kelas umum. Data tersebut akan digunakan sebagai acuan dalam implementasi *web service* sistem LMS dan AIS.

#### 2) Data Sekunder

Data sekunder merupakan data yang diambil melalui studi literatur dan dokumen yang membahas mengenai *web service*. Data tersebut dijadikan sebagai pelengkap pada landasan teori penelitian ini.

### **3.3.2. Teknik Pengumpulan Data**

Penelitian ini menggunakan beberapa teknik pengumpulan data, yaitu:

#### 1) Wawancara

Peneliti melakukan wawancara pada beberapa dosen dan mahasiswa untuk mendapatkan masukan terkait masalah absensi pada sistem LMS dan AIS.

#### 2) Studi literatur

Penelitian ini juga memanfaatkan studi pustaka untuk mengumpulkan data-data yang memiliki relevansi terhadap penelitian ini. Hal ini dilakukan dengan melakukan studi mendalam terhadap buku, jurnal, *web* dan internet.

### **3.4. Instrumen Penelitian**

Instrumen penelitian merupakan tools atau alat dan bahan yang membantu serta menunjang penelitian dalam pengumpulan dan pengolahan data pada

penelitian yang dilakukan. Adapun instrumen penelitian yang digunakan dalam pengumpulan dan pengolahan data pada penelitian ini, yaitu:

- 1) Dokumentasi Moodle LMS
- 2) Dokumentasi AIS
- 3) Perangkat Lunak
  - Sistem Operasi Windows 10
  - Microsoft Office
  - Search Engine (Google Chrome)
  - Visual Studio Code
  - XAMPP, sebagai server lokal
- 4) Perangkat Keras
  - NoteBook Lenovo Thinkped X230
  - Processor Intel Core i3 2.6 GHZ c. RAM 6GB
  - HDD 500 GB
  - Mouse Logitech G103 f. Keyboard *External* Sades

### **3.5. Analisis, Pengujian dan Interpretasi Data**

Analisis merupakan metode yang digunakan untuk mencari kebutuhan sistem yang akan diimplementasi. Pengujian dilakukan untuk menguji apakah sistem yang telah diimplementasi berhasil berjalan.

#### **3.5.1. Analisis *Web AIS* dan LMS STIMIK ESQ**

Pada tahap ini peneliti melakukan analisis pengembangan sistem *web AIS* dan LMS. Sistem menggunakan bahasa pemrograman, framework dan *database* apa yang digunakan oleh sistem untuk jadi dasar pembuatan *web service* dan mudah untuk menghubungkan dua sistem AIS dan LMS.

#### **3.5.2. Pembuatan Alur Rancangan *web service***

Pada tahap ini peneliti melakukan rancangan pembuatan *web service* dari data yang telah didapat dari analisis sistem. Membuat rancangan *database*, table serta variabel yang menjadi penghubung antara sistem AIS dan LMS untuk pembuatan model *web service*.

#### **3.5.3. Pembuatan Model *web service***

Pada tahap ini peneliti melakukan implementasi pembuatan source code pengembangan *web service* dari rancangan model yang telah dilakukan pada tahap sebelumnya.

#### **3.5.4. Analisis Sistem**

Pada tahap ini peneliti melakukan analisis lebih lanjut terhadap sistem yang telah dikembangkan sesuai dengan topik dan pengujian sistem.

## BAB 4

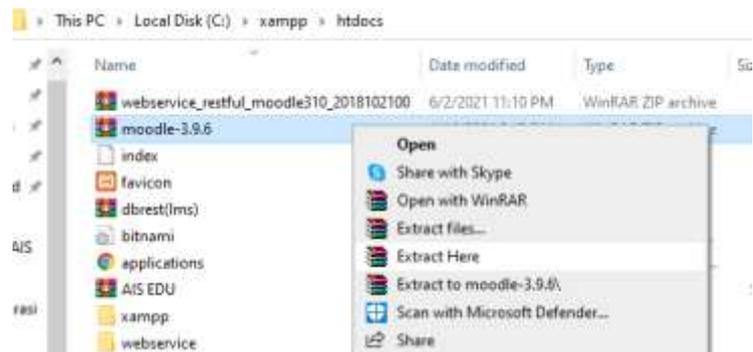
### HASIL PENELITIAN DAN PEMBAHASAN

#### 4.1. Hasil Penelitian

##### 4.1.1. Instalasi Moodle

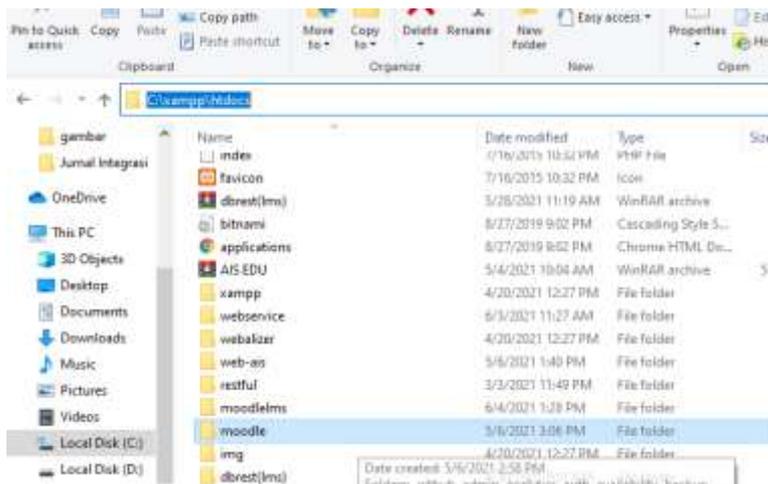
Data Moodle LMS yang digunakan pada kampus STIMIK ESQ tidak bisa digunakan secara *local*. Maka dari itu penulis melakukan instalasi Moodle secara manual yang menyesuaikan dengan spesifikasi Moodle LMS yang digunakan pada STIMIK ESQ. Penulis menggunakan Moodle 3.9 yang didownload pada situs <https://download.moodle.org> dan XAMPP v3.2.4.

- a. Download Moodle 3.9 (sesuaikan dengan versi)
- b. *Extract* file yang telah di download



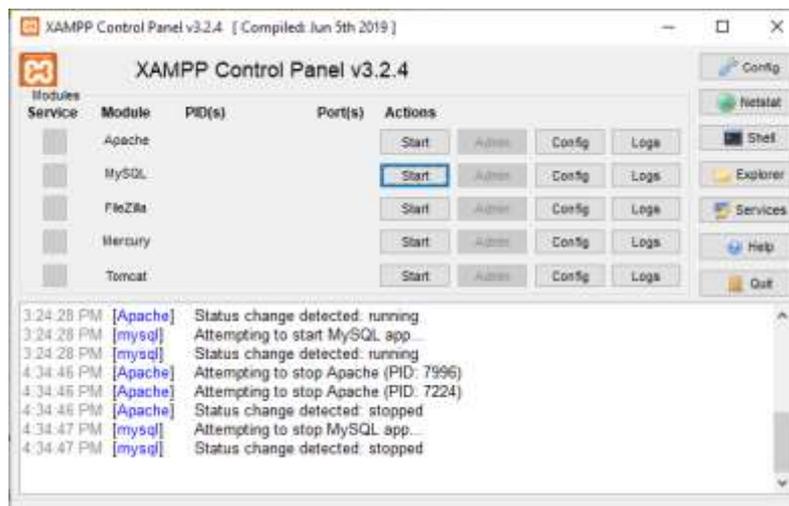
Gambar 4.1 *Extract file Moodle*

- c. Memindahkan seluruh file yang di *extract* di penyimpanan htdocs (c:\xampp\htdocs)



Gambar 4.2 Isi File Moodle

- d. Buka xampp dan aktifkan Apache dan MySQL dengan mengklik tombol Start



Gambar 4.3 XAMPP server

- e. Buka file moodle di browser dengan mengetik <http://localhost/moodle/>  
 f. Kemudian memilih bahasa yang akan digunakan



Gambar 4.4 Memilih Bahasa pada Moodle

- g. Tampilan berikutnya memilih tipe *database*, disini penulis memilih MariaDB

h. Langkah selanjutnya melakukan *database setting*

Gambar 4.5 Database Setting

i. Selanjutnya memastikan kesiapan server pc hingga tidak ada chek berwarna merah. Menyelesaikan semua *requirement* yang dibutuhkan hingga mencapai minimal *requirement* yang dibutuhkan agar tidak terjadi error saat menjalankan sistem, lalu klik *continue*.

Server checks

Name	Information	Report	Plugin	Status
database	mysql (5.5.5-10.1.34-MariaDB)	<b>!</b> must be installed and enabled Wrong \$CFP->dbtype   you need to change it in your config.php file, from 'mysql' to 'mariadb'		Check
php_extension	intl	<b>!</b> must be installed and enabled Intl extension is required to improve internationalization support, such as locale aware sorting and international domain names.		Check
php_extension	xmlrpc	<b>!</b> should be installed and enabled for best results The XMLRPC extension is useful for web services and Moodle networking.		Check
php_extension	soap	<b>!</b> should be installed and enabled for best results Installing the optional SOAP extension is useful for web services and some plugins.		Check
php_setting	opcache.enable	<b>!</b> PHP setting should be changed. PHP opcode caching improves performance and lowers memory requirements. OPcache extension is recommended and fully supported.		Check
unicode		<b>!</b> must be installed and enabled		OK

Server checks instalasi moodle

Other checks

Information	Report	Plugin	Status
mysql_full_unicode_support#file_format	<b>!</b> this test must pass Your database uses Antelope as the file format. Full UTF-8 support in MySQL and MariaDB requires the Barracuda file format. Please switch to the Barracuda file format. See the documentation MySQL full unicode support for details.		Check
mysql_full_unicode_support#large_prefix	<b>!</b> this test must pass For full support of UTF-8 both MySQL and MariaDB require you to change your MySQL setting 'innodb_large_prefix' to 'ON'. See the documentation for further details.		Check
site not https	<b>!</b> if this test fails, it indicates a potential problem It has been detected that your site is not secured using HTTPS. It is strongly recommended to migrate your site to HTTPS for increased security and improved integration with other systems.		Check

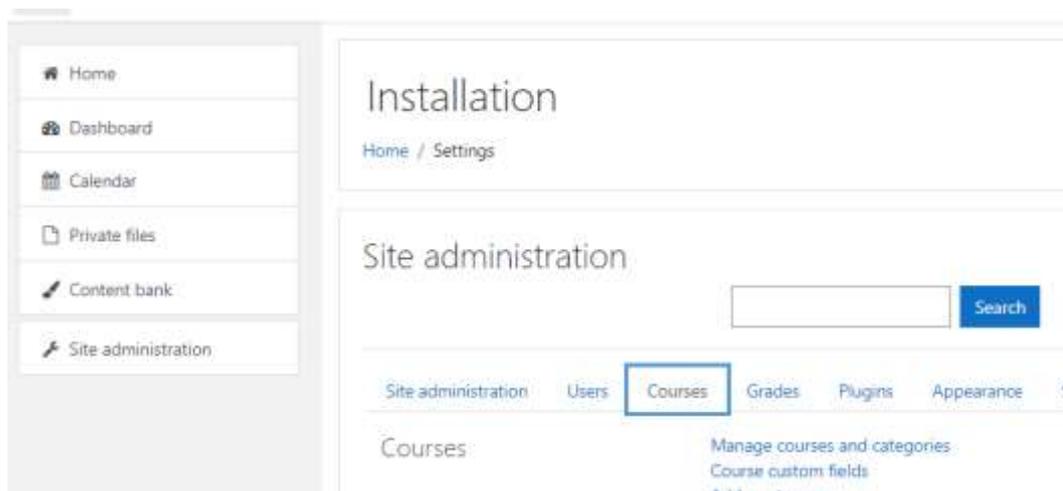
You must solve all the environmental problems (errors) found above before proceeding to install this Moodle version!

Gambar 4.6 Requirement Moodle

j. Langkah selanjutnya tinggal menunggu instalasi moodle dalam waktu 15 menit hingga 30 menit.

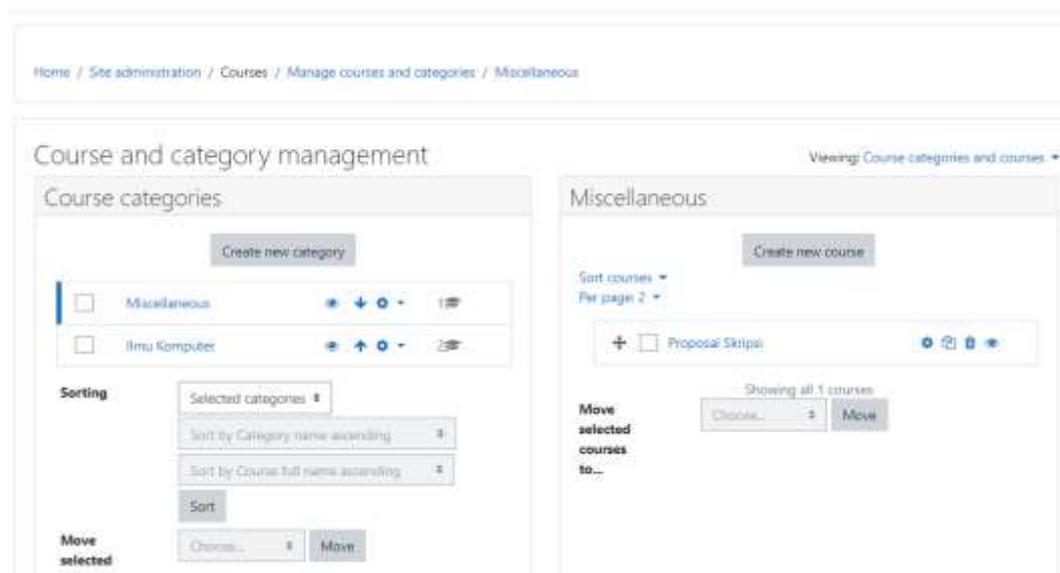
#### 4.1.2. Aktivasi Fungsi *Web Service*

Setelah menginstall moodle dan login dengan akun admin yang dibuat maka langkah selanjutnya adalah membuat *course* atau kelas karena, hal yang dibutuhkan ketika membuat *web service* untuk data absen pada moodle adalah data *course* atau kelas, data mahasiswa, dan data absen persesi kelas.



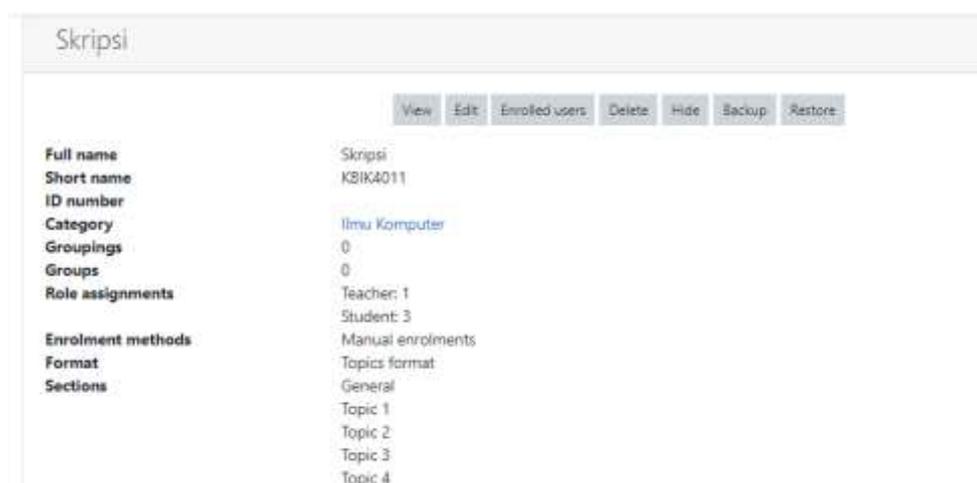
Gambar 4.7 Manage Courses and Categories

Gambar 4.7 adalah tampilan halaman awal moodle ketika sudah login. Kemudian menuju ke menu Site Administration yaitu menu dimana kita mengolah data seperti membuat user, membuat course dan memasukkan plugin baru yang belum kita miliki. Pilih menu *courses* kemudian menu *manage courses and categories*, didalam menu ini kita bisa membuat *course* yang diinginkan.



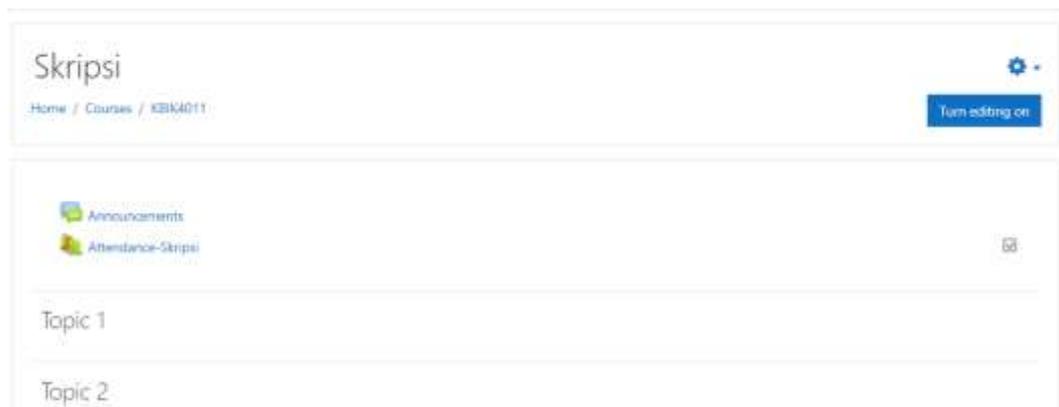
Gambar 4.8 Menu Course

Gambar 4.8 menunjukkan tampilan menu *manage courses and categories*. Pada kotak sebelah kiri kita *create new category* untuk membuat kategori yaitu berupa nama jurusan. Pada kotak sebelah kanan menu *create new course* adalah untuk membuat mata kuliah pada *course* yang telah dibuat.



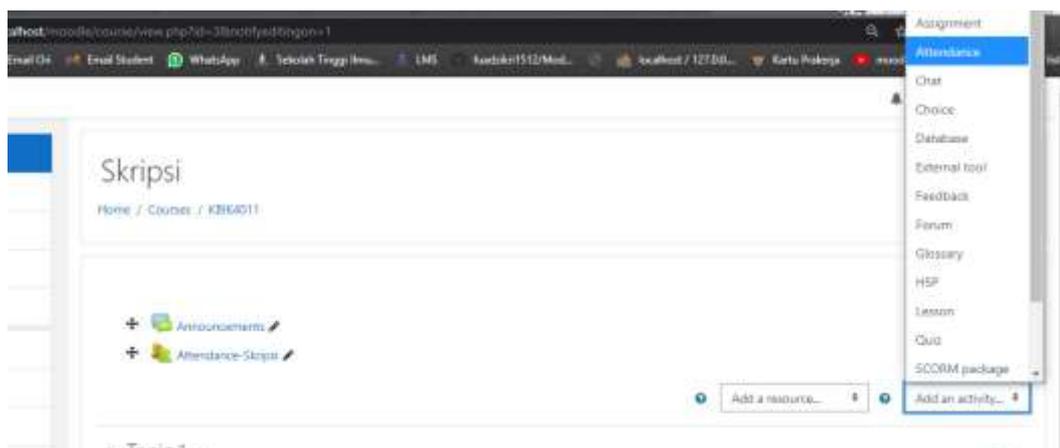
Gambar 4.9 Tampilan Menu Course

Gambar 4.9 menunjukkan *course* Skripsi telah dibuat didalam *category* Ilmu Komputer. Button View berfungsi untuk menampilkan *course* yang telah kita buat. Button *Enrolled users* untuk memasukkan *users* sesuai *role* yang akan dimasukkan yaitu dosen dan mahasiswa.



Gambar 4.10 Tampilan View Course Skripsi

Gambar 4.10 di atas adalah tampilan *view course* skripsi yang telah dibuat. Setelah memasukkan mahasiswa kedalam *course* yang dibutuhkan selanjutnya adalah membuat *attendance* atau absensi kehadiran dengan cara menekan tombol *turn edition on*.



Gambar 4.11 Add Attendance

Kemudian pilih menu dropdown *add an activity* pada Gambar 4.11 lalu pilih *attendance*. Pada step install moodle sebelumnya moodle tidak memberikan plugin *attendance*, maka dari itu harus mendownload plugin *attendance* pada web resmi moodle [https://moodle.org/plugins/mod\\_attendance](https://moodle.org/plugins/mod_attendance). Masuk ke menu Plugin lalu *add* plugin yang telah didownload ke dalam moodle. Hasilnya akan muncul menu *attendance* pada menu dropdown di atas.

Mengambil data dari moodle melalui *web service* membutuhkan izin yang harus diaktifkan dahulu melalui moodle dengan cara masuk ke menu *site administration – plugins – Manage protocols* maka akan muncul tampilan seperti Gambar 4.12.

Active web service protocols

Protocol	Version	Enable	Settings
REST protocol	2020061500		
RESTful protocol	2018102100		
SOAP protocol	2020061500		
XML-RPC protocol	2020061500		

For security reasons, only protocols that are in use should be enabled.

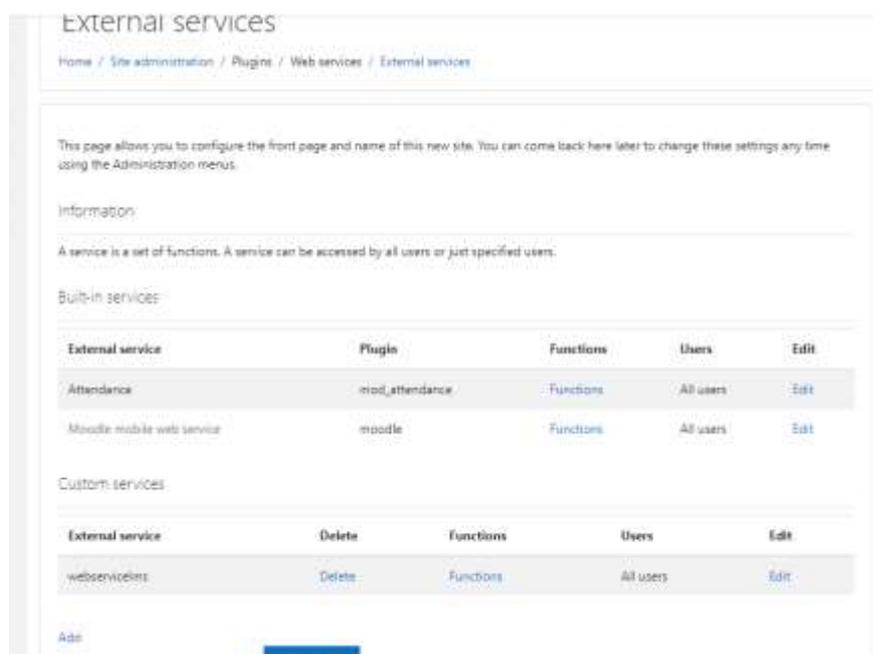
Web services documentation  Default: No  
enablewsdocumentation

Enable auto-generation of web services documentation. A user can access to his own documentation on his security keys page [More details](#). It displays the documentation for the enabled protocols only.

[Save changes](#)

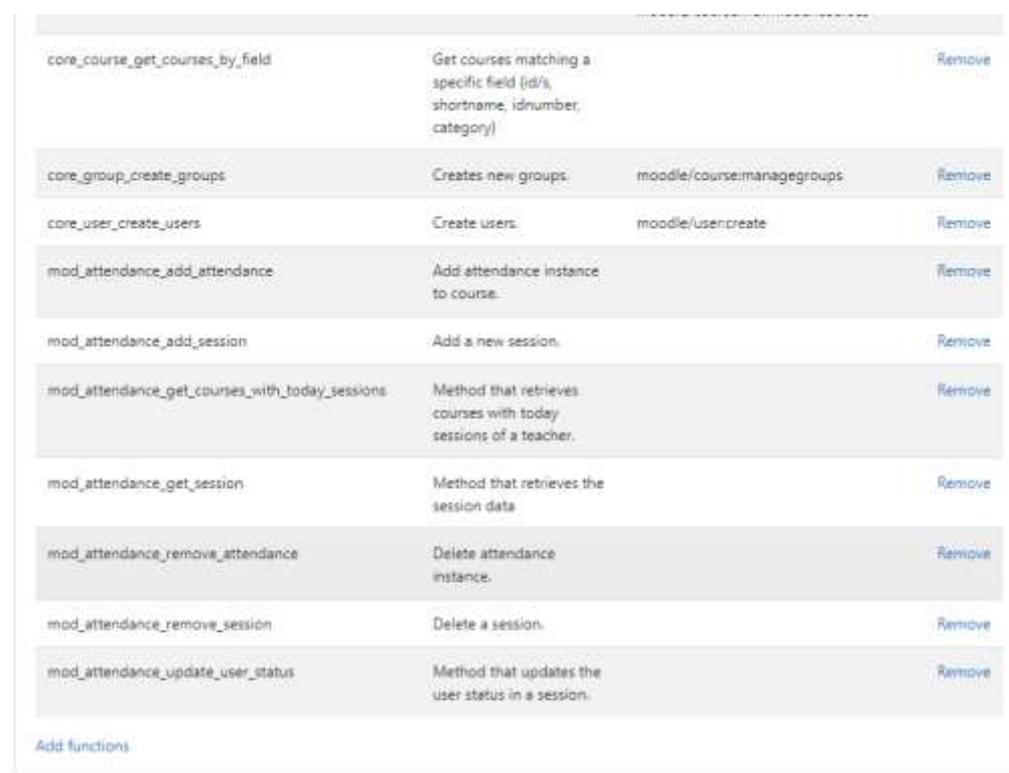
Gambar 4.12 Enable Web Service

Setelah mangaktifkan fungsi *web service* pada moodle langkah selanjutnya adalah memberikan fungsi pada *webservice* agar tidak terjadi error ketika pemanggilan url sesuai fungsi.



Gambar 4.13 Menu External Service

Pada Gambar 4.13 merupakan menu *site administration – plugins – web service – external services*. Langkah pertama yaitu inisiasi membuat satu *webservice* untuk menampung semua fungsi yang akan digunakan. Setelah jadi tekan button *functions* untuk add fungsi lalu akan muncul tampilan seperti **Error! Reference source not found.**



Gambar 4.14 Menu Add Function Web Service

Fungsi yang akan digunakan adalah semua fungsi *core\_* dan *mod\_attendance* dimana *core\_* berfungsi untuk mengaktifkan fungsi crud pada *webservice* dan fungsi *mod\_attendance* untuk mengaktifkan fungsi *attendance* atau absen pada *course* yang digunakan. Langkah selanjutnya adalah mengambil token admin untuk izin penggunaan fungsi pada *web service* seperti pada Gambar 4.15.

Manage tokens

Home / Site administration / Plugins / Web services / Manage tokens

This page allows you to configure the front page and name of this new site. You can come back here later to change these settings any time using the Administration menus.

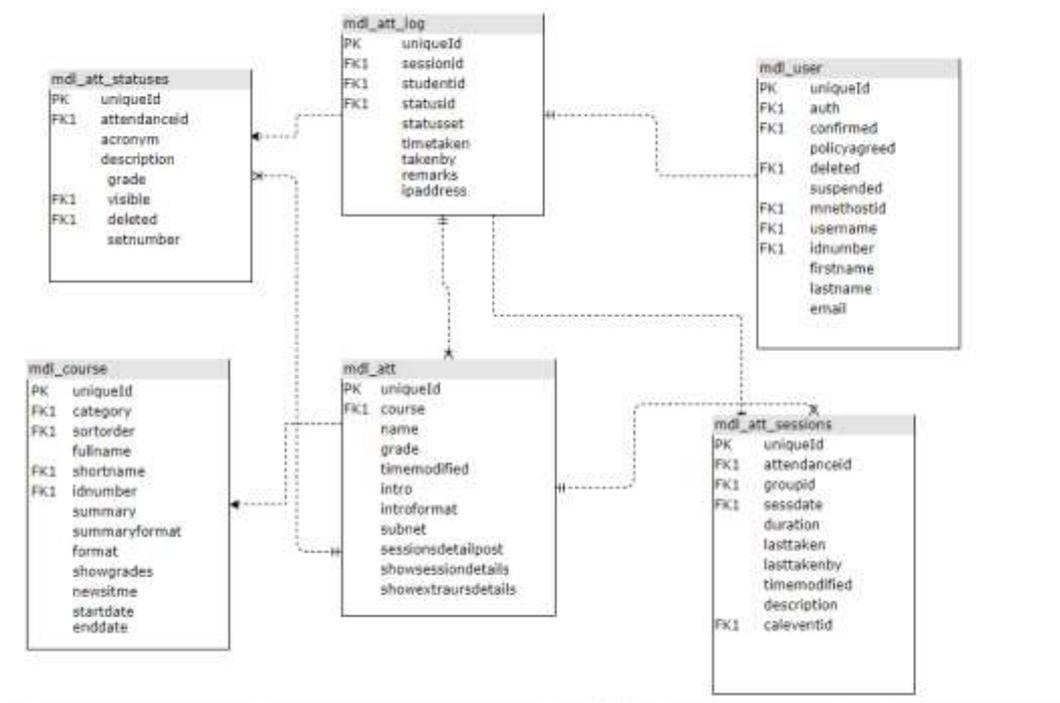
Any token can be deleted, though you can only view tokens that you created.

Token	First name / Surname	Service	IP restriction	Valid until	Creator	Operati
ad54a9b051df29f1c5f643f8b4a8f38	Ahijati Nuraminah	Attendance			fuad zikri	Delete
d11baf3776ad0202fcd31411da105a1	fuad zikri	webservicecms			fuad zikri	Delete
f7cda719dcacd94e1589db5d9b125786	mutashim bilah Missing capabilities: moodle/courseview, moodle/courseupdate, moodle/courseviewhidden, moodle/coursemanagegroups, moodle/usercreate	webservicecms			fuad zikri	Delete

Gambar 4.15 Token Admin

## 4.2. Pembahasan

### 4.2.1. Implementasi *Web Service*



Gambar 4.16 Skema Database

Pembuatan *web service* absensi pada moodle menyesuaikan dengan desain *database* pada LMS dan AIS yang akan digunakan pada Gambar 4.16 **Error! Reference source not found.** Skema Database. Alur pertama adalah menangkap id *course* yang merupakan tempat penampungan data lainnya seperti id session dan id mahasiswa. Setelah id *course* didapatkan barulah kita bisa mengakses semua sesi pada kelas yang dipilih dan absensi setiap mahasiswa yang hadir, terlambat, permisi dan absen pada sesi mata kuliah tersebut. Setelah semua data telah dikumpulkan, data akan di *export* dalam format json ke *database* AIS sebagai penerima data atau penampungan data berikutnya.

```
$url .= "&wsfunction=core_course_get_courses";
```

Gambar 4.17 Fungsi Kode Get Course

Pada Gambar 4.17 merupakan kode pada file index.php yang berfungsi untuk mendapatkan id *course* dengan fungsi

“&wsfunction=core\_course\_get\_courses”. Tampilan UI index untuk mendapatkan *course* seperti pada Gambar 4.18 Fungsi Kode Get Course. Pada tahap ini proses pencarian data *course* berjalan cukup lambat selama kurang lebih 10 detik karena kodingan pencarian data yang masih manual.

The screenshot displays a web interface for a 'WEB SERVICE'. At the top, there is a red header with the text 'WEB SERVICE'. Below this is a yellow bar with the word 'Home'. The main content area is white and contains a section titled 'Input Data'. Inside this section is a green-bordered form titled 'MATA KULIAH'. The form has a sub-header 'Pilih Mata Kuliah'. It contains a 'Token' field with the value 'd11baf3776ad0202ffcd3141fda105a1', a 'Mata Kuliah' dropdown menu, and a 'Search' button. At the bottom of the page, there is a black footer with the text 'Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer'.

Gambar 4.18 Halaman Index.php

Langkah selanjut nya setelah menekan tombol *search* pada UI akan menuju ke halaman `absen.php` yang berisi seluruh pertemuan pada *course* yang di pilih pada menu awal. Memiliki dua *field* yaitu pertemuan dan *action go* yang bila ditekan seperti pada Gambar 4.19 Data Pertemuan.

API

Home

DATA PERTEMUAN

[Pilih Markuk](#)

Pertemuan	Action
pertemuan ke- 1	GO
pertemuan ke- 2	GO
pertemuan ke- 3	GO
pertemuan ke- 4	GO
pertemuan ke- 5	GO
pertemuan ke- 6	GO
pertemuan ke- 7	GO
pertemuan ke- 8	GO
pertemuan ke- 9	GO

STIMIK ESQ

Gambar 4.19 Data Pertemuan

API

Home

DATA ABSEN

DATA SISWA

[Tambah Data](#)

Student Id	Nama Lengkap	Session Id	Status
3	mutashim billah	8	late
6	uta fattah	8	late
5	aidil zartesa	8	late

Export

STIMIK ESQ

Gambar 4.20 Daftar Mahasiswa

Gambar 4.20 Daftar Mahasiswa menunjukkan file `export.php` yang menampilkan data mahasiswa, session id, dan status kehadiran mahasiswa. Session id merupakan id dari sesi course yang telah didapatkan di awal. Tombol

*export* akan mengirim data menuju *database* AIS dan menerima data yang dikirim dari *web service*.

#### 4.2.2. Pengujian *Black Box*

Tahap selanjutnya setelah melewati tahapan implementasi yaitu pengujian sistem. Pengujian sistem dilakukan dengan pengujian *black box*. Pengujian dilakukan untuk memastikan data yang dikirim dan diterima melalui *web service* berjalan dengan baik dan pengujian *black box* bertujuan untuk memastikan bahwa aplikasi berjalan sesuai dengan fungsi. Pada tahapan pertama mahasiswa melakukan *input* absen pada *course* skripsi di *web* LMS dan data absensi direkap pada *database* LMS seperti pada Gambar 4.21.

id	sesssion	studentid	status	statusid	semester	identity
27	1	3	5	5.7.8.8	1623481781	2
28	1	6	5	5.7.8.8	1623481781	2
29	1	5	5	5.7.8.8	1623481781	2
30	2	3	7	5.7.8.8	1623481833	2
31	2	6	7	5.7.8.8	1623481833	2
32	2	5	7	5.7.8.8	1623481833	2
33	3	3	8	5.7.8.8	1623481848	2
34	3	6	8	5.7.8.8	1623481848	2
35	3	5	8	5.7.8.8	1623481848	2
36	4	3	9	5.7.8.8	1623481883	2
37	4	6	9	5.7.8.8	1623481883	2
38	4	5	9	5.7.8.8	1623481883	2
39	15	3	9	5.7.8.8	1623481883	2
40	15	6	9	5.7.8.8	1623481883	2
41	15	5	7	5.7.8.8	1623481883	2
42	5	3	9	5.7.8.8	1623481938	2

Gambar 4.21 Database LMS

Kemudian *web service* mengambil data *course* skripsi untuk pencarian melalui *course* seperti pada Gambar 4.22.

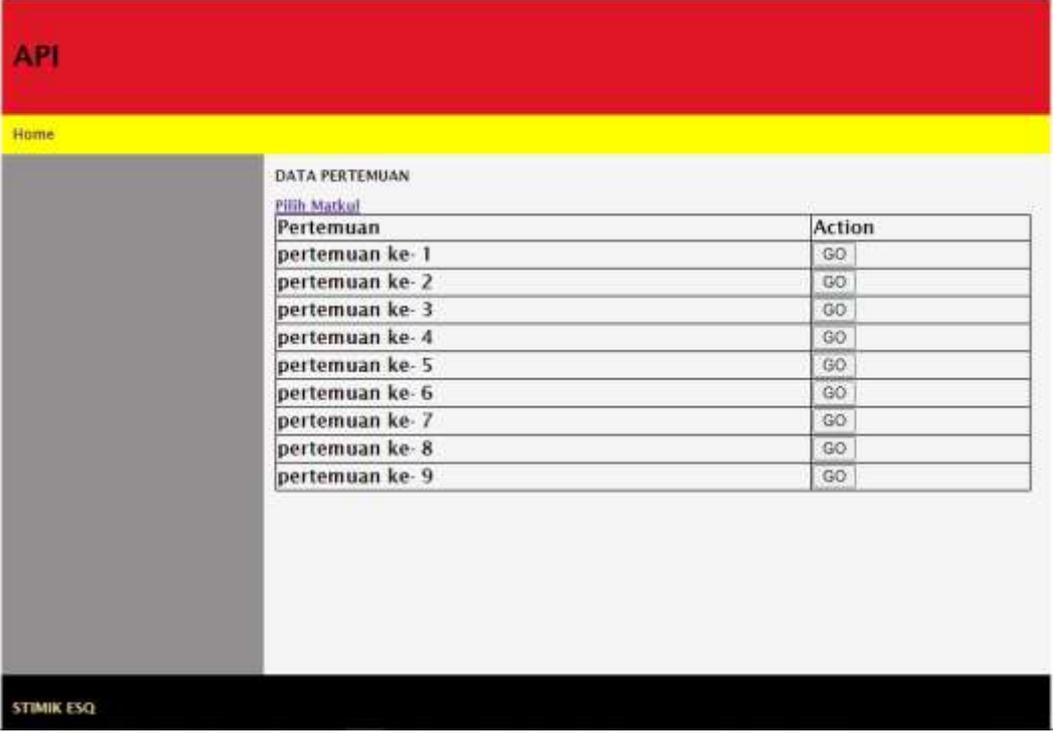
The screenshot shows a web service interface with a red header containing the text "WEB SERVICE". Below the header is a yellow bar with the text "Home". The main content area is divided into a grey sidebar on the left and a white main area on the right. The main area is titled "Input Data" and contains a green box with the following elements:

- Title: "MATA KULIAH"
- Instruction: "Pilih Mata Kuliah"
- Token: "e11ba13776ad0202fcd3141fda105a1" (displayed in a white box)
- Mata Kuliah: A dropdown menu with "Skripsi" selected.
- Search: A button labeled "Search".

At the bottom of the page, there is a black footer with the text "Sekolah Tinggi Ilmu Manajemen dan Ilmu Komputer".

Gambar 4.22 Course Skripsi

Setiap matakuliah atau *course* memiliki sesi atau pertemuan matakuliah. Setelah pencarian *course* skripsi dilakukan, maka langkah selanjutnya adalah pencarian banyaknya pertemuan pada *course* skripsi seperti pada Gambar 4.23.



The screenshot shows a web application interface with a red header containing the text 'API'. Below the header is a yellow bar with the text 'Home'. The main content area is white and contains a table titled 'DATA PERTEMUAN'. Above the table is a link labeled 'Pilih Matakul'. The table has two columns: 'Pertemuan' and 'Action'. The 'Action' column contains a 'GO' button for each row. The rows are labeled 'pertemuan ke- 1' through 'pertemuan ke- 9'. At the bottom of the page, there is a black footer with the text 'STIMIK ESQ'.

Pertemuan	Action
pertemuan ke- 1	GO
pertemuan ke- 2	GO
pertemuan ke- 3	GO
pertemuan ke- 4	GO
pertemuan ke- 5	GO
pertemuan ke- 6	GO
pertemuan ke- 7	GO
pertemuan ke- 8	GO
pertemuan ke- 9	GO

Gambar 4.23 Pertemuan Matakuliah Skripsi

Setelah mendapatkan setiap pertemuan pada matakuliah skripsi, tinggal memilih pertemuan berapa yang akan di *export* ke *database* penerima yaitu *database* AIS. Penulis menampilkan pengujian pada pertemuan ke-1 dan pertemuan ke-2 yang kemudian akan muncul nama mahasiswa yang hadir pada pertemuan tersebut dan status kehadiran mahasiswa seperti pada Gambar 4.24 dan Gambar 4.25.

API

Home

DATA ABSEN

DATA SISWA

[Tambah Data](#)

Student Id	Nama Lengkap	Session Id	Status
3	mutashim billah	1	present
6	uta fattah	1	present
5	aidil zartesyia	1	present

Export

STIMIK ESQ

Gambar 4.24 Data Siswa Pertemuan ke-1

API

Home

DATA ABSEN

DATA SISWA

[Tambah Data](#)

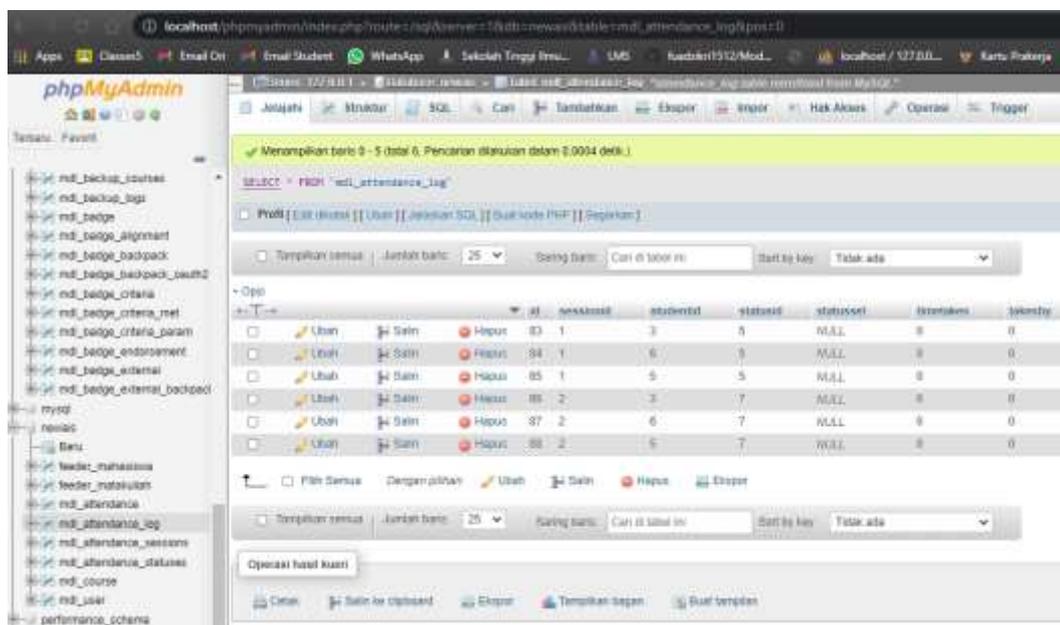
Student Id	Nama Lengkap	Session Id	Status
3	mutashim billah	2	late
6	uta fattah	2	late
5	aidil zartesyia	2	late

Export

STIMIK ESQ

Gambar 4.25 Data Siswa Pertemuan ke-2

Langkah selanjutnya adalah menekan tombol *export* untuk mengirim data absensi mahasiswa ke *database* AIS seperti pada Gambar 4.26.



Gambar 4.26 Database AIS

Data absensi pertemuan ke-1 dan pertemuan ke-2 telah ter-*input* pada *database* AIS yang menunjukkan bahwa pengujian sistem *web service* ini berhasil memindahkan data absensi dari LMS ke AIS.

## **BAB 5**

### **PENUTUP**

#### **5.1. Kesimpulan**

Setelah melakukan analisis, perancangan sistem dan *database*, serta melakukan implementasi *web service* REST untuk melakukan integrasi data absensi dari *Learning Management System* (LMS) ke *Academic Information System* (AIS), dapat diperoleh kesimpulan sebagai berikut:

1. Teknologi *web service* dengan menggunakan metode Representational State Transfer (REST) dapat diterapkan pada platform berbasis *WEB* sebagai penyedia layanan (LMS) dan penerima data (AIS) pada kampus STIMIK ESQ.
2. Dengan terbentuknya model rancangan sistem integrasi *web service* untuk mengintegrasikan data absensi dari *web* LMS ke AIS pada STIMIK ESQ maka dapat diterapkan model rancangan ini secara fungsional terhadap sistem LMS dan AIS

#### **5.2. Saran**

Untuk kepentingan penelitian selanjutnya penelitian ini masih sangat mungkin dikembangkan sebagai berikut:

1. Penelitian ini masih berjalan cukup lambat karena kodingan mencari data session yang cukup banyak pada *course* masih belum efektif mengakibatkan loading yang cukup lama. Oleh karena itu, diharapkan pada penelitian selanjutnya dapat mengolah data yang besar dengan waktu yang sebentar.
2. Belum ada nya pengecekan data jika data sudah ter-*input* memungkinkan terjadinya *double* data ketika dimasukkan dua kali. Cara mengatasi duplikasi data dengan memberi *section* ketika export data dan melakukan pengecekan. Jika data sudah ada maka data tidak akan diexport lagi.

## DAFTAR PUSTAKA

- Atmaja, A. P., & Yulianto, S. V. (2018). Integrated Student Portal Menggunakan Metode Pengembangan Siklus Pendek. *JUPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, 3(1), 24–31. <https://doi.org/10.29100/jipi.v3i1.639>
- AyuningTyas, A., & Ashari, A. (2017). Pemanfaatan Teknologi *Web Service* Untuk Integrasi Sistem Layanan Materi Pelajaran Terdistribusi. *Angkasa: Jurnal Ilmiah Bidang Teknologi*, 8(1), 33. <https://doi.org/10.28989/angkasa.v8i1.130>
- Hidayat, R., & Ashari, A. (2013). Penerapan Teknologi *Web Service* Untuk Integrasi Layanan Puskesmas dan Rumah Sakit. *Bimipa*, 23(1), 64–77.
- Jadid, A. (2017). Rancang Bangun Sistem Absensi Perkuliahan Auto Id Berbasis Rfid Yang Terintegrasi Dengan *Database* Berbasis *Web*. *Kitektro*, 2(2), 59–69.
- Purnamasari, S. D. (2008). *Web Service* Sebagai Solusi Integrasi Data Pada Sistem Informasi Akademik Universitas Bina Darma. *Jurnal Ilmiah MATRIK*, 95(12), 1–14.
- Studi, P., Informatika, T., & Barat, K. (2011). *Implementasi Web Services Sistem Integrasi Data Menggunakan Teknik Replikasi Data*. 1(2), 95–104.
- Raniastri, D. (2013). Implementasi *Web Service* Dengan Metode Representational State Transfer (REST) Untuk Integrasi Data Transaksi di PT. Meliana Perkasa Sejahtera (MPS).
- Perwira, R., & Santosa, B. (2017). Implementasi *Web Service* pada Integrasi Data Akademik dengan Replika Pangkalan Data Dikti. *Telematika*, 14(01), 1–11. <https://doi.org/10.31315/telematika.v14i01.1962>
- Rizal, R., & Rahmatulloh, A. (2019). *Restful Web Service* Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan. *Jurnal Ilmiah Informatika*, 7(01), 54. <https://doi.org/10.33884/jif.v7i01.1004>