

Penerapan JWT untuk Authentication dan Authorization pada Laravel 9 menggunakan Thunder Client

Robertus Laipaka

STMIK Pontianak

Pontianak, Indonesia

e-mail: robertus.laipaka@stmikpontianak.ac.id

Abstrak

JSON Web Token (JWT) merupakan representasi sebuah token web yang berbentuk string digunakan untuk melakukan pembuktian secara autentik dan memberikan hak kepada seseorang untuk menggunakan sistem dan data yang tersimpan didalamnya. Oleh karena itu, maka diperlukan autentikasi token web dengan menggunakan tiga bagian yaitu bagian header, bagian payload dan bagian signature kemudian dikombinasikan menjadi satu token. Penelitian ini menggunakan metode DSRM adalah sebuah metodologi yang berorientasikan desain informasi sistem. Sedangkan metode pengembangan menggunakan RAD. Tujuan penelitian ini adalah menghasilkan aplikasi untuk melakukan otentikasi yang aman menggunakan token web jwt yang dapat dipergunakan untuk merancang berbagai aplikasi yang menggunakan standar keamanan berbasis token web. Dengan adanya perangkat lunak ini maka diharapkan masalah dalam kebijakan akses data di sisi server dapat di otentikasi menggunakan token. Namun informasi yang berada di dalam token dapat diketahui oleh user maupun pihak luar sistem walaupun tidak dapat mengubah nilainya maka tidak dianjurkan memasukan informasi yang rasia ke dalam token. Hasil pengujian pada laravel 9 dan menggunakan thunder client dengan menggunakan skenario register, login, me, logout dan refresh dapat menghasilkan token yang aman karena di eckrip menggunakan secret digital signagture.

Kata kunci: json web tokent, larave 9l, thunder client, RAD

Abstract

JSON Web Token (JWT) is a representation of a web token in the form of a string used to authentically prove and give someone the right to use the system and the data stored in it. Therefore, it is necessary to authenticate web tokens using three parts, namely the header section, payload section and signature section then combined into one token. This study uses the DSRM method, which is an information system design oriented methodology. While the development method using RAD. The purpose of this research is to produce an application to perform secure authentication using a web token jwt that can be used to design various applications that use web token-based security standards. With this software, it is hoped that problems in data access policies on the server side can be authenticated using tokens. However, the information contained in the token can be known by the user or outside the system, although it cannot change its value, it is not recommended to enter confidential information into the token. Test results on laravel 9 and using thunder client using register, login, me, logout and refresh scenarios can generate secure tokens because they are encrypted using a secret digital signature..

Keywords: json web tokent, laravel 9, thunder client, RAD

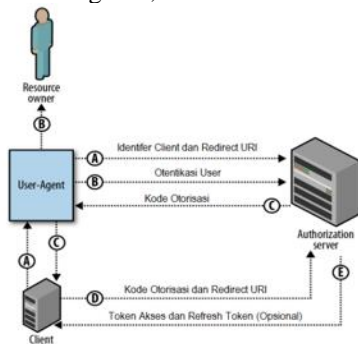
1. Pendahuluan

JSON Web Token (JWT) merupakan representasi sebuah token web yang berbentuk string digunakan untuk melakukan pembuktian secara *autentik* dan memberikan hak kepada seseorang untuk menggunakan sistem dan data yang tersimpan didalamnya [1]. Untuk melakukan autentikasi token web dapat menggunakan dua variabel diantaranya nama pengguna dan kata kunci kemudian di jalankan pada bagian header melalui protokol Http [2], token web ini digunakan untuk mengamankan sistem dan pertukaran informasi menjadi sebuah encode kedalam format JSON [3] menggunakan tanda titik menjadi bagian header yang di encode dengan base64 menggunakan javascript object signing and encryption menjadi payload yang di encode juga dengan base64 json web signature kemudian di tanda tangan digital

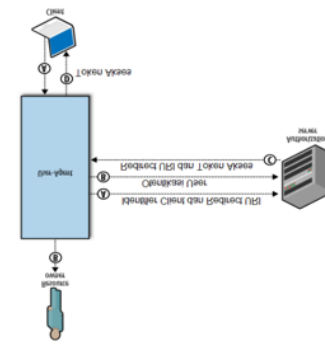
dengan kriptografi json web signature [4]. payload berisi key-value atau claim yang berisi iat (issued At/waktu token dibuat, exp (Expired Time/waktu token kadaluarsa), nbf (Not before/waktu dimana token JWT dinyatakan mulai valid. Dengan demikian masalah dalam kebijakan akses data di sisi server dapat di otentikasi menggunakan token. Namun informasi yang berada di dalam token dapat diketahui oleh user maupun pihak luar sistem walaupun tidak dapat mengubah nilainya maka tidak dianjurkan memasukan informasi yang rasia ke dalam token. Untuk menerapkan json web token (JWT) ini menggunakan algoritma HS256 [5] yang dapat menjaga keamanan data. Penelitian ini bertujuan menerapkan token web untuk *Authentication* dan *Authorization* pada laravel 9 [6] menggunakan *Thunder Client* [7]. Dengan adanya token web ini maka dapat meningkatkan keamanan sistem informasi data dan mengurangi potensi phishing, resiko leaks, dapat menjaga reliabilitas saat password berubah serta dapat mengurangi kebutuhan membuat mekanisme otentikasi yang kompleks pada aplikasi.

2. Metode Penelitian

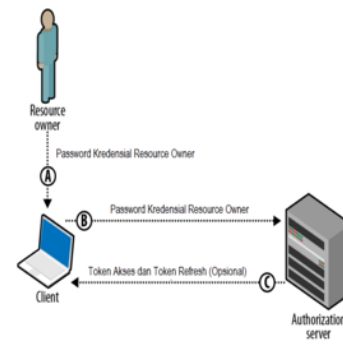
Penelitian ini berbentuk studi kasus dengan metode penelitiannya adalah DSRM (*Design Science Research Methodology*). DSRM adalah sebuah metodologi yang berorientasikan desain informasi sistem. DSRM juga merupakan kerangka prosedur yang digunakan untuk mempermudah penelitian di bidang teknologi informasi yang digunakan sebagai proses pemahaman serta mengulas untuk mengenali dan mengevaluasi hasil penelitian [8]. Framework laravel versi 9 [9], bahasa pemrograman yang digunakan adalah PHP (PHP: Hypertext Preprocessor). Metode pengembangan perangkat lunak RAD (*Rapid Application Development*) karena metode ini sangat menghemat waktu dalam pengerjaannya [10]. Sedangkan mekanisme untuk mendapatkan token menggunakan authorization code, implicit grant, password grand, client credential.



Gambar 1. Authorization code



Gambar 2. Implicit Grant



Gambar 2. Password Grant

Authorization code digunakan ketika ingin akses sistem dalam waktu lama dengan konsep bearer token [11]. Ketika melakukan login informasi user dan password tidak di simpan di client tapi menggunakan expired dengan cara melakukan refresh token Authorization Code. Untuk menguji token menggunakan thunder client yang terdapat pada extension visual code. Adapun fitur yang terdapat pada thunder client berisi semua request pada protokol http dapat dikirim menggunakan GET, POST, PATCH, PUT, DELETE, OPTIONS, HEAD [12]. Selain itu juga thunder client dapat membuat koleksi, menambah aktivitas, membuat token dan mengirim request pada bagian body menggunakan JSON, XML, teks, form encode dan graphql.

3. Hasil dan Pembahasan

Requirements Planning: Pada framework laravel akan digunakan untuk kemudahan dalam pembuatan autentikasi user untuk aplikasi yang akan digunakan untuk menguji token web yang dibangun menggunakan konsep Json Web Token (JWT). Selain itu framework laravel juga menggunakan laravel/ui sebuah package yang memberikan kemudahan dalam membangun sistem, dimana terdapat beberapa hal yang diperlukan untuk proses pembuatan autentikasi terdiri dari routing, view/halaman login, register, reset password, logout, controller, validasi.

3.1. Json Web Token (JWT).

Proses enkripsi token web yang berisi informasi pribadi akan di klaim menjadi encode kedalam bentuk format sring json dan menjadi sebuah payload dari json web signatur (JWS). Kemudian klaim dienkripsi menggunakan tanda tangan digital MAC (Message authentication code). JWT terdapat bagian token header yang memiliki algoritma yang digunakan dan jenis token, yang kedua token payload berisi klaim user dan metadata. Kemudian yang terkakhir token signature digunakan untuk membuat tanda tangan

digital dengan melakukan kombinasi antara encode header, encode payload dan scret. Dapat dilihat pada gambar berikut ini untuk proses encode dan decode [13].

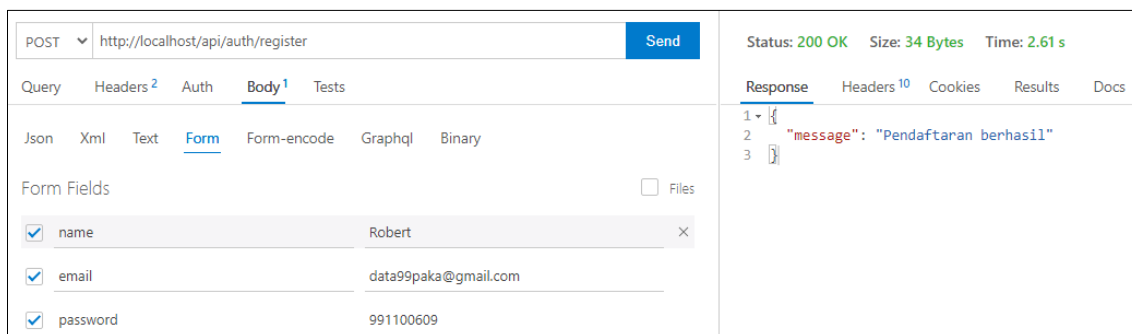
<p>HEADER:</p> <pre style="font-family: monospace; color: red;">{ "alg": "HS256", "typ": "JWT" }</pre> <p style="text-align: center;">Gambar 4. Header</p>	<p>PAYLOAD: DATA</p> <pre style="font-family: monospace; color: purple;">{ "username": "Robertus Laipaka", "email": "rbt99laipaka@gmail.com", "password": "991100609", "iat": 1516239022 }</pre> <p style="text-align: center;">Gambar 5. Payload</p>	<p>VERIFY SIGNATURE</p> <pre style="font-family: monospace; color: blue;">HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload), your-256-bit-secret) <input type="checkbox"/> secret base64 encoded</pre> <p style="text-align: center;">Gambar 6. Signature</p>
<pre style="font-family: monospace; color: black;">var encodedString = base64UrlEncode(header) + "." + base64UrlEncode(payload); HMACSHA256(encodedString, 'secret'); Hasil Encode: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IiJlIiwiaWF0Ij0iMTUxNjM5MDIyLmVudHVzIEExhaXBha2EiLCJlbWFpbCI6InJidDk5bGFpcGFrYUJnbWFpbC5jb20iLCJwYXNzd29yZCI6Ijlk5MTWwMDYwOSIsImhhdCI6MTUxNjM5MDIyLmVudHVzOTAyMn0.rxyxgD0Cm3KTrEQgyCS52Xsu-sIcpTMMoyziGllz0q4</pre>		

3.2. Testing Token Json Web

Proses pengujian token menggunakan thunder client dengan cara melakukan pengujian token yang memiliki beberapa skenario untuk autentikasi user terdiri dari tahapan Register, Login, Me, Logout, Refresh[14].

3.2.1. Tahapan Register

Langkah pertama dengan membuat collection Testing Token JWT kemudian create Request dengan nama Register menggunakan method POST pada url <http://localhost/api/auth/register> yang di dapat dari menggunakan perintah php artisan route:list pada laravel 9. Dapat dilihat pada gambar 7.



Gambar 7. Proses Registrasi Token

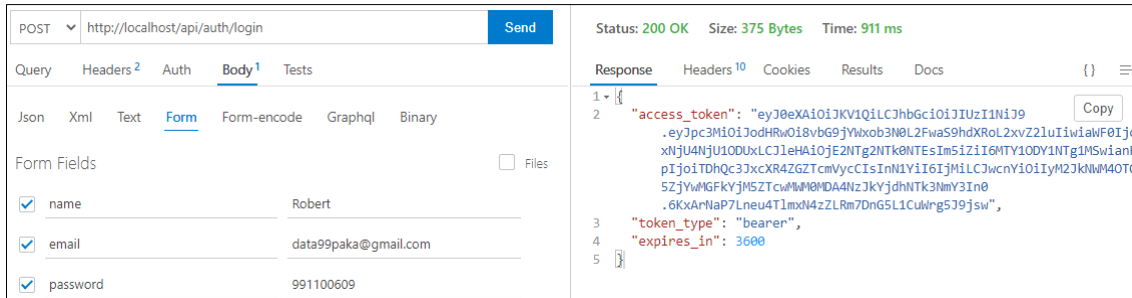
Jika proses registrasi token berhasil maka akan tampil response Message pendaftaran berhasil dan data tersimpan di dalam database. Untuk menguji user yang diinput berhasil maka tahapan selanjutnya adalah tahapan login dengan cara membuat collection login terlebih dahulu.

3.2.2. Tahapan Login

Untuk melakukan pengecekan user yang telah di register selanjutnya harus melakukan login terlebih dahulu dengan menggunakan thunder client, kemudian membuat collection login kemudian create Request dengan nama login menggunakan method POST pada url <http://localhost/api/auth/login> yang di dapat dengan menggunakan perintah php artisan route:list pada laravel 9. Jikan user dan password salah

maka akan mendapatkan response error unauthorized , Jika proses login berhasil maka token akan dibuat pada saat response berhasil dengan mencantumkan akses token {"access_token":

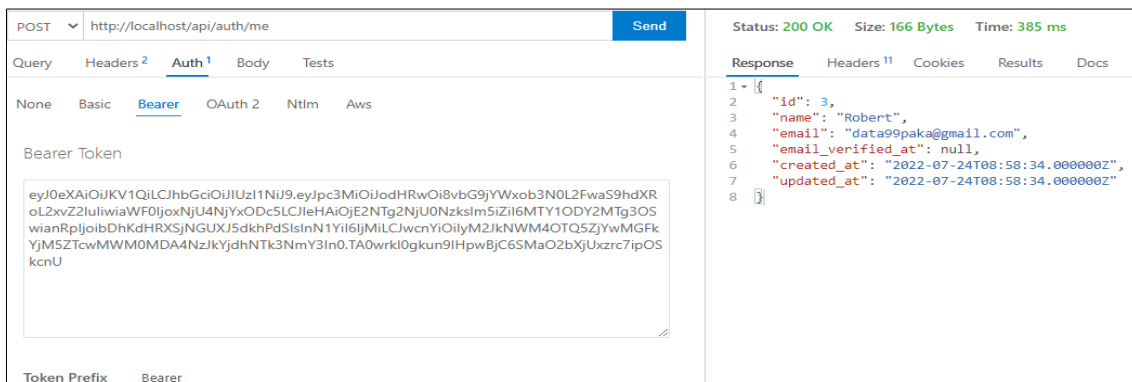
"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYWxob3N0L2FwaS9hdXRoL2xvZ2luIiwiaWF0IjoxNjU0NjU1ODUxLCJleHAiOiJlY2NTg2NTk0NTEsIm5iZiI6MTY1ODY1NTg1MSwianRpljoiTDhQc3JxcXR4ZGZTcmVyciIsInN1YiI6IjMiLCJwcnYiOiIyM2JkNW40OTQ5ZjYwMGFkYjM5ZTcwMWM0MDA4NzJkYjdhNTk3NmY3In0.6KxArNaP7Lneu4TImxN4zZLRm7DnG5L1CuWrg5J9jsw", "token_type": "bearer", "expires_in": 3600}, Status: 200 OK, Size: 375 Bytes ,Time: 911 ms, token ini berlaku selama 60 menit. dapat dilihat pada gambar 8.



Gambar 8 Proses Login Token

3.2.3. Tahapan Me

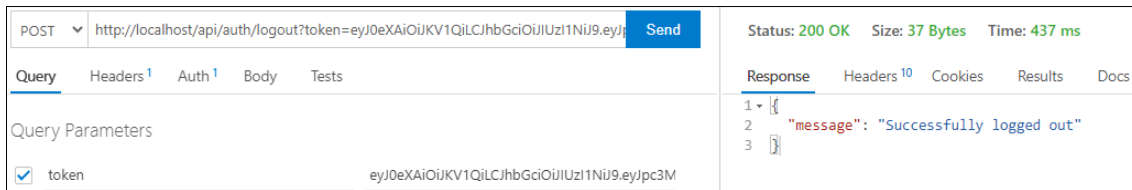
Pada tahapan Me ini adalah proses membaca dan menggunakan token dengan cara membuat collection Me kemudian create Request dengan nama me menggunakan method POST pada url <http://localhost/api/auth/me> yang di dapat dengan menggunakan perintah php artisan route:list pada laravel 9. Ada dua cara untuk menggunakan token me yang pertama menggunakan parameter token dengan value kode token yang sudah di create pada saat tahapan login berhasil dilakukan untuk proses decode token {"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpc3MiOiJodHRwOi8vbG9jYWxob3N0L2FwaS9hdXRoL2xvZ2luIiwiaWF0IjoxNjU0NjU1ODUxLCJleHAiOiJlY2NTg2NTk0NTEsIm5iZiI6MTY1ODY1NTg1MSwianRpljoiTDhQc3JxcXR4ZGZTcmVyciIsInN1YiI6IjMiLCJwcnYiOiIyM2JkNW40OTQ5ZjYwMGFkYjM5ZTcwMWM0MDA4NzJkYjdhNTk3NmY3In0.TA0wrkI0gkun9IHpwBjC6SMA02bXjUxzrc7ipOSkcnU", "token_type": "bearer", "expires_in": 3600}. Menghasikan decode {"id": 3, "name": "Robert", "email": "data99paka@gmail.com", "email_verified_at": null, "created_at": "2022-07-24T08:58:34.000000Z", "updated_at": "2022-07-24T08:58:34.000000Z"}. Cara yang kedua menggunakan Bearer Token dapat dilihat pada gambar 9 Berikut ini.



Gambar 9 Proses Membaca Token

3.2.4. Tahapan Logout

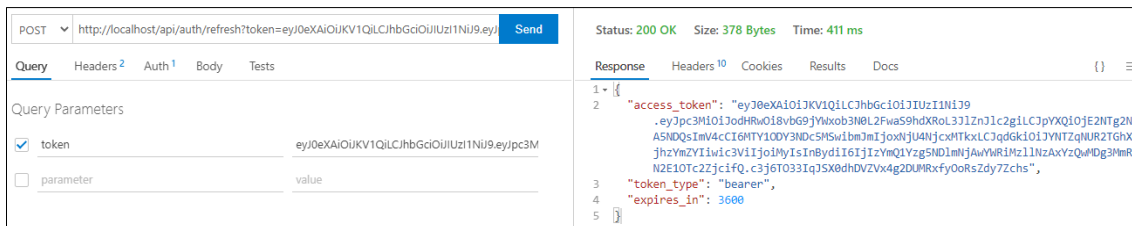
Pada tahapan logout merupakan proses destroy token supaya tidak bisa dipakai lagi dengan cara membuat collection Logout kemudian create Request dengan nama logout menggunakan method POST pada url <http://localhost/api/auth/logout> yang di dapat dengan menggunakan perintah php artisan route:list pada laravel 9. Masukan parameter token dan nilai encode json kemudian lakukan request makan token berhasil di destroy atau logout. Dapat dilihat pada gambar 10 berikut ini.



Gambar 10 Proses Logout Token

3.2.5. Tahapan Refresh

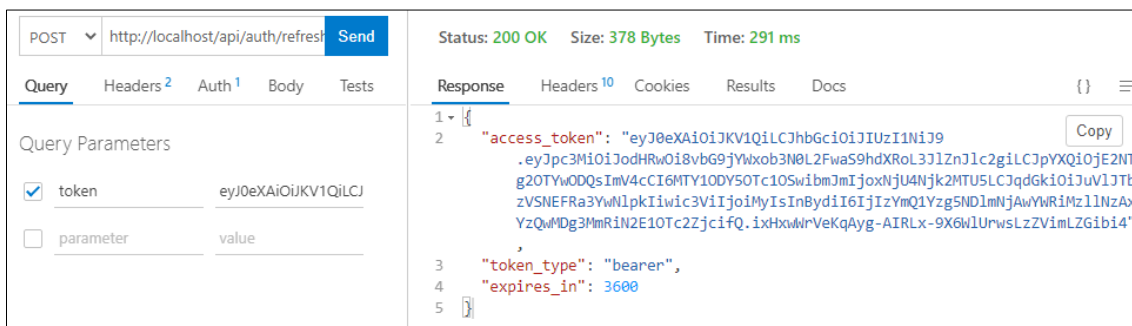
Pada tahapan refresh ini adalah proses membuat dan menggunakan token baru dengan cara membuat collection refresh kemudian create request dengan nama refresh menggunakan method POST pada url <http://localhost/api/auth/refresh> yang di dapat dengan menggunakan perintah php artisan route:list pada laravel 9. Pada saat token baru di buat maka token lama akan otomatis di hapus, jika user tidak ada aktivitas dalam jangka waktu tertentu. Gambar 11 merupakan proses refresh token baru.



Gambar 11 Proses Refresh Token

3.3. Hasil Testing Token Web JSON

Berdasarkan hasil pengujian token web json pada laravel 9 menggunakan thunder client dapat di uraikan kedalam 3 bagian yang di pisahkan oleh tanda koma dan diidentifikasi dengan pemisah dengan tanda titik seperti (rrrrrrrrr.bbbbbbbbbb.ttttttttt). Yang pertama bagian Header terdiri dari dua bagian yaitu bagian untuk mendeklarasikan jenis token (JWT) dan algoritma hashing (HMAC256) diencode menjadi eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ. Kemudian token web payload yang berisi klaim berisi entitas pengguna dan metadata tambahan yang terdiri dari {"id":3,"name":"Robert","email": "data99paka@gmail.com", "email_verified_at": null,"created_at": "2022-07-24T 08:58:34.000000Z", updated_at": "2022-07-24T 08:58:34.000000Z"}. kemudian diencode menjadi eyJpc3MiOiJodHRwOi8vbG9jYXRob3N0L2FwaS9hdXRoL3JlZnJlc2giLCJpYXQiOiJlZjE2NTg2NzA5NDQsImV4cCI6MTY1ODY3NTM5NywibmJmIjoxNjU4NjcxNzk3LCJqdGkiOiI3Z01TeE9tSFM2Nzc4N0RyIiwic3ViIjoIjoiMyIsInBydiI6IjIzYmQ1YzgzNDlmNjAwYWYWRiMzllNzAxYzQwMDg3MmRiN2E1OTc2ZjcifQ. Bagian ketiga token web adalah Signature yang di ambil dari header dan payload yang telah diencode secret seperti (3zTz7C4SWOd4QIwwlBLk2U2B4BzShA1GMRE2vvdQvM8). Untuk mendapat bagian ketiga signature in menggunakan var encodedString = base64UrlEncode(header) + "." + base64UrlEncode(payload); HMACSHA256(encodedString, 'secret'). Untuk Final Hasil Token Web (JWT) dapat dilihat pada gambar 12 berikut ini.



Gambar 12 Proses Create Token Berhasil

4. Kesimpulan

Berdasarkan hasil penelitian pada penerapan token web json (JWT) untuk Authentication dan Authorization pada Laravel 9 menggunakan Thunder Client yang diujikan dengan skenario Register, login,

Me, Logout dan Refresh dapat mentranmisikan informasi dengan aman karena informasi ini dapat *diverifikasi* dan dipercaya karena ditandatangani secara digital. token web JWT ini ditandatangani menggunakan secret dengan algoritma hashing (HMAC256) menggunakan RSA. JWT dapat digunakan pada framework laravel 9 dengan baik dan hasil pengujian menggunakan *Thunder Client* pada saat proses registrasi dengan ukuran Size 378 Bytes membutuhkan time 291 ms. Kemudian pada proses pembacaan Token Me dengan ukuran file 366 Bytes membutuhkan waktu 385 ms. Saat dilakukan destroy token dengan ukuran file sebesar 37 Bytes membutuhkan waktu 437 ms. Saat dilakukan refesh lagi terhadap token dengan ukuran file sebesar 378 Bytes membutuhkan waktu sebesar 411 ms. Kemudian yang terkakhir token signature digunakan untuk membuat tanda tangan digital dengan melakukan kombinasi antara encode header, encode payload dan scret dengan ukuran file 378 Bytes dengan waktu 291 ms dapat membuat token JWT dengan aman. Token web dapat di akses menggunakan token di URL, dengan parameter POST, atau header HTTP. Fleksibilitas JSON Web Token memungkinkan untuk mengautentikasi API dengan cepat dan mudah dengan meneruskan informasi melalui token web. Pada penelitian berikutnya akan menerapkan token web jwt untuk mengamankan traksaksi data API pada objek yang rentan terhadap kemanan informasi data.

Daftar Pustaka

- [1] S. E. Peyrott, "The JWT Handbook," 2018.
- [2] Berner-Lee T, "Network Working Group," 2005. Accessed: Jul. 25, 2022. [Online]. Available: <https://www.w3.org/People/Berners-Lee/>
- [3] Y. Sheffer, D. Hardt, and M. B. Jones, "JSON Web Token Best Current Practices Abstract," 2020. [Online]. Available: <https://www.rfc-editor.org/info/rfc8725>
- [4] N. Rasyada, "Algorithm on Json Web Token for Restful Web Service-Based Authentication," *Journal of Applied Data Sciences*, vol. 3, no. 1, pp. 33–43, 2022.
- [5] M. Jones, "JSON Web Algorithms (JWA)," 2015. [Online]. Available: <http://www.rfc-editor.org/info/rfc7518>.
- [6] J. A. Reyes, "Laravel 9.x," 2022.
- [7] R. Nagar and N. Eckert, "Thunder Documentation," 2022.
- [8] V. K. Vaishnavi and W. Kuechler, "Design Science Research Methods and Patterns Innovating Information and Communication Technology Second Edition," 2015.
- [9] T. Norgaard G, "Build an API with Laravel," 2019.
- [10] Kendall & Kendall, *System Analysis and Design*. 2014.
- [11] Prabath S, *Advanced API Security: Securing APIs with OAuth 2.0, OpenID Connect, JWS, and JWE-Apress*. 2014.
- [12] M. Jones, "JSON Web Token (JWT)," 2015, [Online]. Available: <http://www.rfc-editor.org/info/rfc7519>.
- [13] L. V. Jánoky, J. Levendovszky, and P. Ekler, "An analysis on the revoking mechanisms for JSON Web Tokens," *International Journal of Distributed Sensor Networks*, vol. 14, no. 9, Sep. 2018, doi: 10.1177/1550147718801535.
- [14] A. Warda, P. Putra, A. Bhawiyuga, and M. Data, "Implementasi Autentikasi JSON Web Token (JWT) Sebagai Mekanisme Autentikasi Protokol MQTT Pada Perangkat NodeMCU," 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>