ISSN: 1979-2328

JSON Web Token Implementation for Dynamic Access Rights Authentication in Klinik Pratama UPN "Veteran" Yogyakarta Application Based on RESTful API

Implementasi JSON Web Token Untuk Autentikasi Hak Akses Dinamis pada Aplikasi Klinik Pratama UPN "Veteran" Yogyakarta Berbasis RESTful API

Naufal Rafif Danutirta[1], Yesyua Leon Christy[2]

[1,2] Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

[1]naufalrafif301199@gmail.com, [2]yesyualeon@gmail.com

Keywords: JSON Web Token (JWT); Representational State Transfer (REST); Application Programming Interface (API); authentication

Abstract

***Purpose**: This research was conducted to implement the authentication of Klinik UPN Veteran Yogyakarta application with dynamic access rights using JSON Web Token.*

***Design/methodology/approach**: This research went through several stages, starting from data collection, system requirements analysis, design, implementation, and system testing.*

***Findings/result**: JSON Web Token with dynamic access rights classification*

***Originality/value/state of the art**: Research that applies separation of access rights to JSON Web Token(JWT) and not only aimed at users who have the same right access has never been described before.*

Kata kunci: JSON Web Token (JWT); Representational State Transfer (REST); Application Programming Interface (API); autentikasi

Abstrak

Tujuan: Penelitian ini dilakukan untuk mengimplementasikan autentikasi aplikasi klinik UPN Veteran Yogyakarta dengan hak akses yang dinamis menggunakan JSON Web Token

Perancangan/Metode/Pendekatan : Penelitian ini melalui beberapa tahap, dimulai dari pengumpulan data, analisis kebutuhan sistem, perancangan, implementasi, dan pengujian sistem.

Hasil: JSON Web Token dengan klasifikasi hak akses yang dinamis.

ISSN: 1979-2328

Keaslian / *state of the art*: Penelitian yang menerapkan pemecahan hak akses pada JSON Web Token (JWT) dan bukan hanya ditujukan pada pengguna yang memiliki kesamaan hak akses belum pernah dijelaskan sebelumnya.

1. Pendahuluan

Representational State Transfer (*REST*) merupakan arsitektur standar web yang menggunakan *Hypertext Transfer Protocol* (*HTTP*) sebagai komunikasi data. Layanan ini muncul sebagai alternatif yang ringan dan hemat biaya untuk layanan berbasis *Simple Object Access Protocol* (SOAP) dan dirancang untuk memudahkan konsumsi, komposisi, dan pembangunan layanan berbasis komunitas. Pada survey penelitian [1] diperoleh bahwa layanan RESTful lebih terukur, andal dan terlihat, serta cocok dalam jaringan *wireless* yang mana paketnya ditransmisikan dari node sumber ke tujuan yang dapat berubah-ubah. Namun, layanan RESTful memiliki kekurangan yaitu tidak adanya properti skalabilitas dimana tidak memiliki potensi untuk ditingkatkan sebagai penanganan penambahan beban dan kurangnya tingkat keamanan. Oleh karena itu, diperlukan adanya solusi untuk mengamankan sumber data yang dirancang menggunakan RESTful, salah satunya dengan cara memanfaatkan autentikasi yang lebih aman.

Autentikasi merupakan fitur utama pada suatu aplikasi baik berbasis web maupun *mobile* yang digunakan untuk mencegah akses ilegal pada layanan dan data sensitif. Untuk meningkatkan keamanan yang lebih baik, telah dikembangkan beberapa teknik integrasi autentikasi, salah satunya autentikasi berbasis token[2]–[4]. Autentikasi ini memungkinkan pengguna untuk memasukkan nama pengguna (*username*) dan kata sandi (*password*) untuk mendapatkan token yang telah dibuat oleh *server* agar dapat mengakses sumber tertentu dan dalam jangka waktu yang telah ditentukan.

JSON Web Token (JWT) menjadi solusi dalam meningkatkan keamanan autentikasi pada sebuah aplikasi berbasis RESTful API. Pada dasarnya, JSON Web Token merupakan sarana representasi URL yang ringkas dan aman untuk mewakili klaim yang akan ditransfer antara *client* dan *server* yang dikodekan dalam struktur JSON Web Signature (JWS) dan/atau JSON Web Encryption (JWE). Berdasarkan definisi dari RFC 7159, kumpulan klaim JWT adalah objek JSON yang terdiri dari nol atau lebih pasangan nama/nilai (atau anggota), di mana namanya adalah string dan nilai berupa nilai JSON arbitrer [5]. Struktur objek JSON dikodekan dan dipisahkan oleh titik, diantaranya *header*, yang berisi atribut "type" yang akan diisi dengan JWT dan atribut "alg" dimana merupakan algoritma *hashing* yang digunakan (HS256 atau RSA). Bagian kedua merupakan *payload* dimana berisi objek klaim yang merupakan pernyataan atau pesan tentang suatu entitas, (umumnya entitas pengguna). Bagian terakhir berupa tanda tangan (*signature*), sebagai verifikasi kebenaran sumber pengirim JWT dan memastikan tidak adanya perubahan pesan.

Pada penelitian - penelitian sebelumnya, JWT telah diimplementasikan pada autentikasi aplikasi donor darah [6], aplikasi pengelola bisnis UKM (SIKASIR) yang dapat diakses berbagai platform [7], serta pada perangkat IoT [8]. Pada aplikasi donor darah, autentikasi berbasis token menggunakan JWT berhasil diterapkan pada layanan web dan *backend system blood donors* yang mana sistem ini dapat mengatasi permasalahan interoperabilitas dimana suatu aplikasi dapat saling berinteraksi dengan aplikasi lain melalui protokol yang sama. JWT juga dapat

ISSN: 1979-2328

diimplementasikan pada aplikasi SIKASIR yang menghasilkan adanya informasi pengakses aplikasi tersebut seperti perannya, maupun semua tautan yang diperbolehkan mengakses. Penggunaan JSON Web Token pada autentikasi juga dianggap aman karena siapapun yang mengubah konten token, layanan web akan secara otomatis mengirimkan pesan kesalahan yang mengatakan bahwa token tidak *valid* [9]. Pasalnya, token terintegrasi dengan *timestamp* atau tanda waktu dan fitur kadaluarsa (*expiration*) otomatis untuk memvalidasi integritas yang memungkinkan pembaruan token secara dinamis setiap kali sesi token berakhir. Sedangkan pada perangkat *Internet of Things* (NodeMCU), autentikasi terhadap token yang dikirimkan oleh pengirim *request token* (*publisher*) berhasil dilakukan oleh server yang diimplementasikan dengan menggunakan *framework Node.js*. Jika *publisher* token tidak valid baik kadaluarsa maupun telah diubah, maka *data center* dan *server* melakukan autentikasi dan menampilkan pesan *error*.

Untuk membuat suatu aplikasi klinik yang dapat digunakan di berbagai platform dan tetap aman, dirancang suatu aplikasi berbasis RESTful API dan mengimplementasikan JSON Web Token pada autentikasi pengguna. JSON Web Token (JWT) digunakan karena ukurannya yang relatif kecil sehingga memungkinkan untuk dikirim melalui URL, parameter pengiriman data (HTTP POST), atau *header* HTTP POST, serta adanya kemampuan untuk mengatur batas waktu atau kadaluarsa token tersebut. Implementasi JWT ini bertujuan untuk meningkatkan keamanan data dengan memanfaatkan enkripsi data *username, password,* dan hak akses tiap - tiap pengguna ke dalam suatu token. Sehingga, harapannya JWT dapat mencegah penyusup dalam mengakses data dan jika terdapat pengembang atau ahli yang membuka mode inspeksi pada aplikasi Klinik Pratama UPN "Veteran" Yogyakarta hanya dimunculkan sebuah token pada sesi akses tersebut.

2. Metode/Perancangan

2.1 Kebutuhan Fungsional

Client (pengguna) melakukan autentikasi hanya berupa *login* dengan menggunakan *username* dan *password*. *Username* dan *password* akan diberikan secara *default* bagi para pengguna untuk mengurangi kesalahan fungsi level pada tiap pengguna. Namun, setelah berhasil *login* pertama kali, *client* dengan level resepsionis, perawat, dokter, dan pegawai dapat mengubah *username* dan *password*. Apabila autentikasi dapat dilakukan dan berhasil, JSON Web Token diberikan oleh *server* REST dan kemudian memvalidasi token tersebut. Jika token yang diberikan *valid*, *client* akan diberi data JSON Web Token yang terenkripsi dari *server* REST, sehingga dapat mengakses informasi dan menggunakan aplikasi.

2.2 Perancangan Model

2.2.1. Arsitektur RESTful API

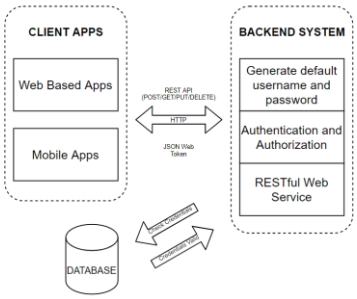
Dalam menangani proses autentikasi dan otorisasi pada RESTful API, diperlukan adanya peran sistem *backend*. Sistem ini berfungsi sebagai penyedia dan pengatur JWT. JWT merupakan enkripsi *public key* dengan secret key milik *back end*. Penerapan autentikasi dan otorisasi menggunakan JWT diusulkan sebagai berikut.

1. Mengatur *username* dan *password default* untuk pengguna,
2. Pengguna mengakses aplikasi dan melakukan *login*,
3. Request terhadap server,

ISSN: 1979-2328

4. Server melakukan verifikasi, ketika data ada di dalam basis data, maka akan dibuat token berdasarkan kunci yang telah disediakan oleh sistem *back end*, dan token akan masuk ke dalam penyimpanan lokal *client*,
5. Setiap ada permintaan baik berupa GET, POST, PUT, ataupun DELETE, token akan ikut dikirimkan. Sehingga, data API lebih aman, dan tidak sembarangan orang bisa mengakses API tersebut.

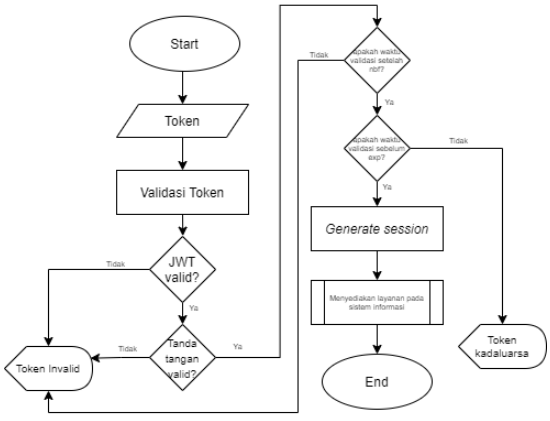
Skema penerapan JSON Web Token pada aplikasi berbasis RESTful API dapat dilihat pada **Gambar 1**.



Gambar 1. Penerapan JSON Web Token pada arsitektur RESTful API

2.2.2. Verifikasi JSON Web Token

Cara kerja JSON Web Token dalam memverifikasi data yang masuk diawali dari penerimaan token. Kemudian token divalidasi dan dideteksi apakah token tersebut valid. Jika ya, maka dilakukan validasi tanda tangan (*signature*), jika tidak, maka token *invalid* dan diberikan pesan error. Setelah itu, waktu validasi dikoreksi apakah setelah nbf. Jika ya, akan dilakukan pengecekan waktu sebelum *expired*, jika tidak, akan dilabeli sebagai token yang *invalid* dan ditampilkan pesan error. Jika waktu tidak melebihi kadaluarsa atau *expired* maka dibuat sesi. Jika tidak, token tersebut sudah kadaluarsa dan tampil pesan error. Setelah sesi dibuat, layanan akan disediakan pada sistem informasi sesuai dengan data yang telah diverifikasi sebelumnya. Alur kerja verifikasi JSON Web Token dapat dilihat pada **Gambar 2**.



Gambar 2. Diagram Alur Verifikasi JWT

ISSN: 1979-2328

2.2.3. Pembagian Level

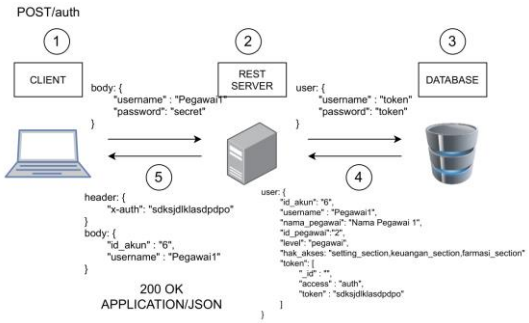
Terdapat 6 peran (level) berbeda dalam aplikasi klinik ini, yaitu, pasien, resepsionis, perawat, dokter, dan pegawai. Setiap level memiliki hak akses masing - masing yang mana khusus riwayat rekam medis hanya dapat diakses oleh level dokter dan pasien. Hak akses pada tiap level akan dirincikan di dalam **Tabel 1**.

Tabel 1. Klasifikasi Hak Akses Tiap Level

No.	Level	Hak Akses
1.	Pasien	Mendaftar pemeriksaan, memonitor antrian, melihat riwayat pemeriksaan, melihat invoice, melihat notifikasi, mengedit profil, melihat kartu periksa.
2.	Resepsionis	Mendaftarkan pasien lama, memonitor antrian, mendaftarkan pasien baru.
3.	Perawat	Mengisi tanda vital pasien, memanggil antrian, memonitor antrian berdasarkan poli.
4.	Dokter	Input diagnosa, input BHP, input resep, menyelesaikan pemeriksaan, melihat riwayat rekam medis pasien, melihat daftar antrian pasien.
5.	Pegawai	Melihat resep dokter, pengurangan stok obat, pembelian obat, melihat daftar obat, menagihkan layanan pemeriksaan pada pasien, melihat keuntungan penjualan.
6.	Admin	Mengatur hak akses pengguna, CRUD data master.

2.3 Perancangan Sistem

Sistem autentikasi aplikasi klinik berbasis RESTful API ini digunakan pada fitur *login*. Sedangkan fitur *logout* dibuat dengan cara penghapusan JWT pada *cookie*. Ada pun penerapan JWT pada fitur login dapat dilihat pada **Gambar 3**.



Gambar 3. Skema Fitur *Login*

Dalam penerapan JSON Web Token pada fitur *login*, proses diawali dengan memasukan data pengguna berupa *username* dan *password*. Setelah, pengguna menekan tombol *login* pada *user interface* yang disediakan, *front end* akan mengirimkan data tersebut (*request*) dengan metode POST terhadap *server* REST. Kemudian, JSON Web Token dibuat secara otomatis menggunakan algoritma SHA256 yang berisikan data *username* dan *password* tersebut. Lalu, token akan didekripsi dan dicocokan datanya dengan data akun dan hak akses yang ada di dalam

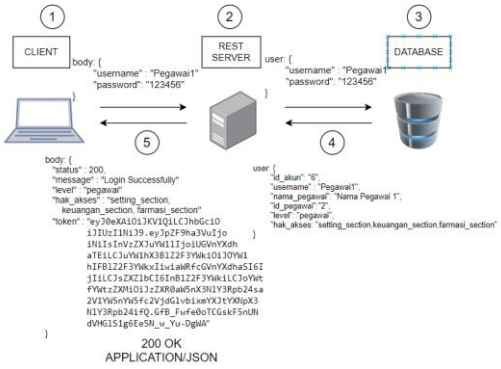
ISSN: 1979-2328

basis data. Selanjutnya, rincian data berupa id akun, *username*, nama pegawai, id pegawai, level pegawai, dan hak akses untuk pegawai dikirimkan kembali ke sisi *front end*. Sehingga, terbentuklah suatu sesi yang dapat digunakan untuk pegawai yang dapat mengakses menu farmasi, keuangan, dan *settings* (data master, *configuration*, dan *permission*).

3. Hasil dan Pembahasan

3.1 Implementasi

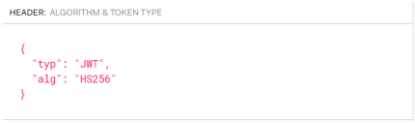
JSON Web Token diimplementasikan pada autentikasi login aplikasi Klinik Pratama UPN "Veteran" Yogyakarta. Adapun mekanisme pembuatan JSON Web Token dapat dilihat pada **Gambar 4.**



Gambar 4. Mekanisme Pembuatan JSON Web Token

Saat melakukan *login*, form data akan dikirim ke server kemudian server akan mencarikan username dan password yang sesuai dengan data tersebut kemudian server akan mengambil seluruh data yang dibutuhkan ke dalam payload. Kemudian, dilakukan proses enkripsi gabungan antara *header*, *payload*, dan *signature.* Hasil dari proses enkripsi gabungan tersebut adalah JSON Web Token.

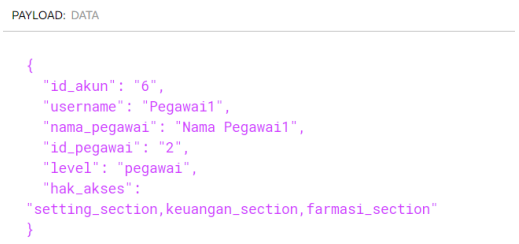
Header merupakan bagian dari susunan JSON Web Token yang berisikan informasi mengenai jenis token serta algoritma yang dipakai dalam pembentukan JSON Web Token tersebut[12]. Dalam pengujian kali ini, algoritma yang dipakai adalah HMAC-256 sehingga data header dapat dilihat pada **Gambar 5**.



Gambar 5. JSON Web Token Header

Payload merupakan data aplikasi yang akan dimasukkan ke dalam JSON Web Token yang berisikan semua data yang dipakai untuk login [13]. Pada pengujian kali ini data yang akan dimasukkan ke dalam JSON Web Token adalah data user yang mana data tersebut akan digunakan untuk memverifikasikan data pada aplikasi. Data aplikasi yang akan dimasukkan akan berbentuk data berformat JSON dapat dilihat pada **Gambar 6**.

ISSN: 1979-2328



Gambar 6. JSON Web Token Payload

Signature merupakan token yang dibuat dengan menggabungkan header, payload, dan token rahasia dan kemudian dienkripsikan menggunakan algoritma base64, kemudian data dienkripsi menggunakan algoritma HMAC-256[14]. Signature digunakan untuk memverifikasi pengirim JWT berasal dari sumber yang benar dan untuk memastikan bahwa pesan tidak berubah [15].

3.2 Hasil Pengujian

3.2.1. Pengujian respon HTTP server

1. Pengujian sukses *login*

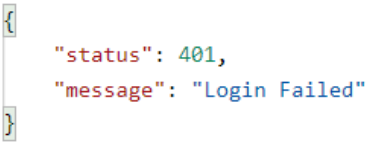
   Pengujian pada data *username* dan *password* yang dimasukkan merupakan data yang sesuai. Hasil keluaran yang diujikan berupa level, hak akses, serta token dapat dilihat dalam aplikasi Postman yang ditunjukan pada **Gambar 7**.

Gambar 7. Hasil Keluaran Token

Dalam pengujian tersebut menggunakan username “Pegawai1” yang mana jika berhasil *login*, level akan teridentifikasi sebagai pegawai dan memiliki hak akses pada *setting_section*, *keuangan_section*, dan *farmasi_section*.

2. Pengujian gagal *login*

Pengujian *login* dengan data *username* benar dan *password* salah. Hasil *message* yang diujikan dalam aplikasi Postman ditunjukan pada **Gambar 8**.



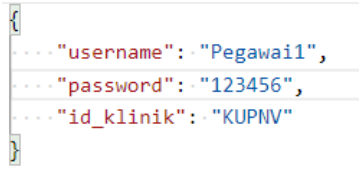
Gambar 8. Pengujian Gagal Login

ISSN: 1979-2328

3.2.2. Pengujian fungsionalitas otentikasi dengan JWT

Penerapan JWT pada POST/auth/login

Pada pengujian kali ini percobaan fungsi login pada aplikasi Postman dijalankan dengan memasukkan variabel data berupa username, password, serta id klinik seperti pada **Gambar 9**



Gambar 9. Data Input Login

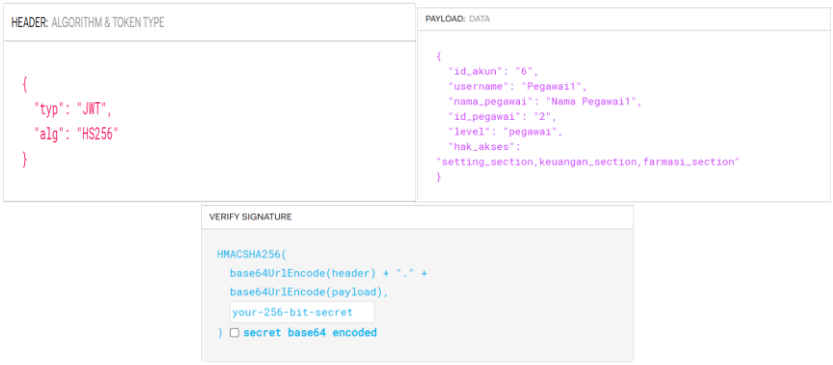
Keluaran yang dihasilkan adalah data berformat JSON yang berisikan level user, hak akses user, serta token seperti pada **Gambar 10.**

Gambar 10. Data Keluaran Login

3.2.3. Pengujian data token

Pengujian validasi token

Untuk melakukan pengujian validasi JSON Web Token, digunakan aplikasi *decode* daring yaitu JWT.io. Token yang didapatkan di-*decode* dan memunculkan hasil seperti pada **Gambar 11**.



Gambar 11. Hasil Deskripsi JWT

Pada **Gambar 11** menunjukan bahwa terdapat id_akun, *username*, nama_pegawai, id_pegawai, level, dan hak_akses di dalam dekripsi JSON Web Token yang cocok dengan data *username* "Pegawai1" di dalam basis data.

4. Kesimpulan dan Saran

Implementasi autentikasi menggunakan JWT meningkatkan keamanan data pada jaringan menggunakan sebuah token yang terbuat dari gabungan antara header, payload, serta signature.

Pada pengujian yang telah dilakukan JSON Web Token berhasil diimplementasikan pada autentikasi, dimana berisikan data *username* dan *password* yang dapat dienkripsi dan kemudian didekripsi menjadi data yang akan dipakai sebagai parameter untuk otorisasi sistem sesuai dengan hak akses penggunanya. Hal ini dibuktikan pada data dengan username "pegawai1" dapat masuk (*login*) ke aplikasi web Klinik UPN "Veteran" Yogyakarta dan bisa mengakses menu utama berupa farmasi, keuangan, dan *settings* serta sistem tidak memunculkan akses apapun untuk menu rawat jalan.

Dalam penelitian ini, penerapan JSON Web Token hanya terdapat pada *login* pengguna dikarenakan konsep *default account* dalam pembangunan aplikasi Klinik Pratama UPN "Veteran" Yogyakarta, yang mana tidak ada autentifikasi token disisi *backend* sehingga proses autentikasi hanya berjalan disisi *frontend*. Oleh karena itu, untuk pengembangan berikutnya autentikasi pada sisi backend dapat diterapkan sehingga dapat meningkatkan keamanan transfer data.

Daftar Pustaka

[1] Garriga, Martin et al , "RESTful service composition at a glance: A survey," in *Journal of Network and Computer Applications, Academic Press*, 2016, pp. 32–53, doi: 10.1016/J.JNCA.2015.11.020.

[2] P. Sahoo, N. K. Janghel and D. Samanta, "Securing WEB API Based on Token Authentication," International Journal on Advanced Electrical and Computer Engineering (IJAECE), vol. 4, no. 2, 2017.

[3] X.-W. Huang, C.-Y. Hsieh, C. H. Wu and Y. C. Cheng, "A token based user authentication mechanism for data exchange in RESTful API," International Conference on Network-Based Information Systems, pp. 601-606, 2015.

[4] A. Bhawiyuga, M. Data and A. Warda, "Architectural Design of Token based Authentication of MQTT Protocol in Constrained IoT Device," 2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA), 2017.

[5] Jones et al (2015), *RFC 6455 JSON Web Token*. Diperoleh 23 Agustus 2021 dari https://datatracker.ietf.org/doc/html/rfc7519.

[6] Gunawan, Rohmat dan Rahmatulloh, Alam, "JSON Web Token (JWT) untuk authetication pada interoperabilitas arsitektur berbasis RESTful web service," *JEPIN (Jurnal Edukasi dan Penelitian Informatika*), 2019, pp. 74 - 70, doi: 10.26418/jp.v5i1.27232.

[7] M. Haekal and Eliyani, "Token-based authentication using JSON Web Token on SIKASIR RESTful Web Service," *2016 International Conference on Informatics and Computing (ICIC)*, 2016, pp. 175-179, doi: 10.1109/IAC.2016.7905711.

[8] Warda, Andri et al, "Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya*, 2018, pp. 584 - 593.

[9] Trivedi, Hiral S dan Patel, Santika J, "Design of secure authentication protocol for dynamic user addition in distributed Internet-of-Things," *Computer Network, Elsevier*, 2020, pp. 107335, doi: 10.1016/J.COMNET.2020.107335.

[10] Satria, Bagus, Kusyanti, Ari, dan Yahya, Widhi, "Implementasi Algoritme Blake2s pada JSON Web Token (JWT) sebagai Algoritme Hashing untuk Mekanisme Autentikasi Layanan REST-API," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, Universitas Brawijaya*, 2018, pp. 6269-6276

[11] Ibarz, Juan Carlos Cruellas, "Bringing JSON signatures to ETSI AdES framework: Meet JAdES signatures," *Computer Standards & Interfaces*, *North-Holland*, 2020, pp. 103434, doi: 10.1016/J.CSI.2020.103434.

[12] Pooja Mahindrakar, Uma Pujeri, "Insights of JSON Web Token," *International Journal of Recent Technology and Engineering (IJRTE),* 2020, ISSN: 2277-3878, Volume-8 Issue-6

[13] Yjvesa Balaj,"A Survey: Token-Based vs Session-Based Authentication " Article September 2017

[14] Pooja Mahindrakar, Uma Pujeri, "Security Implications for Json web Token Used in MERN Stack for Developing E-Commerce Web Application, " *International Journal of Recent Technology and Engineering (IJRTE),* 2020, ISSN: 2249-8958, Volume-10 Issue-1

[15] I. P. A. Pratama, Linawati, and N. P. Sastra, "Token-based single sign-on with JWT as information system dashboard for government," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 16, no. 4, pp. 1745–1751, Aug. 2018, doi: 10.12928/TELKOMNIKA.V16I4.8388.

[16] Bakrim, La Ode dan Salam, " Koneksi jaringan internet menggunakan mode ad-hoc 802.11 pada tumaka kendari," *Jurnal Sistem Informasi dan Sistem Komputer* (*SIMKOM)*, STMIK Bina Bangsa, 2019, ISSN: 2581-1614.