Bit (Fakultas Teknologi Informasi Universitas Budi Luhur) p-ISSN: 1693-9166
Vol. 19, No. 1, April 2022, hlm. 9-16 e-ISSN: 2685-127X

IMPLEMENTASI WEB SERVICE RESTFUL DENGAN AUTENTIKASI JSON WEB TOKEN DAN ALGORITMA KRIPTOGRAFI AES-256 UNTUK APLIKASI PEMINJAMAN LABORATORIUM BERBASIS MOBILE PADA UNIVERSITAS BUDI LUHUR

Gabriel Yoda Gustiegan[1], Painem[2*]

[1,2]Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta
Email: [1]yodaegan@gmail.com, [2]painem@budiluhur.ac.id

(Naskah masuk: 17 Maret 2022, diterima untuk diterbitkan: 17 April 2022)

Abstrak

Peminjaman ruangan laboratorium merupakan salah satu pelayanan yang diberikan oleh LAB ICT Terpadu untuk mahasiswa dan dosen di Universitas Budi Luhur. Saat ini proses peminjaman laboratorium komputer peminjam harus datang langsung ke LAB ICT Terpadu atau memanfaatkan jejaring sosial seperti *Whatsapp* untuk berkomunikasi dua arah dalam melakukan peminjaman lab. Hal tersebut tidaklah cukup efektif untuk melayani pihak peminjam baik dosen atau Unit Kegiatan Mahasiswa. Permasalahan dalam peminjaman lab adalah tumbukan antara jadwal kegiatan perkuliahan dengan unit kegiatan dosen, unit kegiatan mahasiswa serta kegiatan-kegiatan lain yang memerlukan fasilitas lab. Berdasarkan masalah ini diperlukan aplikasi untuk memudahkan peminjaman lab dan koordinasi kepada Asisten LAB ICT berbasis *mobile*. Sementara untuk integrasi dengan sistem yang sudah ada diperlukan *web service* sebagai *backend system* sehingga layanan peminjaman lab dapat diakses oleh berbagai *platform*. Arsitektur yang digunakan pada web *service* menggunakan RESTFUL API, namun masih ada beberapa masalah pada RESTFUL API yaitu mengenai keamanan pada proses otentikasi dan enkripsi data pada data peminjam. Pada arsitektur REST diperlukan metode keamanan yaitu menggunakan JSON *Web Token* dan metode kriptografi *Advanced Encryption Standard*-256 (AES-256).

Kata kunci: *penjadwalan, web service, API, kriptografi, AES-256, laboratorium*

IMPLEMENTATION OF A RESTFUL WEB SERVICE WITH JSON WEB TOKEN AUTHENTICATION AND AES-256 CRYPTOGRAPHIC ALGORITHM FOR MOBILE-BASED LABORATORY LOAN APPLICATIONS AT UNIVERSITAS BUDI LUHUR

Abstract

The loan of laboratory space is one of the services provided by the Integrated ICT LAB for students and lecturers at Budi Luhur University. Currently, the borrower's computer laboratory loan process must come directly to the Integrated ICT LAB or use social networks such as Whatsapp to communicate two-way in conducting laboratory loans. This is not effective enough to serve borrowers, either lecturers or Student Activity Units. The problem with borrowing a lab is the collision between the lecture schedule and the lecturer activity unit, student activity unit and other activities that require lab facilities. Based on this problem, an application is needed to facilitate laboratory lending and coordination to mobile-based ICT LAB Assistants. Meanwhile, for integration with existing systems, a web service is needed as a backend system so that lab loan services can be accessed by various platforms. The architecture used in the web service uses the RESTFUL API, but there are still some problems with the RESTFUL API, namely regarding security in the authentication process and data encryption on borrower data. In the REST architecture, a security method is needed, namely using a JSON Web Token and the Advanced Encryption Standard-256 (AES-256) cryptographic method.

***Keywords**: scheduling, web service, API, cryptography, AES-256, laboratory*

1. PENDAHULUAN

Penjadwalan atau *scheduling* adalah pengalokasian waktu yang tersedia untuk melaksanakan masing-masing pekerjaan dalam rangka menyelesaikan suatu kegiatan hingga tercapai hasil optimal. Persoalan penjadwalan berkaitan dengan beberapa hal seperti tempat atau ruangan, waktu dan dosen pengajar. Dalam pembuatan jadwal untuk mata kuliah, sering dijumpai berbagai persoalan seperti keterbatasan ruang, kapasitas ruang, spesifikasi atau alat penunjang di setiap ruang, dan tumbukan waktu. Pada saat ini, Lab ICT Terpadu Universitas Budi Luhur masih menggunakan sistem manual dalam pembuatan jadwal sehingga sering kali terjadi *human error*.

Pelayanan peminjaman fasilitas ruang laboratorium pada Lab ICT Terpadu Universitas Budi Luhur saat ini masih secara langsung dengan mendatangi ruang asisten untuk melihat jadwal laboratorium yang kosong atau meminta asisten untuk melihat jadwal dan meminjam ruang lab tersebut. Dalam proses peminjaman fasilitas lab banyak dijumpai permasalahan misalnya adalah tumbukan antara jadwal kegiatan perkuliahan dengan unit kegiatan dosen, unit kegiatan mahasiswa serta kegiatan-kegiatan lain yang memerlukan fasilitas lab. Sulitnya untuk mengetahui kapan waktu kosong lab untuk menyelenggarakan kegiatan atau untuk menentukan jadwal kuliah pengganti serta kurangnya prosedur dan koordinasi antara pihak peminjam dengan asisten Lab ICT Terpadu dalam hal peminjaman fasilitas-fasilitas yang ada pada Lab ICT Terpadu.

Dari permasalahan tersebut maka perlu dibangun Sistem Penjadwalan *Web Service* dengan menggunakan metode Rest API dengan Autentikasi JSON Web Token Dan Algoritma Kriptografi AES-256. REST API berfungsi untuk melakukan *request response* antara sistem penjadwalan berbasis android ke sistem berbasis web. Ketika peminjam akan melakukan transaksi peminjaman *system* akan membaca token yang telah dienkripsi oleh algoritma AES-256 kemudian akan di dekripsi agar bisa melakukan transaksi peminjaman. Alasan pemilihan algoritma AES-256 adalah karena ukuran blok yang besar dan ukuran kunci yang lebih panjang, yang mana akan memberikan keamanan lebih dalam jangka panjang.

Penelitian implementasi *web service* sudah banyak dilakukan, contoh penelitian yang sudah dilakukan Sistem Presensi Pegawai Berbasis Web *Service* Menggunakan Metode *Restfull* Dengan Keamanan Jwt Dan Algoritma Haversine[1], SON *Web Token* (JWT) untuk *Authentication* pada Interoperabilitas Arsitektur berbasis RESTful Web Service[2], Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512 [3], Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan[4], Implementasi *Web Service* Pada Sistem Pengindeksan Dan Pencarian Dokumen Tugas Akhir, Skripsi, Dan Praktik Kerja Lapangan[5].

Sistem berbasis web juga menggunakan keamanan JWT *(JSON Web Token)* guna mengamankan data dari pihak-pihak yang tidak bersangkutan

2. METODE PENELITIAN

2.1 *Web Service*

Web service adalah salah satu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan. *Web service* memiliki interface yang dideskripsikan dalam format yang dapat dibaca oleh mesin. *Web Services* juga tidak terikat kepada bahasa pemrograman atau sistem operasi tertentu. Sistem-sistem lainnya berinteraksi dengan *web service* menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML. Definisi diatas diberikan oleh *World Wide Web Consortium* (W3C) yang merupakan badan yang menciptakan dan mengembangkan standar *web service*. Tetapi secara umum, *web service* tidak terbatas hanya pada standar SOAP saja. Salah satu pustaka yang mengulas lengkap tentang *web service* menyebutkan definisi yang lebih umum: *web service* adalah aplikasi yang diakses melalui internet menggunakan protokol standar internet dan menggunakan XML sebagai format pesannya.[6]

2.2 *Representational State Transfer* (REST)

Konsep REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. REST merupakan standar arsitektur komunikasi berbasis web yang selalu digunakan terhadap pengembangan layanan berbasis web. Pada umumnya, *Hypertext Transfer Protocol* (HTTP) berperan sebagai protokol untuk melakukan komunikasi data. Sistem yang menggunakan prinsip-prinsip dari REST dapat disebut dengan "RESTful". Penetapan identifikasi terhadap resource dilakukan oleh *Universal Resource Identifiers* (URIs) atau global ID. Resource diperkenalkan dengan format teks, JSON, atau XML. Pada umumnya, format yang digunakan adalah JSON dan XML[3].

Salah satu kriteria desain web services yang paling sering digunakan adalah restful web services, restful sendiri bekerja dengan cara *resource-oriented*. Pada restful web services client (*requester*) mengakses services yang ditawarkan oleh web server, yaitu dengan cara mengakses URL dari resource menggunakan method pada HTTP. Dalam dunia web API, protokolnya adalah HTTP. API client dapat berinteraksi dengan API dengan mengirimkan berbagai jenis pesan HTTP. Standar HTTP mendefinisikan delapan jenis pesan, yaitu:

a. *GET* (*Method Get* mengambil data dari web server dengan menentukan parameter di bagian URL dari permintaan.
b. *DELETE* (*Method Delete* menghapus sumber daya)

c. *POST* (*Method Post* memanfaatkan badan pesan untuk mengirim data ke *server web*)
d. *PUT* (*Method Put* mirip dengan post memanfaatkan badan pesan untuk mentransfer data)
e. *HEAD* (*Method Head* digunakan untuk mengambil informasi tentang URL dari *web server*)
f. *OPTION* (*Method Option* berguna untuk mencari tahu mana metode HTTP dapat diakses oleh klien)
g. *LINK* (*Method Link* dapat digunakan untuk membuat sambungan jaringan ke *server web* melalui HTTP)
h. *UNLINK* (*Method Unlink* dapat digunakan untuk memutus sambungan jaringan ke server web)

2.3 JSON *Web Token* (JWT)

JWT ini adalah sebuah *token* berbentuk string JSON yang sangat padat (ukurannya), informasi mandiri yang gunanya sendiri untuk melakukan sistem autentikasi dan pertukaran informasi. Karena bentuknya kecil, *token* JWT dapat dikirim melalui URL, parameter HTTP POST atau di dalam Header HTTP, dan juga karena ukurannya yang kecil maka dapat ditransmisikan dengan lebih cepat. Disebut informasi mandiri karena isi dari token yang dihasilkan memiliki informasi dari pengguna yang dibutuhkan, sehingga tidak perlu query ke basis data lebih dari satu kali. Token tersebut dapat diverifikasi dan dipercaya karena sudah di-sign secara digital. Token JWT dapat di-sign dengan menggunakan secret (algoritma HMAC) atau pasangan *public/private key* (algoritma RSA). Proses login yang dilakukan tidak seperti aplikasi website biasa, tetapi menggunakan *session* untuk mengingat yang sedang melakukan proses login. Namun, API hanya menggunakan konsep JWT yang dapat disebut "jot". JWT tidak bergantung pada bahasa program tertentu. Struktur JWT terdiri atas tiga bagian yang dipisahkan oleh titik (“.”), yaitu *header*, *payload*, dan *signature*.[3]

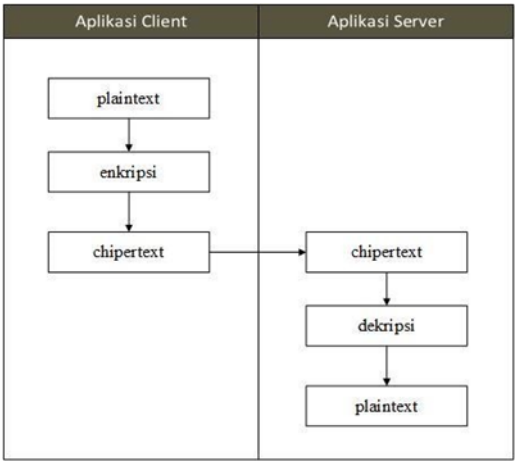
JWT merupakan sebuah token berbentuk string yang terdiri dari tiga bagian yaitu : *header*, *payload* dan *signature* yang digunakan untuk proses otentikasi dan pertukaran informasi. *Token* terdiri dari dua jenis :*token* pembawa dan *token* pemegang kunci. Sedangkan berdasarkan tujuan terdapat dua skema : token identitas dan *token* akses. Cara kerja JWT sama seperti *password*, ketika pengguna berhasil login maka *server* akan memberikan *token* yang disimpan di local *storage* atau *cookies browser* [2].

2.4 Algoritma AES-256

Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128 bit, 192 bit, dan 256 bit [7]. AES memiliki kecepatan enkripsi dan deskripsi tertinggi, berikutnya Blowfish, DES dan IDEA [8]. Algoritma kriptografi AES termasuk dalam klasifikasi algoritma kriptografi kunci simetri, kunci yang digunakan pada enkripsi sama dengan kunci yang digunakan untuk dekripsi [9].

Enkripsi algoritma AES di mulai dengan memasukan XOR plainteks/*state* yang akan di enkripsi dengan *round key*, setelah selesai melakukan XOR plainteks dengan *round key*. Kita lakukan subtitusi dengan s-Box, Setelah itu hasil dari substitusi dengan s-Box selesai. Kita lakukan *shiftrow*. Setelah hasil shiftrow didapat, maka langkah selanjutnya yaitu melakukan *Mix Columns* dengan mengalikan matrik Setelah perhitungan *Mix Column* selesai maka kita melakukan *addroundkey* yaitu melakukan XOR *state* dengan *round key*. Lakukan sampai literasi 10, namun pada saat putaran/literasi yang ke 10, setelah *step shiftrow* lompati step *Mix Column* dan langsung lanjut melakukan XOR hasil *state* saat *shift row* dengan *round key* [10].

Penerapan REST *Web Service* memiliki beberapa kekurangan yaitu tidak ada dukungan standar untuk keamanan dan kebijakan pengaksesan data pada sisi server, untuk mengatasi kekurangan tersebut REST *Web Service* diperlukan sistem autentikasi untuk memberi hak akses data pada REST *Server*. JSON Web Token atau JWT merupakan sebuah *token* berbentuk string yang digunakan untuk melakukan autentikasi dan menjamin integritas pesan yang dikirim oleh salah satu pihak. Pada implementasi JWT dibutuhkan suatu metode yang mendefinisikan cara yang simpel dan independen dari transmisi informasi yang aman antar setiap pihak dengan menggunakan format data objek JSON. Algoritma yang digunakan dalam metode ini adalah algoritma AES-256.



Gambar 1. Desain Komunikasi dari Client ke Server Algoritma AES-256

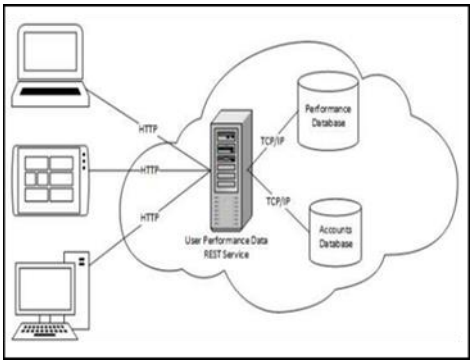
Pada Gambar 1 menjelaskan tentang rancangan proses komunikasi antara aplikasi *client* dengan aplikasi server dalam hal ini REST *Web Service*. Client bermaksud mengirimkan data ke *server* dalam bentuk plaintext. Sebelum dikirimkan, data yang berupa plaintext dienkripsi dahulu dengan algoritma AES-256, kemudian hasil enkripsi berupa *chipertext*,

dikirimkan ke *server*. *Server* menerima data dalam bentuk chipertext, kemudian didekripsi dengan algoritma AES 256, sehingga diperoleh data plaintext. Selanjutnya data plaintext tersebut diteruskan ke proses selanjutnya (disimpan/diolah).

2.5 Cara Kerja REST

Untuk membangun sistem peminjaman Lab menggunakan metode Rest agar dapat digunakan antar *platform* baik web dan *mobile*. Metode REST *web service* menerapkan konsep perpindahan antar state. *State* yang dimaksud disini dapat digambarkan apabila browser melakukan permintaan suatu web, maka *server* akan melakukan pengiriman *state* halaman web yang sekarang ke browser.

Gambar 2 adalah diagram yang menjelaskan cara kerja dari REST *web service*



Gambar 2. REST web service[6]

Pada gambar 2, *Service* menerima adanya HTTP request pada berbagai macam URL *endpoints* dengan berbagai jenis HTTP *request* untuk membedakan antara berbagai macam *action*. *Service* ini memiliki kemampuan untuk mengumpulkan data dari aplikasi klien dengan menggunakan operasi-operasi HTTP dengan adanya *performance* data di bagian *body* pada *request*. *Service* kemudian menerima dan memberi otorisasi pada *request*, kemudian menyimpan data dalam database jika diperlukan.

3. HASIL DAN PEMBAHASAN

3.1 Penerapan Metode

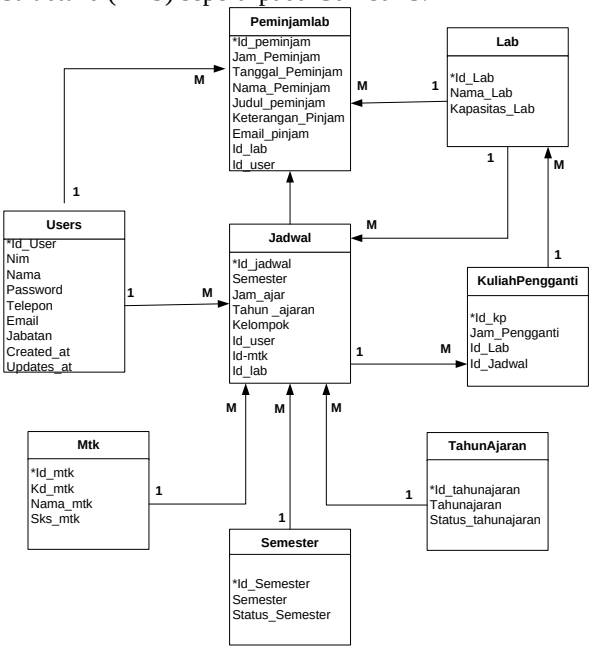
Untuk membangun sistem peminjaman ruang lab menggunakan metode Rest dan algoritma aes-256 dengan autentifikasi JWT sebagai keamanan pertukaran data.Dari metode diatas terdapat beberapa tahapan yang menjadi rancangan utama, rancangan ini sebagai gambaran proses dari enkripsi dan dekripsi yang terdapat pada sistem peminjaman lab.

Pada layanan Web Service terdapat proses enkripsi dan proses dekripsi antara client dan server:

a) Proses enkripsi dilakukan pada saat client mengirimkan data API berupa plaintext.
b) Kemudian data tersebut di enkripsi ke dalam bentuk *chipertext*.
c) Kemudian server menerima data dalam bentuk chipertext dan tersimpan pada database.
d) Proses dekripsi dilakukan pada saat server mengirim data API berbentuk *chipertext* kemudian data tersebut di dekripsi ke dalam bentuk *plaintext*.
e) Sehingga pada web dan mobile menerima data berbentuk *plaintext*

3.2 Rancangan Basis Data

Rancangan basis data Aplikasi Peminjaman Lab digambarkan dengan menggunakan *Logical Record Structure* (LRS) seperti pada Gambar 3.

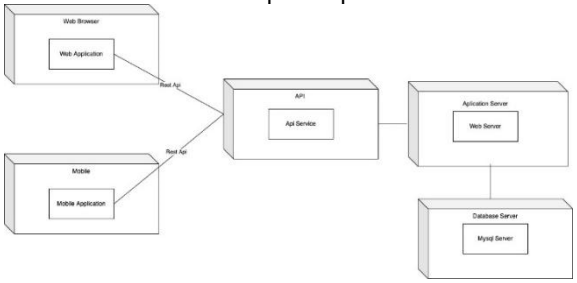


Gambar 3. Logical Record Structure (LRS)

Berdasarkan gambar 3 LRS di atas akan menghasilkan tabel database PeminjamanLab, Lab, User, Jadwal, KuliahPengganti, Mtk, Semester, dan Tahun ajaran.

3.3 Arsitektur Penerapan Web Service

Arsitektur penerapan sistem ini menggunakan model *Three Tier*. *Tier* pertama merupakan arsitektur database, dimana database yang digunakan adalah MySQL. Tier kedua merupakan arsitektur aplikasi beserta API. Sedangkan *Tier* ketiga merupakan *client*, yang diakses melalui desktop browser maupun *mobile*. Arsitektur ditampilkan pada Gambar 4.



Gambar 4. Arsitektur Penerapan Web Service

Adapun spesifikasi *Tier* kedua ditampilkan pada Tabel 1 ini terdiri dari PHP frameworks, Programming languages, Operating *systems* dan UI *frameworks* yang digunakan.

Tabel 1. Spesifikasi Tier Kedua

No	Spesifikasi	Keterangan
1	PHP frameworks	Laravel
2	Web servers	Apache 2.4.6
3	Web Programming languages	PHP 7.3.28
4	Operating systems	CentOS
5	UI frameworks	Bootstrap 4.5.3
6	Mobile Programming Languages	Java

3.4 Rancangan Layanan Web Service

Pada Tabel 2 adalah rancangan ini berisikan tentang rancangan *web service* yang akan digunakan pada sistem Peminjaman Lab. Berikut adalah *Endpoint* yang digunakan.

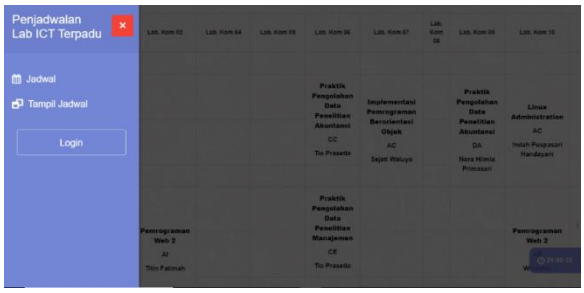
Pada tabel 2 merupakan daftar fungsi atau layanan web service yang akan digunakan pada sistem Peminjaman Lab. Fungsi –fungsi yang digunakan adalah login untuk masuk ke sistem, List Lab untuk mendampilkan daftar nama-nama lab per tanggal dan hari, List Mata Kuliah untuk menampilkan daftar matakuliah yang menggunakan Lab, List Dosen untuk menampilkan daftar nama dosen yang mengajar di Lab, List Kelompok untuk menampilkan id_dosen dan Id Matkul yang menggunakan Lab, List Jadwal harian untuk menampilkan hari, tanggal penggunaan Lab, List Jadwal Sekarang untuk menampilkan jadwal penggunaan Lab sekarang, Enkripsi Data Kuliah Pengganti untuk mengenkripsi ,id_dosen, kelompok, lab, tanggal, jam_ajar, Dekripsi Data Kuliah Pengganti untuk mendekripsi, id_dosen, kelompok, lab,tanggal, jam_ajar, Enkripsi Data Peminjaman Lab untuk mengenkripsi token, id_user, nama,judul, keterangan, tanggal, jam_mulai, jam_selesai, lab,email dan fungsi Dekripsi Data Peminjaman Lab untuk melakukan dekripsi id_user, nama, judul, keterangan, tanggal, jam_mulai, jam_selesai, lab, email.

3.1 Tampilan Layar Aplikasi

Pada Gambar 6 merupakan halaman awal yang terdapat 3 menu yaitu jadwal,jadwal sekarang dan login.

Tabel 2. Rancangan Layanan Web Service

No	Nama Layanan	Metode	Endpoint	Parameter	Keluaran
1	Login	*POST*	/api/login	email, password	Status, Message, Data User Login, *Token*
2	List Lab	*Get*	/api/jadwal/carilab	tanggal, hari	Success, message ,List data Lab
3	List Matakuliah	*Get*	api/jadwal/matkul		Success, message ,List data Matakuliah
4	List Dosen	*Get*	api/jadwal/dosen	id_matkul	Success, message ,List data dosen
5	List Kelompok	*Get*	/api/jadwal/kelompok	id_matkul,id_dosen	Success, message ,List data kelompok
6	List Jadwal Harian	*Get*	api/jadwal/hari-ini	hari,tanggal	Success, message ,List data jadwal
7	List Jadwal Jam Sekarang	*Get*	/api/jadwal/jam-sekarang		Success, message ,List data jadwal
8	Enkripsi Data Kuliah Pengganti	*Post*	/api/jadwal/enkripsikp	token, id_matkul,id_dosen,kelompok,lab ,tanggal,jam_ajar	Success, message,id_jadwal,lab,tanggal ,jamAjar
9	Dekripsi Data Kuliah Pengganti	*Post*	/api/jadwal/dekripsikp	id_user,id_jadwal,lab,tanggal,jam Ajar	Success, message
10	Enkripsi Data Peminjaman Lab	*Post*	/api/pinjam/enkripsipeminjaman	token,id_user, nama,judul, keterangan, tanggal,jam_mulai,jam_selesai,lab ,email	Success, message, id_user, nama,judul,keterangan,tanggal,jam_mulai,jam_selesai,lab,email
11	Dekripsi Data Peminjaman Lab	*Post*	/api/pinjam/dekripsipeminjaman	id_user,nama,judul, keterangan, tanggal,jam_mulai,jam_selesai,lab ,email	Success, message



Gambar 6. Tampilan Halaman Awal Website

Pada Gambar 6 menu jadwal untuk menampilkan jadwal, kuliah pengganti dan peminjaman berdasarkan jam berlangsung. Untuk Halaman Login ketika User masuk di web untuk melakukan login. User diminta untuk memasukan email dan password untuk dapat masuk ke halaman utama. Setelah email dan password terisi kemudian akan dikirim ke REST API untuk dilakukan validasi. Untuk menu login bisa untuk SPV, asisten dan Dosen.

3.2 Tampilan Aplikasi Mobile

Pada Gambar 7 merupakan tampilan aplikasi mobile untuk login , tambah kuliah pengganti dan tambah peminjaman Lab.

Gambar 7. Tampilan menu Login , tambah kuliah pengganti dan Tambah Peminjaman Lab

Tampilan layar pada gambar 5 untuk menu login dapat dilakukan oleh Dosen, supervisor dan Lab. Setelah melakukan login maka salah satu tampilan pada dosen adalah tambah kuliah pengganti dan tambah peminjaman Lab.

3.3 *Pseudocode* atau Algoritma AES

Berikut ini Algoritma proses Enkripsi AES-256 menjelaskan alur proses atau cara kerja algoritma AES-256 untuk menghasilkan *Ciphertext*.

```
1.  Start
2.  Input plaintext dan
    password
3.  AddRoundKey() =
    plaintext XOR password
4.  Rounds = 0
5.  Rounds = Rounds + 1
6.  Proses SubBytes()
7.  Proses ShiftRows()
8.  Proses MixColumns()
9.  Proses AddRoundKey() =
    Current State XOR
    Round Key
10. IF Rounds < 14 THEN
11. Kembali ke baris 5
12. ELSE
13. Proses SubBytes()
14. Proses ShiftRows()
15. Proses AddRoundKey() =
    Current    State    XOR
    Round Key
16. Output plaintext
17. END IF
18. END Process
```

Sedangkan Algoritma proses Dekripsi AES-256 menjelaskan alur proses atau cara kerja algoritma AES-256 untk menghasilkan *Plaintext* sebagai berikut :

```
1.  Start
2.  Input chipertext dan password
3.  AddRoundKey() = chipertext XOR password
4.  Rounds = 0
5.  Rounds = Rounds + 1
6.  Proses InvSubBytes()
7.  Proses InvShiftRows()
8.  Proses InvMixColumns()
9.  Proses AddRoundKey() = Current State
    XOR Round Key
10. IF Rounds < 14 THEN
11. Kembali ke baris 5
12. ELSE
13. Proses InvSubBytes()
14. Proses InvShiftRows()
15. Proses AddRoundKey() = Current State
    XOR Round Key
16. Output plaintext
17. END IF
18. END Process
```

3.4 Pengujian Layanan *Web Service*

Pengujian perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program untuk mengetahui apakah fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan [11]. Black Box Testing sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian *black box* ini menitik beratkan pada fungsi system [12].

Pada pengujian *Layanan Web Service*, pengujian dilakukan mencoba semua endpoint web service dan hasil pengujian pada semua layanan web service Rest Api dengan autentikasi JWT dan Algoritma Aes-256 didapatkan persentase 100%. Seperti yang dijelaskan pada Tabel 3.

Tabel 3. Pengujian Aplikasi Layanan Webservice Dengan Blackbox

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Banyak pengujian	Ke-simpulan	Hasil Akurasi
1	Menampilkan *token* setelah *login*	Pilih *method POST Input* : *Path* URL : api/login *Request Body* : *Email:* yodaegan@gmail.com *Password* : 1711500312	Berhasil mendapatkan data user dan mendapatkan *token*	Sesuai Harapan	*5*	*Valid*	100 %
2	Menampilkan data lab dengan *method GET*	Pilih *method GET Input* : *Path* URL : api/jadwal/lab/{hari}{tanggal}	Server akan merespon dengan mengambil data list lab	Sesuai Harapan	*5*	*Valid*	100 %
3	Menampilkan data matakuliah dengan *method GET*	Pilih *method GET Input* : *Path* URL : api/jadwal/matkul	Server akan merespon dengan mengambil data list Matakuliah	Sesuai Harapan	*5*	*Valid*	100 %
4	Menampilkan data Dosen	Pilih *method GET Input* : *Path* URL : api/jadwal/dosen/	Server akan merespon	Sesuai Harapan	*5*	*Valid*	100 %

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian	Banyak pengujian	Ke-simpulan	Hasil Akurasi
	berdasarkan matakuliah yang dosen ajar dengan *method GET*	{id_matkul}	dengan menampil data list dosen yang sesuai				
5	Menampilkan data Kelompok berdasarkan matakuliah dan dosen dengan *method GET*	Pilih *method GET Input* : *Path* URL : api/jadwal/kelompok/ {id_matkul} {id_dosen}	Server akan merespon dengan menampil data list kelompok yang sesuai	Sesuai Harapan	*5*	*Valid*	100 %
6	Menampilkan Jadwal harian berdasarkan hari dan tanggal dengan *method GET*	Pilih *method GET Input* : *Path* URL : api/ jadwal/hari-ini {hari}{tanggal} *Request Header* : *Authentification:* Bearer token	Server akan merespon dengan menampil jadwal harian yang sesuai	Sesuai Harapan	*5*	*Valid*	100 %
7	Menampilkan Jadwal sekarang dengan *method GET*	Pilih *method GET Input* : *Path* URL : api/ jadwal/jam-sekarang	Server akan merespon dengan menampil jadwal yang sesuai	Sesuai Harapan	*5*	*Valid*	100 %
8	Mengenkripsi data kuliah pengganti dan cek data kuliah pengganti bentrok dengan jadwal yang sudah ada	Pilih *method Post Input* : *Path* URL : api/ jadwal/enkripsikp *Request Header* : *Authentification:* Bearer token *Request Body* : data kuliah pengganti	Server akan merespon dengan menampilkan data kuliah pengganti yang telah dienkripsi	Sesuai Harapan	*5*	*Valid*	100 %
9	Mendekripsi data kuliah pengganti dan meyimpan ke database	Pilih *method Post Input* : *Path* URL : api/ jadwal/dekripsikp/{id_user} *Request Body* : data enkripsi kuliah pengganti	Server akan merespon dengan menampilkan data telah disimpan	Sesuai Harapan	*5*	*Valid*	100 %
10	Mengenkripsi data pengajuan peminjaman lab	Pilih *method Post Input* : *Path* URL : api/ pinjam/enkripsipeminjaman *Request Body* : data kuliah pengganti	Server akan merespon dengan menampilkan data pengajuan peminjaman yang telah dienkripsi	Sesuai Harapan	*5*	*Valid*	100 %
11	Mendekripsi data peminjaman dan meyimpan ke database	Pilih *method Post Input* : *Path* URL : api/ pinjam/dekripsipeminjaman/{id_user} *Request Body* : data enkripsi peminjaman	Server akan merespon dengan menampilkan data telah disimpan	Sesuai Harapan	*5*	*Valid*	100 %

Pada Tabel 3 skenario pengujian yang dilakukan antara lain : *Login* Memakai *Post* bukan *Get* adalah untuk langsung mengirimkan data username dan password ke action login yang terdapat pada web service. Method POST akan mengirimkan data atau nilai langsung ke *action* untuk ditampung, tanpa menampilkan pada URL. Sedangkan method GET akan menampilkan data/nilai pada URL, kemudian akan ditampung oleh *action*. Menampilkan token setelah login, Menampilkan data lab dengan method GET, Menampilkan data matakuliah dengan method GET, Menampilkan data Dosen berdasarkan mata kuliah yang dosen ajar dengan method GET, Menampilkan data Kelompok berdasarkan matakuliah dan dosen dengan method GET, Menampilkan Jadwal harian berdasarkan hari dan tanggal dengan method GET, Menampilkan Jadwal sekarang dengan method GET , Mengenkripsi data kuliah pengganti dan cek data kuliah pengganti bentrok dengan jadwal yang sudah ada, Mendekripsi data kuliah pengganti dan menyimpan ke database, Mengenkripsi data pengajuan peminjaman lab, Mendekripsi data peminjaman dan menyimpan ke database, masing-masing dilakukan pengujian sebanyak 5 kali dengan hasil pengujian sesuai harapan dan hasil akurasi 100%

4. KESIMPULAN

Berdasarkan pengujian aplikasi yang telah dilakukan, maka dapat ditarik beberapa kesimpulan :

a. Membuat sistem peminjaman dan penggunaan lab untuk mengatur jadwal agar tidak bentrok.
b. Sistem ini menggunakan pengaman dengan Autentikasi JWT dan Algoritma AES-256 dengan pengujian *black box* testing berdasarkan web service REST API dan Algoritma AES 256 didapatkan persentase sebesar 100%

DAFTAR PUSTAKA

[1] P. Painem and H. Soetanto, "Sistem Presensi Pegawai Berbasis Web Service Menggunakan Metode Restfull Dengan Keamanan JWT Dan Algoritma Haversine," *Fountain Informatics J.*, vol. 5, no. 3, p. 6, 2020.
[2] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *Jurnal Edukasi dan Penelitian Informatika*, vol. 5, no. 1, p. 74, 2019.
[3] A. Rahmatulloh, H. Sulastri, and R. Nugroho, "Keamanan RESTful Web Service Menggunakan JSON Web Token (JWT) HMAC SHA-512," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 2, 2018.
[4] R. Rizal and A. Rahmatulloh, "Restful Web Service Untuk Integrasi Sistem Akademik Dan Perpustakaan Universitas Perjuangan," *J. Ilm. Inform.*, vol. 7, no. 01, p. 54, 2019.
[5] A. A. G. Y. Paramartha, G. K. Suryaningsih, and K. Y. E. Aryanto, "Implementasi Web Service Pada Sistem Pengindeksan Dan Praktik Kerja Lapangan," *J. Sains dan Teknol.*, vol. 5, no. 2, pp. 1–8, 2016.
[6] S. Sibagariang, "Penerapan Web Service Pada Perpustakaan Berbasis Android," *JurnalMaharjana Inf.*, vol. 1, no. 2, pp. 8–11, 2016.
[7] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," *Eksplora Inform.*, vol. 8, no. 1, p. 52, 2018.
[8] A. P. Nugroho and H. B. Suseno, "Keamanan Data Transaksi Nasabah Pada Aplikasi Bank Sampah Berbasis Web Menggunakan Algoritma AES," vol. 5341, no. April, pp. 9–17, 2020.
[9] A. Kusyanti and K. Amron, "Analisis Perbandingan Algoritma Advanced Encryption Standard Untuk Enkripsi Short Message Service (SMS) Pada Android," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 10, pp. 4281–4289, 2018.
[10] A. Pariddudin and F. Syauqi, "Penerapan Algoritma AES pada QR CODE untuk Keamanan Verifikasi Tiket," *Teknois J. Ilm. Teknol. Inf. dan Sains*, vol. 10, no. 2, pp. 43–52, 2020.
[11] W. N. Cholifah, Y. Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 3, no. 2, p. 206, 2018.
[12] U. Hanifah, R. Alit, and S. Sugiarto, "Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 33–40, 2016.