ISSN:2302-8149
e-ISSN:2540-9719

Implementasi *JSON Web Token* pada *Authentication* dengan Algoritma HMAC SHA-256

Implementation of JSON Web Token on Authentication with HMAC SHA-256 Algorithm

[1]Ficry Cahya Ramdani, [2]Alam Rahmatulloh*, [3]Rahmi Nur Shofa
[123]Informatika, Fakultas Teknik, Universitas Siliwangi
Jalan Siliwangi Nomor 24, Tasikmalaya, Jawa Barat, Indonesia
*e-mail: alam@unsil.ac.id

(***received*:** 28 November 2022, ***revised*:** 27 Desember 2022, ***accepted*:** 28 Desember 2022)

Abstrak

Pertumbuhan teknologi informasi yang semakin cepat dipengaruhi oleh era globalisasi untuk mempercepat akses informasi. Hal ini menimbulkan masalah baru, karena perbedaan harus menghasilkan informasi yang relevan. Tentu saja integrasi sistem akan diperlukan. *Web Service* merupakan solusi integrasi sistem yang tidak mempertimbangkan *platform*, arsitektur, atau bahasa pemrograman yang digunakan dalam sumber yang berbeda. Keamanan pada *web service* dinilai belum diterapkan. Teknologi *JSON Web Token* (JWT) merupakan mekanisme autentikasi pada *web service* dan akan sangat berpengaruh pada hal keamanan data. Implementasi ini mengoptimasi keamanan JWT dengan algoritma HMAC SHA-256. Pengujian dilakukan dengan dua sistem informasi dengan membandingkan ukuran kinerja apabila teknologi JWT diterapkan pada Sistem Informasi Tim Bebersih Masjid. Hasil menunjukkan dalam implementasi JWT pada Windows Server 2019 (VM) sebesar 462,8 ms dengan rata-rata *size data* yang dihasilkan sebesar 8,59 kb. Pengujian pada sistem operasi Windows 10 diperoleh rata-rata kecepatan sebesar 216,25 ms dengan rata-rata *size data* yang dihasilkan sebesar 8,59 kb. Hasil pada Windows Server 2019 (VM) dari pengujian kinerja JWT sendiri mendapatkan hasil tertinggi, dikarenakan penggunaan *virtual machine* yang dinilai memakan RAM yang banyak menjadikan performa yang dihasilkan 2 kali lebih tinggi.

Kata kunci: Algoritma, HMAC SHA-256, *JSON Web Token*, *Web Service*, *Virtual Machine*

Abstract

The rapid growth of information technology is influenced by globalization to accelerate access to information. This creates new problems, as differences must produce relevant information. Of course, system integration will be required. Web Service is a system integration solution that does not consider the platform, architecture, or programming language used in different sources. The security of web service is considered not yet implemented. The JSON Web Token (JWT) technology is an authentication mechanism for web service and will have a significant impact on data security. This implementation optimizes JWT security with the HMAC SHA-256 algorithm. Testing is conducted on two information systems by comparing the performance size when JWT technology is applied to Tim Bebersih Masjid Information System. The results show that the implementation of JWT on Windows Server 2019 (VM) is 462.8 ms with an average data size of 8.59 kb. Testing on the Windows 10 operating system obtained an average speed of 216.25 ms with an average data size of 8.59 kb. The result on Windows Server 2019 (VM) from the JWT performance test itself obtained the highest result, due to the use of virtual machine which is considered to consume a lot of RAM, resulting in performance that is 2 times higher.

Keywords: *Algorithm, HMAC SHA-256, JSON Web Token, Web Service, Virtual Machine*

 ISSN:2302-8149
e-ISSN:2540-9719

1 Pendahuluan

Pertumbuhan teknologi yang cepat menyebabkan persaingan yang ketat dan tingkat percepatan yang semakin inovatif. Era VUCA (*Volatility*, *Uncertainty*, *Complexity*, dan *Ambiguity*) menggambarkan situasi saat pemanfaatan teknologi informasi dituntut untuk bergerak cepat dalam menghadapi perubahan yang mendadak dan tidak dapat diprediksi [1]. Sehingga, perkembangan teknologi informasi perlu melakukan perbaikan strategi untuk mendorong percepatan digitalisasi agar menghindari resiko yang merugikan [2].

Sistem informasi dapat berhasil jika meningkatkan kinerja individu, sedangkan bagi manajer mengurangi pengeluaran menjadi indikator dalam kesuksesan sistem informasi [3]. Sebagian penerapan teknologi masih dikatakan belum sepenuhnya diterapkan, terlihat dari implementasi sistem informasi yang belum menerapkan *web service* [4]. Penerapan *web service* dengan teknologi *JSON Web Token* (JWT) ini akan merubah proses *authentication* dalam meningkatkan keamanan data pada sistem informasi. JWT merupakan sebuah token berbentuk string *JSON* yang sangat pada dalam ukuran dan menghasilkan pertukaran informasi dengan melakukan sistem autentikasi [5].

Permasalahan keamanan menjadi poin penting yang disebabkan adanya kerentanan keamanan dan *authentication* yang disebabkan penerapan arsitektur *Representational State Tranfer* (REST) dan dijalankan melalui *JavaScript Object Notation* (JSON) yang membutuhkan penerapan teknologi JWT. Teknologi JWT akan bekerja dalam meningkatkan keamanan *authentication system* sehingga *user* merasa aman dalam menggunakan sistem tersebut. Selain itu, teknologi JWT akan bekerja dengan algoritma HMAC SHA-256 [6] sebagai penyokong sistem agar menghasilkan fungsi *hash* spesifik yang tidak terbaca.

Menerapkan JWT pada sistem informasi akan sangat efektif dalam hal keamanan informasi. JWT akan menentukan integritas data yang terkirim, sehingga data yang tersedia pada token tidak dapat dimanipulasi. Selain untuk mengamankan data, penerapan JWT pada sistem informasi diharapkan menjadi pemecah permasalahan dalam meningkatkan keamanan yang disebabkan pada saat *authentication* maka akan sulit diakses tanpa adanya token.

Tujuan implementasi JWT pada sistem informasi adalah untuk mensimulasikan keamanan informasi dalam hal autentikasi, mengakses *resource* pada *server* menggunakan REST API, dan melakukan perbandingan ukuran kinerja apabila JWT diterapkan pada dua (2) sistem operasi. Manfaat dari penelitian ini, yaitu JWT akan memastikan integritas data yang dikirimkan kepada *client* dari *server* dan dapat digunakan dalam mengautentikasi atau mengotorisasi dua aplikasi berbeda.

2 Tinjauan Literatur

Penelitian [7] menjelaskan penerapan JWT dalam proses implementasi dari beberapa bagian penerapan *web service*. Perancangan sistem verifikasi berkas kelengkapan terutama pada surat hasil pemeriksaan tes kesehatan secara otomatis dengan menerapkan teknologi *web service*. Verifikasi berkas pemeriksaan dengan menggunakan *QR Code* yang terintegrasi pada RESTFUL API akan menghasilkan data-data yang valid. Kekurangan dari penelitian ini, yaitu belum adanya fitur *backup* data pemeriksaan sebagai upaya dalam keamanan data. Hasil dari penelitian ini, yaitu membantu petugas dalam memverifikasi penumpang secara langsung berdasarkan data pemeriksaan tes kesehatan dari instansi kesehatan.

Penelitian [8] menjelaskan perbandingan algoritma yang diterapkan, antara *Secure Hash Algorithm* (SHA), *Rivest-Shamir-Adleman* (RSA), dan BLAKE. Algoritma SHA merancang fungsi kriptografi penyedia otoritas keamanan internet agar melindungi keamanan data. Algoritma tersebut akan bekerja dengan transformasi data menggunakan fungsi HASH. Algoritma RSA merupakan algoritma yang menggunakan konsep kriptografi kunci publik. Algoritma tersebut akan sulit melakukan cracking pesan karena menggunakan bilangan acak yang dijadikan kunci. Sedangkan algoritma BLAKE merupakan algoritma pengembangan dari algoritma SHA-3. Algoritma ini termasuk algoritma paling sederhana, namun memiliki peluang bermasalah karena memiliki reduksi yang akan menghambat proses keamanan data. Algoritma SHA menjadi yang terbaik dibandingkan dengan 2 (dua) algoritma lainnya karena bekerja untuk melindungi data dengan kuat. Kekurangan dari perbandingan algoritma ini adalah pada algoritma SHA kekurangannya jika *avalanche effect* atau karakteristik algoritma dapat mengubah bagian dari pesan dikirimkan, pada algoritma RSA jika kecepatan operasi yang diterapkan jauh lebih lambat dibandingkan kriptografi simetrik, sedangkan

 ISSN:2302-8149
e-ISSN:2540-9719

algoritma BLAKE jika *fixed-point* mendapatkan waktu yang kurang dari fungsi ideal atau tidak efisien. Hasil penelitian ini akan membandingkan algoritma terbaik dari ketiga algoritma yang dilakukan penelitian untuk implementasi keamanan data.

Kedua penelitian sebelumnya menerapkan teknologi JWT dengan berbagai macam algoritma yang diterapkan. Namun, pada perbandingan algoritma akan menjadi sebuah indikator manakah yang terbaik dalam menerapkan teknologi JWT. Implementasi *authentication* pada JWT menggunakan *framework* CodeIgniter 4 dan RESTful API akan menghubungkan data dari server ke kliennnya dengan pengamanan *authentication* menggunakan teknologi JWT. Selain itu, penerapan teknologi JWT ini menggunakan algoritma HMAC SHA-256. Data yang dihasilkan dalam bentuk *JSON* termasuk pada authentication menggunakan teknologi JWT. Pengujian dilakukan dengan menggunakan dua sistem operasi, yaitu Windows Server 2019 (*virtual machine*) dan Windows 10.

3 Metode Penelitian

Kelengkapan kegiatan penelitian menjadi salah satu hal menarik dengan mengadakan observasi informasi untuk menghasilkan sebuah ilmu pengetahuan. Gambar 1 merupakan tahapan-tahapan penelitian yang dilakukan pada penelitian ini.

Gambar 1. Tahapan Penelitian

3.1. Studi Literatur

Tahap studi literatur akan mencari referensi teori yang berhubungan dengan studi kasus penelitian terkait. Studi literatur dapat dijalankan dengan teknis wawasan yang luas tentang objek yang akan diteliti.

3.2. Analisis Kebutuhan Sistem

Tahap analisis kebutuhan sistem merupakan proses pemecahan masalah pada sistem menjadi bagian-bagian untuk memahami sebuah masalah. Proses pembagian tersebut dilihat dari sistem yang telah berjalan dengan kebutuhan sistem. Selain itu, proses mencari kebutuhan sistem bertujuan untuk mengidentifikasi kekurangan sistem yang kemudian menjadi langkah-langkah perbaikan. Ada 3 (tiga) proses pembagian analisis kebutuhan sistem adalah *hardware requirement* dan *software requirement*.

3.3. Perancangan Model

Tahap perancangan model merupakan tahapan yang dilakukan dalam merancang model yang akan diterapkan pada saat dilakukan implementasi berdasarkan kebutuhan. Konsep perancangan model meliputi pemaparan pola pengembangan dari arsitektur RESTful *web service* dan *flowchart* JWT *verification*.

3.4. Implementasi

Tahap implementasi merupakan tahap perancangan dan penerapan sesuai dengan acuan analisa sistem. Tahap tersebut akan merealisasikan sistem tersebut yang kemudian mengetahui kelebihan dan kekurangan dari sistem setelah diterapkan sesuai kebutuhan. Ada 2 (dua) proses implementasi sistem adalah implementasi pada *login form* sistem informasi dan implementasi *authentication* dengan JWT.

ISSN:2302-8149
e-ISSN:2540-9719

3.5. Pengujian dan Analisis

Tahap pengujian dan analisis merupakan tahap yang menekankan validasi terhadap pada implementasi yang diterapkan untuk menjamin kebutuhan sesuai spesifikasi layak dan dirumuskan dengan baik. Setelah tahap pengujian telah dikerjakan, langkah selanjutnya menganalisis hasil pengujian sistem dengan harapan sesuai dengan rumusan masalah yang dirancang.

4 Hasil dan Pembahasan

4.1. Studi Literatur

Hasil dari studi literatur membahas tentang keamanan *authentication* pada *RESTful API*, kemudian telah dilakukan alternatif keamanan pada teknologi *web service* dengan menggunakan JWT. Fokus penerapan JWT pada implementasi *authentication* menggunakan algoritma HMAC SHA-256.

4.2. Analisis Kebutuhan Sistem

1. *Hardware Requirement*
   Tabel 1 adalah spesifikasi *hardware* yang digunakan dalam pengujian *authentication* pada JWT.

Tabel 1. *Hardware Requirement*

Jenis	Spesifikasi
Processor	Intel Core i3-3217U CPU @ 1.80GHz
RAM	4 GB
Storage	SSD 128 GB

2. *Software Requirement*
   Tabel 2 adalah spesifikasi *software* yang digunakan dalam pengujian *authentication* pada JWT.

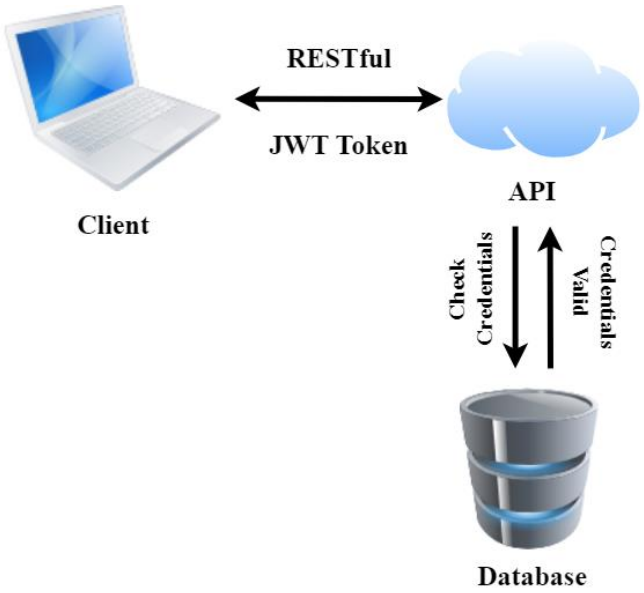
Tabel 2. *Software Requirement*

Jenis	Spesifikasi
Sistem Operasi	Windows 10 Pro 64-bit
Sistem Operasi (VM)	Windows Server 2019 64-bit
Virtual Machine	VMware Workstation 16 Pro
Web Server	Apache 2.4
Database	MySQL 7.4
Bahasa Pemrograman	PHP, HTML, Javascript
Framework	Codeigniter 4
Algoritma	HMAC SHA-256
Web Service	*JSON Web Token* (JWT)
Format data	JSON
Testing	*Postman*

4.3. Perancangan Model

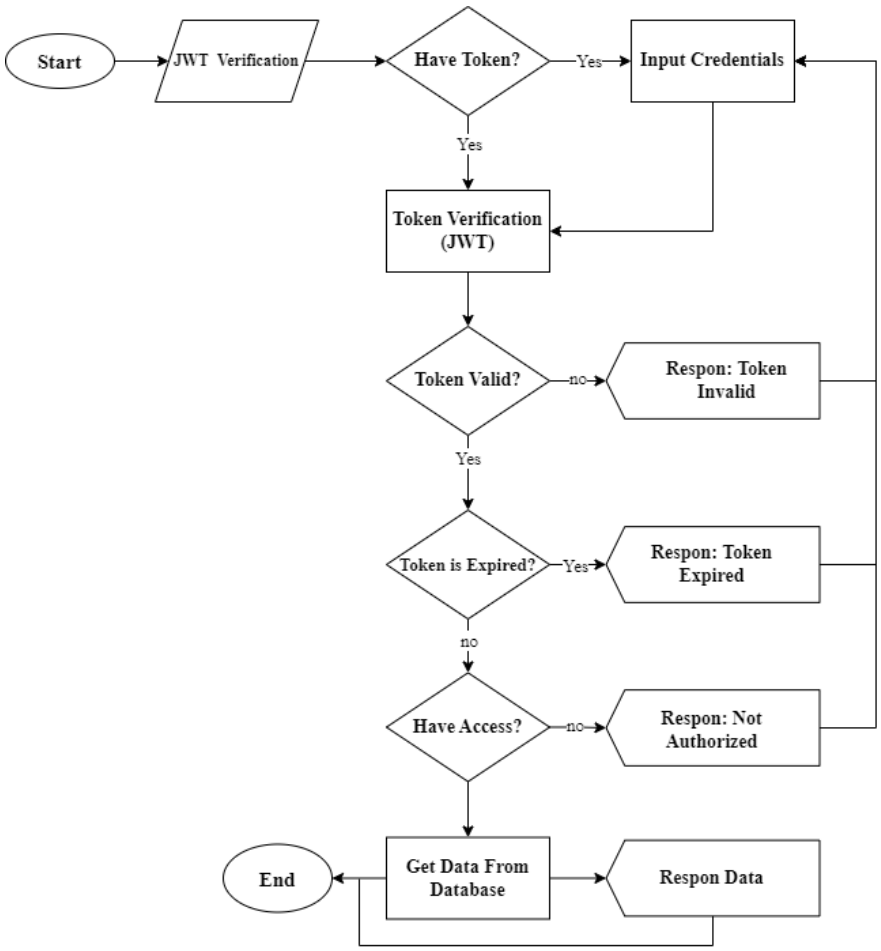
Model API atau *backend system* yang bekerja dalam proses *authentication* dan *authorization* pada sistem informasi. Proses *login* berhasil jika *server* memberikan *response* dalam bentuk token sebagai kunci yang mengakses sumber daya pada *server*. Model yang diusulkan untuk penerapan *authentication* JWT pada arsitektut REST *web service* dapat dilihat pada Gambar 2.

ISSN:2302-8149
e-ISSN:2540-9719



Gambar 2. Penerapan JWT pada arsitektur RESTful Web Service [9]

Verifikasi JWT meliputi jika token belum kadaluwarsa atau *expired* akan memeriksa hak akses yang menyimpan sebuah token pada *payload.* Token mempunyai hak akses untuk sumber daya, *web service* akan merespon dengan sumber daya sesuai yang dibutuhkan *user* [9]. Gambar 3 menjelaskan *flowchart* JWT *verification.*

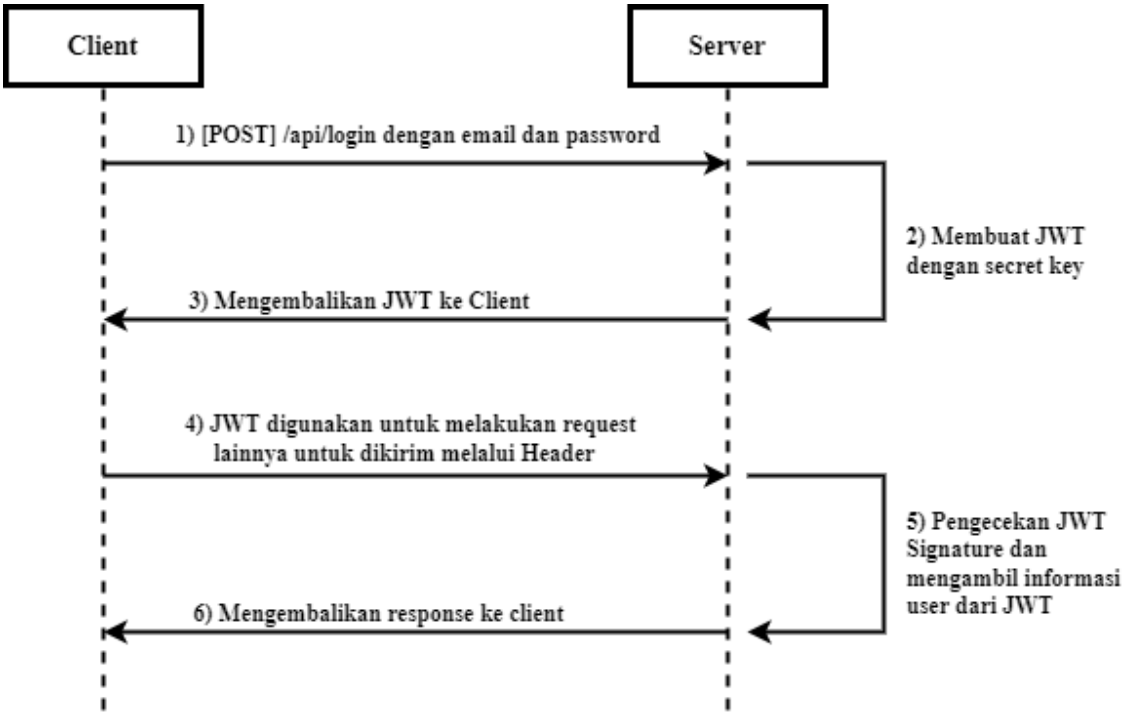


Gambar 3. JWT *Verification* [10]

 ISSN:2302-8149
e-ISSN:2540-9719

4.4. Implementasi Sistem

Penerapan hingga perbandingan ukuran kinerja JWT menggunakan algoritma HMAC SHA-256 dengan merancang sistem informasi berbasis *web*. Perancangan *service* akan memudahkan dalam pengimplementasian ke dalam bahasa pemrograman PHP dengan *framework* CodeIgniter 4. JWT menggunakan algoritma HMAC SHA-256 dan pengujian token dilakukan dari *client* dengan *tools* Postman. Tujuan utama dari penerapan JWT adalah keamanan data antar sistem yang bertukar data dengan membuat permintaan kepada *client* dengan menambahkan token. Cara kerja JWT ditunjukkan pada Gambar 4.



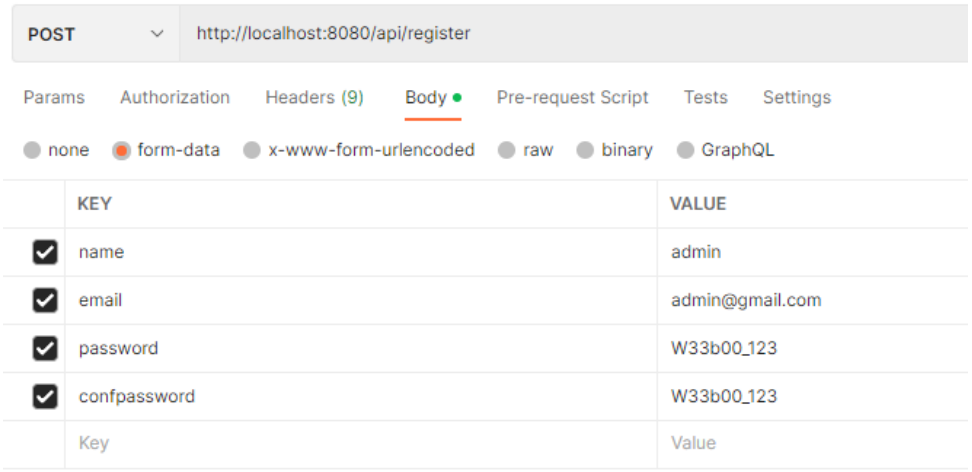
Gambar 4. Cara Kerja JWT

Proses yang diterapkan ada dua (2), yaitu POST dan GET [11]. Proses POST digunakan untuk memberikan parameter yang berisikan *username* dan *password* yang absah agar menciptakan token JWT. Token ini akan sebagai kunci untuk mendapatkan akses *request* selanjutnya. Sedangkan proses GET digunakan untuk menuangkan sebuah token yang diperoleh dari proses POST untuk mendeteksi data yang diperlukan dengan memproses *request* API data Masjid yang telah ada sebelumnya dalam bentuk JSON [12]. Saat proses *request*, token akan dikirimkan kepada HTTP *header* [13]. Proses *authorization* akan diperbaharui setelah HTTP request dilakuan dan mengakses kepada *response header*. Pengujian pada hasil implementasi yang menerapkan parsing data dengan *authentication* JWT [14]. Proses *login* dengan JWT akan dikirim ke server dengan format JSON [15]. HTTP *body* akan diisi dengan *email* dan *password* dengan *form-data* pada aplikasi Postman [16].

4.5. Pengujian dan Analisis

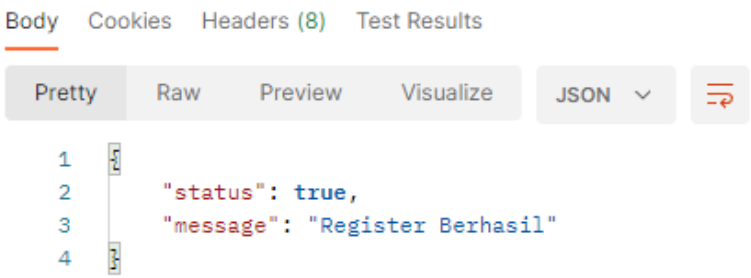
Pengujian pada hasil implementasi yang menerapkan parsing data dengan *authentication* JWT. Proses *login* dengan JWT akan dikirim ke server dengan format *JSON*. Proses *register* dengan JWT akan dikirim ke server dengan format JSON. HTTP *body* akan diisi dengan *username* dan *password* dengan *form-data* pada aplikasi Postman. Gambar 6 merupakan pengujian *register* dengan JWT pada aplikasi Postman.

 ISSN:2302-8149
e-ISSN:2540-9719



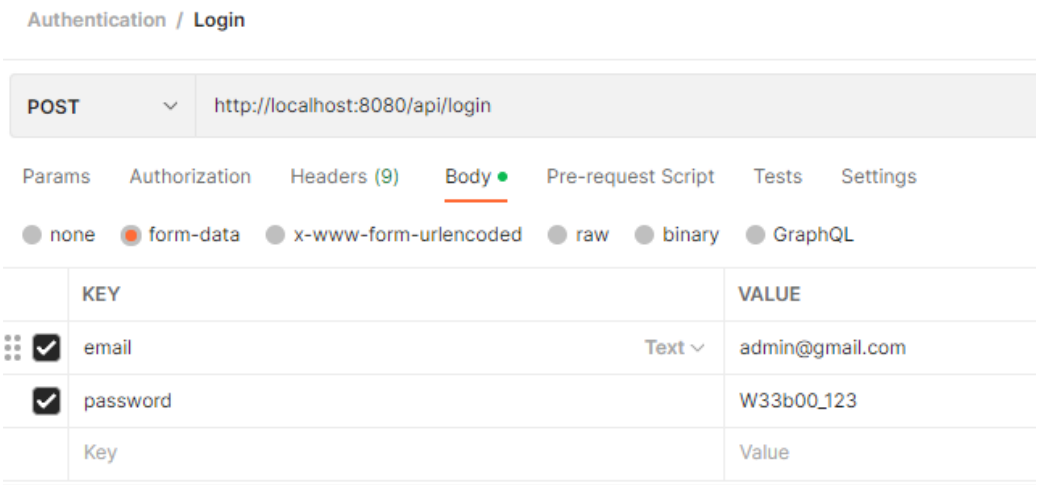
Gambar 5. Pengujian Register dengan JWT

Hasil *authentication* dengan JWT ini memperlihatkan data jika *Registered Successfully*. *Array* yang dihasilkan adalah status "*true*" dengan *message* "Register Berhasil". Hasil *register* dengan JWT menggunakan format *JSON*. *Password* yang dihasilkan saat melakukan *register* telah menerapkan *password hash* agar mengamankan data *user*. *Authorization*.yang dihasilkan saat melakukan *login* menggunakan JWT akan diterapkan pada saat memparsing data Masjid dengan format *JSON*. Gambar 7 merupakan hasil pengujian *register* dengan JWT.



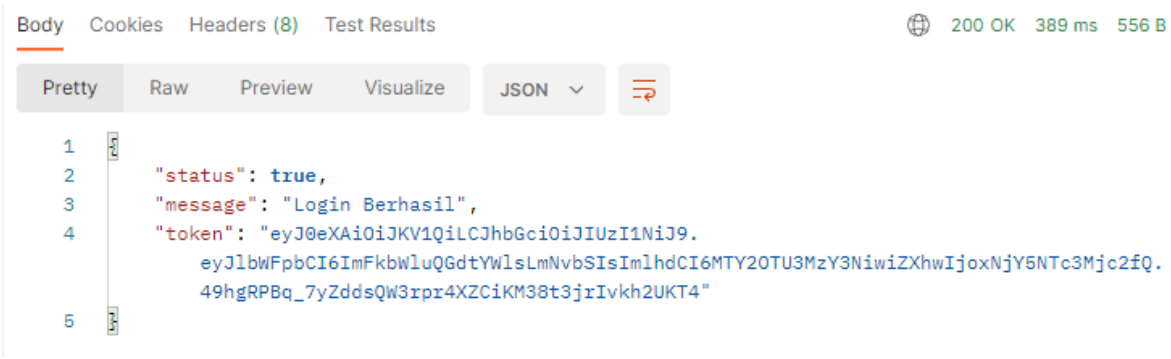
Gambar 6. Hasil Pengujian Register dengan JWT

Pengujian *authentication* pada JWT lainnya adalah *login* dengan JWT. HTTP *body* akan diisi dengan *email* dan *password* dengan *form-data* pada aplikasi *Postman*. Gambar 8 merupakan pengujian *login* dengan JWT pada aplikasi Postman.



Gambar 7. Pengujian *Login* dengan JWT

Pengujian *login* dengan JWT memproses *authentication* yang menghasilkan sebuah token yang menjaga kerahasiaannya. JWT akan mengembalikan hasil pengujian *login* dengan mengirimkan bagian *header* pada *authentication*. Pemeriksaan *signature* pada JWT bekerja dengan rahasia yang mendapatkan sebuah informasi *user* dari JWT. Setelah JWT menyelesaikan pemeriksaannya akan mengeluarkan *respons*e kepada *client* bahwa memperlihatkan hasil pengujian *login* dengan hasil data yang melakukan *login* dengan *access-token* untuk melakukan proses *parsing* data. Gambar 9 merupakan hasil *login* dengan JWT.



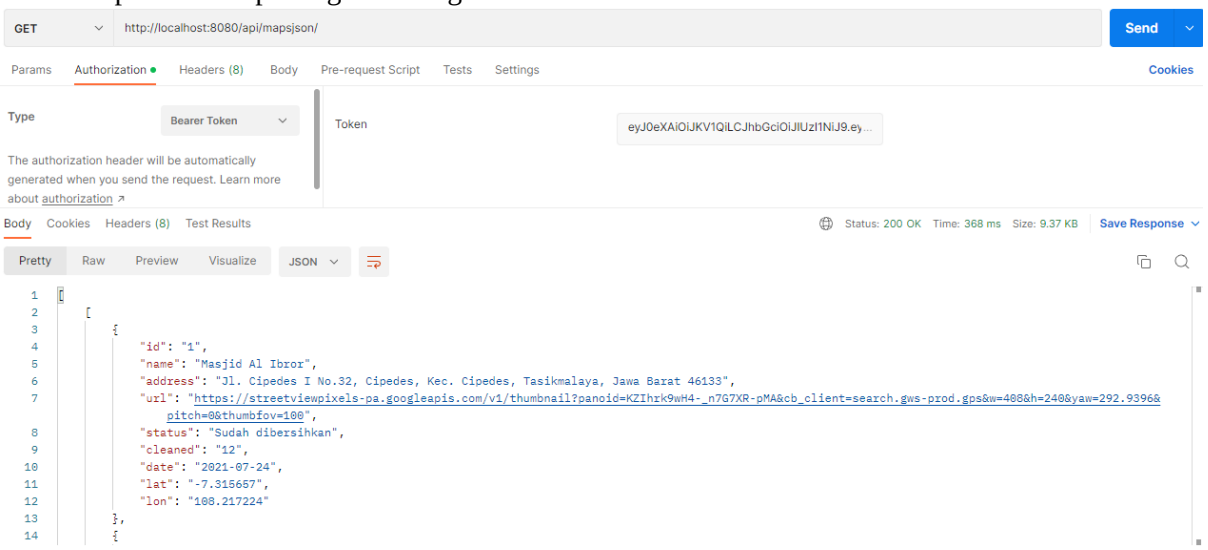
Gambar 8. Hasil *Login* dengan JWT

Pengujian *authorization* akan menguji token asli dengan token modifikasi yang telah diperoleh dari hasil *login* pada JWT. Pengujian ini akan menampilkan parsing data jika menggunakan token asli akan menghasilkan sebuah data Masjid. Tabel 3 merupakan token asli dari hasil *login* pada JWT.

Tabel 3. Token Asli

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6ImFkbWluQGdtYWlsLmNvbSIsImlhdCI6MTY2OTU3MzY3NiwiZXhwIjoxNjY5NTc3Mjc2fQ.49hgRPBq_7yZddsQW3rpr4XZCiKM38t3jrIvkh2UKT4

Token Asli diperoleh saat melakukan *authentication* pada JWT. Token ini akan mengotorisasi untuk melihat data yang tersimpan pada *database* yang ditampilkan dengan format *JSON*. Gambar 10 merupakan hasil parsing data dengan token asli.



Gambar 9. Hasil Parsing Data dengan Token Asli

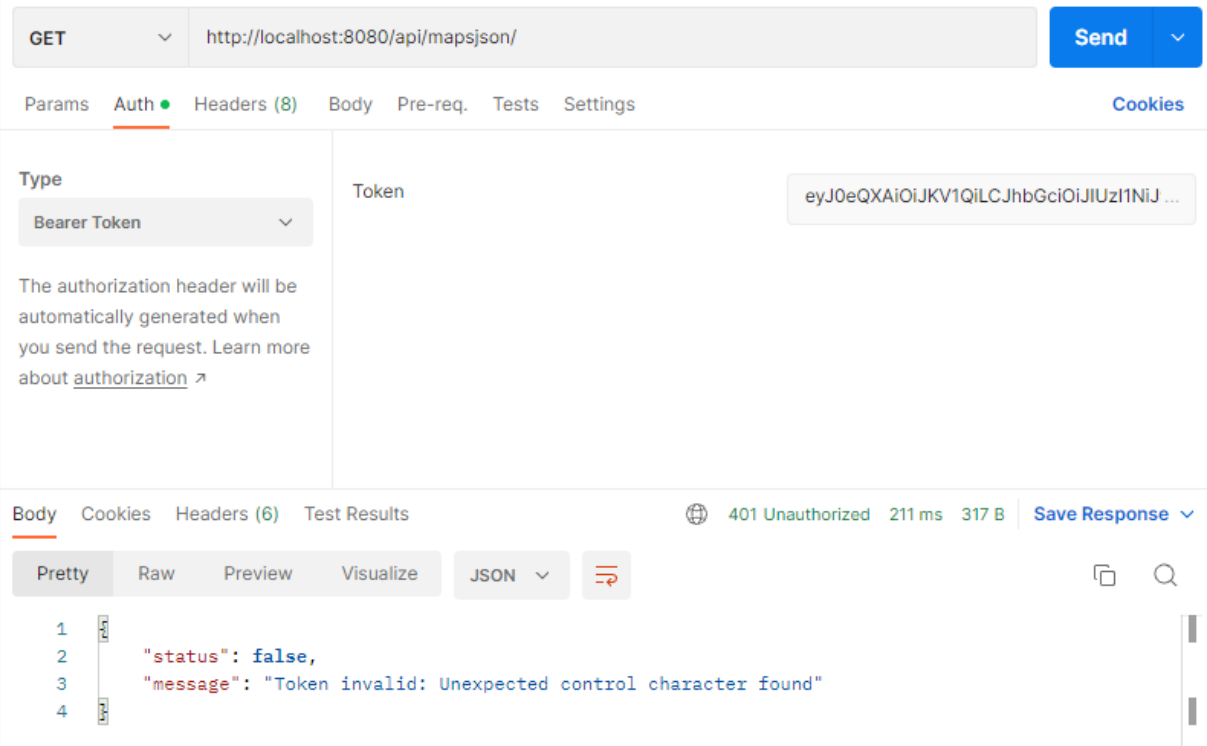
ISSN:2302-8149
e-ISSN:2540-9719

Selain parsing data dengan token asli, pengujian lainnya dengan menggunakan token modifikasi. Perbedaan yang sangat mencolok pada token modifikasi, yaitu menambahkan satu karakter. Token yang tidak dimodifikasi akan menghasilkan pesan token yang valid dan menampilkan data Masjid yang telah diparsing. Tabel 4 merupakan token modifikasi.

Tabel 4. Token Modifikasi

eyJ0eQXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6ImFkbWluQGdtYWlsLmNvbSIsImlhdCI6MTY2OTU3MzY3NiwiZXhwIjoxNjY5NTc3Mjc2fQ.49hgRPBq_7yZddsQW3rpr4XZCiKM38t3jrIvkh2UKT4

Karakter "Q" yang dicatat dengan tebal merupakan karakter modifikasi yang dicantumkan pada sebuah token. Token yang telah dimodifikasi akan mendapatkan pesan token tidak valid dan tidak dapat menampilkan data Masjid yang telah diparsing. Gambar 11 merupakan hasil parsing data dengan token modifikasi.



Gambar 10. Hasil Parsing Data dengan Token Modifikasi

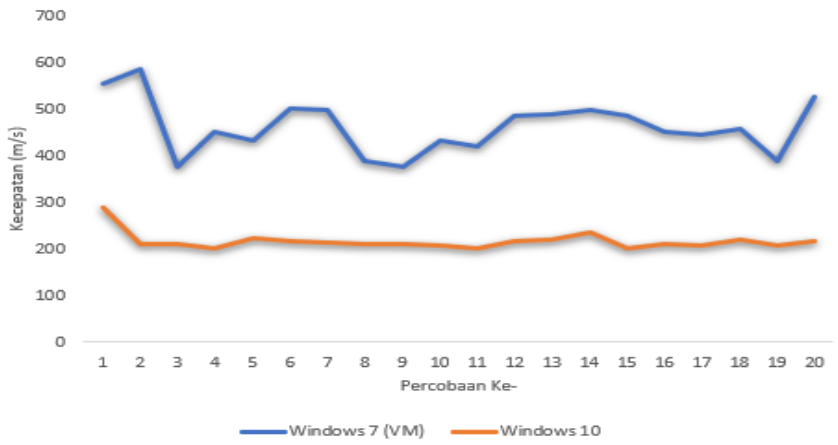
Pengujian kinerja JWT akan dilakukan dengan menggunakan sistem operasi berbeda, yaitu Windows 10 dan Windows Server 2019 (VM). Pengujian ini berdasarkan *response time* dan *size data* yang dihasilkan pada saat menggunakan JWT pada RESTful API Sistem Informasi Tim Bebersih Masjid. Perbandingan tersebut akan dianalisa sebanyak 20 kali percobaan untuk melihat pengimplementasian JWT dengan menggunakan algoritma HMAC SHA-256. Tabel 5 akan menggambarkan hasil perbandingan pengujian JWT pada dua (2) sistem operasi yang berbeda.

ISSN:2302-8149
e-ISSN:2540-9719

Tabel 5. Hasil Perbandingan JWT pada Dua Sistem Operasi

Percobaan	Kecepatan (m/s)		Ukuran (kb)	
	Windows Server 2019 (VM)	Windows 10	Windows Server 2019 (VM)	Windows 10
1.	288	555	8,59	8,59
2.	209	586	8,59	8,59
3.	211	378	8,59	8,59
4.	200	450	8,59	8,59
5.	222	432	8,59	8,59
6.	218	501	8,59	8,59
7.	214	500	8,59	8,59
8.	210	390	8,59	8,59
9.	212	378	8,59	8,59
10.	207	432	8,59	8,59
11.	201	421	8,59	8,59
12.	216	487	8,59	8,59
13.	220	490	8,59	8,59
14.	236	500	8,59	8,59
15.	200	487	8,59	8,59
16.	211	450	8,59	8,59
17.	206	444	8,59	8,59
18.	219	457	8,59	8,59
19.	207	390	8,59	8,59
20.	218	528	8,59	8,59
Rata-Rata	**216,25 m/s**	**462,8 m/s**	**8,59 kb**	**8,59 kb**

Tabel 5 menjelaskan kinerja yang dihasilkan dari JWT dengan algoritma HMAC SHA-256. Percobaan kinerja JWT dilakukan sebanyak 20 kali percobaan untuk melihat hasil terbaik. Percobaan yang dilakukan pada sistem operasi Windows 10 total kecepatan yang dihasilkan, yaitu 4325 ms dengan rata-rata 216,25 ms. Sedangkan pada percobaan yang dilakukan pada sistem operasi Windows Server 2019 (VM) total kecepatan yang dihasilkan, yaitu 9256 ms dengan rata-rata 462,8 ms. Namun, dari dua (2) sistem operasi yang melakukan percobaan hasil *size data* mendapatkan rata-rata 8,59 kb. Hasil *size data* yang sama berasal dari data yang tersedia saat parsing data Masjid yang sama-sama memiliki total 18 data Masjid. Gambar 12 merupakan grafik hasil perbandingan JWT.



Gambar 11. Grafik Hasil Perbandingan JWT

5 Simpulan

Hasil pengujian yang dilakukan pada penelitian ini adalah pengimplementasian *authentication* JWT pada Sistem Informasi Tim Bebersih Masjid telah berhasil diimplementasikan. Hasil pengujian pada *website* jwt.io untuk token yang dihasilkan pada implementasi *authentication* JWT ini

mendapatkan hasil "*signature verified*" dikarenakan token yang dihasilkan selalu berubah dan tidak mengubah isi *header* ataupun *payload*. Hasil pengujian diperoleh rata-rata kecepatan saat parsing data dengan JWT pada Windows Server 2019 (VM) sebesar 462,8 ms dengan rata-rata *size data* yang dihasilkan sebesar 8,59 kb. Pengujian pada sistem operasi Windows 10 diperoleh rata-rata kecepatan sebesar 216,25 ms dengan rata-rata *size data* yang dihasilkan sebersar 8,59 kb. Hasil pada Windows Server 2019 (VM) dari pengujian kinerja JWT sendiri mendapatkan hasil tertinggi, dikarenakan penggunaan *virtual machine* yang dinilai memakan RAM yang banyak menjadikan performa yang dihasilkan 2 kali lebih tinggi.

Pengembangan dalam penelitian selanjutnya, yaitu pengembangan *single sign on* (SSO) atau sistem multi login agar dioptimalisasi pada Sistem Informasi Tim Bebersih Masjid. Penambahan layanan yang terintegrasi menggunakan *single sign on* (SSO) untuk mempermudah organisasi *user* sebagai *single data user*. Selain itu, sangat dibutuhkan dalam *usability testing* tingkat lanjut yang dijadikan sebuah evaluasi yang akan digunakan pada iterasi selanjutnya.

Referensi

[1] P. Hendrarso, "Meningkatkan Kualitas Sumber Daya Manusia di Perguruan Tinggi menuju Era VUCA : Studi Fenomenologi Pada Perguruan Tinggi Swasta," *Prosiding Seminar Stiami*, vol. 7, no. 2, pp. 1–11, 2020.

[2] A. Hidayah, "Tantangan Kaum Freelancer dan Pemerintah Indonesia di Era Perkembangan Teknologi Digital," *RESIPROKAL: Jurnal Riset Sosiologi Progresif Aktual*, vol. 3, no. 1, pp. 92–104, 2021. DOI: 10.29303/resiprokal.v3i1.47

[3] Darmansah and Raswini, "Perancangan Sistem Informasi Pengelolaan Data Pedagang Menggunakan Metode Prototype pada Pasar Wage," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 6, no. 1, pp. 340–350, 2022.

[4] J. Lorasponelsar, A. Zuhdi, and G. B. Santoso, "Sistem Informasi Penelitian Berbasis Cms Wordpress Studi Kasus Lemlit Trisakti Pada Prodi Fti," *Prosiding Seminar Nasional Cendekiawan*, vol. 0, no. 0, pp. 1-44.1–1.44.6, 2019.

[5] G. Y. Gustiegan and Painem, "Implementasi Web Service Restful dengan Autentikasi Json Web Token dan Algoritma Kriptografi Aes-256 untuk Aplikasi Peminjaman Laboratorium Berbasis Mobile pada Universitas Budi Luhur," *Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)*, vol. 19, no. 1, pp. 9–16, 2022.

[6] A. Rahmatulloh, R. Gunawan, and F. M. S. M. S. Nursuwars, "Performance comparison of signed algorithms on JSON Web Token," *IOP Conference Series: Materials Science and Engineering*, vol. 550, no. 1, p. 012023, Aug. 2019 [Online]. DOI: 10.1088/1757-899X/550/1/012023

[7] G. W. Manueke, S. Oei, and W. W. Mamahit, "Implementasi Web Service pada Aplikasi Pemeriksaan Berkas Kelengkapan Penerbangan di Bandara Sam Ratulangi Manado Berbasis Web dan Android," *Global Science*, vol. 2, no. 2, pp. 54–67, 2021.

[8] N. Adianson, Y. Yupianti, and A. Kurniawan, "Analisa Perbandingan Performansi Rsa (Rivest Shamir Adleman) Dan Ecc (Elliptic Curve) Pada Protokol Secure Socket Layer (Ssl)," *Media Infotama*, vol. 11, no. 1, pp. 71–80, 2015.

[9] F. Ramadhani, U. Ramadhani, and L. Basit, "Combination of Hybrid Cryptography In One Time Pad (OTP) Algorithm And Keyed-Hash Message Authentication Code (HMAC) In Securing The Whatsapp Communication Application," *Journal of Computer Science, Information Technology and Telecommunication Engineering*, vol. 1, no. 1, pp. 31–36, 2020. DOI: 10.30596/jcositte.v1i1.4359

[10] R. Gunawan and A. Rahmatulloh, "JSON Web Token (JWT) untuk Authentication pada Interoperabilitas Arsitektur berbasis RESTful Web Service," *Jurnal Edukasidan Penelitian Informatika*, vol. 5, no. 1, pp. 74–79, 2019.

[11] G. A. P. Zaman, "Perancangan Dan Implementasi Web Service Sebagai Media Pertukaran Data Pada Aplikasi Permainan," *Jurnal Informatika*, vol. 11, no. 2, pp. 22–30, 2017. DOI: 10.26555/jifo.v11i2.a6252

[12] B. Adi Pranata, A. Hijriani, and A. Junaidi, "Perancangan Application Programming Interface (Api) Berbasis Web Menggunakan Gaya Arsitektur Representational State Transfer (Rest) Untuk Pengembangan Sistem Informasi Administrasi Pasien Klinik Perawatan Kulit," *Jurnal Komputasi*, vol. 6, no. 1, pp. 33–42, 2018. DOI: 10.23960/komputasi.v6i1.1554

[13] H. Herfandi, M. Julkarnain, and M. Hanif, "Desain dan Implementasi Restful Web Services untuk Integrasi Data dan Aplikasi," *Jurnal Informatika Teknologi dan Sains*, vol. 4, no. 1, pp. 36–41, 2022. DOI: 10.51401/jinteks.v4i1.1529

[14] L. V. Jánoky, P. Ekler, and J. Levendovszky, "Evaluating the Performance of a Novel JWT Revocation Strategy," *Acta Cybernetica*, vol. 25, no. 2, pp. 307–318, 2021. DOI: 10.14232/ACTACYB.289455

[15] A. B. Warsito, A. Ananda, and D. Triyanjaya, "Penerapan Data JSON Untuk Mendukung Pengembangan Aplikasi Pada Perguruan Tinggi Dengan Teknik Restfull Dan Web Service," *Technomedia Journal*, vol. 2, no. 1, pp. 26–36, 2017. DOI: 10.33050/tmj.v2i1.313

[16] W. Galindra Wardhana, I. Arwani, and B. Rahayudi, "Implementasi Teknologi Restful Web Service Dalam Pengembangan Sistem Informasi Perekaman Prestasi Mahasiswa Berbasis Website (Studi Kasus: Fakultas Teknologi Pertanian Universitas Brawijaya)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer; Vol 4 No 2 (2020)*, vol. 4, no. 2, pp. 680–689, 2020.