Jurnal TICOM: Technology of Information and Communication
Volume 12, Nomor 3, Mei 2024 Hal: 108-114 E-ISSN: 2962-7982

Implementasi Web Service Restful API dengan Autentikasi *Personal Access Tokens* dan Algoritma AES 256

Fatchur Shofyan[1*], Rizky Tahara Shita[2]

[1,2]Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, Jakarta, Indonesia
Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan, Jakarta Selatan, 12260
E-mail: [1*]fatchur.shofyan@gmail.com, [2]rizky.taharashita@budiluhur.ac.id
(*: corresponding author)

Abstrak—**Dalam era kemajuan teknologi informasi saat ini, pertukaran informasi yang akurat dan tepat dari satu pihak ke pihak lain dapat dilakukan dengan cepat dan efisien. Namun dengan semakin mudahnya dalam proses pertukaran data disisi lai juga meningkat pula risiko kejahatan didunia maya, seperti pembobolan sistem, pencurian data dan manipulasi data oleh pihak yang tidak bertanggung jawab untuk mendapatkan keuntungan pribadi, serta berdampak kerugian terhadap perusahaan baik secara finansial ataupun bukan finansial. Salah satu solusi yang efektif adalah dengan menerapkan metode autentikasi *Personal Access Tokens* yang dikombinasikan dengan algoritma AES-256 untuk enkripsi dan dekripsi data dalam proses transaksi, baik dari sisi pengguna maupun dari sisi sistem. Hal ini merupakan salah satu solusi yang efektif untuk menjaga keamanan sebuah sistem. Keamanan ini dapat diuji dengan berbagai cara seperti mencoba untuk melakukan *encription* ataupun *decription* dengan menggunakan kata kunci secara acak yang di mana hasil yang diberikan oleh sistem akan selalu salah atau gagal dan juga mencoba menggunakan *Personal Access Tokens* secara acak juga untuk menguji ketahanan sistem yang di mana sistem akan menginformasikan bahwa *user* tidak memiliki hak untuk mengakses data pada sistem. Hasil Pengujian aplikasi keamanan ini menunjukkan keberhasilan dalam peningkatan keamanan aplikasi dengan penerapan algoritma AES 256 pada layanan web *service* yang diakses oleh aplikasi karyawan PT. 3 Consulting Services dengan tetap menjaga keaslian dan keamanan data yang disimpan pada *database*.**

Kata Kunci—**API Token, *Personal Access Tokens*, Web Service, RESTful API, Keamanan API, AES-256.**

Abstract—***In the era of advancing information technology, accurate and precise information exchange between parties can be done quickly and efficiently. However, with the increasing ease of data exchange, the risk of cybercrimes also rises, such as system breaches, data theft, and data manipulation by irresponsible parties seeking personal gain, which can result in financial or non-financial losses for companies. An effective solution to address this challenge is to implement the Personal Access Tokens authentication method combined with AES-256 algorithm for data encryption and decryption in transaction processes, both from user and system perspectives. This approach is an effective measure to maintain system security. Security can be tested in various ways, such as attempting encryption or decryption using randomly generated passwords, where the system will always yield incorrect or failed results, and also by trying to use Personal Access Tokens randomly to test system resilience, wherein the system will indicate that we do not have the authorization to access data. The results of security testing demonstrate the successful enhancement of application security through the implementation of AES 256 algorithm in the web service accessed by employees of PT. 3 Consulting Services, while ensuring the authenticity and security of stored data in the database.***

Keyword—***API Token, Personal Access Tokens, Web Service, RESTful API, API Security, AES-256.***

I. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah membawa tantangan baru, terutama dalam pengelolaan data karyawan dan keamanan sistem secara menyeluruh di berbagai sektor. Terlebih lagi meningkatnya kasus pencurian data dan serangan DDoS pada perusahaan besar yang sudah terjadi beberapa tahun belakang [6]. Pada penelitian ini PT. 3 Consulting Services sebagai perusahaan yang bergerak dalam sektor tersebut diharapkan dapat memanfaatkan teknologi untuk meningkatkan keamanan data karyawan.

Keberadaan sistem dengan keamanan yang masih standar, dapat menimbulkan risiko potensial serangan dari pihak-pihak yang tidak bertanggung jawab. Kondisi ini dapat berdampak pada perusahaan secara finansial karena aplikasi yang berjalan berapa pada ruang lingkup keuangan. Selain itu, ketidakmampuan untuk mengamankan aplikasi dapat merugikan reputasi perusahaan dan kepercayaan karyawan. Beberapa alternatif solusi melibatkan peningkatan keamanan sistem. Alternatif tersebut dapat mencakup pembaruan sistem secara rutin agar selalu menggunakan teknologi yang terbaru, audit keamanan berkala, penerapan enkripsi data.

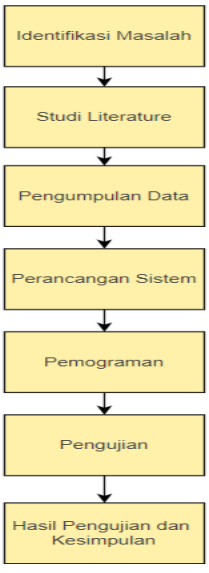
Penelitian terkait dengan kriptografi dilakukan oleh [8] menyatakan bahwa pertukaran data antara dua pihak yang telah dienkripsi menggunakan algoritma AES 256 dapat memudahkan untuk dikelola dan diterapkan pada rumah sakit. Penelitian berikutnya membahas algoritma AES 256 untuk layanan API dari [1] dengan yang memiliki judul "Implementasi Algoritma AES-256 Untuk Pengamanan Layanan API Pada RESTful Dengan Autentikasi JSON Web Tokens" pada penelitian di atas bertujuan untuk melakukan pengamanan pada layanan API yang dibuat. Implementasi

autentikasi dengan JWT pada *web service* [3]. Dimana berbeda dengan penelitian sebelumnya, penelitian ini menggunakan metode *Personal Access Token* sebagai pengaman pada autentikasi, dan juga menabahkan fitur pemantauan log aktivitas dari setiap pengguna yang bermanfaat untuk mengamati kegiatan dari setiap pengguna untuk mengetahui apa saja yang dilakukan selama mengakses sistem karyawan ini dan deteksi di sini jika terjadi serangan DDoS.

Implementasi Algoritma AES 256 GCM untuk pengamanan data pribadi [6]. Terdapat penelitian terdahulu mengenai kasus yang sama, penelitian tersebut dijadikan acuan dan juga sebagai referensi untuk melakukan penelitian ini, beberapa jurnal tersebut seperti yang di atas. Perbedaan dari penelitian saat ini menggunakan metode AES 256 CBC beroperasi dengan mengenkripsi setiap blok data secara terpisah, di mana setiap blok di-XOR-kan dengan blok sebelumnya sebelum dienkripsi. Metode ini membutuhkan IV (*Initialization Vector*) yang unik pada setiap proses untuk memperkuat keamanan. Namun tidak memiliki fitur autentikasi data bawaan, oleh sebab itu bisa dikombinasikan dengan *Personal Access Token* sebagai *layer* untuk autentikasi data.

II. METODE PENELITIAN

Pada penelitian ini berdasarkan masalah pada PT 3 Consulting Services. Di mana sistem yang digunakan pada saat ini masih memakai pengamanan yang masih sangat standar yang dikhawatirkan memiliki potensi besar terhadap pencurian data atau serangan dari pihak tidak bertanggung jawab [5], di mana bisa berakibat merugikan ataupun menjatuhkan nama baik perusahaan. Penerapan metode merupakan rangkaian proses dari aktivitas yang sistematis dan terstruktur yang dilakukan oleh peneliti agar dapat memperoleh hasil dari penelitian yang dilakukan.



Gambar 1. Penerapan Metode

Berdasarkan Gambar 1, untuk melakukan penelitian terdapat prosedur untuk kegiatan selanjutnya dilaksanakan. Dimulai dari mengidentifikasi permasalahan, mempelajari materi untuk mengumpulkan teori-teori yang bisa mendukung penelitian, seperti pada penelitian ini menggunakan metode AES 256 maka yang dipelajari adalah metode tersebut. Kemudian mengumpulkan data, data yang digunakan adalah data karyawan PT 3 Consulting Services. Selanjutnya melakukan perancangan sistem mulai dari perancangan tampilan, *database*, hingga pengamanan data. Kemudian lakukan percobaan dengan metode yang telah dibuat.

A. Pengumpulan Data

Pada tahap penyusunan penelitian ini data yang digunakan harus terperinci dan informasi yang terkait untuk mendukung deskripsi dan pembahasan sesuai dengan tujuan penelitian. Proses pengumpulan data penelitian ini dilakukan dengan metode observasi dan wawancara sebagai cara mendapatkan data yang berkualitas [8]. Sementara itu, dalam proses pemeriksaan, penting dalam penelitian ini untuk menyatukan berbagai masukan dan data dari berbagai sumber, yang nantinya akan dianalisis dan dievaluasi guna memastikan solusi yang sesuai dengan permasalahan yang ada saat ini. Metode penyatuan masukan yang digunakan bagian dalam pemeriksaan ini adalah:

1) *Observasi:* Dilakukan pengamatan langsung di PT. 3 Consulting Services secara langsung bertujuan untuk memperoleh informasi seputar keamanan dan juga metode yang digunakan pada sistem karyawan terpadi [9].

2) *Wawancara:* Proses wawancara dilakukan dengan mengajukan sejumlah pertanyaan secara lisan dan tulisan untuk mengumpulkan informasi yang lebih terinci [9].

3) *Studi Pustaka:* Pengumpulan data dilakukan melalui berbagai sumber yang berhubungan dengan topik yang dibahas, seperti jurnal, buku, artikel, *e-book*, dan situs-situs internet [9].

B. Metode Pengujian Alpha

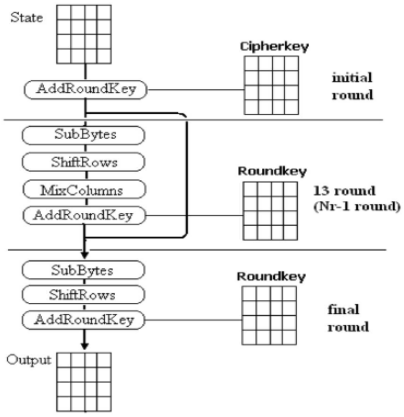
Pengujian alpha, salah satu bentuk pengujian *user acceptance* yang bersifat terbatas, dilakukan secara internal dan dibatasi [10]. Proses ini menuntut peneliti secara eksklusif untuk menguji produk atau aplikasi. Alpha *testing* fokus pada pencarian bug atau kekurangan dasar dalam produk, memastikan fungsi-fungsi dasar telah berjalan dengan baik [9].

C. Penerapan Metode *AES 256*

Enkripsi AES-256, atau *Advanced Encryption Standard* yang memiliki panjang kunci 256-bit, adalah sebuah algoritma enkripsi blok simetris yang diakui sebagai standar keamanan oleh *National Institute of Standards and Technology* [1]. Dikembangkan oleh Vincent Rijmen dan Joan Daemen, AES diterima sebagai pengganti standar enkripsi federal sebelumnya, yaitu *Data Encryption Standard* (DES), pada tahun 2001. Dalam strukturnya, AES memanfaatkan operasi dasar seperti SubBytes, ShiftRows, MixColumns, dan AddRoundKey, yang merupakan bagian dari jaringan substitusi-permutasi.

Pada proses enkripsi AES 256, setelah tahap AddRoundKey, *state* akan mengalami serangkaian tahap yang

disebut dengan *Round Function* yang terdiri dari SubByte, ShiftRows, MixColumns, dan AddRoundKey kembali sebanyak putaran yang diinginkan. Namun pada putaran terakhir, tahap MixColumns tidak dilakukan, sehingga disebut sebagai *Final Round* [9]. Alur proses enkripsi dapat dilihat pada Gambar 2.



Gambar 2 Ilustrasi Proses Enkripsi AES [8]

1) *AddRoundKey:* Pada tahap awal enkripsi AES, AddRoundKey dilakukan menggunakan kunci utama, sementara tahap putaran lainnya menggunakan kunci putaran. AddRoundKey melibatkan operasi XOR pada setiap byte dalam *state*, mempertahankan ukuran 4x4 state. SubBytes, ShiftRows, MixColumns, dan tahap lain mengikuti AddRoundKey, membentuk proses unik yang membedakan AES dari algoritma enkripsi lainnya [2].

2) *SubBytes:* Pada langkah SubByte dalam enkripsi AES, setiap byte di dalam *state* mengalami transformasi dengan menggantinya dengan nilai dari tabel S-Box yang sesuai. S-Box adalah tabel SubByte yang disematkan dalam algoritma AES untuk menghasilkan hasil enkripsi yang aman dan tidak dapat ditebak. Proses ini memastikan keunikan hasil enkripsi AES [6]. Nilai S-Box yang digunakan algoritma AES ditunjukkan dengan nilai hexadesimal pada Gambar 3. Mekanisme penggantian dalam SubByte AES melibatkan pemecahan byte ke dalam dua bagian, misalnya 0xA dan 0x9, kemudian dilakukan substitusi sesuai dengan nilai yang dihasilkan dari perpotongan antara kedua nilai tersebut. Substitusi tersebut kemudian digunakan pada byte dalam *state* AES.

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Gambar 3. S-Box [6]

3) *ShiftRows:* Proses *ShiftRows* dalam enkripsi AES mengubah state dengan melakukan pergeseran searah dan rotasi pada setiap barisnya. Jumlah pergeseran pada setiap baris bergantung pada nilai baris (r), yang mewakili posisi *baris* tersebut [6].

4) *MixColumns:* MixColumns merupakan salah satu tahap kunci dari algoritma Advanced Encryption Standard (AES), yang memungkinkan setiap bit dari data plaintext untuk berdampak pada sebanyak mungkin bit dari data ciphertext [6]. Dalam MixColumns, setiap kolom array/state dikalikan dengan polinom a(x) mod (x4+1), dengan setiap kolom diperlakukan sebagai polinom dengan empat suku pada Galois Field (GF)(28). Polinom a(x) yang digunakan di dalam tahap ini adalah a(x) = {03}x3 + {01}x2 + {01}x + {02}.

Proses dekripsi AES 256 dilakukan dengan cara yang berlawanan pada setiap tahapnya dibandingkan dengan proses enkripsi. Tahap-tahap dalam dekripsi AES adalah InvSubBytes, InvShiftRows, InvMixColumns, dan AddRoundKey [6]. Proses dekripsi AES memungkinkan untuk mendapatkan teks awal kembali dari teks terenkripsi.

D. *Web Service*

Web service adalah teknologi yang memungkinkan aplikasi untuk berbagi data dan berinteraksi melalui internet, tanpa tergantung pada platform atau bahasa pemrograman tertentu. Ini memfasilitasi integrasi aplikasi dengan menggunakan pesan SOAP yang umumnya dikirim melalui protokol standar seperti HTTP dalam bentuk XML [2], *Web Services* merupakan suatu fitur yang disediakan oleh penyedia layanan berupa layanan informasi untuk sistem yang lain sehingga antar sistem dapat saling berinteraksi melalui *services* yang diberikan [4].

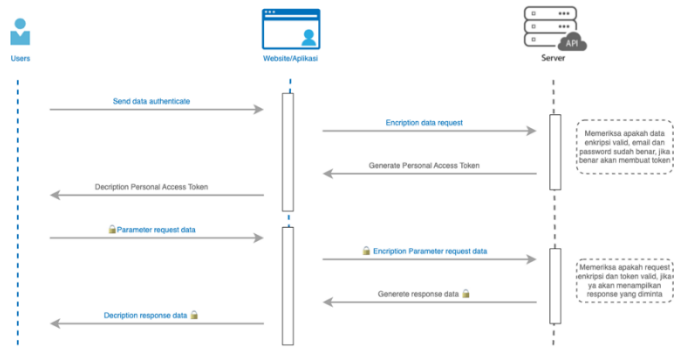
E. *RESTful API*

RESTful API adalah sebuah API yang menggunakan arsitektur REST untuk mengatur komunikasi antara client dan server [7]. Ini memungkinkan aplikasi untuk mengakses, mengubah, dan menghapus data melalui HTTP requests, serta menggunakan URIs untuk mengidentifikasi sumber daya. RESTful API mendukung format data yang berbeda, seperti JSON atau XML [3], dan sering digunakan dalam pembangunan aplikasi mobile dan web yang membutuhkan akses data melalui internet. Keunggulan utamanya adalah fleksibilitas dan kapabilitas yang tinggi, yang memudahkannya diterapkan pada berbagai jenis aplikasi dan platform.

F. *Arsitektur Sistem*

Sistem karyawan terpadu ini terdiri dari 2 sisi yaitu sisi klien dan sisi, yang di mana pada sisi bagian klien bisa dibedakan menjadi 2 platform yaitu aplikasi (mobile) dan website. Interaksi antara 2 bagian ini menggunakan metode RESTful web service melalui protokol HTTP [10]. Yang di mana untuk pertama kali klien harus mendapatkan sebuah token terlebih dahulu untuk mengakses sistem ini dengan cara melakukan proses autentikasi, setelah autentikasi berhasil dijalankan maka server akan memberikan respons berupa data yang memiliki format JSON, yang di mana data tersebut bisa

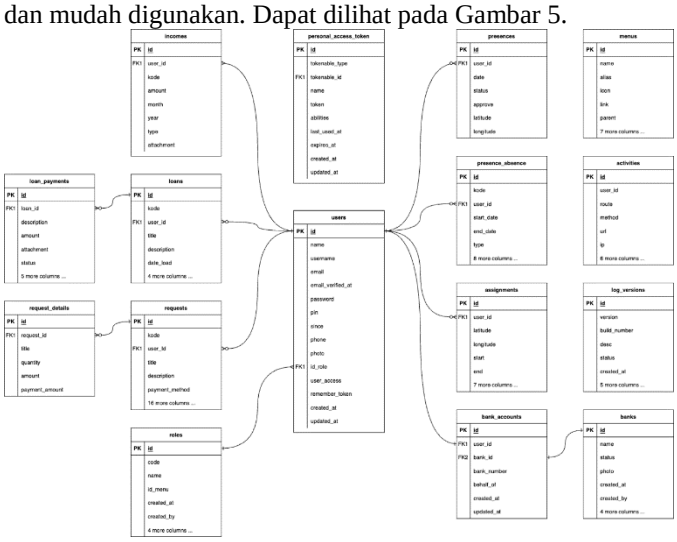
digunakan oleh klien sesuai dengan kebutuhannya [11]. Berikut merupakan gambar dari sistem otorisasi menggunakan *Personal Access Tokens*. Dan rancangan arsitektur sistem dapat dilihat pada Gambar 4.



Gambar 4. Arsitek Sistem

G. Rancangan Basis Data

Proses perancangan sistem basis data melibatkan identifikasi kebutuhan pengguna, pembuatan model konseptual dan logis, implementasi fisik, pengujian, serta pemeliharaan dan peningkatan sistem, sementara rancangan *layer* meliputi desain antarmuka pengguna yang ramah, logika bisnis, pengaturan akses data, dan penyediaan layanan tambahan untuk memastikan sistem basis data yang efisien dan mudah digunakan. Dapat dilihat pada Gambar 5.



Gambar 5. Rancangan basis data

H. *Rancangan Web Service*

Berikut ini adalah desain *web service* yang akan menjelaskan tentang konsep RESTful API. Setiap API yang diakses akan mengirimkan sebuah respons. Dan secara umum berikut skema respons API yang dibangun. Pada Tabel 1 mencantumkan rancangan *web service* yang akan dibangun.

TABEL I
RANCANGAN WEB SRVICE

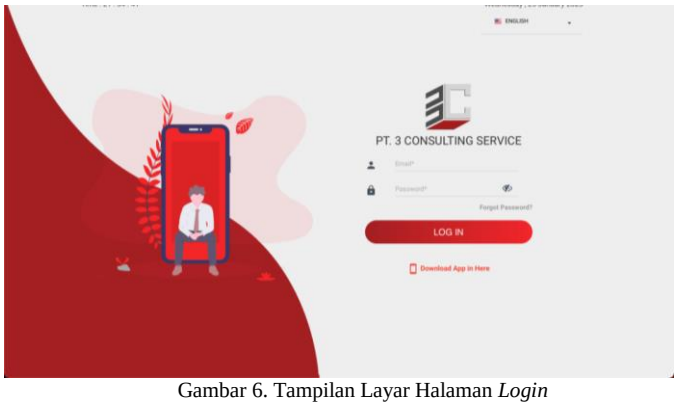
No	Nama Variabel	Tipe Variabel	Deskripsi
1	*success*	*Boolen*	Berisikan *true* jika berhasil, dan *false* jika gagal atau terjadi kesalahan ataupun *error* pada sistem API
2	*message*	*String*	Berisikan informasi berhasil atau gagal dan bersifat opsional
3	data	*Array*	Berisikan *list* data, atau *object* sebagai respons
4	*token*	*String*	Berisikan *token* yang digunakan untuk semua layanan

III. HASIL DAN PEMBAHASAN

Bagian ini berisi analisis, hasil implementasi, pengujian, dan pembahasan dari topik penelitian, yang biasanya dikembangkan berdasarkan metodologi penelitian. Pada bagian ini, biasanya, terdapat penjelasan yang disertai dengan gambar, tabel, dan bentuk visual lainnya untuk memperjelas hasil penelitian.

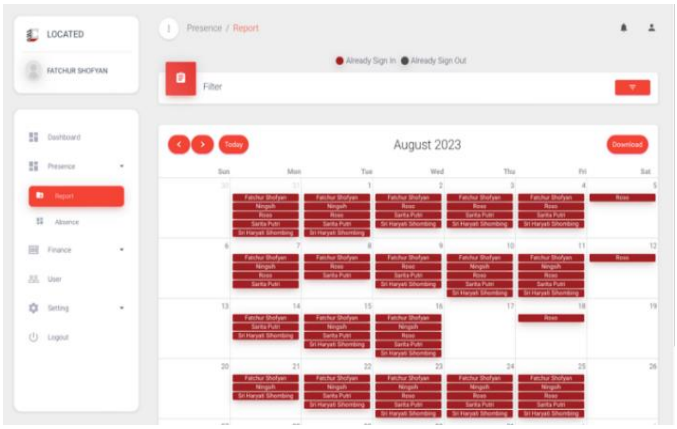
A. *Tampilan Layar*

1) *Tampilan Layar Halaman Login:* Tampilan layar *form login* yang digunakan pada sistem yang telah dibuat dapat dilihat pada Gambar 6.



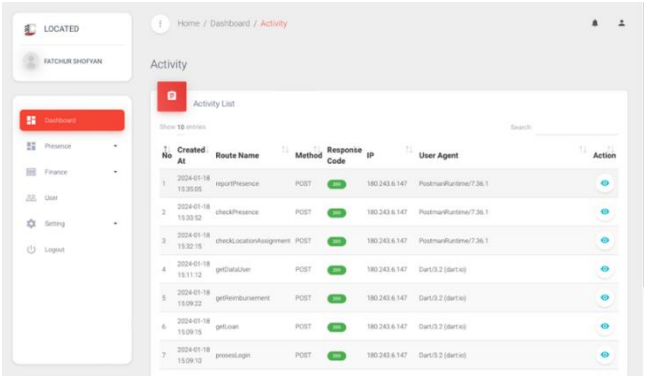
Gambar 6. Tampilan Layar Halaman *Login*

2) *Tampilan Layar Halaman Absensi:* Pada Gambar 7 menampilkan tampilan layar pada halaman absensi yang berisikan *list* berbentuk kalender untuk mempermudahkah dalam membaca secara periode per bulan.



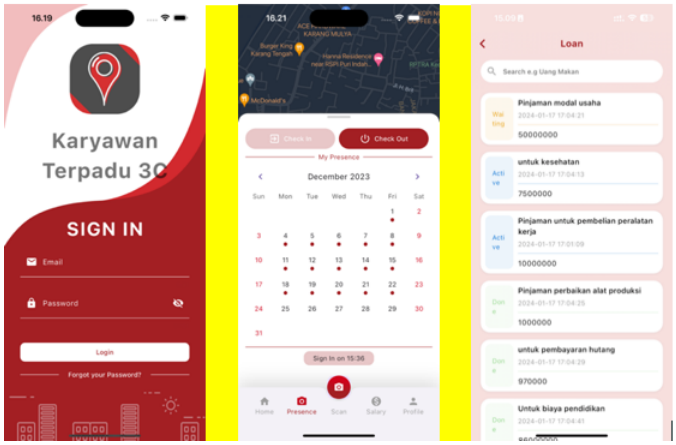
Gambar 7. Tampilan Layar Halaman Absensi

3) *Tampilan Layar Halaman Activity:* Halaman *activity* menampilkan daftar aktivitas yang dilakukan oleh setiap *user* sesuai dengan yang ditampilkan pada Gambar 8.



Gambar 8. Tampilan Layar Halaman *Activity*

4) *Tampilan Aplikasi Mobile:* Pada Gambar 9, terlihat tampilan aplikasi *mobile* yang menunjukkan halaman *login*, absensi, dan pinjaman pada aplikasi yang telah dikembangkan.



Gambar 9. Tampilan Aplikasi *Mobile*

B. *Pengujian*

Pada tahap pengujian terdapat 2 komponen pengujian yang dilakukan antara lain pengujian ketahanan sistem dan juga pengujian pengamanan enkripsi dan dekripsi.

1) *Pengujian Ketahanan Rancangan Program:* Pada Tabel 2 disajikan hasil pengujian ketahanan dari program yang telah dibuat, di mana dilakukan dengan mengakses data secara serentak oleh beberapa *user*. Dengan kriteria pengujian yang dilakukan oleh 15 pengguna yang melakukan proses sesuai dengan skenario secara bersamaan dengan melakukannya secara berulang sebanyak 2 hingga 3 kali. Yang di diharapkan persentase keberhasilan di atas 75% dan dengan waktu rata-rata setiap skenario di bawah 3 detik.

2) *Pengujian Proses Enkripsi dan Dekripsi:* Pada Tabel 3 menampilkan hasil pengujian alpha untuk proses enkripsi dan dekripsi yang telah di lakukan pengujian sebelumnya. Dengan cara setiap skenario akan memasukkan data sampel yang telah disediakan dan akan menjalankan pada bagian masing-masing proses yang diharapkan sistem dapat memberikan timbal balik yang positif yang di mana dapat memproses hasil enkripsi tersebut dan juga mengembalikan kepada pengguna yang melakukan permintaan.

Berdasarkan Tabel 2 yang merupakan ketahanan sistem dan rata-rata waktu yang diperlukan dari setiap proses dan Tabel 3 menampilkan tabel terkait proses terjadi enkripsi dan dekripsi yang menunjukkan penambahan proses enkripsi dan dekripsi ini tidak terlalu memakan banyak waktu pada setiap prosesnya.

TABEL 1
HASIL PENGUJIAN KETAHANAN RANCANGAN PROGRAM

No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Banyak Pengujian	Persentase Keberhasil-an	Waktu Enkripsi & Dekripsi
1	Melakukan *request* pengecekan versi secara bersamaan	Server berhasil memproses permintaan pengecekan versi dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Hasil sesuai dengan yang diharapkan dapat memproses permintaan data versi aplikasi secara bersamaan dengan waktu yang singkat	30	100%	2,527 Detik
2	Melakukan proses *login* secara bersamaan	Server berhasil memproses permintaan *login* dan memberikan respons terhadap permintaan yang dilakukan secara bersamaan dalam waktu kurang dari 3 detik	Berhasil menjalan semua permintaan *login* secara bersamaan dengan waktu yang singkat	30	100%	0,560 Detik
3	Melakukan *request* GET data *user* secara bersamaan	Server berhasil memproses permintaan data *user* dan memberikan balikan terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Berhasil memproses permintaan secara bersamaan dengan waktu yang singkat	45	100%	1,179 Detik
4	Melakukan *request* GET data *income user* secara bersamaan	Server berhasil memproses permintaan data gaji dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Hasil Sesuai dengan yang diharapkan dapat memproses pemintaan secara bersamaan dengan waktu yang singkat	45	100%	0,923 Detik
5	Melakukan *request* GET data pinjaman *user* secara bersamaan	Server berhasil memproses permintaan data pinjaman dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Tidak sesuai dengan harapan dikarenakan hanya 33% yang berhasil terproses. Walau dengan rata-rata waktu proses	45	33%	2,008 Detik

			masih di bawah 3 detik			
6	Melakukan *request* GET data permohonan *user* secara bersamaan	Server berhasil memproses permintaan data permohonan dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Sesuai dengan yang diharapkan yaitu dapat memproses secara bersamaan dengan waktu yang secara singkat	45	100%	1,928 Detik
7	Melakukan *request* pengecekan kehadiran secara bersamaan	Server berhasil memproses pengecekan kehadiran dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Berhasil memproses permintaan secara bersamaan dengan waktu yang singkat	45	100%	0,402 Detik
8	Melakukan proses kehadiran secara bersamaan	Server berhasil memproses kehadiran pengguna dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Tidak sesuai dengan harapan dikarenakan hanya 33% yang berhasil terproses. Walau dengan rata-rata waktu proses masih di bawah 3 detik	45	33%	1,562 Detik
9	Melakukan proses ganti kata sandi secara bersamaan	Server berhasil memproses perubahan kata sandi dan memberikan respons terhadap permintaan secara bersamaan dalam waktu kurang dari 3 detik	Berhasil memproses permintaan secara bersamaan dengan waktu yang singkat	45	100%	0,625 Detik

TABEL 2
HASIL PENGUJIAN ALPHA TESTING ENKRIPSI DAN DEKRIPSI

No	Skenario Pengujian	*Test Case*	Hasil Yang Diharapkan	Hasil Pengujian
1	User mengisi *form login*	JIFckFZ/Wmi3+49PeXCQs6JscT PvWEMG/lRJra8Q0AUcE8wVon Q45HDrvk+e3gLIGV0c3v16pg6er +Ke5JKTik1NYSLV+zjdduaeY5CWIUw=	Berhasil melakukan dekripsi dan memberikan balikan data *user* dan *token*	Sistem berhasil melakukan dekripsi terhadap data yang di beri dan memberikan kembali data sesuai dengan yang di minta oleh pengguna
2	User membuka menu absensi	jCyysA0ClaiwQRINKl2MYg==	Server akan melakukan proses dekripsi dan merespons dengan memberikan data lokasi absensi *user*	Sesuai dengan yang diharapkan sistem dapat memberikan respons dengan baik terhadap data yang diminta
3	User meminta riwayat data absensi	jCyysA0ClaiwQRINKl2MYg==	Server melakukan proses *dekripsi* lalu memberikan *response* dengan data absensi	Hasil sesuai dengan yang harapan, yaitu dapat memproses data enkripsi dengan baik
4	User melakukan absensi secara *online*	7n1fPVg9+HonejwyOwOkTEbKwl1u/ Yi5Hr6VlWjnqgXXLj73jx2vocpb4eb YctvLZ75esttxLLHWcpg/ETz2jfRgJ aab1uvtd1igCo3KQ6NNjqoYeS1qdeP fwMePrDxfyG0uBT7KFux1KbYikPa gPGFdCNPTzS7N5e2J7+2cv3kNh4d MIyKvzHU4RNckhDHt	Server akan melakukan proses *dekripsi* dan akan menyimpan data tersebut pada *database*	Sistem dapat melakukan proses dekripsi dengan baik dan memberikan umpan balik terhadap *user*
5	User memilih menu gaji dan memilih tahun yang dinginkan	jCyysA0ClaiwQRINKl2MYg==	Server akan memberikan respons dengan mengambil data gaji yang disimpan dalam *database*.	Sesuai dengan yang diharapkan sistem dapat memproses dengan baik
6	User memilih menu ganti *password* dan mengisi *form* yang ada	xkF4YSCH7eO7hbh8DWc/4E2Jbop 8XyrM87ulHIKtj9FWM+zKpWO0M 6oESXGXmf77Ma50hooJTowXo5hq CmtHhommtoXcPsR+Hxpe6Bf789luR YzmGaqQWGCYJ2dehxul		

IV. PENUTUP

Berdasarkan diskusi dan data dari hasil wawancara di PT. 3 Consulting Services, dapat diambil kesimpulan dan saran yang bermanfaat bagi semua pihak yang terlibat dalam menghadapi permasalahan. Pertama, mengimplementasikan penggunaan *Personal Access Tokens* pada setiap proses dapat menghindari akses yang tidak sah dari pihak luar yang ingin mencoba mengakses sistem. Kedua, penggunaan enkripsi dan dekripsi dengan algoritma AES 256 CBC pada setiap data yang dikirim maupun diterima membuat peningkatan pada keamanan sistem, sehingga tidak dapat menghindari pemalsuan data dan tetap menjaga keaslian dari data. Terakhir, dengan cara menyimpan setiap transaksi atau proses yang dijalankan ketika mengakses API dapat dengan mudah mendeteksi aktivitas sistem ataupun juga memantau kegiatan dari setiap pengguna dan dapat mendeteksi jika terdapat serangan DDoS pada sistem. Langkah-langkah ini secara bersama-sama berkontribusi pada penguatan infrastruktur keamanan dan memastikan integritas operasi di PT. 3 Consulting Services, melindungi dari ancaman dan kerentanan potensial.

UCAPAN TERIMA KASIH

Terima kasih kepada manajemen PT. 3 Consulting Services atas kesempatan yang telah diberikan untuk melakukan penelitian dan membantu menyelesaikan masalah perusahaan ini..

REFERENSI

[1] A. Ardiansyah and M. Kurniasih, "Implementasi Algoritma AES-256 Untuk Pengamanan Layanan API Pada Restful Dengan Autentikasi JSON Web Tokens," *Seminar Nasional Inovasi Teknologi (SNITek)*, pp. 315-327, 2019.

[2] G. Y. Gustiegan and P. Painem, "Implementasi Web Service RESTFul Dengan Autentikasi JSON Web Token Dan Algoritma AES 256 CBC Untuk Aplikasi Peminjaman Laboratorium Berbasis Mobile Pada Universitas Budi Luhur," *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 19, pp. 9-16, 2022.

[3] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada interoperabilitas arsitektur berbasis RESTful Web Services.," *Jurnal Edukasidan Penelitian Informatika (JEPIN)*, vol. 5, pp. 74-79, 2019.

[4] I. Kusuma, A. Susanto and I. U. W. Mulyono, "Implementasi RESTFul Web Service Dengan Otorisasi OAUTH 2.0 Pada Sistem Pembayaran Parkir.," *SIMETRIS*, vol. 10, pp. 391-404, 2019.

[5] A. Azanuddin, S. Yakub and J. Prayudha, "Implementasi Keamanan Citra Menggunakan Algoritma AES-128 Dengan Aplikasi Client-Server," *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, vol. 7, pp. 51-61, 2022.

[6] E. S. Marsiani, I. Setiadi and A. Cahyo, "Implementasi Sistem Keamanan AES 256-Bit GCM Guna Mengamankan Data Pribadi," *Jurnal Rekayasa Komputasi Terapan (JRKT)*, vol. 01, pp. 108-114, 2021.

[7] A. Y. Nashikhuddin, J. Karaman and Y. Litanianda, "Implementasi Api Restful Dengan Json Web Token (JWT) Pada Aplikasi E-Commerce Thrifty Shop Untuk Otentikasi Dan Otorisasi Pengguna," *Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, vol. 7, pp. 239-246, 2023.

[8] R. S. Rohman, D. A. Firmansah and E. Ermawati, "Sistem Informasi Decrypt Respon Bridging BPJS Kesehatan Dengan Algoritma AES 256," *JURNAL RESPONSIF*, vol. 04, pp. 142-151, 2022.

[9] R. Febrianto and S. Waluyo, "Implementasi Algoritme Kriptografi Advanced Encryption Standard (AES-256) Untuk Mengamankan Data Base Penilaian Karyawan Pada KJPP NDR," *Bit* (*Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 20, pp. 44-49, 2023.

[10] P. Painem and H. Soetanto, "Sistem Presensi Pegawai Berbasis Web Service Menggunakan Metode Restfull Dengan Keamanan JWT Dan Algoritma Haversine," *Fountain of Informatics Journal*, vol. 05, pp. 6-11, 2020.

[11] F. I. Pratama, Mustagfirin and A. Fachreza, "Rancang Bangun Aplikasi Presensi Multi Event Dengan QR-Code Berbasis Restful Web Service," *Jurnal Teknologi dan Sistem Informasi (JURTEKSI)*, vol. 7, pp. 15-22, 2020.